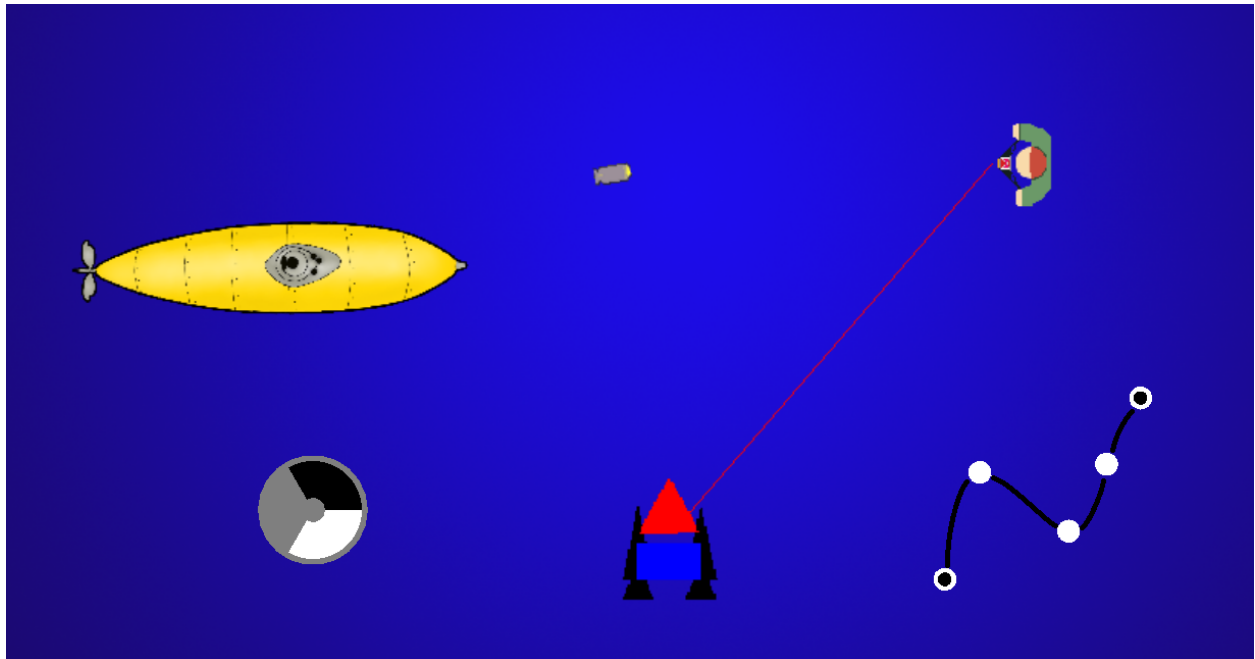




CHALMERS



Fantastiska spelmekaniker och var man hittar dem

En utforskning av unika designmöjligheter inom fysikbaserade videospel

Kandidatarbete inom Data- och Informationsteknik

ÅKE ANDERSSON
THOMAS JONSSON DAMGAARD
ALGOT NILSSON
ANNA ROMEBORN
RAZMUS STRANDELL

KANDIDATARBETE DATX02-18-42

Fantastiska spelmekaniker och var man hittar dem

En utforskning av unika designmöjligheter inom fysikbaserade videospel

ÅKE ANDERSSON
THOMAS JONSSON DAMGAARD
ALGOT NILSSON
ANNA ROMEBORN
RAZMUS STRANDELL



CHALMERS

Institutionen för Data- och Informationsteknik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sweden 2018

Fantastiska spelmekaniker och var man hittar dem
En utforskning av unika designmöjligheter inom fysikbaserade videospel
ÅKE ANDERSSON
THOMAS JONSSON DAMGAARD
ALGOT NILSSON
ANNA ROMEBORN
RAZMUS STRANDELL

© Åke Andersson, Thomas Jonsson Damgaard, Algot Nilsson, Anna Romeborn, Razmus Strandell,
2018.

Handledare: Staffan Björk, Inst. för Data- och Informationsteknik
Examinator: Olof Torgersson, Inst. för Data- och Informationsteknik

Kandidatarbete DATX02-18-42
Institutionen för Data- och Informationsteknik
Chalmers Tekniska Högskola
412 96 Göteborg
Telefon +46 31 772 1000

Omslagsbild: En kombinationsbild av de fem minispel som har utvecklats under projektets gång.

Institutionen för Data- och Informationsteknik
Chalmers Tekniska Högskola
Göteborg, Sweden 2018

Sammandrag

Syftet med detta projekt har varit att utforska designmöjligheter i videospel baserade på fenomen från sällan använda fysikområden. För att hitta intressanta utforskade fysikområden som skulle kunna användas har en dokumentation av tidigare spel skapats, varefter en analys av deras fysikanvändning har utförts. Därefter gjordes en kartläggning av dessa, vilken användes för att hitta de olika fysikområdena som vi ville utforska. För att nå projektets syfte har fem egna implementationer i form av minispel skapats, där fenomen från de olika fysikområdena vi valt att utforska lyfts fram som den centrala spelmekniken. Dessa fem minispel utforskar stråloptik, kvantfysik, lagrangesk mekanik, relativitetsteori, och strömningsmekanik. Spelidéer som utforskar fyra andra fysikområden utvecklades även, utan motsvarande implementationer. Dessa fysikområden är kretslära, elektriska fält, elektromagnetism, och fourieroptik. Implementationerna visar att det finns mindre utforskade fysikområden att skapa intressant gameplay kring. Dock visar projektet att användningsmöjligheten med de olika fysikområdena varierar. Störst potential ses i relativitetsteori och kvantfysik; där finns fenomen som inspirerar till spännande nya mekaniker. Vår slutsats är därmed att det finns designmöjligheter i att även utnyttja sällan använda fysikaliska fenomen i videospel.

Abstract

The purpose of this project has been to explore possibilities in the design of video games based on phenomena from seldom used areas of physics. To find interesting, seldom used areas of physics that could potentially be used, a documentation of previously existing games was created. After that, an analysis was made of their use of physics. A mapping of these results was then created, which was used to find the different areas of physics we wanted to explore. To achieve the purpose of the project, five mini-games have been created. Phenomena from the areas of physics chosen to explore were highlighted as the central game mechanics in each of these games. These physics explored in these games are optics, quantum physics, lagrangian mechanics, the theory of relativity, and fluid mechanics. Ideas for games exploring four other areas of physics were also developed, without any resulting implementations. These areas are circuit theory, electric fields, electromagnetism, and fourier optics. These implementations show that there exist less explored areas of physics that can be utilized to create interesting gameplay. The degree of possible utilization, however, varies between areas of physics. The greatest potential is seen in the theory of relativity and quantum physics; these contain phenomena that inspires the emergence of new exciting mechanics. We conclude that there exist possibilities to utilize seldom used areas of physics in the design of video games.

Tack till

Först och främst vill vi ge ett stort tack till vår handledare Staffan Björk för hans inspiration och hjälp under hela projektets gång. Han har åkt mellan två campus för vår skull och har haft mycket att säga för att vägleda oss i designprojektens tidigare obekanta värld.

Vi vill även tacka alla som har testat våra spel och gett värdefull feedback. Det gav oss möjligheten att se dem i ett nytt ljus och genomföra viktiga förbättringar.

Tack!

Åke, Thomas, Algot, Anna och Razmus
Göteborg, Maj 2018

Innehåll

1	Introduktion	1
1.1	Syfte	1
1.2	Avgränsningar	1
2	Bakgrund	2
2.1	Fotbollsspel	2
2.1.1	Slutsats från observationerna	2
2.2	Pusselspel	3
2.2.1	Slutsats från observationerna	4
2.3	First Person Shooter (FPS)	4
2.3.1	Slutsats från observationerna	6
2.4	Racingspel	6
2.4.1	Slutsats från observationerna	7
2.5	Plattformsspel	7
2.5.1	Slutsats från observationerna	8
2.6	Action-Adventure	8
2.6.1	Slutsats från observationerna	9
2.7	Olika fysikområden i videospel	9
2.7.1	Mekanik dominerar spelbranschen	9
2.7.2	Optik används som pussel inom videospel	9
2.7.3	Aerodynamik bidrar till en bättre realism på detaljnivå	10
2.7.4	Brist på användning av kvantfysik i spelbranschen	10
2.7.5	En avsaknad av relativistiska effekter i videospel	11
2.7.6	Spelmarknaden attraheras inte av elektromagnetism	12
3	Teori	12
3.1	MDA (Mechanics-Dynamics-Aesthetics)	12
3.2	Newtonsk mekanik	13
3.3	Relativitetsteori	13
3.3.1	Speciell relativitetsteori	13
3.3.2	Allmän relativitetsteori	14
3.4	Strömningsmekanik	14
3.5	Kvantfysik	15
3.6	Lagranges ekvationer	16
3.7	Optik	16
3.7.1	Stråloptik	16
3.7.2	Fourieroptik	17
3.8	Kretslära	18
3.9	Elektromagnetism	19
4	Genomförande	20
4.1	Tidsplanering och översikt	20
4.2	Dokumentation och analys	21
4.3	Mjukvaruutveckling	22
4.4	Framtagande av spelidéer	22
4.5	Implementering av minispelen	23
4.5.1	Utvecklingen av Spacetimewar	23
4.5.2	Utvecklingen av Hydrocurve	24
4.5.3	Utvecklingen av Tic Tac Ket	25
4.5.4	Utvecklingen av Lagom Action	26
4.5.5	Utvecklingen av A Maze in Light	28

5	Resultat	31
5.1	Implementationer	31
5.1.1	Spacetimewar	31
5.1.2	Hydrocurve	32
5.1.3	Tic Tac Ket	34
5.1.4	Lagom Action	35
5.1.5	A Maze in Light	37
5.2	Ej fullständigt implementerade spelprototyper	38
5.2.1	Kretslära	38
5.2.2	Elektriska fält	38
5.2.3	Elektrofall - The Electro Game	40
5.2.4	Fourieroptik	41
5.3	Kartläggning	41
5.4	Våra observationer om möjligheten att använda fysik i spel	42
5.4.1	Möjligheten att använda relativitetsteori för gameplay	42
5.4.2	Möjligheten att använda strömmingsmekanik för gameplay	43
5.4.3	Möjligheten att använda kvantfysik för gameplay	43
5.4.4	Möjligheten att använda Lagrangeformalismen för gameplay	44
5.4.5	Möjligheten att använda stråloptik för gameplay	44
6	Diskussion	45
6.1	Diskussion av resultat	45
6.1.1	Diskussion av analysen av tidigare spel	45
6.1.2	Diskussion av Spacetimewar	45
6.1.3	Diskussion av Hydrocurve	46
6.1.4	Diskussion av Tic Tac Ket	46
6.1.5	Diskussion av Lagom Action	47
6.1.6	Diskussion av A Maze in Light	47
6.1.7	Diskussion av ej implementerade spelprototyper	48
6.2	Diskussion av genomförande	49
6.2.1	Diskussion av dokumentation och analys	49
6.2.2	Diskussion av mjukvaruutveckling	49
6.2.3	Diskussion av avgränsningar	50
6.2.4	Diskussion av utvecklingen av specifika implementationer	50
6.3	Validitet och generaliserbarhet	52
6.4	Etiska och samhällreliga aspekter	52
6.5	Framtida arbete	53
7	Slutsatser	53
Bilaga A	Dokumentation över tidigare spel	62
A.1	Pusselspel	62
A.1.1	Portal	62
A.1.2	Bridge Builder	62
A.1.3	Crayon Physics Deluxe	62
A.1.4	Peggle	63
A.1.5	Monster Strike	63
A.1.6	The Talos Principle	63
A.1.7	The Incredible Machine	64
A.1.8	The Quantum Game	64
A.2	Fotbollsspel	64
A.2.1	Pele's Soccer	64
A.2.2	Super Mario Strikers	65
A.2.3	FIFA 14	65
A.3	First Person Shooters (FPS)	65
A.3.1	Wolfenstein 3D	65
A.3.2	Halo: Combat Evolved (Halo CE)	66

A.3.3	Call of Duty 4: Modern Warfare (CoD4)	66
A.3.4	Battlefield 4 (BF4)	66
A.3.5	Counter Strike: Global Offensive (CS:GO)	67
A.4	Racingspel	67
A.4.1	Space Race	67
A.4.2	Super Mario Kart	68
A.4.3	Grand Theft Auto V (GTA V)	68
A.4.4	Need for Speed (NFS)	68
A.5	Plattformsspel	69
A.5.1	Mario Bros., Super Mario Bros. och Super Mario 64	69
A.5.2	Braid	69
A.5.3	Broforce	70
A.5.4	Temple Run	70
A.5.5	Defy Gravity	70
A.5.6	The Electric Shocktopus	71
A.6	Action-Adventure	71
A.6.1	Adventure	71
A.6.2	Castlevania: Harmony of despair	72
A.6.3	Mirror's Edge	72
A.6.4	Zelda: Breath of the wild	72
A.7	Spel som är svåra att dela in i genrer	72
A.7.1	Velocity Raptor	72
A.7.2	Quantum Moves	73
A.7.3	A Slower Speed Of Light	73
A.7.4	Plasma Pong och Fluid Table-Tennis	73
A.8	Övriga spel	74
A.9	Sammanställning	74

Ordlista

Action-adventure Arcade Shooter	Genre som blandar element från action- respektive äventyrs-genren. En typ av First Person Shooter spel som centreras kring att man får poäng genom att skjuta ned motståndare.
Branch	“Gren” där en viss version av projektets källkod finns. Används för att underlätta då flera jobbar på samma projekt.
Bullet hell Fast Paced	Spelgenre där stora mängder projektiler kontinuerligt regnar ned mot spelaren. En term som används för att förklara en typ av spelmiljö som strävar efter snabba förändringar och en kort tid mellan olika interaktioner.
FPS	(First Person Shooter) Spelgenre som kännetecknas av att spelaren har en förstahandsperspektiv och ofta håller ett vapenliknande föremål till hands. Kallas även för förstapersonsskjutare.
Free-to-play	Ett spel som kan spelas helt gratis, dock ibland med visst innehåll som kan köpas för pengar.
Gameplay Git workflow	Spelupplevelse samt spelarens interaktion med spelmekaniken. Rekommendation för hur man kan arbeta med Git på bästa sätt för maximal produktivitet.
GR	Allmän relativitetsteori.
HSV	Hue-Saturation-Value, ett sätt att ange färger.
Indiespel	Ett spel tillverkat av enskilda individer, eller en mindre grupp, oberoende av någon större speltillverkare.
JSON	JavaScript Object Notation, ett filformat.
Open-source	Programvarans källkod är öppen för allmänheten.
Platform-adventure	Plattformsspel med inslag från action-adventure genren.
Power ups	Speciella föremål som spelare kan samla på och aktivera för att utöka sina förmågor.
Puzzle Shooter	En typ av First Person Shooter spel som baseras på att spelaren skall lösa pussel för att avancera till nästa nivå.
Ray casting	Metod för testning om en stråle skär en yta.
Real-time	En attribut till spel som innebär att spelet pågår i realtid utan att vänta på något spelare.
RNG	(Random Number Generator) En term som används för att beskriva slump genererade värden i ett videospel.
Spray Pattern	Sprutmönster, är en term som används för att beskriva rekylmönstret hos olika vapen inom spelgenren First Person Shooters.
SR	Speciell relativitetsteori.
Tactical Shooter	En typ av First Person Shooter spel som bygger på att spelaren måste lösa uppdrag som involverar någon form av taktisk tillvägagångssätt.
Turn-based	En attribut till spel som innebär att spelare turas om att vara aktiva.
Tutorial	Video eller manual med (oftast) stegvisa instruktioner för att åstadkomma eller lära sig en specifik sak.

1 Introduktion

Videospelens historia kan sägas börja år 1961-62, då det som ofta räknas som det första datorspelet skrevs: *Spacewar!*, där två rymdskepp ska försöka skjuta ned varandra medan de försöker undvika att sugas upp av en stjärna i mitten av skärmen. Även om det klassas som det första spelet i modern mening var *Spacewar!* egentligen bara gjord som en demonstration för vad en tidig dator, PDP-1, kunde åstadkomma[1]. Det riktiga kommersiella genombrottet för videospel kom, enligt *NE.se*, först cirka tio år senare i samband med lanseringen av hemdatorer[2], då man fick möjlighet att spela spel som det välkända *Pong* i hemmet[3]. Sedan dess har spelindustrin fortsatt att växa. Enligt *SuperData Research* spelade en tredjedel av jordens människor free-to-play-spel, den del av spelmarknaden som var störst under året 2017. Marknadens omsättning var då över 108 miljarder US-Dollar[4].

Med en så stor marknad och så många som spelar videospel är det viktigt för varje spel att ha något som låter det sticka ut ur mängden, så att det finns en anledning för en konsument att spela just det spelet, istället för att välja ett annat. Detta kan bland annat vara genom att spelet berättar en gripande berättelse, ser vackert ut, eller har någon intressant spelmekanik. Ett av många sätt att hitta en intressant spelmekanik, eller att få det som händer i sitt spel att se mer realistiskt ut, är att utgå från den verkliga världen och använda någon form av fysik i sitt spel.

Fysik har använts i videospel otaliga gånger, men ofta handlar det om mekanik, som att karaktärer som hoppar snart faller ned till marken igen. Spel där annan fysik än mekanik har en betydande roll är dock ganska ovanliga. Därför utforskar detta projekt möjligheten att göra spel med annan fysik än enbart mekanik samt utforska om realismen i denna fysik behöver kompromissas bort något för att göra spelet mer underhållande. Förhoppningen är att hitta mindre utforskade spelmekaniker och fysikområden som kan utgöra ett underhållande spel, samtidigt som spelaren kanske kan lära sig något om fysik bara genom att spela spelet.

1.1 Syfte

Arbetets huvudsakliga syfte har varit att utforska designmöjligheter i videospel baserade på fysik, med fokus på sällan använd fysik. Det har därför varit ett delsyfte att undersöka hur frekvent olika fysikområden har använts i redan existerande spel. Designmöjligheterna i gameplay från olika fysikområden har undersökts genom att implementera små spel, ett för varje fysikområde. Fokus har legat på fysiken och hur den kan användas, samt att undersöka hur den kan påverka spelupplevelsen, snarare än att utveckla bra spel.

1.2 Avgränsningar

På grund av brist på tid och resurser har detta projekt uteslutit vissa faktorer. Vid dokumentationen av fysiken inom existerande spel har ett urval gjorts som möjligtvis inte täcker alla aspekter eller spelmekaniker. Implementationerna har främst fokuserat på fysiken och är därmed inte kompletta spel. Dessutom har endast ett fåtal olika fysikområden realiserats i minispelen, då varje spel främst visar på en typ av fysik. Valen av fysikområden har då gjorts för att illustrera fysik som inte är så välrepresenterad i tidigare spel, eller för att använda fysiken i någon ny spelmekanik. De använder inte heller sig av stora nätverk utan spelas alla lokalt, eventuellt med delad skärm. Grafiken i implementationerna är enkel och står inte i fokus. Vi valde därför att begränsa oss till 2D-grafik.

Implementationerna har inte studerats med stora användartester på grund av tidsbrist, men enklare tester har genomförts. Detta leder till att en statistiskt signifikant slutsats inte kan dras. Det påverkar dock inte syftet med implementationerna alltför mycket då de grundar sig i att försöka utvidga användningen av fysik och se på möjligheter och svårigheter. De småskaliga användartesterna hade istället som syfte att försäkra att implementationerna är allmänförståeliga.

2 Bakgrund

Bakgrunden till ämnet videospel presenteras genom en analys av existerande spel. Analysen omfattar fysiken inom olika spelgenrer, olika fysikområden inom videospel samt en analys angående en kvalitativ eller kvantitativ användning av fysik inom videospel. Se Bilaga A för att läsa den fullständiga dokumentation av existerande spel.

2.1 Fotbollsspel

Fysiken i fotbollsspelen har enligt våra observationer överlag ideala förhållanden och tar inte hänsyn till faktorer som till exempel aerodynamiska krafter som luftmotstånd. Fotbollsspelen tycks bygga mycket på klassisk mekanik med kraftlagar, kollisioner och impulser. Ett av de första tilläggen inom spelgenren var *Pele's Soccer* (1981)[5] som byggde på en enkel spelmekanik med tydliga begränsningar hos spelaren. Spelaren styrde samtliga fotbollsspelare ute på planen samtidigt och fotbollsspelarens karaktär representerades i form av en färgad cirkel. Fysiken uppstod då spelaren tillförde en impuls hos fotbollen via en kollision. Denna impuls gav upphov till en förändring i rörelsemängden hos fotbollen vilket resulterar i att bollen åker med en viss hastighet längs med impulsens riktning.

Ett exempel på ett mer modernt fotbollsspel är *FIFA 14* (2013)[6]. I *FIFA 14* kontrollerar spelaren 3D-figurer vars karaktärer efterliknar verkliga fotbollsspelare med unika fördelar och svagheter. Fysiken vid övergång från en 2D-värld till en 3D-värld kompliceras då man behöver ta hänsyn till ytterligare en dimension för krafter men även ta hänsyn till rotationer. Då spelaren tillför en impuls till fotbollen sker detta via en animation i spelet där man tydligt kan observera en kontakt mellan spelarens kropp och bollen. I tidigare spel representerades istället detta förenklat med en cirkulär representation av spelaren som kolliderade med fotbollen. Var någonstans på spelarens kropp som bollen får kontakt avgör impulsens storlek jämfört med de tidigare spelmekanikerna som enligt våra observationer inte tog hänsyn till den faktorn. Exempelvis visas detta om spelaren i *FIFA 14* nickar bollen med huvudet eller sparkar bollen med foten, där man kan observera en markant skillnad i hur stor den tillförda impulsen blir.

FIFA 14 kom även med en ny förbättrad fysik för fotbollens rörelse i luften[7]. Det nya tillägget omfattade en korrekt motståndskoefficient till luftmotståndet på fotbollen[7]. Detta tillägg inkluderade ett nytt fysikområde för fotbollsspel som kallas för aerodynamik[8]. En korrekt hantering av aerodynamiska krafter leder troligtvis till en mer realistisk visualisering av föremål som färdas genom en fluid. *FIFA 14* hanterar även exempelvis den så kallade Magnuseffekten, som talar om hur kroppar tenderar att böja av från sin färdriktning om den utsätts för ett roterande moment[9]. Om man tillför en rotation hos fotbollen, genom att "skruva" bollen, kan fotbollen få en ny färdriktning som uppkommer av Magnuseffekten. En dynamisk friktion mellan markytan och bollen har också implementerats i *FIFA 14*. Denna dynamiska friktion är väderberoende. Om planytan är våt efter nederbörd glider bollen mer, vilket påverkar hur den studsar. Detta är fysik som försvårar situationen för spelaren som måste ta detta i beaktande då den till exempel agerar målvakt.

Fotbollsspel enligt våra observationer fokuserar inte entydigt på realistisk fysik för att skapa en verklighetstrogen spelupplevelse. Ett tydligt motexempel är *Super Mario Strikers* (2005)[10] som inte strävar efter en realistisk fysik utan istället en surrealistisk spelupplevelse som grundas i genren fantasy. I *Super Mario Strikers* har spelaren möjlighet att välja bland Nintendos kända spelfigurer som *Mario* och *Luigi* som alla har unika för- och nackdelar. Fysiken i spelet övergår från realistisk till surrealistisk då man inkluderar spelmekaniker som *Super Strike*. Vid en *Super Strike* upphör gravitationen och andra ofysikaliska fenomen uppkommer som till exempel att konservering av laddning upphör då bollen blir kraftigt laddad.

2.1.1 Slutsats från observationerna

Resultaten från våra observationer visar på att majoriteten av fotbollsspelen troligtvis strävar efter en realistisk spelupplevelse[11]. Om speldesignen är mer inriktad mot fantasy tros fysiken vara mindre viktig då man troligtvis strävar efter en mer surrealistisk spelupplevelse. Enligt våra observationer utgörs fysiken i fotbollsspel i stora drag av mekanik med små undantag som till exempel luftmotstånd hos fotbollen, vilket beskrivs av aerodynamik. I modern tid har man exempelvis infört Magnuseffekten för att bättre illustrera hur bollen kan skruvas i luften, dynamisk friktions-

konstant mellan markytan och bollen vid olika väderförhållanden samt arbetat för att utveckla en mer realistisk representation av luftmotstånd.

2.2 Pusselspel

Ett flertal pusselspel observerades vara fysikbaserade och tros förlita sig helt på realistisk fysik för att utmana spelaren. På så vis ligger fysiken i centrum för gameplay och speldesign. Spelen observerades utmana användaren att lösa problem som efterliknar ingenjörspromblematik. Det krävs dock ingen fördjupad förståelse av fysiken eller stora beräkningar för att lösa problemen. Spelaren förlitar sig på sin intuition och rättar sig efter spelreglerna. Om denna intuition inte täcker all problematik, behöver spelaren experimentera sig fram. Därmed visar sig fysiken vara relevant även för spelreglerna, vilket bidrar till att spelaren får en kvalitativ och kvantitativ förståelse för fysiken. Med kvantitativ förståelse menas exempelvis att spelaren kan bestämma storlekar på krafter, hastigheter och riktningar. Exempelvis i *Half-Life*[12] behöver spelaren placera ett tungt objekt på en knapp placerad längs markytan för att öppna en tidigare otillgänglig dörr. Om spelaren placerar mindre objekt på knappen kommer knappen inte att tryckas in då den inte utsätts för en tillräckligt stark kraft. Därmed behöver spelaren välja en tillräckligt stor massa för att knappen ska tryckas in.

Bridge Builder (2000)[13] är den första av sin sort av pusselspel som grundas i att spelaren utmanas till att skapa en hållfast konstruktion. I *Bridge Builder* har spelaren möjlighet att anta olika utmaningar som alla har en gemensam faktor, att lösa ingenjörslignande problem inom brobygge som grundas i hållfasthetslära och mekanik. Först måste spelaren bygga en bro över ett hinder. För att senare testa om bron är hållbar nog passerar ett antal fordon över den. Om bron klarar av att transportera alla fordon utan att rasa har spelaren lyckats med utmaningen. Som verktyg har spelaren till exempel tillgång till kablar för att skapa hängbroar. Spänningskraften som uppkommer i kablarna förstärker brons konstruktion, såvida spelaren har placerat kablarna på ett korrekt sätt. Efter sin succé skapades en hel familj av pusselspel med liknande koncept som tillgavs namnet “*Bridge Builder games*”[14].

Ett annat exempel på ett “*Bridge Builder game*” är *Crayon Physics Deluxe*[15]. Återigen observerades att realistisk fysik tycks ha en central roll i spelmekniken. Spelaren måste skapa konstruktioner för att transportera ett runt objekt fram till en stjärna och vid kontakt med stjärnan får spelaren poäng och går vidare. Mekanik står för största delen av användningen av fysik med bland annat olika moment där spelaren måste skapa ett vridmoment kring en balk för att skjuta iväg det runda objektet över ett hinder. Vridmomentet kring balken beskrivs likadant som funktionen hos en gunga där objektet åker iväg på grund av en fiktiv centrifugalkraft[16] som uppstår av vridmomentet. Den fiktiva centrifugalkraften kan beskrivas som den kraft som trycker iväg ett föremål som utsätts för en rotation. Med riktningen “iväg” menas riktningen ut från rotationsriktningen.

Portal[17] och *The Talos principle*[18] är två exempel på en blandning av spelgenrerna *First Person Shooter (FPS)* och pusselspel. Dessa två spel observerades till att vara väldigt lika på det sättet att spelaren befinner sig i en 3D-värld och att spelaren håller något typ av föremål i handen som används flitigt under spelets gång. I *Portal* håller spelaren en så kallad *Portal Gun* där spelaren har möjlighet att öppna två portaler och transportera olika föremål eller sig själv mellan dessa portaler omedelbart. I *The Talos Principle* måste spelaren plocka upp olika sorters föremål, däribland ett stativ med en kristall på. Kristallen används bland annat för att reflektera olika lasrar. Spelmekniken är väldigt lik i båda spelen; klara av olika utmaningar i form av fysikbaserade pussel i en 3D-värld. Ett exempel på en användning av fysik i *Portal* är då spelaren utmanas till att ta sig från en plattform till en annan där plattformarna är separerade av en fallgrop. Spelaren kan lösa problemet bland annat genom att placera en portal nere i fallgropen och en portal längs väggen till den första plattformen. Då spelaren faller ned i den första portalen accelererar spelaren tack vare gravitation och sedan skickas spelaren ut ur den andra portalen. På sådant sätt “flyger” spelaren över till den andra plattformen. Denna typ av fysik beskrivs som mekanik. I *The Talos Principle* kan man finna ett exempel på en användning av fysik inom ett annat fysikområde, nämligen optik. Spelaren utmanas att ta sig förbi en låst dörr. Dörren låser upp sig om den blir träffad av en laser. Spelaren har inte tillgång till en bärbar laser utan tvingas istället att använda sig av en utplacerad laser. Spelaren måste rikta om strålen med hjälp av kristallen som sitter på ett stativ som spelaren bär på. Strålen kan böjas av med hjälp av kristallen och detta fenomen beskrivs tydligt inom optiken. Dock görs detta enkelt för spelaren, då strålen träffar rätt så länge det finns en korrekt

länk mellan lasern och måltavlan, så spelaren behöver inte tänka på att vrida kristallen i exakt rätt riktning.

Två andra exempel på pusselspel som är fysikbaserade är *Monster Strike*[19] och *Peggle*[20] vars fysik baserar sig på kollisioner mellan olika kroppar. Båda spelen har en liknande spelmekanik som bygger på hantering av elastiska stötar. Elastiska stötar är en del av fysikområdets mekanik och förklaras med impulser. I *Peggle* utmanas spelaren till att beskjuta ett minfält med ett begränsat antal kulor. Antalet minor överstiger kraftigt mängden ammunition som spelaren har tillgängligt. Spelaren behöver därmed träffa flera minor med samma kula. Detta kan bland annat lösas genom att med rätt utgångsvinkel och utgångshastighet studsas kulan mot en mina så att den studsar till en annan mina och så upprepas processen till dess att kulan har åkt utanför banan. Kulan startar ovanför minfältet och påverkas konstant av en gravitation som drar kulan nedåt. Då minorna efter en elastisk stöt behåller sin statistiska position betyder detta enligt konserveringen av rörelsemängd att kulan studsar iväg med en liknande fart med endast en marginell skillnad. *Monster Strike* fungerar på ett liknande sätt. Dock är minfältet där fiender, och kulan som skjuts iväg är spelarens karaktär som ständigt påverkas av en friktion från markytan istället för av gravitation. Spelaren har här möjlighet att tillföra en impuls hos sin karaktär som senare kolliderar med statistiska fiender som med bara en marginell skillnad bibehåller sin statistiska position efter kollision.

2.2.1 Slutsats från observationerna

Resultaten från dokumentationen av pusselspel visar enligt våra observationer tydligt på att fysiken har en relevans och ibland även en central roll i speldesign och gameplay. Dock tycks klassisk mekanik dominera användningen av fysik men till exempel *The Talos Principle* är ett undantag där andra fysikområden, i detta fall optik, är närvarande. Slutsatsen från våra observationer blir att för pusselspel har fysiken troligtvis en betydelsefull påverkan på gameplay och speldesign som tros leda till en positiv upplevelse för spelaren. Fysiken visar sig även ibland ha en avgörande roll. Detta gäller för de pusselspel som mer eller mindre efterliknar ingenjörskonst som exempelvis *Bridge Builder*. För dessa spel blir spelupplevelsen troligen försämrad om fysikmotorn inte kan presentera en realistisk fysik. Spelaren tvingas ofta förlita sig på sin intuition och anpassa sig efter ett misslyckande. Detta leder även till att spelaren får en både kvalitativ och kvantitativ förståelse av fysiken.

2.3 First Person Shooter (FPS)

Fysikens relevans inom *First Person Shooter*-spel har enligt våra observationer varierat kraftigt. I början på 90-talet tycks FPS-spel inte haft något behov av en realistisk fysik. FPS-spelen tros ha haft som uppgift att leverera en actionfylld dramatik som involverade bland annat granatkastning och vapenhantering i ett förstahandsperspektiv istället för att förmedla en verklighetstrogen spelmiljö. *Wolfenstein 3D*[21] anses vara det första tillverkade First Person Shooter-spelet och lanserades år 1992. *Wolfenstein 3D* saknade överlag realistisk fysik som exempelvis rekyll hos vapen. Rekyll kallas den kraft som uppkommer då man avfyrar en kula från ett gevär, och förklaras med Newtons tredje lag[22]. Kraften fungerar som en impuls som slår tillbaka mot individen som avfyrade kulan, i en motsatt riktning längs vapnets eldrör.

Allt eftersom utvecklades FPS-genren och olika varianter av FPS-spel, som exempelvis *Arcade Shooter*, *Tactical Shooter* och *Puzzle Shooter*, uppkom. *Portal*[17] är ett exempel på en *Puzzle Shooter* som enligt våra observationer tycks vara helt beroende av en korrekt realistisk fysik för att dess spelmekanik ska fungera. För just FPS-genren *Puzzle Shooter* verkar fysiken ha en stor relevans, då många av de pussel som spelen erbjuder bygger på fysikaliska delmoment som till exempel att placera ett tungt objekt på en plattform. Det observerades också att på senare tid har fysiken spelat en stor roll i att skapa en realistisk spelmiljö för spelaren i många olika typer av *First Person Shooters*.

Ett exempel på en kontrast mellan realistisk fysik och gameplay är *Halo: Combat Evolved*[23] och *Battlefield 4*[24] som båda är av typen *Arcade Shooter*. I *Halo: Combat Evolved* tycks fysiken enligt våra observationer inte ha en avgörande roll i gameplay då spelet inte försöker skapa en realistisk miljö som spelaren ska kunna relatera till, utan strävar efter en ultramodern miljö där till exempel vapen inte påverkas negativt av rekyll. Raka motsatsen observerades för *Battlefield 4* som enligt spelutvecklarna strävar efter att skapa en realistisk gameplay som spelaren kan relatera

till i en modern krigföringsmiljö[25]. Exempelvis observerades denna kontrast tydligt i hur spelen hanterar rekyl. *Halo: Combat Evolved* hanterar rekyl endast som en animation hos spelets vapen, som inte påverkar spelarens precision eller vapenhantering. Detta gör det alltså enklare för spelaren att hantera vapnen. Tvärtom gäller *Battlefield 4* som visar på en relativt realistisk rekyl. Med relativt menas att relativt andra FPS-spel existerar ett försök till en realistisk rekyl även om den är kraftigt dämpad jämfört med en verklig rekyl. Om spelaren avlossar ett helautomatiskt gevär påverkas spelaren av en rekyl som drar gevärspipan uppåt och på så sätt påverkar spelarens precision och vapenhantering negativt. Dock kan spelaren hantera rekyl om den tar hänsyn till denna avböjning, genom att exempelvis börja sikta nedåt för att kompensera för den kraft som drar gevärspipan uppåt. I verkligheten avstår man oftast från att avfyra helautomatiska gevär utan ett stativ för att rekyl gör det nästan omöjligt att skjuta med precision på långa avstånd[26]. Rekyl kan anses vara ett stort hinder vid vapenhantering över långa avstånd där precision är en viktig förutsättning. Därmed kan en helt fysikaliskt korrekt rekyl påverka gameplay negativt. En realistisk men approximativ rekyl kan istället bidra till en blandning av realism och positivt gameplay.

Counter Strike: Global Offensive (CS:GO)[27] är ett spel som är lite av en blandning av de mer taktiska *Tactical Shooters* och de mer hektiska *Arcade Shooters*, som observerades till att bygga på en delvis realistisk fysik. Vid granatkastning påverkas granaten av en gravitationskraft som kan beskrivas av ett newtonskt kraftsystem. Med ett newtonskt kraftsystem menas klassisk mekanik som grundas i Newtons lagar[22]. Dock observerades att spelmotorn i *CS:GO* inte tar hänsyn till vindar och väder. Konsekvensen av detta är att kastparabeln för granaten befinner sig i ett idealt tillstånd där ingen dynamisk avböjning existerar. En mer korrekt fysik skulle ta hänsyn till eventuella vindar. Dock kan detta påverka gameplay negativt då spelaren inte längre har en fullständig kontroll över hur kastparabeln ser ut. Därmed kan en delvis realistisk kastparabel utan en dynamisk avböjning påverka gameplay positivt och bidra till en känsla av realism.

CS:GO, *Battlefield 4* och *Call of Duty 4: Modern Warfare (CoD4)*[28] är tre olika *Arcade Shooter*-spel som alla utspelar sig i en spelmiljö som efterliknar modern krigsföring. Dock observerades en markant skillnad mellan spelen i hur man väljer att avspejla en realistisk fysik. Exempelvis har alla tre spelen olika sätt att representera rekyl hos vapen. Enligt våra observationer har *CS:GO* den mest korrekta representationen av rekyl då rekylen är väldigt stark och gör det näst intill omöjligt för spelaren att skjuta på långa avstånd med precision. Dock är rekylen inte slumpad som i verkligheten. Rekyl är ett kaotiskt system som i verkligheten nästan aldrig upprepar sig identiskt utan har små avböjningar. I *CS:GO* är rekyl fix i en så kallad *Spray pattern* istället för slumpad[29] medan i *Battlefield 4* och *CoD4* är den mer slumpad. *CoD4* och *Battlefield 4* har däremot en markant "snällare" rekyl som inte straffar spelarens vapenhantering lika hårt. En intressant observation som kan bidra till denna analys är då *Valve*, ägare av *CS:GO*, valde att implementera en slumpad rekyl[30]. Efter återkoppling från användare valde *Valve* att upphäva implementationen och återgå till ett fixt spray pattern då användarna ansåg att denna implementation påverkade gameplay negativt. En fysikaliskt korrekt rekyl bidrog här till en försämrad spelupplevelse.

En jämförelse mellan *CoD4* och *Battlefield 4* visar enligt våra observationer att båda spelen har en väldigt förenklad men delvis realistisk rekyl som inte är fullständigt fysikaliskt korrekt. Dock har enligt våra observationer *CoD4* en snällare rekyl jämfört med *Battlefield 4*. Detta troligtvis på grund av att *Battlefield 4* strävar efter en mer realistisk spelmiljö överlag jämfört med *CoD4* som troligtvis strävar efter en "fast paced" och actionfylld spelmiljö där förenklingar bidrar till en positiv men förenklad gameplay. *Dice*, spelutvecklarna bakom *Battlefield 4*, skriver att *Battlefield 4* strävar efter att förmedla en verklighetstrogen och rolig spelmiljö samt spelmekanik[25] vilket troligtvis är en anledning till varför *Battlefield 4* observerades ha en mer realistisk fysik jämfört med *CoD4*. Bland dessa förenklingar är vapenhantering där vapenrekyl implementeras till att vara minimal men delvis realistisk. *Battlefield 4* har däremot istället en mer realistisk spelmiljö överlag, med exempelvis dynamiska väderförhållanden som påverkar friktion för fordon, omfattande explosioner där byggnader kan ta skador eller helt förstöras samt aerodynamiska effekter från roderblad hos helikoptrar. De aerodynamiska effekterna hos rotorbladen kan observeras då man flyger en helikopter eller ett jetflygplan tillräckligt nära ett träd som påverkas av en kraftig vind. Även andra dynamiska effekter som vågor längs vattenytan existerar och kan påverka spelupplevelsen om spelaren exempelvis simmar eller åker med en båt.

2.3.1 Slutsats från observationerna

Fysikens relevans inom *First Person Shooters* visade sig enligt våra observationer inte vara entydig. Det observerades att spelen har olika mängd realistisk fysik som tros påverka gameplay både positivt och negativt. I allmänhet observerade vi att användningen av realistisk fysik ökat inom FPS-spel med åren då fler spelutvecklare väljer att skapa realistiska *Arcade Shooters* och *Tactical Shooters*. Exempel där realistisk fysik har avgörande roll kunde identifieras med exempel som *Portal* och uppföljaren *Portal 2*. *Wolfenstein 3D*, som sägs vara det första FPS-spelet, hade ingen realistisk fysik utan fokuserade på en actionfylld dramatik med ett förstahandsperspektiv.

Slutsatsen blir att fysiken varierar väldigt kraftigt mellan olika typer av *First Person Shooter*-spel. Det finns tydliga exempel där en realistisk fysik är avgörande för gameplay och exempel där en avsaknad av realistisk fysik inte påverkar gameplay negativt.

2.4 Racingspel

Spelgenren racingspel har tre signifikanta subgenrer vid namn *Arcade-style racers*, *Racing simulators* och *Kart racing games*. Alla tre subgenrer tycks ha haft en markant utvecklingsprocess överlag, som involverar en förbättring i hanteringen av fysik. Tidigt på 70-talet lanserades de första racing-spelen och bland dessa var *Space Race*[31] som lanserades år 1973 för spelkonsolen *arkad*. *Space Race* tillhörde subgenren *Arcade-style racers* och hade en väldigt enkel spelmekanik som näst intill helt saknade fysikaliska fenomen. Troligtvis fokuserade *Space Race* på att leverera en mer dynamisk konfliktkänsla hos spelaren istället för att försöka leverera en mer realistiskt baserad spelmiljö. Racingspel karaktäriseras ofta av just sin konfliktbaserade och tävlingsbaserade dynamik.

Spelserierna *Grand Theft Auto (GTA)*[32] och *Need for Speed (NFS)*[33] tillhör båda subgenren *Racing simulators* och tycks ha haft en tydlig utvecklingsprocess med varje iteration. Vi observerade att för varje iteration fanns någon form av utveckling av grafiken men också en tydlig förbättring inom användningen av fysik. Exempelvis kunde detta observeras genom att göra en jämförelse mellan det första och det senaste tillägget i spelserien *Need for Speed*. *Road & Track Presents: The Need for Speed*[34] släpptes år 1994 och är det första spelet i serien och *Need for Speed: Payback*[35], spelseriens senaste tillägg, släpptes år 2017. Vid jämförelse mellan dessa två kan man exempelvis observera i det senaste tillägget att bilarna nu tar skada efter en kollision, hur stötar mellan bilar blivit allt mer inelastiska och hur väderförhållanden nu påverkar friktionen mellan däck och markytan. Att bilen tar skada innebär att den efter kollision inte bibehåller sin tidigare fysikaliska kropp utan eventuella deformationer kan tillkomma samt kan även bilrutor krossas och strålkastare kan gå sönder. Argumentet att skifta från elastiska stötar till mer inelastiska stötar mellan bilar vid kollision grundas troligtvis i att man vill skapa en mer realistisk representation av bilrockor, då verkliga bilrockor är näst intill helt inelastiska. Samma typ av funktionalitet observeras även i spelserien *Grand Theft Auto*. Det senaste tillägget i spelserien *Grand Theft Auto* är *Grand Theft Auto V*[36] och lanserades år 2013. *Grand Theft Auto V* hanterar även, utöver det som nämndes tidigare för *Need for Speed: Payback*, kraftiga inbromsningar som i värsta fall resulterar i att föraren åker ut genom bilrutan. Denna typ av utveckling och förbättring inom användningen av fysik grundas troligtvis på att man vill leverera en allt mer verklig representation av racing.

Förutom dessa exempel på racingspel som använder realistisk fysik finns spel med mer surrealistisk fysik. Ett tydligt exempel på detta är *Super Mario Kart*[37] som lanserades år 1992 och tillhör subgenren *Kart racing games*. *Super Mario Kart* har en speldesign som bygger på fantasy, det vill säga att det finns inslag av exempelvis magi och påhittade varelser. Fysiken i spelet grundas i klassisk mekanik. Surrealismen tillkommer då man exempelvis inkluderar *power ups*, hanterar kollisioner som elastiska stötar och försummar skador på fordonet efter en eventuell kollision. Exempel på *power ups* är svampar som kan, utan att skada fordonet, accelerera det långt över dess egna kapacitet. Skador som skulle kunna uppkomma är dock exempelvis skador på drivaxeln om bildäcken roterar med ett allt för snabbt vridmoment. Med "försummar skador på fordonet efter kollision" menas att fordonet efter kollisionen bibehåller sin tidigare fysikaliska kropp istället för att kroppen får eventuell deformation.

2.4.1 Slutsats från observationerna

Enligt våra observationer noterades en utvecklingsprocess inom racingspel som omfattade en utveckling för att skapa mer realistisk fysik. Överlag visade sig racingspel inte nödvändigtvis vara beroende av en realistisk fysik. Dock gav en mer realistisk hantering av fysik ett troligtvis bättre resultat för racingspel inom subgenren *Racing simulators* där man strävar efter en verklig representation av racing. Förbättringar inom realistisk fysik omfattar exempelvis mer inelastiska stötar vid kollision och att fordon tar skada vid en kraftig kollision och därmed inte bibehåller sin fysikaliska form.

2.5 Plattformsspel

Plattformsspel är en bred spelgenre som karaktäriseras av sitt sätt att hantera spelplaner som en uppbyggnad av plattformar, statiska, dynamiska samt rullande spelramar och spelnivåer som är ändliga eller oändliga. Spelgenren är uppbyggd av subgenrer som bland annat *Puzzle-platform game*, *Run-and-gun platform game*, *Comical action game* och *Endless running game*[38]. Våra observationer är att plattformsspel tenderar att domineras av fysikområdet klassisk mekanik. Eftersom klassisk mekanik är läran om hur kroppar påverkas av krafter samt hur kroppar rör sig[39], är det rimligt att spelmekaniken inom plattformsspel är beroende av rörelse- och kraftssystem för att leverera sin speldesign. Ett exempel kan vara tyngdaccelerationen som troligtvis fungerar bra med en spelplan uppbyggd av plattformar och som kan ha en positiv effekt på gameplay. Denna hypotes grundas på de observationer som vi har gjort inom plattformsspel.

Super Mario Bros.[40] lanserades år 1985 av spelföretaget *Nintendo* och sålde globalt över 40 miljoner kopior[41]. Spelmekaniken i *Super Mario Bros.* involverar bland annat att man ska hoppa över hinder, ta sig förbi eller besegra motståndare genom att hoppa på dem och hoppa mellan olika plattformar för att klara av spelets olika nivåer. Fysiken i spelet baseras på klassisk mekanik med exempelvis elastiska stötar och en gravitationskraft som drar spelaren nedåt mot markytan. *Super Mario Bros.* verkar inte ha fysik som central del för att leverera en bra spelupplevelse. Dock använder *Super Mario Bros.* sig delvis av fysik för att skapa en utmaning för spelaren där det huvudsakliga exemplet grundas i att spelaren tvingas att hoppa mellan plattformar för att avancera och där gravitationskraften används som en problematik för spelaren.

Spelkritikern *IGN* påstår att *Super Mario Bros.* är ett av tidernas bästa spel[42] av anledningar som bland annat dess positiva inverkan på spelkollapsen 1983[43] och dess unika speldesign och enkla spelkontroller. Möjligtvis kan spelets stora succé haft ett starkt inflytande på spelgenren och därmed ha bidragit till att många av de spel som vi har observerat inom spelgenren bygger på en liknande spelmekanik och speldesign. Därmed skulle en hypotes kunna vara att *Super Mario Bros.* troligen starka inflytande lett till att många spelutvecklare inte har experimenterat med mer exotisk fysik.

Spelföretaget *Nintendo* lanserade år 1996 sitt första 3D-baserade Mario-spel vid namn *Super Mario 64*[44]. *Super Mario 64* omfattade nu en helt interaktiv 3D-värld. I en recension från *Highsnobiety*[45], en livsstil blogg med sponsrat innehåll, skriver man att *Super Mario 64* hade en revolutionerade kamera-användning för sin tid som underlättade för spelaren när den ska navigera runt. *Highsnobiety* skriver även att *Super Mario 64* kan anses vara bland de viktigaste spelen inom spelbranschens historia och var väldigt betydelsefull och banbrytande för 3D-plattformsspel. Trots sin goda kritik omfattar *Super Mario 64* en delvis felaktig fysik. Det främsta exemplet som observerades inträffar då spelaren vid ett tillfälle ska kasta iväg spelets huvudsakliga antagonist, *Bowser*. Kastet efterliknar ett verkligt släggkast där spelaren håller i *Bowser* och sedan skapar ett vridmoment för att sedan vid rätt rotationshastighet och riktning kasta iväg antagonisten. Den felaktiga fysiken uppstår då spelaren precis kastar iväg *Bowser*, där det observerades att det förekommer en icke-kontinuitet. Spelarens karaktär, *Mario*, bromsas in omedelbart istället för att långsamt sakta in. En felaktig användning av fysik kan leda till att man skapar en felaktig förståelse hos spelaren, vilket vi anser kan vara problematiskt ur ett etiskt perspektiv.

Ovan nämndes att plattformsspel bland annat kan karaktäriseras av att ha oändliga spelnivåer. Denna typ av funktionalitet förekommer för spel inom subgenren *Endless running game*. Exempelvis klassas *Temple Run (2011)*[46] som ett *Endless running game*. *Temple Run* har endast en spelnivå och har en spelmekanik som centreras kring att spelaren ska springa så långt som möjligt. Spelaren stöter dock på olika typer av hinder längs vägen där spelaren bland annat kan hoppa

över eller huka sig förbi alla hinder. Spelets fysik centreras även här i klassisk mekanik. Dock använder spelet sig av en väldigt svag tyngdacceleration vilket leder till att kastparablen som spelarens karaktär följer vid ett hopp blir ovanligt lång. Detta val grundades troligtvis på att man som spelutvecklare ville forma kastparabeln efter spelets egna hinder för att balansera gameplay istället för att förmedla någon form av realism. Spelet saknar även någon form av friktion eller inbromsning när spelarens karaktär glider under ett hinder. Spelarens karaktär har alltså ett starkt fotfäste när karaktären springer men har samtidigt ingen friktion när karaktären glider längs marken, vilket antyder att det har gjorts ett inkonsekvent val av friktion mellan karaktärens kropp och markytan. *Temple run* är ett *Endless running game* som enligt vår observation troligtvis inte baseras på realistisk fysik utan istället tycks forma sin fysik efter bekvämligheten för spelaren.

Braid[47] är ett pusselbaserat plattformsspel som lanserades mellan år 2015-2016 och tillhör subgenren *Puzzle-plattform game*. Spelets spelmekanik observerades främst bygga på klassisk mekanik men har även inkluderat ett mer unikt inslag av tidsmanipulation. Med tidsmanipulation menas att spelaren har möjlighet att resa tillbaka i tiden för att ändra på sina handlingar. Denna typ av "fysik" kan knappast benämnas som realistisk men en hypotes inom speciell relativitetsteori visar på en möjlighet till tidsmanipulation. Exempelvis skulle detta inträffa för information som färdas snabbare än ljusets hastighet, vilket idag anses vara omöjligt med undantag för mindre hypoteser. Dock observerades att *Braid* har flera fysikbaserade pusselmoment som grundas på klassisk mekanik. Spelmekniken hos *Braid* tenderar alltså att vara beroende av en realistisk fysik för att kunna leverera sin speldesign. Pusselspel observerades generellt i denna undersökning vara mer beroende av en realistisk fysik än andra spelgenrer, vilket konstaterades i avsnitt 2.2.

2.5.1 Slutsats från observationerna

Enligt våra observationer tenderar plattformsspel att domineras av klassisk mekanik. En hypotes skulle vara att det framgångsrika *Super Mario Bros.* kan ha haft ett starkt inflytande på marknaden och därmed inspirerat många spelutvecklare att använda liknande spelmekanik. Små exempel på användning av exotisk fysik observerades. Våra observationer visade att plattformsspel överlag inte nödvändigtvis är beroende av realistisk fysik men att vissa fysikaliska fenomen som exempelvis gravitation är nödvändiga för att speldesignen med plattformar ska fungera.

2.6 Action-Adventure

Det första existerande spelet inom denna genre anses vara *Adventure* (1979-1980) som släpptes till spelkonsolen *Atari 2600*[48]. Grafiken var simpel och den spelbara karaktären var inte mer än en kvadrat som kunde flyttas omkring. Spelaren kan använda sig av olika objekt för att lösa pussel och besegra fiender. Ett av dessa objekt är en magnet som används för att attrahera nycklar placerade på ställen som spelaren annars inte kommer åt. Denna användning av fysik är dock den enda vi kan hitta i detta spel, och den är enbart kvalitativ och inte helt realistisk.

Castlevania: Harmony of despair (2010) är ett 2D-action-adventure-spel tillhörande subgenren *platform-adventure* [49]. Trots att det är ett relativt modernt spel finns det inte mycket som tyder på att realistisk fysik har använts. Hinder i form av pendlar finns, men dessa är helt opåverkade av fysikaliska fenomen som luftmotstånd eller kollision med objekt eller karaktärer. En av de spelbara karaktärerna har en förmåga som tillåter denne att med hjälp av magneter suga sig fast i olika metallobjekt för att ta sig fram till vissa ställen, vilket är en kvalitativ men inkorrekt implementation av elektromagnetism. Utöver dessa tillämpningar samt en simulering av gravitation som är väldigt orealistisk, observerades inte spelet använda sig av fysikaliska fenomen alls.

Ett annat spel som också tillhör *platform-adventure*-genren är *Mirror's Edge* (2008) [50]. Spelets grafik är i 3D och använder sig av realistisk fysik för bland annat visuella effekter. Exempel på dessa är hur glas betar sig när ett fönster krossas, hur vind påverkar löv i omgivningen samt hur olika plastdraperier påverkas av att någon springer igenom dem. Spelet går ut på att man med rörelser inspirerade av parkour tar sig mellan olika byggnader och gör framsteg i spelet. Målet tycks ha varit att eftersträva realism, dock med många undantag för att ge bättre gameplay. Exempelvis återhämtar sig spelaren snabbt efter fall från höga höjder vilket gör att man inte behöver tänka lika mycket på de fysiska skador som kan uppkomma i samband med de olika hoppen. Vissa rörelser som karaktären kan utföra är inte heller möjliga att göra i verkligheten, exempelvis hur denne springer och hoppar mellan väggar. Om karaktären bär på ett vapen så blir den även långsammare; tyngre

vapen leder till mer begränsad rörelse vilket är fysikaliskt korrekt. Den huvudsakliga fysiken i det här spelet, utöver den visuella biten, visar sig alltså vara hur spelaren rör sig och tar skada vid fall och kollisioner, enligt våra observationer. Dessa bedöms vara någorlunda realistiska.

Inga av de ovannämnda titlarna använder sig av fysikaliska fenomen som en central del av gameplay, vilket misstänks vara en representativ bild av titlar inom genren generellt från samma tidsperioder. På senare år har däremot fysik använts som central del av gameplay även för spel i denne genre. *Zelda: Breath of the wild* (2017)[51] hyllades snabbt av både kritiker och spelare och blev utnämnd till årets bästa spel 2017 av *D.I.C.E awards*[52][53]. Spelet har en kraftig fysikmotor som tillåter spelaren att experimentera med olika typer av fysik för att lösa pussel, besegra fiender och ta sig framåt i spelet. Våra observationer visade att de har använt sig av både icke-realistisk fysik för att passa spelets grafikdesign samt mer verklighetstrogen fysik; målet tycks ha varit att skapa en fysikmotor med bra balans mellan realism och positiv påverkan på gameplay. Exempel på implementerad inkorrekt fysik i spelet är ragdollfysik för animeringar då en varelse dör, dynamisk väderfysik samt användandet av magneter för att flytta föremål. Bland användandet av korrekt fysik finns "gungbräde-fysik" då exempelvis en stor sten släpps på en gungbräda med spelaren stående på andra änden, hur vind påverkar objekt, samt spridning av eld.

2.6.1 Slutsats från observationerna

Resultaten från denna analys antyder att fysiken i action-adventure-genren oftast varit där för visuella effekter, snarare än för en praktisk påverkan på spelmekaniken. Att fysiken har en central del av spelens gameplay inom denna genre verkar vara sällan förekommande, och de undantag har varit då man infört fysikpussel i stil med de som nämnts under rubriken pusselspel ovan. Införandet av fysik som ett vanligt stridselement är en relativt ny innovation och endast tiden kan avgöra om detta kan komma att bli en standard för framtida spel inom action-adventure-genren.

2.7 Olika fysikområden i videospel

I följande delkapitel presenteras olika fysikområden och hur de används i videospel.

2.7.1 Mekanik dominerar spelbranschen

Enligt våra observationer dominerar klassisk mekanik spelmarknaden markant. Näst intill alla fysikbaserade videospel tycks ha större eller mindre delmoment baserade på mekanik. Den stora användningen av klassisk mekanik sträcker sig exempelvis över spelgenrer som *First Person Shooters*, *pusselspel*, *plattformsspel* och *racingspel*. Klassisk mekanik kan kort formuleras till läran om hur kroppar förflyttar sig och påverkas av krafter[39]. Inom klassisk mekanik beskrivs exempelvis gravitation, elastiska och inelastiska stötar, rörelsemängd, rörelsemängdsmoment och Newtons lagar. På grund av sin starka dominans inom spelbranschen har vi valt att inte göra en detaljerad analys av mekanikens inflytande. Detta då en detaljerad analys av mekanikens dominans inte bidrar till vårt huvudsakliga syfte, utan istället fokuserar vi på att utforska andra mer exotiska användningar av fysik.

2.7.2 Optik används som pussel inom videospel

Optiska fenomen förekommer till exempel i videospel för att tillföra en typ av pusselmoment, ofta genom att spelaren ska vrida en spegel till rätt vinkel för att en ljusstråle ska träffa en måltavla. Ibland är detta huvudpoängen i spelet, medan det i andra fall är en mindre del, bland många andra. *Fireboy and Watergirl 2 In the Light Temple* [54] använder ljusguidning tillsammans med olika knappar och spakar som behöver tryckas på i rätt ordning för att banorna ska klaras. Spelaren styr ljuset genom att vrida på speglar eller knuffa reflekterande block till rätt positioner, vilka båda begränsas av banans utformning. *Lego Pirates of the Caribbean*[55] använder speglar vid enstaka tillfällen för att leda ljus rätt så att till exempel dörrar ska öppnas [56], vilket görs genom att spelaren vrider speglarna i diskreta steg. I båda dessa spel har spelaren alltså begränsat inflytande över hur strålen kan styras. Det är ännu tydligare i *The Talos Principle*[18], som använder stråloptik i pusselmoment där spelaren endast länkar en ljuskälla till en slutpunkt med hjälp av en kristall på ett stativ som den kan hålla i. Reflektionerna sker automatiskt med korrekt vinkel inuti kristallen om bara spelaren har skapat en korrekt fungerande länk mellan ljuskällan och slutpunkten.

När det gäller spel där ljusledning är huvuddelen av spelet tycks det vara vanligast att spelaren ska placera ut speglar och vrida dem rätt för att belysa olika saker, till exempel i spelet *Prisme* [57]. Spegelns verkar alltså vara de vanligaste optiska komponenterna, men även till exempel linser och prismor används i *Optika* [58]. Det gör att det spelet troligen i större utsträckning kan användas för att spelaren ska förstå och lära sig om optik, medan spelen som bara använder speglar ofta inte kräver någon större fysikförståelse.

Många optikbaserade spel visar ljuset som en rak stråle, som inte sprids någonting, vilket inte är fysikaliskt korrekt, speciellt inte då ljuskällan i många fall tycks vara en vanlig lampa. Även då koherenta källor som lasrar används visas oftast hela ljusstrålen som ett streck istället för att man bara skulle se var den träffar, vilket inte är helt korrekt men som underlättar för spelaren. Detta är alltså tydliga exempel på när fysiken har kompromissats med för att göra bra spel.

2.7.3 Aerodynamik bidrar till en bättre realism på detaljnivå

Aerodynamik är läran om hur gaser rör sig och även hur föremål i rörelse påverkas av den gasen den innesluts av[8]. Aerodynamik är en mindre gren inom fysiken som tillhör strömningsmekaniken, men kan visa sig ha en betydelsefull inverkan på gameplay som baserar sig på realism. I dokumentationen av existerande spel hittades två distinkta fall där aerodynamik hade en inverkan på gameplay. Dessa två exempel var *FIFA 14* och *Battlefield 4*. I *FIFA 14* försöker man skapa en realistisk representation av verklig fotboll medan i *Battlefield 4* försöker man istället skapa en verklig representation av modern krigföring. Det är två väldigt åtskilda spelmiljöer men som båda försöker skapa en realistisk och verklig bild.

I *FIFA 14* uppstår aerodynamiken hos fotbollen där den påverkas av ett luftmotstånd när den färdas över fotbollsplanen. I jämförelse med sina företrädare inkluderade *FIFA 14* en mer korrekt luftmotståndskoefficient vilket ledde till en mer realistisk inbromsning hos fotbollen. I *FIFA 14* har man också ett dynamiskt vädersystem som skiftar från match till match och som påverkar de aerodynamiska effekterna. Utöver de aerodynamiska effekterna påverkas även friktionen mellan fotbollen och markytan. Denna mer detaljerade användning av fysik kan leda till en mer realistisk representation av verklig fotboll. *Financial Post* skriver i en recension från år 2013 att *FIFA 14* är "det mest realistiska sportbaserade spelet på marknaden"[59] och menar tydligt att den förbättrade och mer detaljerade användningen av fysik har konsekvent lett till en bättre realism. Dock hävdar *Financial Post* att förbättringen inom fysik och realism har lett till en sämre gameplay. Den alltmer realistiska representationen av verklig fotboll skapar en allt mer komplicerad manövrering för spelaren som även måste ta hänsyn till flera dynamiska händelser.

I *Battlefield 4* uppstår aerodynamiken hos rotorbladen i spelets helikoptrar. Om spelaren exempelvis svävar i tillräckligt låg höjd och tillräckligt nära ett träd blir konsekvensen att trädet påverkas av en kraft från luften. Denna kraft efterliknar en vanlig blåst och om trädet påverkas av en tillräckligt stark blåst faller trädet ned. Denna detaljerade användning av fysik har ingen större inverkan på gameplay utan kan mer ses som en förbättring av den realistiska faktorn. *Battlefield 4* strävar efter att leverera den mest realistiska, roliga och verkliga spelupplevelsen inom modern krigföring och detta tillägg av aerodynamik kan anses bidra positivt till denna vision[25].

Överlag är aerodynamik en mindre utforskad spelmekanik men förekommer ibland på detaljnivå. Aerodynamik är en mindre gren inom fysik som helhet och detta kan vara en anledning till att den inte används mer frekvent.

2.7.4 Brist på användning av kvantfysik i spelbranschen

Enligt våra observationer tenderar kvantfysik vara ett mindre utforskat fysikområde inom videospel. En trolig anledning kan vara att kvantfysik överlag inte är särskilt intuitivt i jämförelse med andra fysikområden som exempelvis klassisk mekanik och optik. Dock observerades några få exempel på mindre spel vars spelmekanik var baserade på kvantfysik.

Quantum Moves (2012)[60] är ett fysikbaserat simuleringsspel där spelaren utmanas till att förflytta en nedkyld atom utan att förlora information om dess position. Informationen om atomens position visas visuellt av en sannolikhetsfunktion som talar om hur sannolikt det är att atomen befinner sig inom ett visst intervall. Beroende på hur spelaren förflyttar atomen minskar sannolikheten att atomen befinner sig inom det området som spelaren förutspått. Enligt våra observationer är *Quantum Moves* troligtvis skapat i syfte att förmedla en kvalitativ och kvanti-

tativ förståelse av hur kvantfysik används. Dock anser vi att det behövs en viss förförståelse av kvantfysik innan man som spelare kan skapa en förståelse av kvantfysikaliska system.

Ett exempel på ett fysikbaserat pusselspel som bygger på kvantfysik är *The Quantum Game (2016)*[61]. *The Quantum Game* har en spelmekanik som grundas i att spelaren utmanas till att transportera en foton från sin foton-källa till en ljusdetektor via speglar och polarisationsfilter. Utmaningen ligger i att försöka undvika hinder i form av minor som exploderar vid direkt kontakt med en foton. Då fotonen träffar en glasskiva, som är ett av objekten spelaren placerar ut för att leda fotonen till ljusdetektorn, är det 50% chans att den reflekteras respektive transmitteras. Därmed kan spelaren inte vara helt säker på att dess lösning uteslutar möjliga fall där fotonen kolliderar med en mina. Denna typ av osäkerhet reflekterar väl hur mycket kvantfysiken grundas på sannolikhetslära. Efter våra observationer tros syftet bakom speldesignen grundas på att förmedla en kvalitativ och kvantitativ förståelse av kvantfysik.

Enligt våra observationer verkar det som att kvantfysik förekommer mer sällan i videospel på grund av att kvantfysik är ett svårt koncept att förmedla via gameplay. I de videospel som vi observerade där kvantfysik fanns noterades ett stort fokus i att försöka förmedla en kvalitativ och kvantitativ förståelse. Dock anser vi att det tenderade finnas en problematik inom just videospel med kvantfysikaliska fenomen som spelmekanik. Problematiken anses ligga i att spelaren behövde ha någon form av förförståelse av kvantfysik för att förstå spelkonceptet.

2.7.5 En avsaknad av relativistiska effekter i videospel

Enligt våra observationer förekommer relativistiska effekter väldigt sällan inom videospel trots att våra observationer antydde att klassisk mekanik mer eller mindre dominerade spelmarknaden. Klassisk mekanik fungerar bra för att beskriva hur föremål rör sig då de inte rör sig i närheten av ljusets hastighet. Då ett föremål rör sig närmare ljusets hastighet förekommer relativistiska effekter som beskrivs av speciell relativitetsteori. I en marknad som tycks domineras av klassisk mekanik visade det sig efter våra observationer att väldigt få spelutvecklare experimenterar med ljusets hastighet.

A Slower Speed Of Light (2012)[62] är ett förstapersonsspel vars spelmekanik bygger på att spelaren ska påverkas av relativistiska effekter. Spelaren har möjlighet att fritt utforska en 3D-värld där det huvudsakliga syftet är att försöka "fånga" ljuset. För att spelaren ska kunna hinna ikapp ljuset, som färdas med ljusets hastighet, har spelaren möjlighet att sakta ned ljuset genom att samla på sfärer som finns slumpmässigt utplacerade. När ljusets hastighet bromsas in börjar alltså spelarens egna hastighet att närma sig ljusets hastighet, vilket enligt speciell relativitetsteori skulle innebära att relativistiska fenomen skulle inträffa, som exempelvis tidsdilation. Enligt våra observationer tycks *A Slower Speed Of Light* försöka förmedla hur vardagliga fenomen som färger och längder påverkas då man rör sig tillräckligt nära ljusets hastighet vilket vi menar skulle leda till en kvalitativ förståelse av relativistiska effekter. Fysiken tycks alltså ha en direkt påverkan på gameplay.

Ett annat exempel på ett spel vars spelmekanik observerades grundas på relativistiska effekter var *Velocity Raptor (2011)*[63]. Spelaren introduceras först till en 2D-värld som enklast beskrivs med klassisk mekanik, men som senare inkluderar relativistiska effekter då spelets huvudsakliga antagonist sänker ljusets hastighet. När ljusets hastighet bromsas in påverkar detta spelarens observation av omgivningen som nu får tillskott av relativistiska effekter som exempelvis längdkontraktion, på grund av att spelarens egna hastighet närmar sig ljusets hastighet. Spelutvecklarna påstår själva[64] att spelets fysik är realistisk och korrekt, vilket stämmer överens med våra observationer. Fortsättningsvis skriver även spelutvecklarna att ett av syftena bakom spelet var att försöka förmedla en visuell representation av hur man möjligtvis skulle observera relativistiska effekter och hur dessa skulle påverka ens intuition. Eftersom spelet först inleds med en spelmekanik som baseras på klassisk mekanik som senare övergår till en som baseras på speciell relativitetsteori, visar detta på hur man som spelare troligtvis behöver ändra spelstil för att bättre anpassa sig till relativistiska effekter.

Sammanfattningsvis visar våra observationer att speciell relativitetsteori används ytterst småskaligt i mindre implementerade spel.

2.7.6 Spelmarknaden attraheras inte av elektromagnetism

Enligt våra observationer verkar det finnas en brist på användning av elektromagnetism inom spelbranschen. En möjlig hypotes är att elektromagnetism är mindre intuitivt än exempelvis mekanik, som enligt våra observationer dominerar marknaden. Elektromagnetism är läran om elektrostatiska och magnetostatiska fenomen som gemensamt beskrivs av Maxwells ekvationer[65]. Dock hittades några exempel på spel vars spelmekanik grundades i elektromagnetism.

The Electric Shocktopus (2015)[66] är ett plattformsspel vars spelmekanik bygger på tre olika krafter: elektrisk kraft, magnetisk kraft och gravitationskraft. Spelet utvecklades av *TestTubeGames* som är ett mindre spelföretag vars vision är att föra fram fysikaliska fenomen i videospel på ett "roligt" sätt[67]. *The Electric Shocktopus* observerades ha en realistisk representation av elektriska och magnetiska krafter och påstår själva att spelmekaniken grundas på korrekt fysik. Dock finns mindre approximationer och undantag inom energikonservering, där anledningen bakom avvikelserna från korrekt fysik påstås grundas i att förbättra gameplay[68]. I spelet förekommer det elektriska krafter vid utplacerade laddningar, där spelarens karaktär spontant kan byta laddning för att attraheras eller repelleras av den utplacerade laddningen. Denna spontana ändring i laddning förekommer vanligtvis inte i naturen. Elektriska krafter förekommer även i de elektriska fält som spelaren kan aktivera. Spelaren kan även åka igenom magnetiska fält där det kan förekomma magnetiska krafter om spelaren är laddad. Dessa krafter verkar enligt högerhandsregeln, vilket är en regel som förklarar åt vilket håll en kraft påverkar en laddning som rör sig i ett magnetfält.[69]

Enligt våra observationer tenderar magneter inom videospel att på ett missvisande sätt representera magnetism. Magneter används ibland i videospel, men i de fall vi observerat att magneter använts, exempelvis *Zelda: Breath of the Wild*, har magneternas inverkan överdrivits kraftigt.

Sammanfattningsvis är elektromagnetism en stor gren inom fysiken och, förutom användandet av kraftigt överdrivet effektiva magneter, verkar det enligt våra observationer finnas en brist på användningen av realistisk elektromagnetism i videospel.

3 Teori

Här presenteras teori som är relevant i projektet. Först introduceras begrepp från det så kallade MDA-ramverket för speldesign. Därefter presenteras den fysik som har använts i implementationerna, samt relevant teori för de idéer som vi utvecklade men som inte ledde till faktiska implementationer. Fysikområdet newtonsk mekanik är även inkluderat eftersom det frekvent används i fysikmotorer för spel.

3.1 MDA (Mechanics-Dynamics-Aesthetics)

För att analysera speldesign på ett mer systematiskt sätt än att beskriva videospel och dess underhållningsvärde med termer som "roliga" eller "inte roliga", finns ramverket MDA (Mechanics-Dynamics-Aesthetics), utvecklat av Robin Hunicke, Marc LeBlanc och Robert Zubek. Det delar in speldesign i tre delar: mekanik, dynamik och estetik. Mekaniken beskriver grunden i ett spel, som dess regler och underliggande datastrukturer. Dynamiken handlar om hur spelet fungerar när det spelas, till exempel huruvida spelare har incitament att samarbeta. Slutligen beskriver estetiken hur spelare reagerar på spelet, och vad de får ut av spelupplevelsen.[70]

Som spelutvecklare är det vanligt att man ser ett spel från mekaniken, via dynamiken, till estetiken, då det i första hand är mekaniken som man kontrollerar när man skapar ett spel, medan dynamiken och ännu mer estetiken ofta kan vara svåra att förutsäga. Som spelare å andra sidan är perspektivet det motsatta, där man framförallt upplever estetiken, till viss del dynamiken och till ännu mindre del mekaniken. För att skapa bra och omtyckta spel är det därför viktigt att som spelutvecklare även ha det perspektivet, så att man kan leva upp till spelares förväntan.

Det är alltså bra att börja med att titta på estetiken om man vill göra ett spel utifrån spelares efterfrågan. När man analyserar estetiken i ett spel försöker man i någon mån analysera hur roligt ett spel är, och på vilket sätt det är roligt. Spel kan upplevas som roliga av många olika anledningar, därför är det viktigt att ha andra begrepp för det, för att beskriva vad som gör spelet underhållande. Inom MDA brukar man framförallt prata om åtta olika typer av underhållning, som presenteras i Tabell 1.[71]

Tabell 1: Åtta typer av underhållning i spel

1.	Sensation	Ger upphov till känslor hos spelaren
2.	Fantasy	Fantasier som spelaren inte upplever i verkliga livet
3.	Narrative	Spelaren upplever en berättelse, berättad av spelet eller spelaren själv
4.	Challenge	Utmaningar som spelaren behöver klara för att ta sig vidare
5.	Fellowship	Skapar gemenskap mellan spelare
6.	Discovery	Spelaren får utforska exempelvis en virtuell värld eller olika sätt att spela
7.	Expression	Spelaren får utforska egenskaper hos sig själv, exempelvis genom skapande
8.	Submission	Tidsfördriv som inte kräver någon större tankekraft hos spelaren

De flesta spel innehåller flera av dessa aspekter, i olika proportioner, och genom att undersöka dessa kan man bättre förstå varför olika spel är populära i olika situationer. Utifrån detta kan sedan dynamiken planeras för att olika estetiker ska uppnås, och mekaniken utvecklas för att dynamiken och estetiken ska realiseras i ett faktiskt spel. På så vis kan alltså en spelutvecklare se spelets uppbyggnad ur en spelares perspektiv, och skapa spel utifrån spelares efterfrågan.

3.2 Newtonsk mekanik

Newtonsk mekanik är ett grundläggande fysikområde vars ekvationer beskriver hur kroppar rör sig genom rummet. Exempel på användningsområden i spel kan vara hur en kanonkula rör sig i en båge genom luften, hur tärningar studsar och rullar på bordet, eller hur en kedja rör sig i vinden. Det finns tre lagar i newtonsk mekanik, *Newtons lagar*:

1. En kropp befinner sig i vila eller rör sig med konstant hastighet om summan av krafterna på den är noll.
2. En kropps massa gånger dess acceleration är lika med kraften på den.
3. När en kropp påverkar en annan kropp med en kraft, påverkas den själv av en lika stor motriktad kraft.

I fysikmotorer följs inte alltid lagarna exakt. Ett exempel är att spelaren påverkas av sin tyngdkraft utan att kroppen som orsakar kraften, jorden, påverkas av en motsatt kraft. Det är en god approximation eftersom jordens massa är mycket större än spelarens. Den första och andra lagen tolkas i fysikmotorer som differentialekvationen

$$\frac{d^2\vec{x}}{dt^2} = \frac{1}{m}\vec{F}, \quad (1)$$

där \vec{x} är positionen för kroppen som simuleras. Kraften \vec{F} kan bero på nästan vad som helst, ofta på positionen för aktuell och andra kroppar. För tyngdkraften är den konstant med riktning nedåt.

3.3 Relativitetsteori

Den speciella och den allmänna relativitetsteorin är fysikaliska modeller för rörelse som ersätter newtonsk mekanik. Skillnaden är märkbar för höga hastigheter ($v \approx c$) och nära stora massor ($r \approx \frac{GM}{r^2}$). Den speciella relativitetsteorin utgår från att alla observerar samma hastighet för ljuset. Den allmänna relativitetsteorin uttrycker gravitation som en geometrisk egenskap hos rymden. För djupare läsning hänvisas till Rindler[72].

3.3.1 Speciell relativitetsteori

Speciell relativitetsteori förenar rum och tid. Om två händelser sker vid tid t_1 och t_2 samt läge \vec{x}_1 och \vec{x}_2 , skulle observatörer i Newtons värld vara överens om skalärerna $|\vec{x}_1 - \vec{x}_2|$ och $t_1 - t_2$, det vill säga att de är *invariant*. Den första invarianten är den euklidiska distansen, och den andra säger något om i vilken ordning händelserna sker. I Einsteins värld, däremot, finns det bara en invariant:

$$\Delta s^2 = c^2(t_1 - t_2)^2 - |\vec{x}_1 - \vec{x}_2|^2. \quad (2)$$

Om $\Delta s^2 < 0$ sägs avståndet vara *rumsligt*. Om två händelser har ett rumsligt avstånd, kan det visas att alla observatörer inte är överens om vilken som händer först.

Om vi definierar en vektor med fyra element som

$$x^\alpha = (ct, x, y, z), \quad (3)$$

kan ekvation (2) skrivas på differentialform som

$$ds^2 = \eta_{\alpha\beta} dx^\alpha dx^\beta = \sum_{\alpha=1}^4 \sum_{\beta=1}^4 \eta_{\alpha\beta} dx^\alpha dx^\beta, \quad (4)$$

där $\eta_{\alpha\beta} = \text{diag}(1, -1, -1, -1)$ är *Minkowskimetriken*, och vi har använt Einsteins summationskonvention för att dölja summatecknen.

3.3.2 Allmän relativitetsteori

Allmän relativitetsteori utgår från speciell relativitetsteori, men byter ut Minkowskimetriken $\eta_{\alpha\beta}$ (grekiska bokstavsindex går över en tids- och tre rumsdimensioner) mot en allmän metrik $g_{\alpha\beta}$ vars komponenter som fält är relaterade till massor i närheten. Fältekvationerna är icke-linjära och kan som regel inte lösas analytiskt. Som en kvalitativ regel blir tidskomponenterna lägre och rumskomponenterna högre nära massor. Långt ifrån massor är $g_{\alpha\beta} = \eta_{\alpha\beta}$.

Metriken $g_{\alpha\beta}$ påverkar kroppars rörelseekvationer. Detta stycke presenterar relationen utan härledning. Först behövs *Christoffelsymbolerna*, koefficienter som definieras utifrån $g_{\alpha\beta}$:

$$\Gamma_{\beta\gamma}^\alpha = \frac{1}{2} g^{\alpha\delta} \left(\frac{dg_{\gamma\delta}}{dx^\beta} + \frac{dg_{\delta\beta}}{dx^\gamma} - \frac{dg_{\beta\gamma}}{dx^\delta} \right), \quad (5)$$

där δ summeras över enligt Einsteins summationskonvention. Rörelseekvationen för en kropp blir då

$$\frac{d^2 x^\alpha}{ds^2} + \Gamma_{\beta\gamma}^\alpha \frac{dx^\beta}{ds} \frac{dx^\gamma}{ds} = A^\alpha, \quad (6)$$

där A^α är en yttre pålagd acceleration. Om kroppen bär med sig en fyrvektor V^α , till exempel rörelsemängdsmoment, utvecklas den i tiden enligt *Fermi-Walker transport*:

$$\frac{dV^\alpha}{ds} + \Gamma_{\beta\gamma}^\alpha \frac{dx^\beta}{ds} \frac{dV^\gamma}{ds} = g_{\beta\gamma} V^\gamma \left(\frac{dx^\beta}{ds} A^\alpha - A^\beta \frac{dx^\alpha}{ds} \right). \quad (7)$$

Med $V^\alpha = \frac{dx^\alpha}{ds}$ reduceras denna ekvation till rörelseekvationen.

Ett problem som kan lösas analytiskt är att beräkna $g_{\alpha\beta}$ för en punktmassa. Lösningen är den så kallade Schwarzschildmetriken. Med teckenkonvention $(+ - - -)$, $c = 1$ och polära koordinater $x = (t, r, \theta, \phi)$ skrivs den

$$ds^2 = g_{\alpha\beta} dx^\alpha dx^\beta = \sqrt{1 - \frac{2GM}{r}} dt^2 - \sqrt{1 - \frac{2GM}{r}}^{-1} dr^2 - r^2 d\theta^2 - r^2 \sin^2(\theta) d\phi^2. \quad (8)$$

Fysikaliskt beskriver Schwarzschildmetriken ett svart hål. Det finns en koordinatsingularitet vid ytan $r = 2GM$, den kallas *händelsehorisonten*, och har två intressanta egenskaper. Den första är att en kropp som befinner sig innanför inte kan ta sig ut hur mycket den än accelereras. Den andra är att ljus inte heller kan ta sig ut, vilket leder till att det svarta hålet är helt svart.

3.4 Strömningsmekanik

För kroppar som färdas i en vätska ger vätskan ett motstånd på kroppen enligt

$$\vec{F} = -\frac{1}{2} \rho C_D A |\vec{v}| \vec{v}, \quad (9)$$

där ρ är vätskans densitet, \vec{v} kroppens hastighet, C_D motståndskoefficienten och A tvärsnittsarean på kroppen [73]. Notera att kraften \vec{F} är motriktad hastigheten \vec{v} . C_D beror på kroppens geometri,

men även materialet och friktionen på ytan. Det beror även på Reynoldstalet, en dimensionslös storhet som beskriver flöde i en vätska och anger om flödet är laminärt (långt Reynoldstal) eller turbulent (høgt Reynoldstal). Ekvation (9) gäller för turbulent flöde. Reynoldstalet ges av

$$\text{Re} = \frac{\rho v L}{\mu}, \quad (10)$$

där ρ är vätskans densitet, v hastigheten relativt kroppen, L en typisk längd och μ vätskans dynamiska viskositet[74].

3.5 Kvantfysik

Kvantfysiken utökar antalet tillstånd som ett system kan befinna sig i, jämfört med den klassiska fysiken. Tag som exempel ett kastat mynt. I klassisk fysik kan det befinna sig i två tillstånd, krona (Kr) och klave (Kl). I kvantfysik däremot, kan det befinna sig i alla tillstånd på formen

$$c_1 |\text{Kr}\rangle + c_2 |\text{Kl}\rangle, \quad c_1, c_2 \in \mathbb{C}. \quad (11)$$

$|\text{Kr}\rangle$ och $|\text{Kl}\rangle$ är här basvektorer som motsvarar de klassiska tillstånden. De komplexa koefficienterna c_i motsvarar sannolikheterna att observera myntet i de klassiska tillstånden. Exempelvis är sannolikheten att observera Kr

$$\frac{|c_1|^2}{|c_1|^2 + |c_2|^2}. \quad (12)$$

För att uttrycka ett stort system i flera mindre system används tensorprodukten. Med mynt som exempel igen kan systemet med två kastade mynt betraktas. Deras tillstånd är

$$c_1 |\text{Kr}_1\rangle + c_2 |\text{Kl}_1\rangle \quad \text{och} \quad c_3 |\text{Kr}_2\rangle + c_4 |\text{Kl}_2\rangle. \quad (13)$$

Detta kan med tensorprodukt uttryckas som

$$(c_1 |\text{Kr}_1\rangle + c_2 |\text{Kl}_1\rangle) \otimes (c_3 |\text{Kr}_2\rangle + c_4 |\text{Kl}_2\rangle) = c_1 c_3 |\text{Kr}_1 \text{Kr}_2\rangle + c_1 c_4 |\text{Kr}_1 \text{Kl}_2\rangle + c_2 c_3 |\text{Kl}_1 \text{Kr}_2\rangle + c_2 c_4 |\text{Kl}_1 \text{Kl}_2\rangle, \quad (14)$$

där $|\text{Kr}_1 \text{Kr}_2\rangle = |\text{Kr}_1\rangle \otimes |\text{Kr}_2\rangle$ motsvarar det klassiska tillståndet där båda mynt visar Kr. Tensorprodukten är i allmänhet en produkt av två kvantfysikaliska tillstånd som ger ett tredje tillstånd. För två godtyckliga tillstånd

$$|X\rangle = \sum_i c_i |X_i\rangle \quad \text{och} \quad |Y\rangle = \sum_j d_j |Y_j\rangle \quad (15)$$

är tensorprodukten

$$|X\rangle \otimes |Y\rangle = \sum_i \sum_j c_i d_j |X_i \otimes Y_j\rangle, \quad (16)$$

där $X_i \otimes Y_j$ är det klassiska tillståndet där både X_i och Y_j är sanna. Den fysikaliska tolkningen av tensorprodukten är tillståndet som består av de två tillstånden utan interaktion mellan dem.

En observation kollapsar ett tillstånd, vilket innebär att termer försvinner från det. För enkla system, till exempel ett mynt, försvinner alla termer förutom den som motsvarar det observerade tillståndet. I allmänhet verkar en observation på ett tillstånd $|X\rangle$ enligt

$$|X\rangle \rightarrow \left(\sum_Y |Y\rangle \langle Y| \right) |X\rangle, \quad (17)$$

där Y är alla tillstånd som är kompatibla med observationen, och $\langle A|B\rangle$ i detta fall utvärderas till 1 om A och B är samma klassiska tillstånd, och 0 annars. Som ett minimalt exempel betraktas återigen systemet med två mynt. Om det första myntet observeras vara Kr multipliceras tillståndet från vänster med

$$|\text{Kr}_1 \text{Kr}_2\rangle \langle \text{Kr}_1 \text{Kr}_2| + |\text{Kr}_1 \text{Kl}_2\rangle \langle \text{Kr}_1 \text{Kl}_2|, \quad (18)$$

vilket ger tillståndet

$$c_1 c_3 |\text{Kr}_1 \text{Kr}_2\rangle + c_1 c_4 |\text{Kr}_1 \text{Kl}_2\rangle. \quad (19)$$

Tolkningen av det resulterande är att det första myntet helt säkert är Kr, men att det fortfarande finns en osäkerhet i vilken sida det andra myntet visar.

3.6 Lagranges ekvationer

Lagranges ekvationer är en alternativ formulering av mekaniken som gör det lättare att hantera generaliserade koordinater. Istället för att lösa kraftekvationerna med kartesiska koordinater x_i :

$$m\ddot{x}_i = F_i(x, \dot{x}, t), \quad (20)$$

införs den så kallade *Lagrangianen* $L = L(q, \dot{q}, t)$, med dimension energi och beroende på generaliserade koordinater. Ekvationerna som löses blir istället då (*Euler-*)*Lagranges ekvationer*:

$$\frac{d}{dt} \left(\frac{dL}{dq_i} \right) - \frac{dL}{dq_i} = 0. \quad (21)$$

Lagranges ekvationer kan härledas från *Hamiltons princip* (även kallad minsta verkans princip), som säger

$$\delta S = 0, \quad \text{där} \quad S = \int_{t_1}^{t_2} L dt. \quad (22)$$

Hamiltons princip ger en annan tolkning av fysiken. Givet ett systems initial- och sluttillstånd, följer systemet den utveckling där S , kallad *verkan*, är stationär, det vill säga $\delta S = 0$.

Om krafterna är konservativa, det vill säga kan skrivas på formen

$$F_i = -\frac{dV}{dx_i}, \quad (23)$$

för någon potential $V = V(x)$, blir Lagrangianen

$$L = T - V, \quad (24)$$

där T är den kinetiska energin.

Som ett enkelt exempel tas en endimensionell harmonisk oscillator. Kraftekvationen är

$$m\ddot{x} = F = -kx. \quad (25)$$

Motsvarande Lagrangian är med $q = x$

$$L_{\text{osc}} = \frac{1}{2}m\dot{q}^2 - \frac{1}{2}kq^2, \quad (26)$$

där de två termerna tydligt kan läsas av som kinetisk energi och lägesenergi.

3.7 Optik

Det finns två sätt att behandla optik, läran om ljuset, på. Stråloptik, även kallad geometrisk optik, antar att ljuset färdas som små partiklar, kallade *fotoner*. Fourieroptik, även kallad vågoptik, behandlar ljuset som elektromagnetiska vågor.

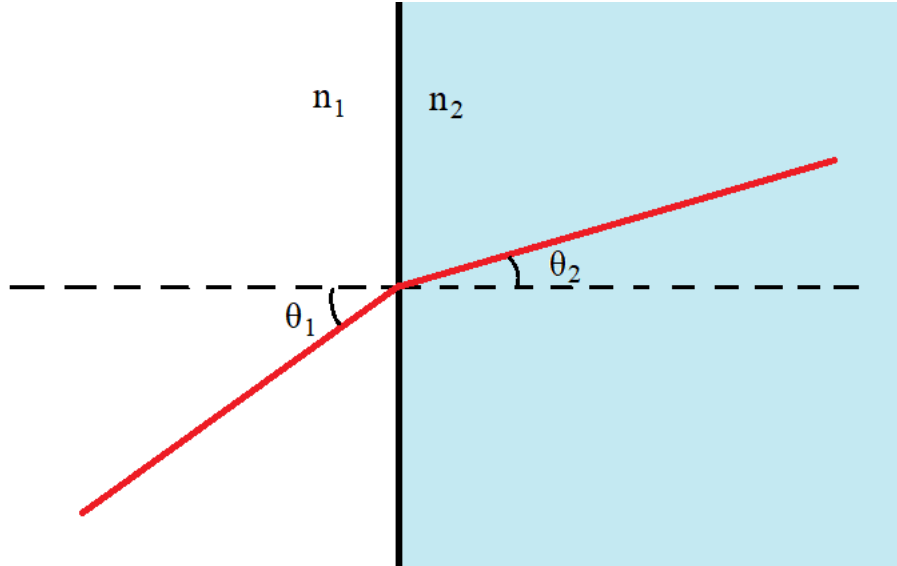
3.7.1 Stråloptik

Stråloptik behandlar ljus som en stråle, en idealiserad modell av ljusets väg. En viktig del av stråloptik handlar om hur laserstrålar bryts när de träffar en spegel. Strålen infaller mot spegeln med en infallsvinkel θ_i mot normalen, och reflekteras med en vinkel θ_r , även den mot normalen. Enligt reflektionslagen är $\theta_i = \theta_r$. [75]

Ett liknande fenomen uppkommer när en ljusstråle färdas genom gränsytan mellan två olika material, och därmed bryts och får en ny riktning. Detta sker enligt Snells lag

$$n_1 \sin \theta_1 = n_2 \sin \theta_2, \quad (27)$$

där n_1 och n_2 är mediernas brytningsindex, och θ_1 och θ_2 är strålens vinkel mätt mot normalen [76], se Figur 1. Om ljuset färdas från ett optiskt tunnare material till ett optiskt tätare bryts strålen mot normalen. I det omvända fallet bryts strålen från normalen och det konstateras då att vinkeln θ_2 aldrig kan bli större än 90° , alltså då $\sin \theta_2 = 1$. Detta innebär att för infallsvinklar större än gränsvinkeln $\theta_g = \arcsin \frac{n_2}{n_1}$ sker totalreflektion, det vill säga att allt ljus reflekteras i gränsytan. [76]



Figur 1: Snells lag anger hur en ljusstråle bryts i gränssytan mellan två material med olika brytningsindex. Detta sker enligt $n_1 \sin \theta_1 = n_2 \sin \theta_2$, där n_1 och n_2 är mediernas brytningsindex, och θ_1 och θ_2 är strålens vinkel mätt mot normalen.

3.7.2 Fourieroptik

Fourieroptiken behandlar ljus som vågor, till skillnad från stråloptiken, där ljus behandlas som en stråle av partiklar. Ljus är elektromagnetiska vågor, så de relevanta ekvationerna är Maxwells ekvationer utan källtermer:

$$\nabla \cdot \vec{D} = 0 \quad (28)$$

$$\nabla \cdot \vec{B} = 0 \quad (29)$$

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \quad (30)$$

$$\nabla \times \vec{H} = \frac{\partial \vec{D}}{\partial t}, \quad (31)$$

där \vec{D} är det elektriska förskjutningsfältet, \vec{B} är den magnetiska flödestätheten, \vec{E} är det elektriska fältet, och \vec{H} är den magnetiska fältstyrkan. För en omfattande förklaring av vad dessa fält innebär, se Jackson[77]. Symbolerna $\nabla \cdot$ och $\nabla \times$ är differentialoperatorerna **div** och **curl**, för definition av dessa hänvisas till Persson-Böiers [78]. I ett homogent isotropt material kan dessa fyra ekvationer bli en:

$$\nabla^2 \vec{E} - \frac{1}{c^2} \frac{\partial^2 \vec{E}}{\partial t^2} = 0, \quad (32)$$

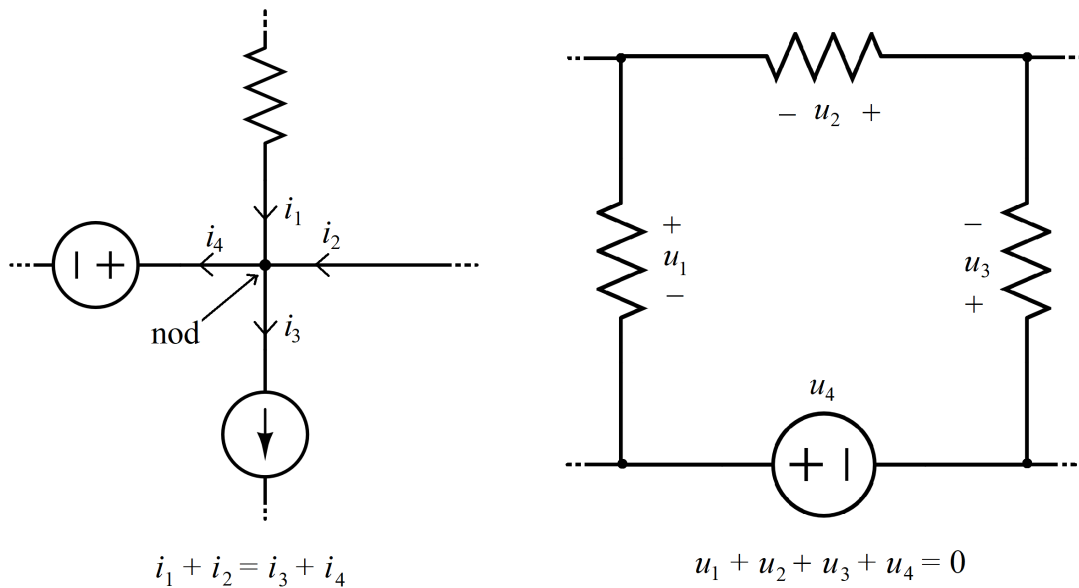
där c är ljushastigheten i materialet[77]. Detta är den så kallade vågekvationen. Observera att en planvåg med amplitud \vec{E}_0 och vågvektor \vec{k} ,

$$E_{\vec{k}}(\vec{x}, t) = \text{Re } \vec{E}_0 e^{-i|\vec{k}|ct - i\vec{k} \cdot \vec{x}}, \quad (33)$$

löser vågekvationen. Detta används av den numeriska metoden PAS (Propagation of Angular Spectrum).

PAS löser problemet där monokromatiskt ljus infaller från ett plan, och det är av intresse hur det propagerar ut från planet. Matematiskt innebär det att vi vet $\vec{E} = \text{Re } \vec{E}_0 \exp(-i\omega t)$ på planet, där ω är vinkelfrekvensen på ljuset, och att vi är intresserade av \vec{E} utanför planet. Tricket i PAS är att med en tvådimensionell fouriertransform dela upp randvillkoret i ett kontinuum av planvågor, vilka har en analytisk lösning.

Med numeriska metoder som PAS kan ljus realistiskt simuleras nära optiska komponenter, till exempel linser, dubbelspalter, och gitter. En sådan simulering är för beräkningstung för att se användning i grafisk rendering.



Figur 2: Vänstra bilden illustrerar Kirchhoffs strömlag, som säger att summan av alla strömmar in i en nod är lika med summan av strömmarna ut från den. Högra bilden illustrerar Kirchhoffs spänningslag, som säger att summan av spänningarna över komponenterna i en sluten krets är lika med noll.

3.8 Kretslära

För att förstå hur ström och spänning fördelar sig i en likströmskrets används Ohms lag samt Kirchhoffs lagar. De kan även användas mer generellt i växelströmskretsar som förutom resistorer även kan innehålla kondensatorer och induktorer, om växelströmmen och spänningen har konstant frekvens. Då används $j\omega$ -metoden för att skriva komponenternas kapacitans respektive induktans som ett komplext tal, impedans, varefter denna impedans kan räknas med på samma sätt som resistans i en likströmskrets.[79]

Ohms lag beskriver hur ström och spänning förhåller sig för en linjär resistor, nämligen enligt

$$U = R \cdot I, \quad (34)$$

där U är spänningen, R resistansen och I strömmen[79]. För en växelströmskrets ersätts resistansen med impedansen Z , och Ohms lag kan då användas även för kondensatorer och induktorer.

Kirchhoffs strömlag säger att summan av alla strömmar in i en nod, alltså en knutpunkt i kretsen, är lika med summan av strömmarna ut från den. Det vill säga

$$i_1 + i_2 + \dots + i_n = 0, \quad (35)$$

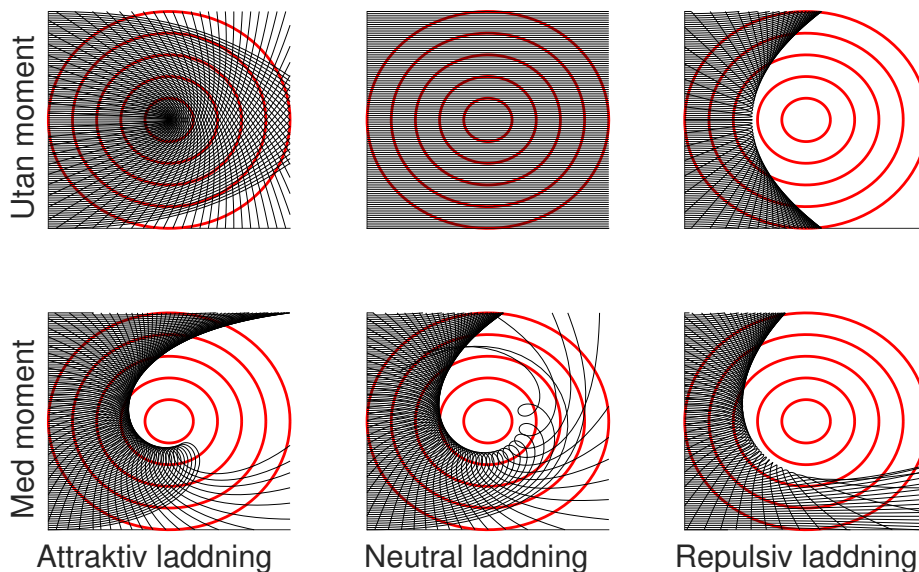
där i_k betecknar en nodström, som anges med positivt tecken om den går in i noden och med negativt om den går ut från den[79], se Figur 2.

Kirchhoffs spänningslag anger hur spänning fördelar sig i en sluten krets, nämligen att summan av spänningen över komponenterna i kretsen är noll, enligt

$$u_1 + u_2 + \dots + u_n = 0, \quad (36)$$

där u_k betecknar en spänning över en komponent[79], se Figur 2.

Dessa förutsätter en ideal krets, på så vis att ledarna är perfekta och all impedans är samlad i idealiserade elektriska komponenter.



Figur 3: Hur laddande partiklar påverkas av elektromagnetiska fält. Figurens sex celler visar hur partiklarna som infaller från vänster böjs av. I den vänstra kolumnen har mittpunkten en attraktiv punktladdning, i den högra en repulsiv. I den undre raden har mittpunkten ett magnetiskt dipolmoment, i riktning normal mot pappret. De röda cirklarna markerar avstånd från mittpunkten.

3.9 Elektromagnetism

Elektromagnetism är läran om hur laddade partiklar växelverkar mellan varandra med en attraherande eller repellerande kraft. Denna kraft har fått namnet *Lorentzkraft* och uttrycks som,

$$\vec{F}_{\text{Lorentz}} = q \left(\vec{E} + \vec{v} \times \vec{B} \right) \quad (37)$$

där \vec{F} är den resulterade kraften, q partikelns laddning, \vec{v} partikelns hastighetsvektor, \vec{B} det magnetiska fältet och \vec{E} det elektriska fältet[80]. Det elektriska och magnetiska fältet härleds av Maxwells ekvationer, som presenterades i avsnitt 3.7.2. Två exempelkällor till fält presenteras i nästa stycke. De härleds i [77].

En typisk källa till det elektriska fältet är en *punktkälla*. En punktkälla med elektrisk laddning q ger upphov till fältet

$$\vec{E}(r, \theta, \phi) = \frac{q}{4\pi\epsilon_0} \frac{1}{r^2} \hat{r}, \quad (38)$$

där ϵ_0 är permittiviteten, en konstant[81]. En motsvarande källa för det magnetiska fältet är *dipolkällan*, med tillhörande fält

$$\vec{B}(r, \theta, \phi) = \frac{\mu}{r^3} \left(2 \cos(\theta) \hat{r} + \sin(\theta) \hat{\theta} \right), \quad (39)$$

där μ är det längden av det magnetiska momentet[82]. Här antas det magnetiska momentet vara riktat längs med nordpolen ($\theta = 0$) i det sfäriska koordinatsystemet.

Som en tillämpning studeras hur laddade partiklar påverkas av en kombinerad punktkälla och dipolkälla. Kraftekvationen blir $F_{\text{Lorentz}} = ma$. Den löses i MATLAB med den inbyggda funktionen *ode45*[83], med olika värden på q och μ . Lösningarna presenteras i Figur 3. En intressant observation är att stora μ kan deflektera partiklar från källan trots att q attraherar dem.

Uppgift \ Vecka	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Uppstart																				
Implementeringsdetaljer				X																
Planeringsrapport					X															
Inläsning																				
Fysik implementerad							X													
Halvtidsredovisning								X												
Dynamik																				
Multiplayer / AI																				
Iterativ spelutveckling																				
Dokumentation av tidigare spel																				
Skriftlig rapport																			X	X
Utställning																			X	
Slutredovisning och opposition																				X
Egenbedömning								X												X

Figur 4: Veckovis tidsplanering för projektet, så som vi planerade vid projektets början. De färgade rutorna visar under vilka veckor vi planerade att arbeta med en viss uppgift, där färgen anger vilken kategori uppgiften tillhör. Implementering omfattar allt som rör utvecklingen av produkten, och med externa resultat menas de uppgifter som har en deadline satt "utifrån". Kryssade rutor representerar deadlines.

4 Genomförande

I följande kapitel beskrivs hur projektets genomfördes. Detta innefattar vår planering och hur väl vi följde den, hur dokumentationen av existerande spel har gått till, våra val av arbets sätt och mjukvara samt genomförandet av de olika implementationerna som gjordes under projektets gång.

4.1 Tidsplanering och översikt

I detta avsnitt presenteras vår tidsplanering samt hur väl vi följde den i form av en översikt av projektet över tid. Processen för de olika arbetsområdena kommer sedan att förklaras vidare i kommande sektioner.

Figur 4 visar den tidsplanering som lades upp i början av projektet, medan Figur 5 visar hur vi faktiskt har fördelat arbetet tidsmässigt. Varje arbetsuppgift beskrivs av en process som i många fall avslutas med en deadline. Uppgifterna är indelade i fyra områden: förberedelser, implementering, dokumentation av spel och externa resultat.

Förberedelser innefattar de delar vi behövde genomföra för att kunna göra de senare delarna, såsom planering och inläsning av teori. Implementeringsdetaljer syftar till val av spelmotor samt val av fysik och dynamik till spelen. Inläsning innefattar relevant teori samt hur spelmotorn vi skulle använda fungerar. Inläsning av fysik som är relaterad till implementationerna ingår istället i utvecklingsprocessen för dessa.

Implementering innefattar utvecklingen av våra egna spel. I planeringen är den uppdelad i fyra delprocesser: fysik, dynamik, multiplayer/AI och iterativ spelutveckling. Dynamiken syftar till det som utvecklar en fysiksimulering till ett faktiskt spel. Multiplayer eller en datorstyrd motspelare genom AI var något som vi övervägde att göra, men det var inte helt bestämt. I vår faktiska tidsöversikt är Implementering istället uppdelat efter de fem minispel som vi har utvecklat under projektets gång: Spacetime war, Hydrocurve, Tic Tac Ket, Lagom Action och A Maze in Light.

Dokumentation av spel innefattar allt som har med vår dokumentation och analys av tidigare spel att göra, som att hitta de olika spelen, analysera fysiken i dem och kartlägga vilka fysikområden som har använts mer frekvent i redan existerande spel.

Externa resultat beskriver de uppgifter som har en deadline satt "utifrån", alltså de inlämningar och redovisningar som vi behövde göra i projektet.

Uppgift \ Vecka	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Uppstart																				
Implementeringsdetaljer				X																
Planeringsrapport					X															
Inläsning																				
Halvtidsredovisning								X												
Spacetimewar																				
Hydrocurve																				
Tic Tac Ket																				
Lagom Action																				
A Maze in Light																				
Dokumentation av tidigare spel																				
Skriftlig rapport																			X	X
Utställning																			X	
Slutredovisning och opposition																				X
Egenbedömning								X												X

Figur 5: Veckovis tidsöversikt över projektets gång. De färgade rutorna visar under vilka veckor vi arbetade med en viss uppgift, där färgen anger vilken kategori uppgiften tillhör. Implementering omfattar allt som rör utvecklingen av våra minispel, och med externa resultat menas de uppgifter som har en deadline satt "utifrån". Kryssade rutor representerar deadlines.

En stor skillnad mellan planeringen och det slutliga arbetet är att vi från början tänkte implementera endast två spel parallellt under hela projektet, något som senare ändrades till att göra fler minispel för att på så vis kunna utforska fler fysikområden. Detta syns tydligt i våra två tids-scheman. Eftersom vi planerade att utveckla samma spel under hela projektets gång planerade vi att lägga mycket tid på iterativ spelutveckling. Denna del blev sedan förminskad då inte lika stort fokus lades på att utveckla implementationerna till bra spel, samt att den delades upp i mindre delar, en för varje minispel. Detsamma gäller för uppgifterna att implementera fysiken och dynamiken; de blev också komprimerade till varje implementation. Implementeringsdetaljer innefattade dessutom endast val av fysik och dynamik till de två första spelen, medan de för de senare spelen valdes i samband med deras uppstart. På samma sätt blev inläsning något vi gjorde mer under hela projektets gång, då varje ny implementering behövde börja med en inläsningsfas. Detta syns inte så tydligt i schemat då det är inkluderat i utvecklingen av minispelen. Vidare är multiplayer något som implementerades i vissa minispel men inte i andra, beroende på deras mål, och AI är inte implementerat för något av dem. Förutom de färdigutvecklade spelen utvecklade vi några spelprototyper som inte blev implementerade. Dessa utvecklades under hela projektets gång, och visas inte i schemat.

4.2 Dokumentation och analys

Dokumentation och analys av fysikanvändning i redan existerande spel påbörjades tidigt i projektet och utökades kontinuerligt under hela projektets gång. Syftet med detta var att ge oss en bild av vilka områden som redan har utforskats samt för att se hur de tillämpat fysiken, exempelvis om de valt att följa fysikens lagar korrekt eller om fokus bara har varit att få det att se snyggt ut. Spelen som analyserades var en blandning av spel som vi själva kände till sedan innan, och spel som söktes fram på nätet eller i vissa fall tillhandahölls av vår handledare. När vi analyserade de olika spelen försökte vi först och främst hitta dokumentation på nätet om den underliggande fysiken. Vi försökte även spela spelen själva för att mer utförligt kunna utforska och analysera de olika tillämpningarna av fysik. Detta var dock endast möjligt i de fall där någon i gruppen redan ägde spelet i fråga, alternativt om det fanns tillgängligt att spela gratis på nätet. I de fall där vi analyserade spel som vi inte hade tillgång till genomfördes analysen istället genom att vi kollade på gameplay-videor från spelet på Youtube.

Resultaten från denna analys kartlades i en matris för att på bästa sätt kunna besluta om vilka fysikområden vi ville börja studera. På så vis kunde vi konstatera vilka områden som var underrepresenterade eller till och med inte representerade i vissa fall, för att sedan utforska om det fanns möjlighet att skapa någon intressant spelmekanik utifrån dessa.

4.3 Mjukvaruutveckling

För att utforska om de fysikområden som visade sig underrepresenterade kan bidra till en förbättrad användarupplevelse i spel skapades implementeringar i form av olika fysikbaserade minispel. Projektgruppen delades in i olika arbetsgrupper, där grupperna utvecklade varsitt minispel parallellt. Mjukvaruutvecklingen arbetades med enligt ett agilt arbetssätt med veckovisa sprintmöten. Detta innebar att vi arbetade iterativt och inkrementellt under hela projektets gång, med frekvent leverans av körbara prototyper. Vid sprintmötena diskuterades gruppernas resultat, och därefter bestämdes hur vi skulle arbeta vidare med implementationerna. Användartester utfördes även på de olika implementationerna. De som testade spelen var en blandning av oss själva, vänner och familjemedlemmar.

Programmeringen gjordes uteslutande i Java med hjälp av ramverket *libGDX* [84]. Anledningen till valet av *libGDX* framför andra mer populära spelmotorer, såsom *Unity*[85] och *Unreal Engine*[86], var först och främst att den tillåter programmering i Java, vilket var det enda programmeringsspråk som alla i gruppen hade någon tidigare erfarenhet av. Bland de spelmotorer och ramverk som tillåter programmering i Java var *libGDX* vårt förstahandsval då det har stöd för både 2D- och 3D-grafik, det finns massor med dokumentation och tutorials för det, samt att det är rekommenderat av många Java-spelutvecklare [87]. Andra alternativ som tillät programmering i Java som vi kollade på var *LWJGL*[88], *Slick2D*[89] samt *jMonkeyEngine*[90]. Dessa valdes bort då de antingen ansågs sakna bra dokumentation och tutorials alternativt att de inte uppdaterats de senaste åren. Stora delar av den implementerade fysiken gjordes med hjälp av det populära open-source fysikbiblioteket *Box2D*, som var fullt kompatibelt med *libGDX*. [91]. Där skapar man objekten som kroppar, som sedan olika krafter, impulser eller vridmoment kan läggas på. Utifrån detta löses rörelsekvationerna med semi-implicit Euler för att bestämma hur kropparna ska röra sig[92]. Detta gjorde implementeringen av fysik enklare, då vi endast behövde skriva in krafter och liknande, och inte lösa några differentialekvationer själva.

Projektets källkod delades på GitHub då några i gruppen hade erfarenhet av tjänsten sedan tidigare. Vi arbetade enligt en typ av feature branch Git workflow, där varje implementation arbetades med på en separat branch. Även nya funktioner till de olika implementationerna arbetades med på separata branches. Detta arbetsflöde valdes då det var smidigt och välanpassat till att ungefär hälften av gruppen saknade tidigare erfarenhet av Git.

Från början var idén att utforska ett fåtal fysikområden, implementerade i endast två spel. Detta speglas i de spel som utvecklades i början av projektet, *Hydrocurve* och *Spacetimewar*, där mer tid lagts på att göra dem till underhållande spel än vad som gjordes på de spel som vi utvecklade senare. Ungefär en månad in i implementeringsfasen kände vi att vi hade tillräckligt med tid för att kunna utforska fler fysikområden. En tanke var då att man skulle kunna lägga till nya spelmekaniker baserad på andra typer av fysik till våra redan existerande spel. Vi valde slutligen att de olika utforskade fysikområdena istället skulle resultera i flera olika minispel; detta då det gav oss möjligheten att fritt utforska ett antal olika fysikområden utan att behöva begränsa oss till fysik som passade in i ett redan existerande spel. Då vi ville att fokus skulle vara på utforskningen och inte på implementeringen valde vi även att undersöka och presentera vissa spelidéer som inte resulterade i någon implementation. Detta gjorde att vi kunde spara tid och således utforska så många områden som möjligt.

4.4 Framtagande av spelidéer

För att finna spelidéer baserade på ett fysikområde, listade vi först några välkända fenomen för det fysikaliska området. Som ett exempel innehåller kvantfysiken bland annat fenomenen kvantifiering, interferens, och superpositioner. Sedan undersöktes dessa fenomen och vilken roll spelaren kunde ha i dem. Som exempel kan spelaren i ett spel om elektromagnetism ha en laddning, kunna skjuta en laser, eller kunna flytta på komponenter som påverkar ett elektromagnetiskt fält. Sist gjordes ett subjektivt val: vilket fenomen ska implementeras och vilken roll ska spelaren ha. Eftersom vi

ville sprida ut implementationerna över så många fysikområden så möjligt, valde vi att bara ta vidare en idé per område till implementation. Några idéer som inte blev implementationer men ändå gav insikter kommer att presenteras i resultatet.

4.5 Implementering av minispelen

I kommande sektioner presenteras hur utvecklingen av de olika implementationerna har gått till.

4.5.1 Utvecklingen av Spacetimewar

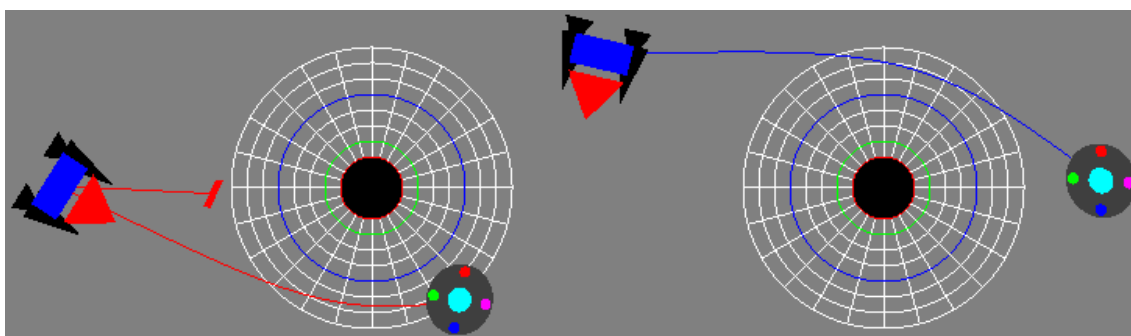
Allmän relativitetsteori är ett intressant område i fysiken, men som används väldigt sällan i spel. Därför beslutade vi att göra ett spel där detta område inom fysiken används. Eftersom det är svårt att märka av relativistiska effekter förutom i vissa extrema fall eller områden, beslutades det tidigt att detta spel skulle utspela sig i rymden runt ett svart hål. Svarta hål ger nämligen upphov till påtagliga relativistiska fenomen.

Med spelets fysik och miljö vald återstod det att välja vad spelet skulle gå ut på. Eftersom spelet skulle utspela sig i rymden runt ett svart hål gick våra tankar till *Spacewar!*, som utspelar sig i samma miljö, men som inte inkluderar någon relativistisk fysik. Då inga negativa aspekter i att basera ett spel på *Spacewar!* hittades, och det dessutom sparar värdefull tid att inte behöva hitta på ett spel helt från grunden, blev det så. Spelet fick namnet *Spacetimewar*, vilket anspelar på att rum och tid byts ut mot rumstid i relativitetsteori.

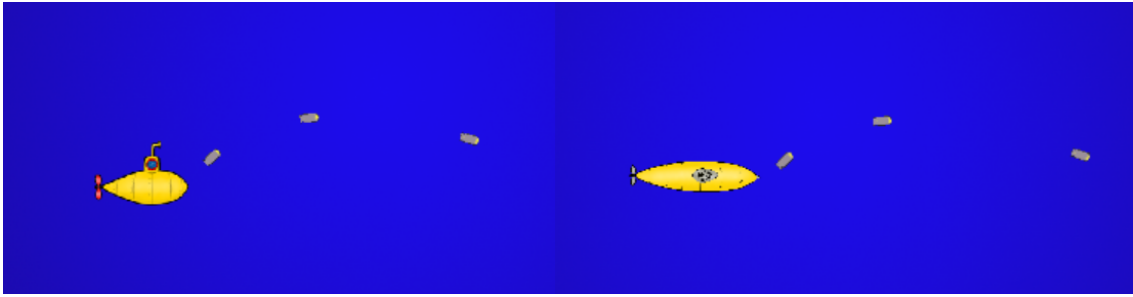
Med målet att göra ett *Spacewar!*-liknande spel med realistisk allmän relativitetsteori i fokus påbörjades implementationen. För att göra realistisk fysik krävs en hel del matematiska uträkningar. Majoriteten av implementationstiden gick därför till att skapa nya Java-klasser för en mängd olika matematiska objekt och metoder. Exempel på objekt är tensorer av godtycklig rang, ljussignaler, och relativistiska fysikaliska kroppar; exempel på metoder är Gram-Schmidt-processen generaliserad för godtyckliga metriker med signatur $(1,3)$, Fermi-Walker transport av tensorer, och Lorentz-boosting uttryckt i en lokal tetrad.

Tidigt i processen skapades en snabb demonstrationsversion av det tänkta spelet, där allt till en början representerades av kuber. Denna enkla version användes för att få en visuell representation av vad matematiken resulterade i, för att påbörja den slutgiltiga grafiken och för att dela upp skärmen i två delar. Även olika tidiga alternativ för hur spelet skulle kontrolleras testades här.

Grafiken i *Spacetimewar* är helt uppbyggd av 3D-modeller från libGDX-klasserna *ModelBuilder* och *MeshPartBuilder*, vilka användes för att enkelt göra en tredimensionell rymdfärja till implementationen. Kameran som *Spacetimewar* använder är dock libGDX kamera-klass *OrthographicCamera*, vilket innebär att spelet i alla bemärkelser ser ut att vara i 2D. Valet att trots 2D-vyn använda 3D-grafik kommer främst ifrån att *Spacetimewar* var en av de två första av våra implementationer. Eftersom den andra implementationen, *Hydrocurve*, skulle vara helt i 2D, men libGDX stödjer både 2D- och 3D-grafik, sågs det som nyttigt att utforska libGDX möjligheter att göra spel i 3D också. Resultatet kan ses i Figur 6.



Figur 6: Exempelbild på *Spacetimewar*. Vänstra delen är ur rymdfärjans perspektiv, högra delen ur UFO:ts. Rymdfärjan har precis avfyrat en röd kvadratisk projektil mot UFO:t, men UFO:t vet det inte än på grund av informationsfördröjning, och projektilen ser platt ut på grund av längdkontraktion. De färgade kurvorna illustrerar banan för den foton som just nu observeras.



Figur 7: Sett från sidan kan en krökt projektilbana misstas för en följd av gravitation, men sett ovanifrån är risken mindre. Det är en anledning till att Hydrocurve har en ovanifrån-vy.

När grafiken och matematiken var färdig fungerade spelet som det var tänkt, men en del mindre ändringar utfördes för att göra det mer lättspelat. Bland annat förbättrades kontrollerna och ett nytt spelbart rymdskepp i form av ett UFO, med annorlunda kontroller, lades till. Därefter diskuterades att lägga till fler saker i spelet, men vi ansåg att spelet redan uppföll sitt syfte att visa hur allmän relativitetsteori kan påverka gameplay. Därför lämnades spelet ifred och nya implementationer påbörjades.

4.5.2 Utvecklingen av Hydrocurve

Ubåtsimplementationen är ett spel där två ubåtar har som mål att skjuta på varandra med projektiler med krökta banor. Detta görs genom att projektilerna har en egen drivkraft i en riktning som avviker från färdriktningen, vilket gör att projektilen svänger av. Tillsammans med Spacetimewar utforskar den olika fysikaliska källor till krökning som kan ge en innovativ spelmekanik, vilket var en anledning till att vi utvecklade implementationen. Dessutom ger vattnet en effekt på projektilerna och ubåtarna i form av vattenmotstånd som vi inte sett i så många spel tidigare då de flesta utspelar sig i vakuum eller luft. Denna implementation namngavs *Hydrocurve*, för att koppla till både vattnet och projektilernas krökning.

Implementationen av fysik i Hydrocurve är gjord med hjälp av fysikmotorn Box2D. Ubåtarna och projektilerna implementeras som kroppar som kan påverkas av att man lägger på olika krafter, impulser eller vridmoment på dem. Detta gjorde vi för att underlätta vid implementeringen av fysik, då vi inte behövde lösa rörelseekvationerna själva utan endast skriva in krafter och liknande. Kropparna skapades med hjälp av programmet *Physics Body Editor* där man kan generera polygoner utifrån en given textur till en JSON-fil [93]. Texturen är själva bilden på objektet, till exempel en ubåt. Den färdiga klassen *BodyEditorLoader* användes och modifierades för att ladda in dessa JSON-filer till spelet. Detta användes för att på ett smidigt sätt få kropparnas form att stämma överens med formen på deras texturer, så att till exempel kollisionerna som registreras följer ubåtens eller projektilens form. Grafiken för spelet ritades av oss med hjälp av *Paint Tool SAI* [94] samt *Photoshop*. Dessa program användes främst då vi kunde få mycket hjälp från vänner med flera års erfarenhet av att arbeta med dem båda. Spelplanen skapades i *Tiled Map Editor* [95]. Den huvudsakliga orsaken till valet av detta program var att det användes i de tutorials för libGDX som vi har kollat på. Andra alternativ övervägdes, exempelvis *Overlap2D*, men avfärdades efter mindre testning då det ansågs vara onödigt krångligt att få att fungera med den valda spelmotorn.

Det är två huvudsakliga fysikaliska fenomen som kommer in i Hydrocurve. Det första är vattenmotståndet som påverkar ubåtarna såväl som projektilerna, och det andra är projektilernas egna drivkraft, som ger upphov till deras krökta banor. I början av projektet var tanken att ubåtarna skulle visas från sidan, vilket innebar att även gravitation och lyftkraft behövde räknas med. För ubåtarna skulle detta i de flesta fall inte göra så stor skillnad, då de skulle hitta jämvikt och "sväva" på önskad höjd tills spelarna vill flytta dem. För projektilerna däremot innebar det att de kunde få en krökt bana av gravitationen istället för av sin egna drivkraft. För att krökningen då inte skulle misstas för gravitation, valde vi istället att ha en ovanifrån-vy på spelet, vilket innebar att vi inte längre behövde ta hänsyn till gravitation och lyftkraft. Figur 7 visar hur projektilens krökning lätt kan se ut som en följd av gravitation om vyn är från sidan, medan den inte gör det med vy ovanifrån.

I samband med valet att ändra spelarvy ändrade vi också styrningen av ubåtarna. Från början styrdes de genom att helt enkelt flyttas i riktningen som spelaren valde med piltangenterna. Detta ändrades till att istället vrida ubåten med höger- och vänster-tangenterna, och förflytta den framåt och bakåt med upp- respektive ned-tangenterna. Detta gjordes för att få en mer realistisk styrning av ubåtarna.

Vattenmotståndet bestämdes enligt ekvation (9), då flödet antogs vara turbulent efter följande uppskattning av Reynoldstalet, $Re = \frac{\rho v L}{\mu}$. För vatten gäller att $\mu \approx 10^{-3} \text{ Pa s}$ och $\rho = 1000 \text{ kg/m}^3$. Ubåten är i storleksordningen 10 m och färdas i storleksordningen i 10 m s^{-1} . Detta ger $Re \approx 10^8$, vilket är högt nog för att flödet ska vara turbulent.

Projektilerna drivs av en kraft med en riktning som avviker aningen från färdriktningen, vilket ger upphov till en krökning av färdbanan. Hur mycket dess riktning avvek samt storleken på kraften balanserades så att önskad krökning åstadkoms. Då det är två ubåtar i spelet som ska skjutas på varandra prioriterade vi att det skulle gå snabbt för en spelare att välja riktning och skjuta, istället för att ha gott om tid att ställa in siktet ordentligt. Detta gjorde vi genom att ha tre olika lägen på projektilerna, som anger vilken krökningsvinkel den kommer ha, det vill säga om den kommer vika av åt höger, vänster eller inte alls. Detta ändras enkelt genom ett knapptryck. Dessutom kan spelaren välja att skjuta ut projektilen rakt fram, åt vänster eller åt höger, vilket styrs med en knapp per riktning. Att kombinera dessa på olika sätt ger alltså upphov till många olika projektilbanor som spelaren kan testa för att nå sitt mål.

För att utveckla spelupplevelsen till ett spelbart spel lade vi till så att en ubåt tar lite skada vid varje träff av en projektil, vilket visas med en mätare för varje ubåt. När mätaren når botten visas en explosion på dess plats, för att göra det tydligt att den andra spelaren har vunnit och spelet är slut.

4.5.3 Utvecklingen av Tic Tac Ket

Ett fysikområde som inte har använts i de spel som vi har undersökt är kvantfysik. Därför valde vi att göra ett spel som involverade fenomenet sammanflätning. Detta avsnitt berättar i kronologisk ordning hur spelets mekaniker skapades, och vad som motiverade dem.

Kvantfysik är svårt att räkna på. Mängden information om ett system växer exponentiellt med storleken på systemet, en stor skillnad mot den linjära tillväxt som klassisk fysik har. För att det implementerade spelet skulle bli begripligt för människor och spelbart i realtid, valdes därför ett väldigt enkelt spel att "kvantifiera": tre i rad. Implementationen namngavs *Tic Tac Ket*, efter engelskans *Tic Tac Toe* och kvantfysikens *ket* ($|\cdot\rangle$).

Tre i rad spelas på ett bräde med N rutor. En ruta kan innehålla en vit bricka, en svart bricka, eller vara tom. Det finns därför 3^N möjliga klassiska tillstånd som ett bräde kan befinna sig i. Basen för tillståndsrummet innehåller därför så många element. Ett kvantfysikaliskt tillstånd kan därför representeras som en summa av klassiska tillstånd med komplexa c -koefficienter framför, se ekvation (11) för ett exempel. Det naiva sättet att lagra informationen om c -koefficienterna skulle vara två flyttal för varje c_i : realdelen och imaginärdelen. För ett $N = 4 \times 4$ spelbräde blir detta ungefär 86 miljoner flyttal, vilket är onödigt mycket. En stor förbättring är att istället lagra informationen glest, vilket innebär att alla nollskilda c_i lagras, tillsammans med sina index.

En konsekvens av att det finns många speltillstånd är svårigheten att visualisera det nuvarande tillståndet. Det är inte praktiskt möjligt att lista de tusentals klassiska tillstånden tillsammans med lika många c_i -koefficienter. Ett enkelt men begränsat alternativ är att visa brädet, och i varje ruta visa väntevärdet av vad som ligger där. Om en ruta har 50% chans att ha en svart bricka på sig, och annars har en vit bricka på sig, visas en grå bricka på rutan. Detta har nackdelen att sammanflätning inte syns; de två tillstånden

$$|W_1 W_2\rangle + |B_1 B_2\rangle \quad \text{och} \quad |W_1 B_2\rangle + |B_1 W_2\rangle,$$

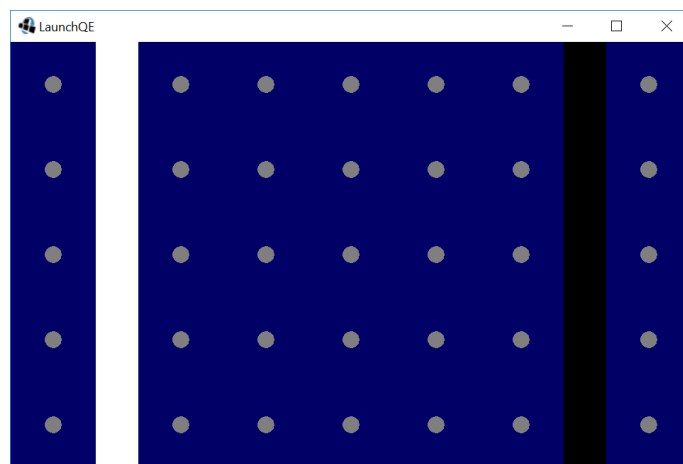
där $|W_1 B_2\rangle$ är tillståndet med en vit bricka på ruta ett och en svart bricka på ruta två, visas båda som två grå brickor, trots att de är sammanflätade på olika sätt. Som en kompromiss implementerades båda visualiseringar samtidigt. Spelaren ser väntevärdet som en färg, och alla klassiska tillstånd som ett cirkeldiagram.

I klassiskt tre i rad kan en spelare på sin tur placera ut en bricka i sin färg på en tom ruta. Den naturliga analogen i kvantfysik är att tensormultiplicera brädets tillstånd med en bricka som ligger

i en superposition av flera rutor. Med denna definition, om två brickor placeras ut så att deras möjliga positioner inte överlappar, beter de sig som om de hade varit ensamma, och helt oberoende av varandra. Som en avvägning mellan frihet och enkelhet, implementerades utplacering som att spelaren först väljer n ($n \geq 1$) rutor fritt, och alla c_i för brickan blir sedan $\frac{1}{\sqrt{n}}$. Eftersom interferens inte uppkommer i spelet, är det inte en begränsning att lagra c_i som reella tal.

Något i vår implementation som saknar klassisk motsvarighet är observation. En spelare kan på sin tur, istället för att placera ut en bricka, observera en ruta. Tillståndet kollapsar då, enbart i den spelarens perspektiv, så att den observerade rutan befinner sig i ett klassiskt tillstånd.

Det upptäcktes tidigt i spelttestandet att det vid utplacering är bäst att välja $n = 1$, eftersom spelare då säkert vet var brickan hamnar. Detta är oönskat ur ett designperspektiv, eftersom spelet då degenererar till sin klassiska version. För att ändra på detta infördes en ny regel: När spelaren väljer vilka rutor brickan ska hamna i, måste exakt en tom ruta i *handen* väljas för placering. Handen är en ny zon där brickor kan hamna, men som inte på något sätt bidrar till att vinna spelet. Varje spelare har en egen hand, så en spelare får inte göra något med motståndarens hand. Syftet med handzonen är att avskräcka spelare från att välja n för litet. Chansen att en placering lyckas, det vill säga inte hamnar i handen är $1 - \frac{1}{n}$, givet att ingen sammanflätning har skett. Figur 8 visar hela spelplanen, brädet och händerna.



Figur 8: Den tomma spelplanen, med 35 platser. De två 5×1 -kolumnerna på varsin kant är händerna och 5×5 -gittret i mitten är brädet.

Ett problem med handen är att den över tid fylls upp, vilket gör det omöjligt att placera ut en bricka. Det implementerades därför ett tredje drag: städning. En spelare kan spendera en tur på att ta bort en bricka som befinner sig i handen.

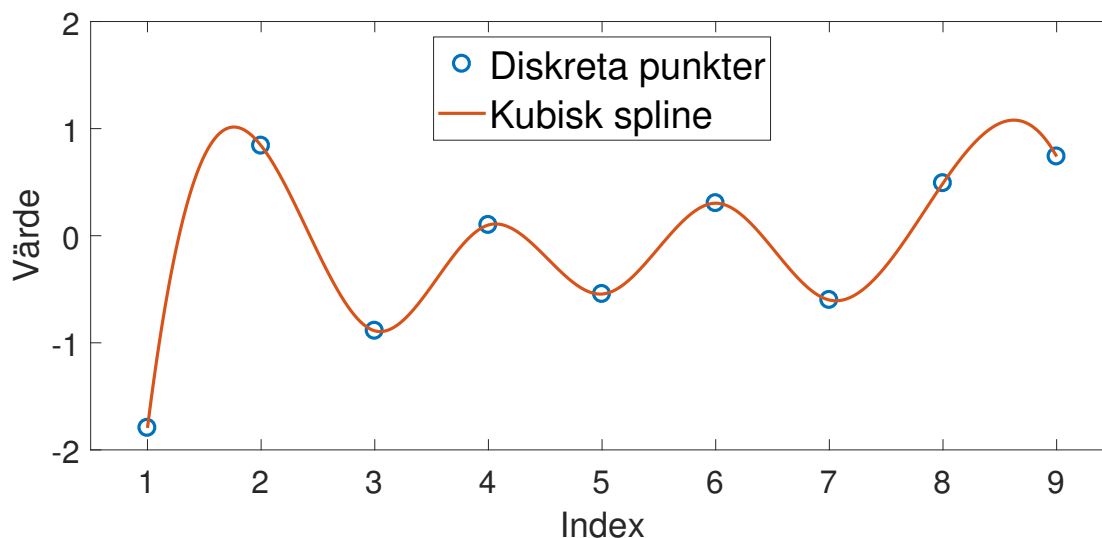
Ett problem som uppstår är bestämmandet av vem som vinner. I den klassiska versionen vinner den som först får tre i rad. Det finns två "naturliga" sätt att översätta detta. Det första är att den som först får tre i rad i *någon* term vinner. Detta är inte ett bra kriterium; den spelare som börjar har en trivial vinnande strategi. Det andra är att den som först får tre i rad i *varje* term vinner. Detta är ännu sämre; spelet hamnar då lätt i ett tillstånd där ingen kan vinna.

Ett tredje vinstkriterium valdes istället: när alla termer har tre i rad av någon färg vinner den spelare vars chans att ha tre i rad är högst. Detta kriterium har inga uppenbara brister, och har fördelen att det i och med detta kan mätas hur nära slutet spelet är. Chansen att någon spelare har tre i rad börjar på noll, förväntas växa med varje drag, och när den når ett är spelet slut.

När ovannämnda vinstkriterium hade implementerats, bedömdes spelet vara färdigt. Tic Tac Ket utvecklades huvudsakligen under en vecka, presenterades under ett veckomöte, och ändrades efter kritik veckan därpå.

4.5.4 Utvecklingen av Lagom Action

Ett område, eller snarare formalism, som enligt våra observationer aldrig har använts i spel är lagrangesk mekanik, troligtvis på grund av dess abstrakta natur. Därför valde vi att göra ett spel som handlade om minsta verkans princip. Minsta verkans princip är ett lustigt namn, eftersom verkan



Figur 9: Demonstration av interpolationsmetoden kubisk spline. Metoden tar in en lista av tal, representerade av de blå cirklarna. Metoden ger en funktion med kontinuerlig andraderivata som skär alla punkter, representerad av den brandgula linjen. Det behöver inte nödvändigtvis vara heltalsindex på x-axeln, men det gör matematiken enklare.

inte behöver vara varken minimal eller maximal, endast stationär. Implementationen namngavs därför *Lagom Action*. Detta avsnitt berättar i kronologisk ordning hur spelets mekaniker skapades, och vad som motiverade dem.

Analytisk mekanik är abstrakt, så det finns stor frihet i valet av system. För att göra visualiseringen enkel, valdes en partikel i en potential. Det är ett system vars komplexitet kan vara allt mellan trivial och kaotisk, beroende på valet av potential.

Spelarens uppgift valdes att vara att, givet en potential samt start- och slutposition och -tid, välja en kurva som förbinder positionerna. Kurvan betygsätts efter hur väl den följer minsta verkans princip:

$$\delta S = 0.$$

Det fanns här två designval som behövde göras: hur spelaren ska kunna välja kurvan, och hur avvikelsen från minsta verkans princip ska kvantifieras.

Den första idén för valet av kurvan var att låta spelaren rita en egen med pekaren. Problemet med denna idé är att det inte går att avgöra partikelns hastighet. Jämför som exempel de två partikelbanorna

$$(x, y) = (t, t) \quad \text{och} \quad (x, y) = (t^2, t^2), \quad 0 \leq t \leq 1.$$

De ser i xy -planet båda ut som en rät linje från $(0, 0)$ till $(1, 1)$, men bara en har konstant hastighet. Det räcker därför inte med att rita en kurva; någon anvisning om tid behövs också.

Den andra idén byggde vidare på den första genom att sätta ut etiketter på kurvan som anger tiden. Detta löser problemet med att avgöra hastigheten. Ett kvarstående problem är att det är svårt att justera en redan ritad kurva, eftersom den innehåller så många punkter. Det var också svårt att kvantifiera avvikelsen från minsta verkans princip.

Den tredje och sista idén var att låta spelaren placera ut enbart etiketterna, nu kallade noder, och att interpolera en kurva mellan dessa noder. Spelaren kan nu finjustera kurvan med små ändringar i nodernas position. Interpolationsmetoden valdes till *kubisk spline*[96], den enklaste linjära interpolationen som garanterar en kontinuerlig andraderivata. Den är intressant eftersom partiklar i kontinuerliga potentialer har kontinuerliga andraderivator hos alla lägeskoordinater. Figur 9 visar ett exempel på hur kubisk spline ser ut.

Det återstod att kvantifiera avvikelsen från minsta verkans princip. δS är differentialen till en funktional, så det går inte att mäta dess längd direkt. Vad som istället kan göras är att derivera

verkan S med avseende på en perturbation r , detta ger en mätbar projektion av δS :

$$\frac{\delta S}{\delta r} = \frac{d}{d\lambda} \int L(q + \lambda r, \dot{q} + \lambda \dot{r}, t) dt \Big|_{\lambda=0}$$

Avvikelsen kvantifierades därför som *root-sum-square* av flera projektioner. Perturbationerna valdes till att vara alla derivator till splinen med avseende på rörliga noders koordinater. Detta är samma sak som att se verkan som en funktion av alla rörliga noders positioner, och beräkna längden på gradienten till denna funktion. Denna kvantifiering av avvikelsen är garanterad att vara noll för något val av nodernas positioner. Derivatorna i gradienten beräknades numeriskt med den symmetriska formeln

$$f'(x) = \frac{f(x + \delta) - f(x - \delta)}{2\delta}.$$

Med de två designvalen gjorda, återstod det att välja potentialer. Den första som testades var

$$V_{\text{fri}} = 0,$$

vilket motsvarar lagrangianen

$$L_{\text{fri}} = T + V_{\text{fri}} = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2).$$

Lösningen blir en rak bana. Det kan tyckas vara trivialt, men V_{fri} kan vara en bra första potential för spelaren att stöta på.

Andra potentialer som testades var

$$\begin{aligned} V_{\text{osc}} &= \frac{1}{2}k(x^2 + y^2), \\ V_{\text{sadel}} &= \frac{1}{2}k(x^2 - y^2), \\ V_{\text{källa}} &= \frac{q}{4\pi} \frac{1}{x^2 + y^2}. \end{aligned}$$

De krökte partikelbanor på olika sätt. Med syfte att visa upp potentialen som funktion av läge, färgades bakgrunden efter värdet på V . Detta sätt att visualisera potentialen valdes eftersom det låter spelaren förstå potentialen utan att behöva se det matematiska uttrycket.

Nära projektets slut påbörjades arbete för att implementera flerpartikelsystem i LM, men det avbröts på grund av svårigheter i att grafiskt representera växelverkan mellan partiklar.

Vid det här laget kunde spelaren se potentialen, gissa en kurva, och få veta hur nära minsta verkans princip gissningen var. Implementationen bedömdes därför vara färdig.

4.5.5 Utvecklingen av A Maze in Light

Optikimplementationen är ett labyrintspel, där spelaren riktar en laserstråle för att träffa olika knappar som låser upp dörrar och gör att spelaren kommer vidare. För att åstadkomma detta finns diverse speglar och glasområden i labyrinten, där laserstrålen kan reflekteras och brytas för att få önskad riktning. Med koppling till labyrinten och laserstrålen fick spelet namnet *A Maze in Light*.

Att vi valde att göra detta spel grundar sig i att vi såg brister i redan existerande spel av samma karaktär, som beskrevs i avsnitt 2.7.2. Dels såg vi att brytning mellan olika material, och därmed användningen av Snells lag, inte används särskilt ofta, och dels konstaterades att när en laserstråle används för ljusguidning med exempelvis speglar, brukar hela strålen ritas ut, medan det är mer korrekt att bara se var den träffar. Detta är två viktiga aspekter som vi tog med oss i utvecklingen av optikimplementationen.

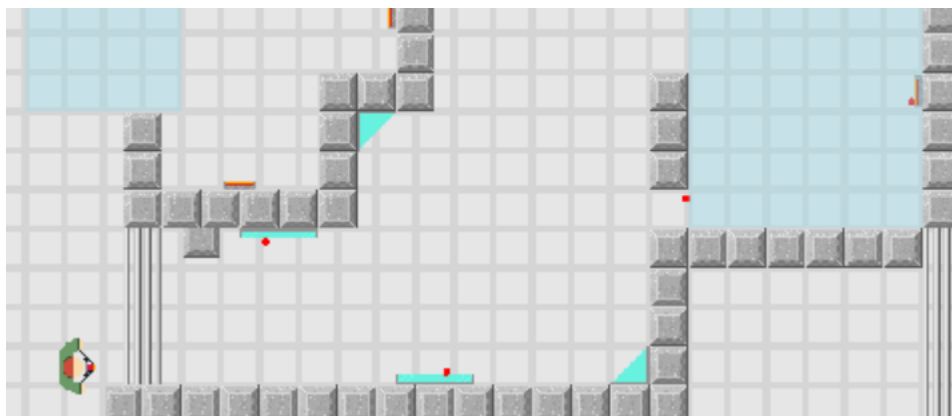
Implementationen av fysik gjordes precis som i Hydrocurve med fysikmotorn Box2D, och kroppar skapades även här med Physics Body Editor, i de fall kropparna hade en mer avancerad form än en rektangel.

För realismens skull ville vi att laserstrålen skulle ritas ut utan fördröjning, och för detta krävs ett snabbt sätt att avgöra var strålen skulle träffa en yta, så som en vägg eller en spegel. Som en första ansats tänkte vi använda den kollisionsdektionsmetod som används för kroppar,

så som spelarens kollision med väggar. Möjligheten att använda den undersöktes genom att låta en osynlig kropp färdas i hög hastighet i laserstrålens riktning, varefter kollision detekterades och strålen kunde ritas ut, men eftersom kroppens hastighet inte kunde bli i närheten av tillräckligt hög visade sig detta vara ej genomförbart. På liknande sätt undersöktes om kollisionen kunde detekteras genom att bygga på kroppar på varandra i linje med strålens riktning tills dess kollision detekteras, men även denna metod visade sig vara alldeles för långsam.

Slutligen använde vi oss av Box2Ds ray casting-funktion, som är en snabb och effektiv metod som uppfyllde våra krav. Ray casting-funktionen fungerar så att man väljer en startpunkt och en slutpunkt, där startpunkten är laserns position och slutpunkten är en punkt som nås från startpunkten i den riktningen man vill skjuta åt på ett avstånd som är utanför spelplanen. Denna metod anropas flera gånger och hittar således alla fixturer som befinner sig mellan de båda punkterna. Varje fixtur motsvarar ett kollisionsobjekt, och varje kropp består av en eller flera fixturer. Metoden returnerar kollisionspunkten, normalriktningen för den träffade ytan samt vilken fixtur som träffades, vilket används för att veta om strålen ska stanna, reflekteras eller brytas. För varje fixtur returnerar den även ett värde som motsvarar hur nära startpunkten den befinner sig, vilket används för att hitta det närmsta objektet som strålen möter.

När punkten för laserstrålens första kollision hittats ritas en punkt ut där, istället för att rita ut hela strålgången. Detta val gjordes för att det är mer realistiskt, då man i verkligheten bara ser var en laserstråle träffar och inte hela vägen dit. Hur denna strålgång ser ut visas i Figur 10. Om den träffade ytan är en spegel eller glas görs därefter en ny ray casting i den reflekterade eller brutna strålens riktning.



Figur 10: Strålgång ritas genom prickar där laserstrålen träffar.

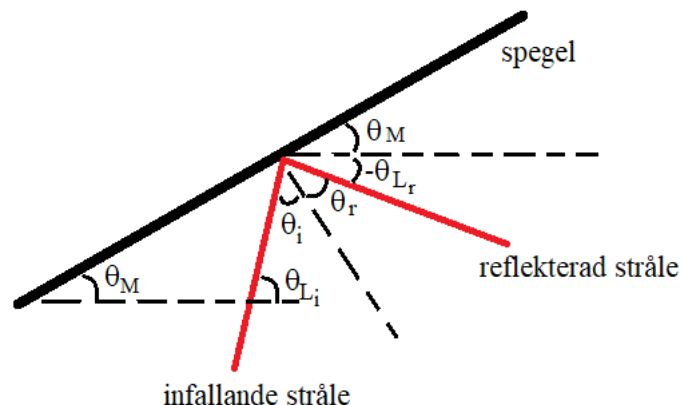
Speglarna byggdes upp av flera fixturer, en för den reflekterande ytan och en för resten av kroppen. Detta gjordes för att på ett enkelt sätt avgöra om strålen träffade rätt sida av spegeln för att reflekteras eller inte.

För att utnyttja reflektionslagen, det vill säga att för en stråle som träffar en spegel är reflektionsvinkeln lika stor som infallsvinkeln, och bestämma i vilken riktning laserstrålen skulle gå efter en reflektion behövdes följande beräkningar. Laserstrålens riktning samt spegelns rotation definierades med hjälp av vinklar mätta från en horisontell axel, se Figur 11. Här beskriver θ_{L_i} infallande laserstrålens vinkel, θ_{L_r} reflekterad strålens vinkel och θ_M spegelns vinkel. Yttervinkelsatsen ger $\frac{\pi}{2} - \theta_i + \theta_M = \theta_{L_i}$. Vidare ger den räta vinkeln mellan spegeln och normalen $\theta - \theta_{L_r} + \theta_M = \frac{\pi}{2}$. Med $\theta_i = \theta_r$ fås alltså

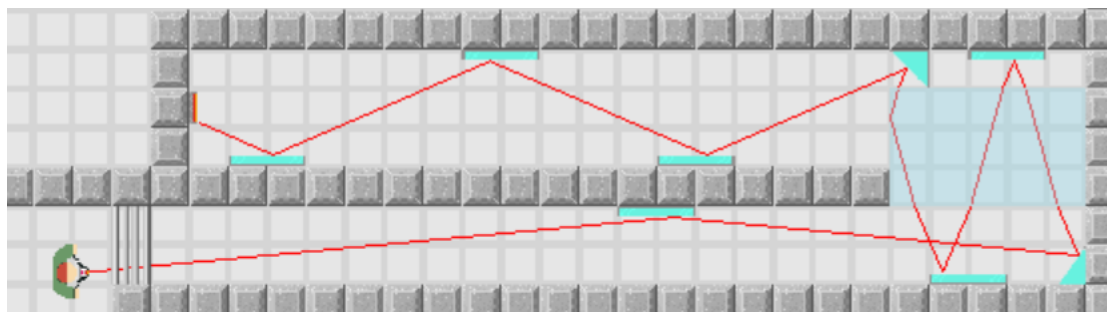
$$\theta_{L_r} = 2\theta_M - \theta_{L_i}.$$

Snells lag, ekvation (27), utnyttjas för att bestämma hur laserstrålen ska färdas när den träffar gränsytan mellan luft och glas. Normalen till sidan av glaset som träffats tas fram för att kunna bestämma strålens vinkel mot normalen, varefter den nya vinkeln beräknas med Snells lag, och konverteras tillbaka till att vara mätt från en horisontell axel.

I fallet då strålen färdas ut från glaset behöver även totalreflektion beaktas, då det kan inträffa när ljus går från ett optiskt tätare till ett optiskt tunnare medium. Detta görs genom att undersöka om infallsvinkeln är tillräckligt stor för att ge totalreflektion, och i så fall bestäms strålens nya riktning istället på samma sätt som för speglarna.



Figur 11: Hur en laserstråle reflekteras mot en roterad spegel. Spegelns och laserstrålarnas vinklar är mätta från en horisontell axel. Då $\theta_i = \theta_r$ fås reflekterade strålens vinkel som $\theta_{L_r} = 2\theta_M - \theta_{L_i}$.



Figur 12: Hur strålgången ritas ut när en knapp träffas. Då visas hela strålgången för att göra det tydligare för spelaren när den träffar, samt ge ökad förståelse för hur det blev en träff.

Vi valde att låta strålen antingen endast brytas eller endast totalreflekteras i glaset, medan det är mer realistiskt att den både bryts och reflekteras upp till vinkeln då totalreflektion sker. Om vi skulle följt det skulle det innebära att strålen hade delats upp i en reflekterad och en bruten stråle med olika intensiteter, och eftersom strålen ofta bryts flera gånger i samma glas skulle det bli många olika strålar vilket troligen skulle försvårat mycket för spelaren. Speciellt gör valet att endast rita ut var strålen träffar det svårt, då det skulle vara svårt att avgöra vilka prickar som hör till samma stråle. Därför valde vi denna förenkling av fysiken.

Punkten där laserstrålen lämnar ett glas hittas inte direkt med ray casting som tidigare, då kollisioner med en fixtur bara detekteras på fixturens utsida. Ett möjligt sätt att lösa detta är att använda ray casting “baklänges”, så att strålens kollision med glasets detekteras från utsidan. När ray casting-funktionen träffade flera glasobjekt visade det sig dock svårt att se till att brytningen skedde i rätt glas, då det inte längre var den närmaste kollisionen som skulle användas som vid den vanliga “framlänges” ray castingen. Därför löstes detta istället genom att ha en fixtur på varje sida på glaset, samt att hålla koll på om strålen var inuti glas eller inte. På så vis kunde ray castingen göras som vanligt.

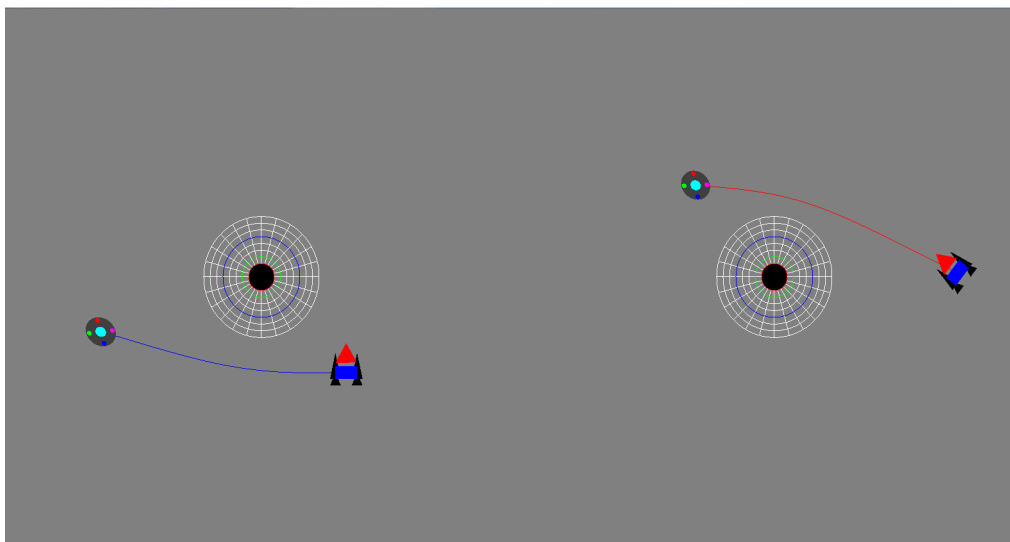
Användartesterna som utfördes under utvecklingen av A Maze in Light lyfte fram en del intressanta problem i mån av speldesign, främst att det var svårt att förstå fysiken som leder till lösandet av pusslarna då laserstrålen inte ritas ut samt att man i vissa fall inte ens märkte när man träffat en knapp. Detta resulterade i valet att göra så att hela laserstrålen ritas ut då spelaren träffat en knapp. Ett exempel på sådan strålgång visas i Figur 12.

5 Resultat

Detta kapitel presenterar vårt resultat och innehåller fyra sektioner. Först presenteras de fem minispel som vi har utvecklat, vilket följs av de idéer vi hade som inte fullföljdes till fullständiga implementationer. Därefter presenteras den kartläggning av fysikområden i spel som är resultatet av vår dokumentation. Slutligen presenteras de observationer som har kunnat göras i koppling till syftet, alltså hur olika fysikområden kan användas i videospel.

5.1 Implementationer

I kommande sektioner presenteras de fem minispel som har utvecklats under projektets gång: Spacetimewar, Hydrocurve, Tic Tac Ket, Lagom Action och A Maze in Light.



Figur 13: Exempelbild på Spacetimewar. Vänstra skärmen visar UFO:ts riktiga position, och var UFO:t ser att rymdfärjan är. Högra skärmen visar var rymdfärjan är och var det tror att UFO:t är. Båda objekt rör sig moturs runt det svarta hålet i mitten.

5.1.1 Spacetimewar

Spacetimewar är ett spel som utspelar sig i rymden där två spelare styr varsitt rymdskepp och försöker skjuta ned det andra. Spelet är tänkt att spelas på två separata skärmar, men använder sig i nuläget av en delad skärm. De två spelarna styr två olika rymdskepp: Ett UFO och ett skepp som är mer lik en rymdfärja. En bild av hur implementationen Spacetimewar ser ut finns i Figur 13.

Kontrollerna för de två olika rymdskeppen i Spacetimewar skiljer sig. Rymdfärjan måste rotera sin spets åt det håll den vill accelerera. Eftersom den skjuter från spetsen kan den alltså inte sikta åt ett håll och accelerera åt ett annat. UFO:t däremot kan accelerera åt vilket håll som helst utan att rotera, och kan snurra runt för att sikta sina skott. UFO:t kan alltså skjuta åt vilket håll som helst, hur den än accelererar. Då rymden inte har något som exempelvis luftmotstånd eller friktion som bromsar skeppen saktar de inte ned bara för att de slutar accelerera. Det finns inte heller några bromsar. För att sakta ned måste de alltså accelerera åt motsatt håll, vilket gör skeppen något svårstyrda.

Fysiken som förekommer i Spacetimewar är gravitation och vissa fenomen från allmän relativitetsteori. Främst märkbar från den senare är informationsfördröjning, men man kan även se viss längdkontraktion då ett skepp närmar sig det svarta hålet. Informationsfördröjningen gör att den ena spelaren inte ser var den andre är, utan bara var den har varit. Detta gör det svårt för spelare att dels sikta på motståndaren, men också att undvika skott som motståndaren skjutit eftersom även projektilernas sedda position påverkas av informationsfördröjningen.

Grafiken i Spacetimewar är simpel 3D-grafik gjord av libGDX-klasserna *ModelBuilder* och *MeshPartBuilder*. Kamera-klassen som används är dock libGDX:s *OrthographicCamera*, vilket innebär

att 3D-grafiken visas i ett perspektiv som gör att grafiken i alla bemärkelser ser ut att vara 2D-grafik.

5.1.2 Hydrocurve

I denna implementation, Hydrocurve, har båda spelarna var sin ubåt på vardera sida av spelplanen. Målet är att skjuta projektiler mot varandra tills endast en ubåt återstår. På spelplanen finns en mängd hinder som försvårar detta, vilket hänger samman med spelets huvudsyfte: att skjuta projektiler med olika fysikaliska krökningar för att träffa ett mål. Detta görs medan du undviker samma typ av projektiler från din motståndare genom att navigera under vattnet och utnyttja hindren och projektilernas krökningsvinklar till din fördel. Projektilerna kröks genom att de har en egen drivkraft som har en annan riktning än färdriktningen, vilket gör att projektilbanan böjs av. Se Figur 14 för ett exempel på hur det kan se ut när spelet spelas.

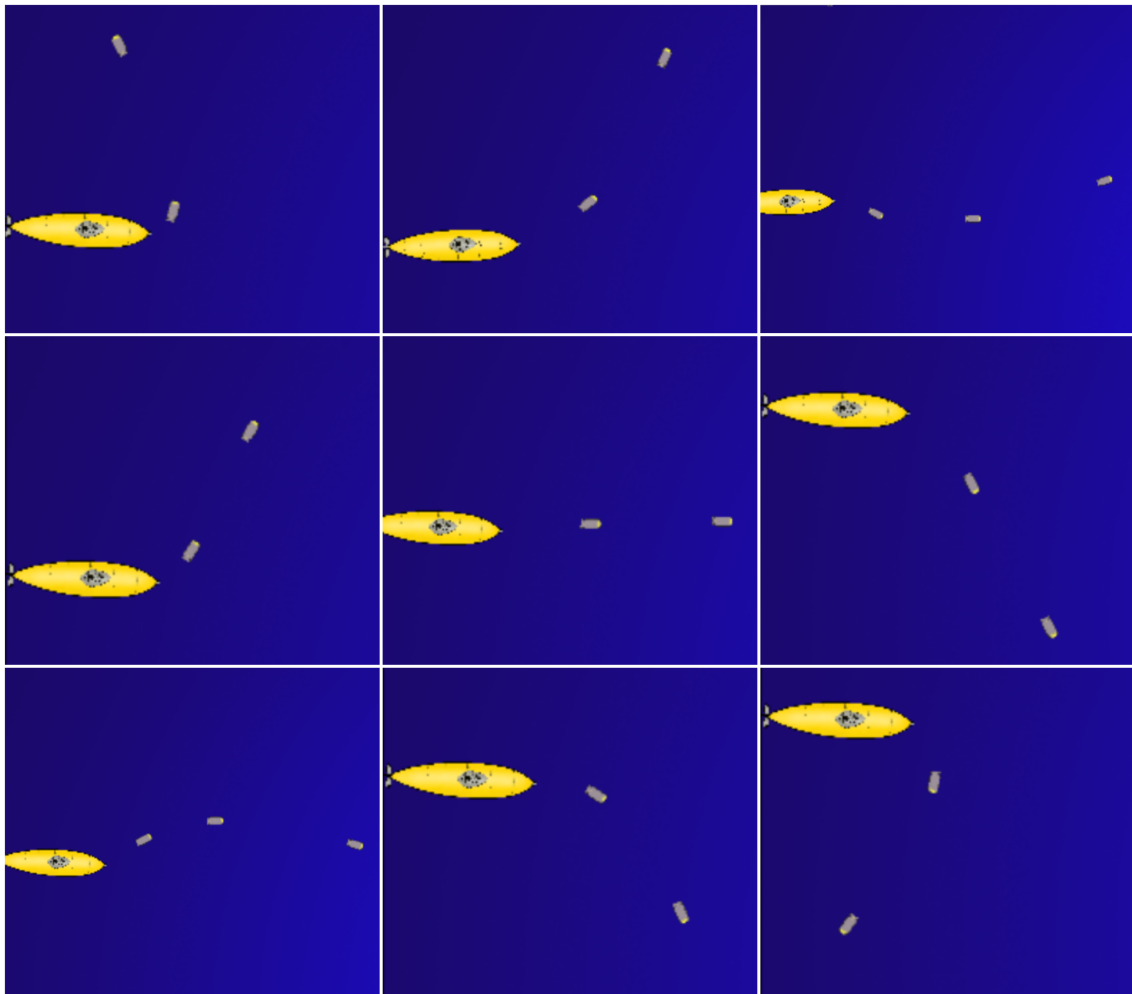


Figur 14: Exempelbild på Hydrocurve. Den vänstra ubåten utnyttjar här en krökt projektilbana för att träffa den andra ubåten.

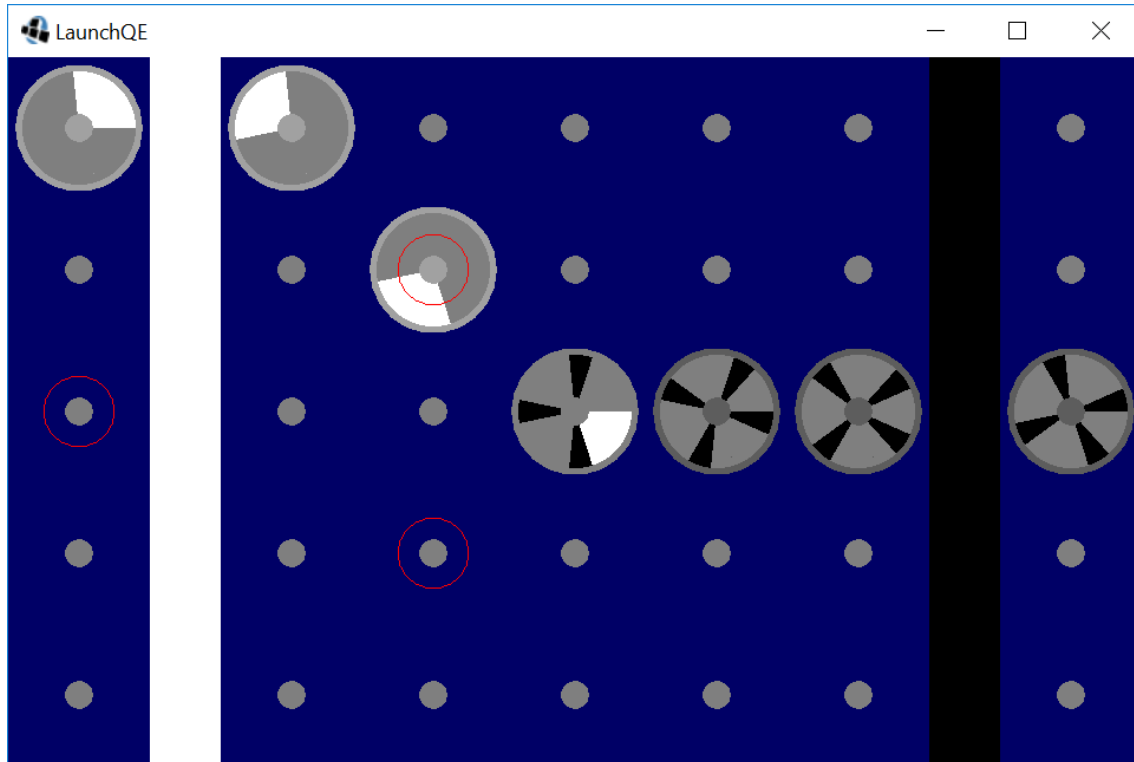
Då vi ville att fokus skulle ligga på projektilerna och deras krökning samtidigt som det skulle gå snabbt för spelaren att skjuta valde vi att implementera ett system där spelaren kan välja olika kombinationer av förutbestämda projektilbanor. Med olika knappar kan spelaren välja om projektilen ska skjutas ut åt vänster, höger eller rakt fram. Utöver det har ubåten även ett läge, som spelaren kan ändra med en knapp, som avgör vilken egen krökning projektilerna kommer att ha: vänster, höger eller ingen. Genom att kombinera dessa fås flera olika resulterande projektilbanor, utan att spelaren behöver ägna någon längre tid på att ställa in sikte. Se Figur 15 för de nio olika möjliga banorna.

För att inte krökningarna ska misstas för gravitation har spelet en ovanifrån-vy, vilket leder till en förenkling av den fysikaliska modellen i avseende att det enda som påverkar ubåtarna är vattnets motstånd, och för projektilerna är det bara motståndet och deras egna drivkraft.

När en ubåt träffas av projektiler tar den skada tills den slutligen exploderar och den andra spelaren har vunnit. Hur mycket "liv" en ubåt har kvar visas med en mätare för varje ubåt i övre hörnen på skärmen, se Figur 14. Bredvid dessa mätare finns även en pil som indikerar vilken projektilkrökning spelaren har valt, alltså åt vilket håll projektilens riktning kommer böjas av åt.



Figur 15: Olika möjliga projektilbanor i Hydrocurve. Kolumnerna anger utgångsriktningen, det vill säga om projektilen skjuts ut åt vänster, rakt fram eller åt höger. De olika raderna visar krökningen, alltså om projektilen viker av åt vänster, inte alls eller åt höger.



Figur 16: Hur ett speltillstånd i Tic Tac Ket kan se ut. Spelplanen har 35 platser. De två 5×1 -kolumnerna på varsin kant är händerna och 5×5 -gittret i mitten är brädet. De små grå diskarna är tomma platser, och de stora är platser där brickor kan finnas. De röda cirklarna är en markering inför en placering, ett drag som kommer att definieras snart.

5.1.3 Tic Tac Ket

Tic Tac Ket låter två spelare spela en kvantmekanisk variant av tre i rad. Båda spelare använder samma dator. Eftersom spelarna har olika information, måste den andra titta bort när den första gör ett drag. Detta underlättas genom att visa en blank skärm när turen går över. Figur 16 visar hur spelet kan se ut.

Tic Tac Ket har en spelplan med 35 platser för brickor, varav 25 av dessa ligger i mitten och kallas för *brädet*. De resterande 2×5 platserna kallas för *händer*, med en hand för varje spelare. Spelplanen i Tic Tac Ket kan även generaliseras till att ha $n^2 + 2n$ platser, varav n^2 är brädet, men $n = 5$ valdes i en avvägning mellan spelkomplexitet och beräkningskomplexitet.

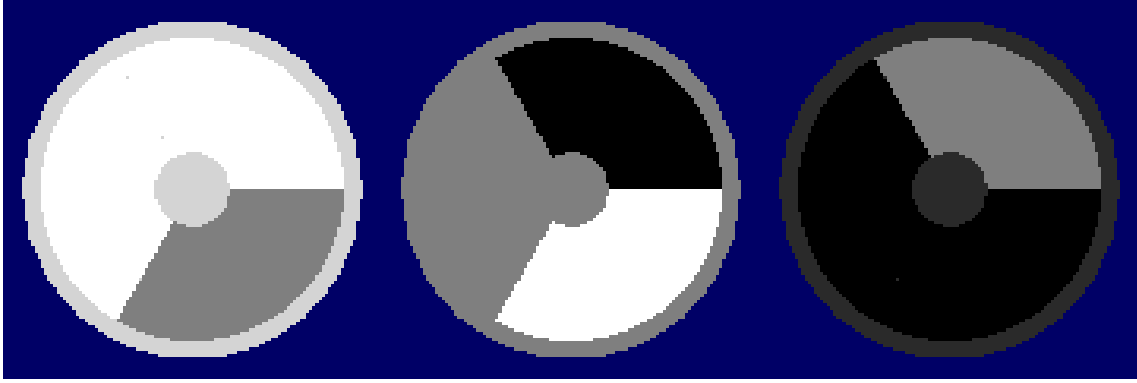
De två spelarna turas om att göra ett av tre drag: placering, observering, eller städning.

Placering Spelaren väljer en tom plats på sin hand och valfritt antal platser på brädet. Spelplanens tillstånd tensormultipliceras med superpositionen av brickan utplacerad på alla valda platser.

Observation Spelaren väljer en plats på sin hand eller en plats på brädet. Den valda platsens innehåll (vit bricka, svart bricka eller tom) kollapsar till ett av de klassiskt möjliga alternativen, och spelplanens tillstånd projiceras ned på det underplan som är kompatibelt med observationen.

Städning Spelaren väljer en bricka i handen, som då tas bort. Detta drag kan inte användas på en plats vars innehåll är en blandad superposition.

Spelet tar slut när varje möjligt klassiskt tillstånd är färdigspelat. Ett klassiskt tillstånd definieras vara färdigspelat om en eller båda spelare har tre brickor i rad. Vinnaren är den som har högst chans att vinna om hela brädet skulle observeras.



Figur 17: Brickors utseende för tillståndet (40) i Tic Tac Ket. Brickorna innehåller tre delar: den yttre ringen, den inre kärnan, och cirkelsektorerna däremellan. Ringen och kärnan färgläggs viktat efter sannolikheten att brickan på den platsen är vit eller svart. Cirkelsektorerna motsvarar termerna från det globala tillståndet. Mer precist är en cirkelsektors färg brickans färg i just det tillståndet.

På grund av de komplexa speltillstånd som kan uppstå är den grafiska presentationen inte trivial. Tic Tac Ket ritar först ut alla 35 platser. För varje plats och term i spelplanens tillstånd, ritas sedan en cirkelsektor på platsen. Dess färg är platsens innehåll i termen, dess tjocklek är sannolikheten för termen, och dess startvinkel är den lägsta som inte krockar med tidigare cirkelsektorer. Slutligen ritas för varje plats en cirkel och en cirkelskiva längs med platsens kant respektive i dess mitt, med färgen som är medelvärde av platsens alla cirkelsektorsers färg. Figur 17 visar ett minimalt exempel, med spelplanstillståndet:

$$|W_1B_2\rangle + |W_1B_3\rangle + |W_2B_3\rangle, \quad (40)$$

där $|W_1B_2\rangle$ är tillståndet med en vit bricka på ruta ett och en svart bricka på ruta två. Detta är ett av de enklaste tillstånd som inte är ett produkttillstånd. Spelare kan lätt läsa av i Figur 17 att plats 2 har lika stor chans att innehålla en vit bricka, svart bricka, eller att vara tom.

5.1.4 Lagom Action

Lagom Action är ett pusselspel för en spelare. Målet är att, givet start- och slutposition, välja en bana för en partikel som så gott som möjligt uppfyller minsta verkans princip:

$$\delta S = 0.$$

Den kinetiska termen är alltid

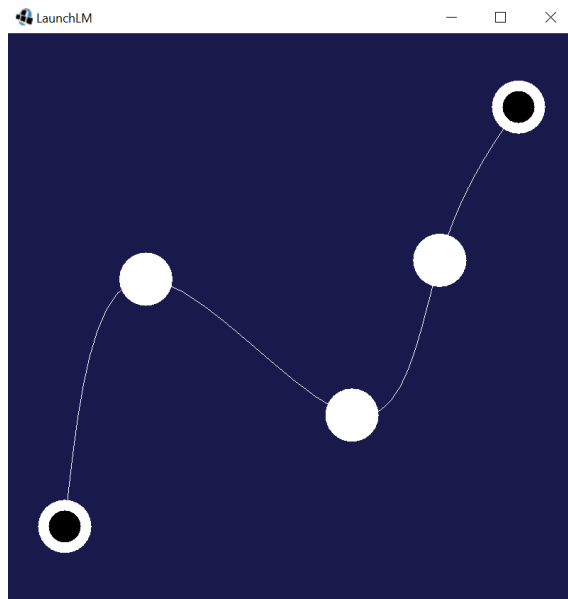
$$T = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2),$$

energin för en fri partikel. Den potentiella termen varierar i spelets olika nivåer.

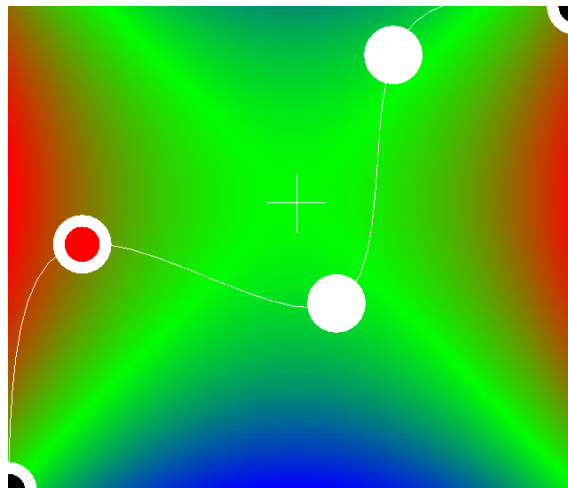
Spelaren väljer en bana genom att flytta *nod*er. En kurva interpoleras sedan genom noderna med en kubisk spline. Figur 18 visar ett exempel med fem noder.

Potentialen förmedlas till spelaren genom att färglägga bakgrunden. Figur 19 visar hur detta ser ut. Fördelen med att förmedla potentialen grafiskt är att spelaren inte behöver sätta sig in i matematiken bakom den.

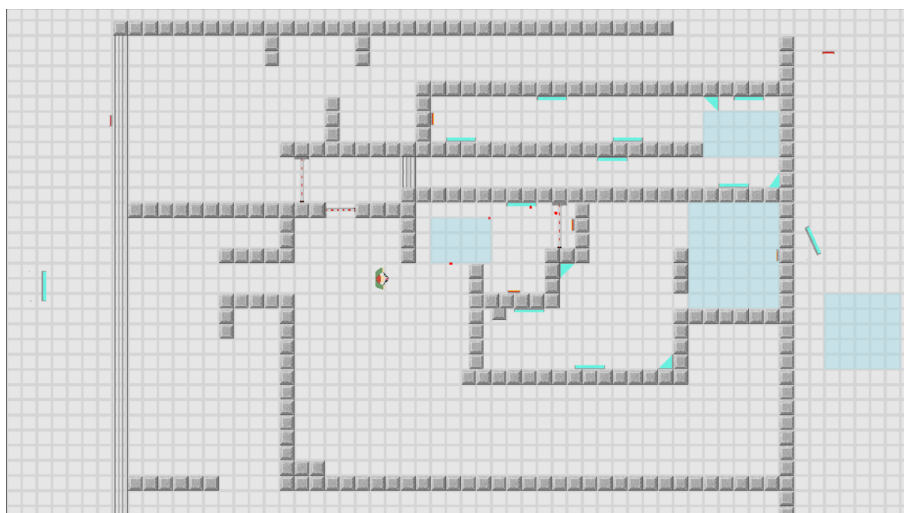
Det finns inget hårt krav för att vinna en nivå i Lagom Action. Varje bana betygsätts efter hur nära minska verkans princip den kommer. Målet blir att komma så nära noll som möjligt.



Figur 18: En partikelbana i Lagom Action. Den tunna vita kurvan är en kubisk spline från de fem noderna. De två noderna med svarta insidor kan inte flyttas av spelaren.



Figur 19: Den färglagda bakgrunden för potentialen $V(x, y) = \frac{1}{2}(x^2 - y^2)$. Potentialen är intressant eftersom den i punkten $(x, y) = (0, 0)$ har en stationär punkt som varken är ett maximum eller minimum. En sådan punkt kallas för en sadelpunkt. Färgen skalas så att potentialens lägsta värde målas blå, och dess högsta värde röd. Det vita krysset markerar sadelpunkten.



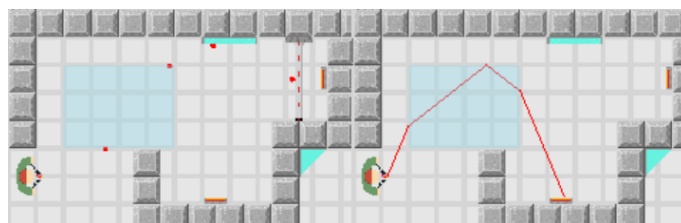
Figur 20: Exempelbild på A Maze in Light. Spelaren använder en laserstråle som den kan reflektera bryta i olika speglar och glas för att träffa knappar och på så vis komma vidare i labyrinten.

5.1.5 A Maze in Light

A Maze in Light är ett spel där man styr en karaktär som ska navigera sig genom en labyrint. Låsta dörrar gör det omöjligt att ta sig igenom vissa delar, och för att låsa upp dessa måste man träffa olika knappar med spelarens laserstråle. Detta görs med hjälp av olika objekt utplacerade runt om i labyrinten. En typ av objekt som vi har inkluderat i spelet är speglar. Dessa kan ha olika vinklar, vilket resulterar i att spelaren måste anpassa skjutvinkeln för att strålen ska reflekteras rätt för att träffa en specifik knapp. En annan typ av objekt som finns är rektangelformade glasobjekt av olika storlekar. I dessa bryts laserstrålen i gränsytan och ges en ny riktning, vilket även det kan behöva utnyttjas av spelaren för att träffa vissa knappar. Spelaren skjuter genom att trycka med musen i riktningen som strålen ska skjutas. Se Figur 20 för en exempelbild av spelet.

Speglarna gjordes i två varianter, en rektangulär, och en triangelformad, anpassad till att vara placerad i hörn. Dessa har båda en fix storlek, medan glasområdena har valfri storlek, men endast i rektangulär form. Dessutom är vissa väggar i labyrinten i form av galler, där spelaren inte kan gå igenom men däremot skjuta laserstrålen emellan. Att träffa en knapp kan förutom att öppna en dörr även omvandla vanliga väggar till gallerväggar, för att öppna för nya delpussel i labyrinten.

A Maze in Light utnyttjar alltså fysikområdet stråloptik, närmare bestämt reflektion i speglar och Snells lag - brytning i gränsyta mellan olika medier. När strålen träffar en glasyta antingen bryts eller totalreflekteras den, istället för att mer realistiskt både brytas och reflekteras för vinklar mindre än gränsvinkeln för totalreflektion. På så vis hålls strålen intakt i en enda stråle, vilket underlättar för spelaren. Strålgången visas genom att röda prickar ritas ut där den träffar, både i spegel- och glasytor och där den slutligen "stannar", vilket är mer realistiskt än att rita ut hela strålgången som ett streck. När laserstrålen träffar en knapp ritas dock hela strålen ut, dels för att spelaren tydligare ska märka när den träffar rätt, och dels för att ge en ökad förståelse för fysiken genom att spelaren kan se strålgången tydligare. I Figur 21 visas de två alternativen av strålgång.



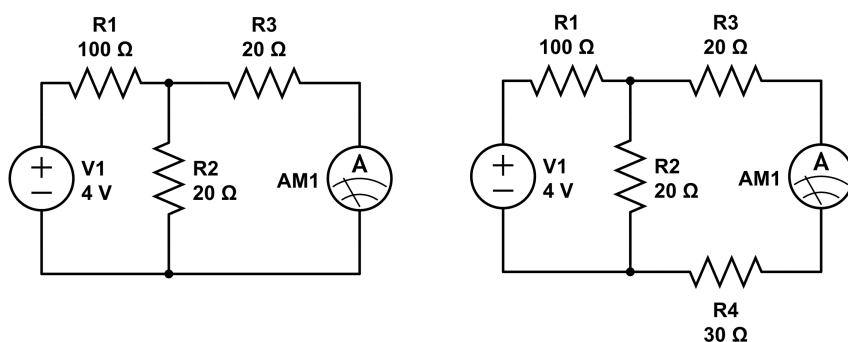
Figur 21: Två sätt att rita ut strålgången i A Maze in Light. Den vänstra visar endast prickar där lasern träffar, vilket är så det ser ut vanligtvis. När spelaren träffar en knapp däremot visas hela strålgången med streck, som till höger.

5.2 Ej fullständigt implementerade spelprototyper

I kommande avsnitt presenteras de spelprototyper som har utvecklats under projektets gång men som inte ledde till några fullständiga implementationer.

5.2.1 Kretslära

Ett område inom fysiken som visade sig inte så välrepresenterat i datorspel är kretslära. En idé vi hade som aldrig blev implementerad var att använda detta till en form av pusselspel. Tanken är att det skulle vara ett spel där spelaren får bygga egna kretsar alternativt placera in saknade komponenter i halvfärdiga kretsar, för att nå specifika mål. Målen kan vara saker som att spänningen eller strömmen ska anta ett visst värde vid ett visst ställe i kretsen, vilket kan illustreras genom att ha en volt- eller amperemeter där som man ska få att visa ett visst värde. Ett enkelt exempel på ett sådant pussel visas i Figur 22. Det skulle också vara möjligt att ge spelaren i uppgift att bygga en krets med en specifik funktion, som ett lågpasfilter eller en förstärkarkrets. Detta begränsas naturligtvis av vilka komponenter som görs tillgängliga i implementationen, vilket till stor del avgörs av hur avancerade beräkningar man kan tänka sig ska utföras.



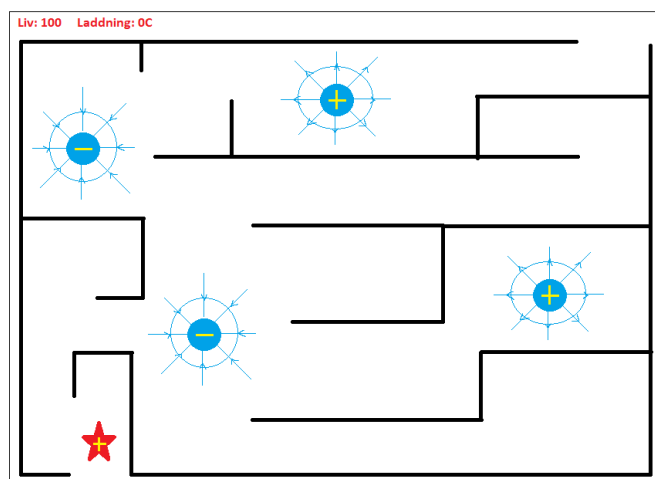
Figur 22: Ett exempel på ett enkelt pussel i elnätsspelet. Spelaren får kretsen till vänster tillsammans med ett antal komponenter som den kan placera ut för att få amperemetern att visa 1 A. Detta kan lösas genom att placera ut en resistor med en resistans på $30\ \Omega$, R_4 i högra kretsen.

Detta skulle till största del resultera i ett spel som ger förståelse för elektriska lagar såsom Ohms lag och Kirchhoffs lagar, snarare än intressant gameplay. Detta kan liknas med simulatorer som låter en bygga en krets, varefter den kan beräkna ström och spänning i olika punkter i kretsen. Skillnaden är de specifika utmaningarna man får som spelare, som kan ge större motivation att lära sig, men det är nog ändå en implementation som framförallt skulle användas för just lärande snarare än underhållning. Det är dock möjligt att det skulle kunna ingå som minipussel i ett större spel, speciellt om nivån hålls relativt enkel.

5.2.2 Elektriska fält

Ett annat område vi såg inte var särskilt använt i spel var elektriska fält och laddningar. Vi hade ett antal olika idéer på spel som använde sig av detta som en central del av spelmekaniken, men inget av dem resulterade i en implementation.

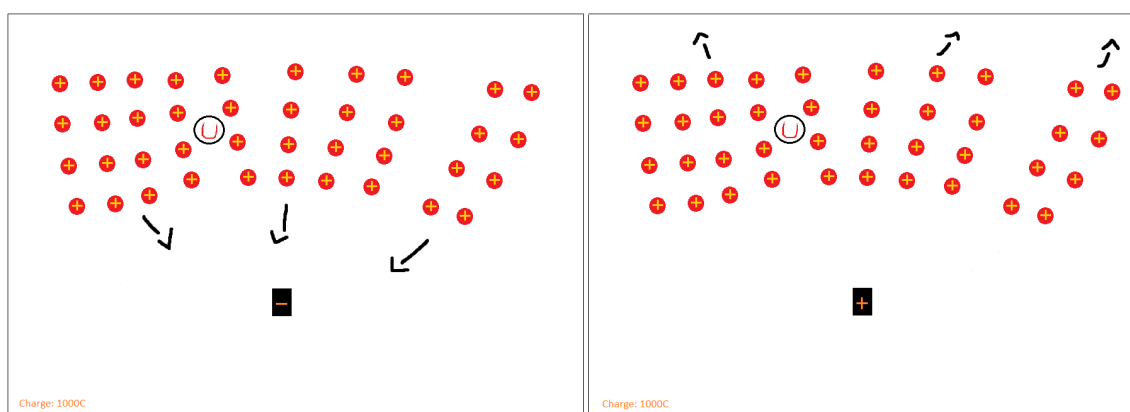
En av idéerna var ett pusselspel bestående av en labyrint och en spelarstyrd karaktär. Olika delar av labyrinten skulle bestå av elektriska fält med olika laddningar och styrkor; spelaren kan sedan påverka styrkan samt tecknet på sin egna laddning till en kostnad av lite av ens liv för att kunna ta sig igenom de olika passagerna. Målet var att navigera sin karaktär genom labyrinten utan att man får slut på liv. Tanken var att spelaren på förhand skulle försöka lista ut bästa sättet att nyttja laddningarna för att ta sig igenom labyrinten innan den spelarstyrda karaktären kastas in i den. Ett antal möjliga strategier för att klara varje nivå skulle finnas, vilket resulterar i att spelaren får använda sig av sina kreativa och utforskande sidor samtidigt som förståelse för den underliggande fysiken är den främst avgörande faktorn. Exempel på hur spelet hade kunnat se ut visas i Figur 23.



Figur 23: Exempel på hur labyrintspelet skulle kunna se ut. Den röda stjärnan är spelaren, och de cirkulära figurerna är de elektriska fälten.

Spelet skulle vara enkelt att skapa, och genom att designa labyrinter av olika utseende och storlek hade det förmodligen även varit enkelt att bygga ut. Ett färdigt spel hade kunnat bestå av flera nivåer med ökande svårighetsgrad som tillåter spelaren att få ökad förståelse för elektriska fält och laddningar i takt med att hen tar sig igenom spelet.

En annan spelidé vi hade inom samma fysikområde var en utökning på ett klassiskt "bullet hell"-spel[97] där projektiler regnar ned på spelaren som denne ska försöka undvika. Träffas man av en projektil förlorar spelaren lite av sina liv; när dessa når noll så förlorar man. Vår nya föreslagna spelmekanik var att införa laddningar till just dessa projektiler som skjuts mot spelaren. Då spelaren blir träffad i detta spel tar denne inte enbart skada utan spelarens laddning ändras även, vilket resulterar i att det blir lättare att undvika projektiler av en laddning, och svårare att undvika de av motsatt laddning. För att göra det hela mer intressant så kan spelaren till exempel byta tecken på sin egna laddning, för att ge spelaren en mer aktiv del av fysikaspekten i spelet. Power ups i form av exempelvis magneter skulle även regna ned med jämna mellanrum för att göra spelet mer intressant och utmanande. Exempelbild på detta spel demonstreras i Figur 24.



Figur 24: Exempel på hur bullet-hell-spelet hade kunnat se ut. I den vänstra bilden har spelaren (den svarta rektangeln) negativ laddning, medan den i högra bilden har en positiv laddning.

Det resulterande spelet hade gett spelaren både kvantitativ och kvalitativ förståelse för den underliggande fysiken då både riktning och värde på de olika laddningarna påverkar de attraherande och repellerande krafterna. Mängden projektiler samt deras placering hade kunnat öka med tiden och således göra spelet svårare ju längre man spelat. Detta hade kunnat resultera i en lagom inlärningskurva för att kunna ta till sig fysiken.

5.2.3 Elektrofall - The Electro Game

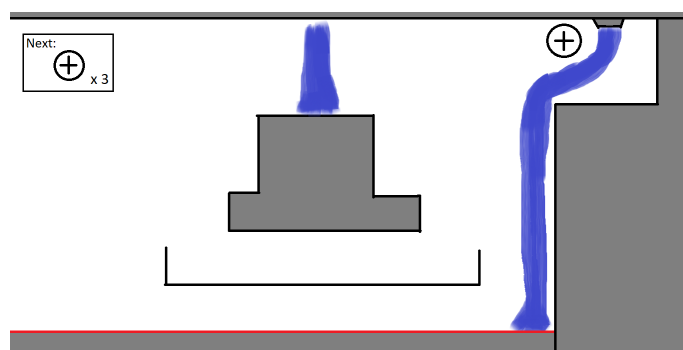
Elektrofall är tänkt att vara ett 2D-spel som bygger på elektromagnetism. Spelaren introduceras till ett flöde av vatten som rinner ned från toppen av en fallgrop. Spelaren har en sidovy. Vattnet kan antas rinna rakt ned utan större spridning, det vill säga att det är en tjock vattenstråle. Målet med spelet är att följa strålen ned till botten av fallgropen. Problemet ligger i att det existerar ett flertal hinder i vägen som absorberar vatten. Spelaren måste bygga någon typ av konstruktion för att leda vattnet förbi alla hinder. Detta görs med hjälp av den så kallade Lorentzkraften som beskrivs i avsnitt 3.9.

Genom att placera ett laddat föremål i närheten av vattenflödet kan en attraktiv kraft skapas mellan det rinnande vattnet och det laddade föremålet som leder till en avböjning hos det rinnande vattnets riktning. Denna typ av attraktiv kraft kommer att användas för att leda vattnet ned från fallgropen och förbi alla hinder. Eventuellt kan andra fluider användas för att ge upphov till en repellerande kraft.

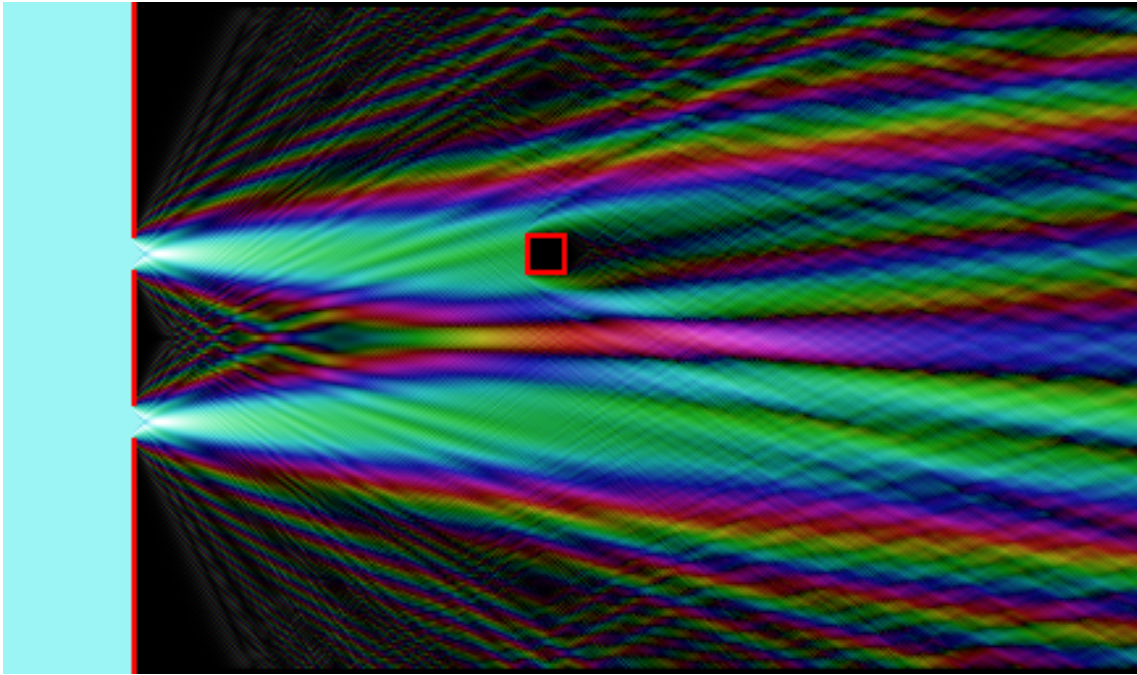
Spelet är tänkt att vara av typen pusselspel och fysiken centreras kring elektromagnetism. Spelmekniken bygger på att man med smarta val ska kunna leda vattenflödet förbi alla hinder utplacerade på kartan och på sådant sätt kunna avancera till nästa nivå.

För att underlätta implementationen kan man anta att vattenflödet har någon typ av laddning som approximeras till att vara konstant över hela vattenstrålen. Sedan har spelaren möjlighet att placera ut laddade föremål som ger upphov till en attraktiv kraft. Denna kraft kan balanseras till att ge upphov till en viss förflyttning i sidled. För att detta ska vara möjligt behöver den attraherande kraften från laddningen vara tillräckligt stark jämfört med gravitationskraften.

Spelet kan försvåras på många olika sätt. Spelaren kan få laddningar i en viss ordning och måste utifrån den givna laddningen lösa problemet, en idé som är inspirerad från Tetris. Spelaren kan även få olika storlekar på laddningarna som förflyttar vattenstrålen 1-3 "rutor". Se Figur 25 för visualisering. Spelaren har en fix mängd laddningar och kan därmed inte förkasta laddningar som inte passade in i spelarens strategi. Målet är att leda vattenstrålen längst ned i fallgropen genom att leda den förbi alla hinder med så få antal utplacerade laddningar som möjligt. Spelaren blir bedömd på hur sparsam den har varit, om spelaren har lyckats leda vattnet ned och tiden det tog att klara nivån. Om spelaren klarar att leda ned vattenstrålen utan att vara sparsam och precis klara kravet så kan spelaren få en stjärna och gå vidare till nästa nivå, medan om spelaren klarar kravet att ta sig längst ned i fallgropen och samtidigt är väldigt sparsam kan spelaren få upp till tre stjärnor. Spelaren har även möjlighet att tävla med sina tre stjärnor mot andra spelare genom att jämföra tiden det tog att klara utmaningen. Frågan om spelet ska vara helt RNG-baserat, det vill säga med helt slumpade laddningar, eller om det ska existera ett mönster för varje enskild nivå är svår. En blandning där de första tre laddningarna är statiska för varje nivå och resterande är slumpade skulle kunna vara en möjlighet.



Figur 25: En kort skiss kring hur spelet *Elektrofall* möjligtvis skulle se ut. Målet är att transportera vattnet ned till den röda linjen. För att ändra riktningen hos vattenflödet används elektriska laddningar. Spelaren blir slumpvalt tilldelad en laddning att placera ut och får poäng beroende på hur sparsam spelarens slutgiltiga lösning är. Vattnet absorberas när den får kontakt med den gråa konstruktionen och spelaren behöver därmed rikta om strålen för att leda vattenflödet förbi hindret. Detta visualiseras av vattenstrålen till höger. Vattenstrålen till vänster (centralt i skissen) visualiserar hur vattnet absorberas av den gråa konstruktionen.



Figur 26: Ljus passerar genom optiska komponenter från vänster. Ljusamplituden beräknas med PAS, och illustreras i färg. Färgen i bilden anges i HSV. H -värdet är ljusets fas dividerat med 2π . S -värdet är $1 - I^2$ och V -värdet är $2I - I^2$, där I är intensiteten normerad så att dess högsta värde är 1. Konsekvensen av dessa formler är att noll intensitet visas som svart, max intensitet visas som vitt, och mellanliggande intensiteter visas som till en färg som beror på faser. De optiska komponenterna är en dubbelspalt och en ljusblockerande kloss, båda utritade i som röda heldragna linjer. Topp- och bottenkanterna på bilden är reflekterande.

5.2.4 Fourieroptik

Fourieroptiken skiljer sig från vågoptiken genom att erkänna vågnaturen hos fotoner. Vi hade därför idéer om att göra en spelmekanik av ett fenomen som är beroende av fotonens vågnatur. Exempel på detta är diffraktion och realistisk linsfokusering.

Ett pusselspel skulle kunna gå ut på att monokromt ljus infaller från vänster. Spelaren får placera ut optiska komponenter, till exempel linser, spalter, speglar, och gitter. Målet är att få tillräckligt mycket intensitet på ett litet målområde.

Visualieringen av ljuset kan göras på flera sätt. Eftersom ljuset är monokromt är det frestande att färglägga i den färgen, med opacitet (alpha) proportionellt med intensiteten. Ett annat alternativ är att låta färgvärdet (talet V i HSV) vara proportionellt mot intensiteten, och låta nyansen (talet H i HSV) motsvara faser på ljuset. Figur 26 visar hur det kan se ut när färgen motsvarar faser.

5.3 Kartläggning

Detta kapitel innehåller sammanställningen av vår dokumentation av gamla spel. För att sammanfatta vilka de mest vanliga fysikområden i spel är, och samtidigt jämföra med de områden som har undersökts, presenteras allt i Tabell 2. En sammanställning av alla spel som använts för denna kartläggning kan hittas i avsnitt A.9 i Bilaga A.

Tabell 2: Histogram över spelmekaniker, efter spelgenre och fysikområde. Varje rad motsvarar ett fysikområde, och varje kolumn en spelgenre. Siffran i parentes efter spelgenren indikerar hur många spel i den genren som har undersökts. Ett spel kan innehålla spelmekaniker från flera fysikområden.

Fysik \ Spelgenre	Spelgenre									
	Fotbollsspel (4)	Pusselspel (14)	FPS (5)	Racingspel (6)	Action-Adventure (5)	Plattforms spel (11)	Övriga spel (8)	Totalt (53)	Våra spel (5)	Våra speldéer (5)
Mekanik	4	10	4	5	4	11	6	44	0	0
Optik	0	4	0	0	1	0	0	5	1	1
Elektromagnetism	0	0	0	0	3	2	0	5	0	3
Relativitetsteori	0	0	0	0	0	0	2	2	1	0
Kvantfysik	0	1	0	0	0	0	1	2	1	0
Strömningsmekanik	1	1	1	0	0	0	3	6	1	0
Termodynamik	0	0	0	0	0	0	1	1	0	0
Kretslära	0	0	0	0	0	0	0	0	0	1

5.4 Våra observationer om möjligheten att använda fysik i spel

I detta avsnitt presenteras de observationer som har gjorts gällande möjligheten att implementera fysik från olika områden. De är tänkta att svara på syftet, vilket är att utforska designmöjligheterna i gameplay för olika fysikområden. Resultaten är subjektiva eftersom vi inte har kunnat mäta designmöjligheten för ett fysikområde kvantitativt.

5.4.1 Möjligheten att använda relativitetsteori för gameplay

Det som skiljer relativitetsteori från newtonsk mekanik är beteendet vid höga hastigheter. I relativitetsteori finns det en högsta hastighet, och vid jämförbara hastigheter blir tidsdilation ett påtagligt fenomen.

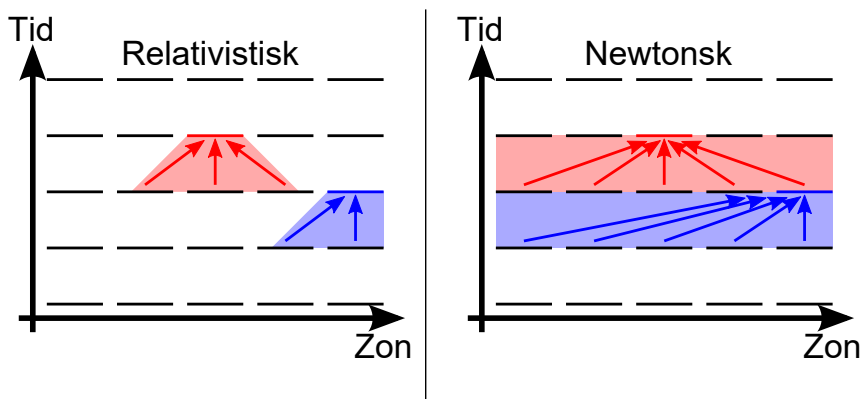
Existensen av en högsta hastighet kan vara ett användbart verktyg i speldesign, men det är inget nytt. Racingspel har alltid haft en högsta hastighet motiverad av bilmotorns begränsning. Ett universum med en högsta hastighet är enklare beräkningsmässigt, eftersom information från en händelse sprids i en kon, istället för att påverka hela universum för senare tider. Detta är en fördel då universumet är indelat i lokala zoner. Figur 27 visar hur nödvändig information begränsas av en högsta hastighet.

Konsekvensen är att fysikaliska fenomen blir enklare att simulera realistiskt om de behandlas relativistiskt.

Då det finns en högsta hastighet, begränsar den även information, vilket innebär att information om avlägsna platser alltid är fördröjd. Att få fördröjd information om motståndaren kan implementeras som mekanik, både i real-time och turn-based konfliktspel. Spelare måste då extrapolera information om motståndaren utifrån den givna fördröjda informationen.

Fenomenet tidsdilation utforskades inte utförligt i Spacetimewar, men kan vara en intressant mekanik i ett racingspel. Vid höga hastigheter minskar upplevd tid (egentid) per tid i referenssystemet. En konsekvens blir att föraren får kortare reaktionstid och mycket längre bromssträcka vid höga hastigheter. Detta är sant även i newtonsk mekanik, men det blir mer extremt i relativitet.

Krökning av rummet är ett fenomen som fanns med i Spacetimewar, men blev där inte en påtaglig mekanik. Fenomenet tillåter intressant världsdesign, som inte skulle vara möjlig i euklidisk geometri. För att få större frihet i design av världen kan krökningen låtas vara godtyckligt vald, istället för att bero på massa.



Figur 27: Rumstidsdiagram över information. Den vertikala axeln innehåller de diskreta tidsstegen, och den horisontala innehåller zoner i rummet. De skuggade områdena markerar händelser som kan påverka zonen i nästa tidssteg. I det relativistiska fallet, med en högsta hastighet, behövs bara information från närliggande zoner. Det är en fördel över det newtonska fallet, där information från alla zoner behövs för att simulera fysiken realistiskt.

5.4.2 Möjligheten att använda strömningsmekanik för gameplay

Att implementera korrekt strömningsmekanik och vattenmotstånd är ofta svårt, då det är många faktorer som spelar in och som man då behöver ta hänsyn till. Att beskriva luft- eller vattenmotstånd är något som ofta görs genom experiment och uppmätt data. Motståndskoefficienten C_D beror på många olika faktorer så som kroppens geometri och material, men även Reynoldstalet som varierar med vätskans hastighet relativt kroppen. Detta gör att C_D inte är en konstant, men den kan ofta approximeras som konstant om hastigheten hålls inom ett tillräckligt litet intervall. Att bestämma dess värde är ändå svårt, speciellt om kroppen har en komplicerad form. Ofta kan den istället approximeras som enklare former, för att då kunna använda uppmätta värden från tabeller. Värt att notera att det då ofta inte tas hänsyn till material, så värdet blir ännu mer osäkert. Det är dessutom naturligtvis enklare att beskriva en motståndskraft för ett föremål som rör sig rakt framåt, men om det samtidigt svänger eller roterar, blir det svårare.

Det finns alltså många svårigheter att beräkna storleken på hydrodynamiska eller aerodynamiska motståndskrafter, och många approximationer behöver göras. Om den barriären passeras finns det dock inga större hinder att implementera detta i gameplay. Att implementera dessa krafter realistiskt kan dock innebära sämre spelupplevelse då det kan bli alltför komplicerat för spelaren. Detta sågs dels angående luftmotstånd och andra realistiskt implementerade fysikaliska fenomen i *FIFA 14* i avsnitt 2.7.3, dels i vår implementation Hydrocurve, där styrningen blev ett svårt moment, även om det inte enbart var en följd av vattenmotståndet.

5.4.3 Möjligheten att använda kvantfysik för gameplay

Möjligheten att använda kvantfysik som en spelmekanik begränsas av komplexitet på två sätt.

Den första begränsningen kommer från beräkningskomplexiteten och lagringsutrymmet, som båda växer exponentiellt med systemets storlek. Detta sätter en övre gräns på hur stora system som kan simuleras. Dock finns det sätt för att nå längre under vissa omständigheter. Som exempel, se på spelplanen i Tic Tac Ket. Den kräver egentligen 10^{53} byte för att lagra ett tillstånd, men med gles informationslagring används vanligtvis mycket mindre. Problemet med gles informationslagring är att lagringsutrymmet växer över tid, och till slut kan bli något större än dess naiva motsvarighet.

Ett möjligt sätt att minska komplexiteten är att införa en spelmekanik där spelare vill undvika komplexitet. Komplexitet kan reduceras med observationer. I Tic Tac Ket kan en observation i bästa fall dividera lagringsutrymmet med tre.

Den andra begränsningen kommer från svårigheten att presentera ett tillstånd för ett komplext system. Om all information om tillståndet presenteras blir det överväldigande för spelare. Om informationen istället förenklas till några få variabler som till exempel väntevärden, går djupare information som exempelvis korrelation förlorad. I Tic Tac Ket prövades båda alternativ. Det mest användbara alternativet var väntevärdena.

5.4.4 Möjligheten att använda Lagrangeformalismen för gameplay

Lagrangeformalismen skiljer sig från de andra undersökta områdena. Den är inte ett fysikaliskt område, utan ett sätt att uttrycka tidsutvecklingen av system från ett godtyckligt område. På grund av att det inte finns något associerat fenomen, är det svårt att basera en mekanik på området.

Designrymden för lagrangeformalismen bedöms vara begränsad. Implementationen Lagom Action var baserad på kanske det enda "fenomenet" i området, minsta verkans princip, och där upplevde vi att designmöjligheterna snabbt tog slut. Visserligen kan potentialen göras godtyckligt komplicerad, och partikelbanan kan ges fler noder, men det är bara artificiell svårighet som inte gör pusselaspekten djupare.

I varje fall där lagrangeformalismen kan användas, är det bättre att använda det underliggande fysikområdet, om möjligt. Tag som exempel potentialen $V(x) = x^2$. Den är svår för en spelare att visualisera eftersom den är en formel utan fysikalisk kontext. Om spelet istället skulle presentera till exempel en stämgaffel, har spelaren en intuition för hur den beter sig. Det är därför att föredra konkreta system över abstrakta.

5.4.5 Möjligheten att använda stråloptik för gameplay

Stråloptik är något som har använts i många spel tidigare, ofta i pussel spel där optiken är huvudfokus i gameplay. Det har också använts i andra typer av spel, där det istället har varit ett delmoment bland många andra utmaningar som spelaren stöter på. Stråloptik är i dessa spel ofta begränsad till reflektion i speglar, vilket alltså verkar vara ett välfungerande koncept. Det kan förstås genom att reflektion av en laserstråle är ganska intuitiv även om spelaren saknar vidare kunskap om fysiken, speciellt när hela strålgången ritas ut. För mer avancerade optiska komponenter krävs ofta större förståelse för den bakomliggande fysiken, vilket gör dem svårare att använda i spel.

Vår implementation, A Maze in Light, innehåller förutom speglar även glasområden där laserstrålen kan brytas eller totalreflekteras. Hur strålen bryts är ett lättförståeligt fenomen; även om spelaren inte känner till det sedan tidigare förstås det enkelt från en tydlig illustration i spelet. Totalreflektion är ett något mindre intuitivt koncept, vilket gör att det behöver introduceras för spelaren på ett tydligare sätt, för att spelaren ska förstå hur det kan nyttjas. Ifall kunskap om den bakomliggande fysiken saknas kan det lätt uppfattas som slumpvist när totalreflektion respektive brytning sker i en glasyta. Om det däremot introduceras så att spelaren får den nödvändiga förståelsen, anses fenomenet kunna användas i gameplay och skapa en större variation i spel som använder optik.

Ett problem som uppkommer med brytning mellan olika medier är dock att strålen egentligen både bryts och reflekteras i ytan så länge infallsvinkeln är lägre än gränsvinkeln för totalreflektion. För att illustrera detta realistiskt behöver strålen delas upp, och illustreras med olika intensiteter. I A Maze in Light, där strålgången endast visas som prickar där den träffar, ansågs detta bli alltför komplicerat för spelaren, då det inte går att se vilka punkter som hör till samma stråle. Om strålgången istället ritas med ett streck blir det mer hanterbart, men beroende på hur det implementeras kan det fortfarande bli onödigt komplicerat och förvirrande för spelaren. Det är alltså ett tydligt fall där en kompromiss mellan realistisk fysik och bra gameplay kan behöva göras.

Valet att rita ut hela strålgången, som gjorts i de tidigare spel som vi har undersökt, förstås genom att det gör det enklare för spelaren att förstå hur laserstrålen färdas. Att istället, mer realistiskt, endast rita ut var strålen träffar är något vi valde att göra, och som fungerade väl i vår implementation även om det försvårade för spelaren. Så beroende på spelets syfte är även detta ett fall där en kompromiss mellan fysik och gameplay kan behövas.

6 Diskussion

I kommande avsnitt presenteras diskussionen av våra resultat och vår metod, en analys av etiska och samhällsliga aspekter kopplade till projektet samt en diskussion kring hur vårt arbete kan kopplas till ett större sammanhang, dels i hur generellt vårt resultat är och dels hur framtida arbete inom området kan se ut.

6.1 Diskussion av resultat

Detta avsnitt diskuterar vad vår dokumentation av tidigare spel kom fram till, och utvärderar därefter både de implementerade spel och de ej fullständigt implementerade spelprototyper som detta projekt har resulterat i.

6.1.1 Diskussion av analysen av tidigare spel

Kartläggningssmatrisen antyder att mekanik är det fysikområde som är överlägset mest representerat inom spelbranschen idag. Detta har antagligen mycket att göra med att all rörelse och förflyttning av kroppar är mekanik, vilket är grundläggande funktioner i de allra flesta observerade videospel. Man kan se att pusselspel sticker ut som den kategorin där flest fysikaliska fenomen förekommer som central del av gameplay. Analysen visar även att det ofta är inslag av denna typ av pussel i andra spel som ger upphov till intressant användning av fysik i dessa. I de allra flesta spel verkar fokus ha varit på att göra användarupplevelsen så bra som möjligt, och realistisk fysik har då implementerats i de fall där den kan bidra till just detta. För att implementera realistisk fysik kan det ofta krävas att tunga beräkningar görs. Det verkar därför som att begränsningar i tekniken har varit en stor bidragande faktor till att realistisk fysik blivit alltmer framträdande i modernare spel. Vi har observerat fall där en mer realistisk fysik har implementerats men har lett till en sämre spelupplevelse för användarna enligt kritiker. Exempelvis var detta fallet för *FIFA 14* [6] där man kunde observera en tydlig utveckling hos användningen av fysik [7] men där kritiker ansåg att utvecklingen bidragit till en negativ påverkan på gameplay [59]. Inom exempelvis plattformsspel utformade vi en hypotes att *Super Mario Bros.*, ett väldigt framgångsrikt plattformsspel, kan ha haft en väldigt starkt inflytande på spelgenren och bidragit som inspiration till många utvecklare som skapat plattformsspel med en liknande spelmekanik. Hypotesen fortsätter med att en undersökning kring mer exotisk användning av fysik inom plattformsspel kan ha överskuggats på grund av *Super Mario Bros.* troligtvis starka inflytande. Det har observerats att spelutvecklare har experimenterat med en mer realistisk fysik, som har kritiserats både negativt och positivt.

Dokumentationen och analysen av existerande spel är baserade på ett subjektivt urval av videospel och kan inte ses som en fullständig representation av spelmarknaden. För att skapa en mer rättvis representation av spelmarknaden studerades olika spelgenrer som enligt oss skulle täcka en signifikant del av spelmarknaden. Dock är även spelgenrererna baserade på ett subjektivt urval där vi möjligtvis kan ha uteslutit spelgenrer där fysik förekommer mer frekvent.

6.1.2 Diskussion av Spacetimewar

Spacetimewar togs främst fram för att undersöka om relativitetsteori kan bidra till gameplay i spel. Detta på grund av att ytterst få spel verkade använda sig av allmän relativitetsteori, vilket kan ses av dokumentationen, och eftersom vi kunde se potential för att det skulle lägga intressanta speldynamiker. För att de relativistiska fenomenen skulle vara någorlunda rimliga valdes rymden, runt ett svart hål, som platsen där spelet skulle utspela sig. Därmed gick tankarna snabbt till att göra något som liknar *Spacewar!*, fast med allmän relativitetsteori tillagd som spelmekanik.

De främsta synliga effekter som uppkom av den implementerade relativitetsteorin var längdkontraktion och informationsfördröjning. Längdkontraktionen gav mest bara upphov till att formen på skeppen och projektilerna ändrades, inte någon ny dynamik. Informationsfördröjningen, däremot, påverkade gameplay väldigt mycket.

Informationsfördröjning är fenomenet som gör att den signal med information som alla objekt skickar ut, och som en mottagare bland annat kan använda för att exempelvis se objektet, inte når fram till en mottagare förrän mottagaren är inom en viss räckvidd. Denna räckvidd utökas med ljusets hastighet, det vill säga en konstant hastighet. Detta innebär att en mottagare som är dubbelt så långt ifrån objektet som en annan, fångar upp signalen efter dubbelt så lång tid som

den mottagare som är närmre objektet. Därmed får den mottagare som är längre bort även mer utdaterad information än den andre. I Spacetimewar syns detta fenomen väldigt tydligt, eftersom avstånden mellan skeppen, i spelets värld, är flera ljussekunder - alltså väldigt stort. Därmed når inte skeppens signaler med information det andra skeppet förrän efter signalen färdats en stund. Det svarta hålets närvaro förstärker det här något eftersom dess enorma gravitation ger upphov till tidsdilation, vilket i detta fall innebär att en sekund för ett skepp nära det svarta hålet kan vara flera sekunder för ett skepp längre bort. Informationsfördröjningen hade skett ändå, eftersom skeppens avstånd från varandra är så stort, men det svarta hålet förstärker det något, i synnerhet då ett skepp är väldigt nära det svarta hålet.

Underhållningsvärdet i Spacetimewar enligt MDA är främst *Challenge*, precis som i *Spacewar!*, eftersom det är en utmaning att manövrera sitt skepp och sikta och skjuta på motståndaren. Den tydliga informationsfördröjning gör dock att det blir väldigt mycket svårare att träffa motståndaren med sina skott, eftersom spelarna inte vet var den andra är, bara vad den varit, såvida de inte är väldigt nära varandra. Detta gör att det blir ganska meningslöst att försöka skjuta motståndaren från långa avstånd, eftersom chansen att träffa är väldigt liten. Istället blir spelarna nästintill tvungna att försöka leta upp varandra för att kunna träffa varandra. På så nära avstånd blir det dock inte särskilt utmanande att träffa det andra skeppet, vilket gör att utmaningen blir väldigt mycket mindre. Så ironiskt nog blir utmaningen mindre av att den tillagda fysiken gjort spelet mer utmanande, vilket också gör att spelet blir mindre roligt.

Även om målet med implementationen anses vara nått, finns det många brister med spelet. Till exempel saknas många funktioner som kan förväntas av ett spel, som en vinstskärm eller att spelet startas om när någon har vunnit. Det är dessutom lite svårare än det kanske borde vara att styra skeppen, och det finns nog en del balansering av spelet att göra, till exempel att öka hastigheten på skeppens skott eller kanske öka gravitationspåverkan från det svarta hålet, som nu inte märks så mycket. Det finns alltså en del att jobba vidare på i spelet, men eftersom det huvudsakliga finns där, alltså en fungerande demonstration av allmän relativitetsteori i spelet, lades ingen mer tid på att göra Spacetimewar till ett fullt fungerande och "bra" spel. Den nuvarande implementationen anses räcka för att bidra till projektets syfte.

6.1.3 Diskussion av Hydrocurve

Hydrocurve är ett spel som utmanar spelaren genom att skjuta krökta projektiler för att träffa sitt mål. Detta kan kopplas till den fjärde typen av underhållning inom MDA, *Challenge*. Spelet innehåller dessutom en viss del av *Discovery*, då spelaren får utforska nya sätt att skjuta projektiler, och kan skjuta runt hinder med de krökta projektilbanorna. Detta kräver en viss förståelse för fysiken för att man ska kunna förutse hur projektilen kommer färdas, men förståelsen är mestadels kvalitativ då det räcker att förstå vilket håll projektilen kommer krökas av åt och spelaren inte reglerar krökningsvinkeln själv. När det gäller den andra delen av fysiken, vattnets motstånd mot ubåten, varken kräver det eller ger så mycket förståelse för spelaren. Dessutom är fysiken i spelet förenklad, då approximationer har gjorts i vattenmotståndet, till exempel angående kropparnas form. Då motståndskraften ändå inte märks så tydligt i spelet ansåg vi att dessa approximationer var rimliga i förhållande till hur mycket det påverkade spelupplevelsen.

Valet att försöka implementera realistisk styrning av ubåtarna resulterade i en entydlig åsikt bland speltestare om att det var svårt att styra; vissa användare ansåg detta vara den absolut svåraste aspekten i spelet. Det var även få av användarna som kände av att fysiken var den centrala delen av spelet, möjligtvis då fokus istället behövde läggas på styrningen. Att veta hur man skulle kombinera de olika krökningarna med de möjliga riktningarna var initialt ett svårt moment för de flesta speltestarna, dock verkade inlärningskurvan vara ganska brant. En majoritet av användarna verkade tycka att spelet är roligt i något avseende. Man kan även se potential för att lägga till denna typ av krökta projektiler i redan existerande spel, som exempelvis olika spel inom FPS-genren. Detta hade då kunnat ge en ökad utmaning och intressantare gameplay.

6.1.4 Diskussion av Tic Tac Ket

Syftet med Tic Tac Ket var att undersöka möjligheten att använda kvantfysik för gameplay.

Från speltestande har vi funnit att spelet har tre faser, med mjuka övergångar. Det finns tumregler för vad ett bra drag är i dessa faser.

I början av spelet är det viktigast att placera ut brickor, men inte så viktigt var de hamnar. Ett typiskt bra drag är att placera ut brickan på många platser, så att risken att den observeras i handen minskar.

När flera brickor ligger ute ökar värdet av att observera brädet. Eftersom en observation kollapsar tillståndet, men bara ger information till en spelare, gör det att spelaren som har observerat ligger steget före. Observationen kan också användas på platser i handen, för att göra den platsen tom, vilket tillåter ett till placeringsdrag. Om spelaren tidigare har placerat brett, det vill säga placerat enskilda brickor på många platser, är risken för att finna en bricka i en plats i handen liten.

När tillräckligt många brickor ligger ute för att tre-i-rad-konfigurationer ska vara möjliga, övergår fokus till att skifta sannolikheterna för dessa fördelaktigt. Verktyg för detta är observationer av brädet eller smala utplaceringar, det vill säga att bara platsen i handen och en eller två platser till väljes. Om smala utplaceringar görs för mycket blir händer fulla med brickor, och städning måste användas.

Efter begränsad speltestning på två timmar har inga triviala vinnande strategier funnits, vilket är ett gott tecken. Ett problem under speltestningen var att upprepade breda utplaceringar inte kunde göras, eftersom beräkningstiden till slut blev för stor. Det kan därför finnas oupptäckta strategier som involverar många breda drag.

6.1.5 Diskussion av Lagom Action

Lagom Action är ett enkelt abstrakt spel där spelaren väljer en bana och får ett betyg för det. Den resulterande speldynamiken blir en iterativ process där spelaren

1. Får se betyg för den nuvarande banan.
2. Listar ut vad i banan som begränsar betyget.
3. Ändrar banan för att kompensera.

Detta liknar speldynamiken för spel i brobyggargenren, där spelaren iterativt

1. Testar om bron håller.
2. Observerar var bron går sönder.
3. Läger till byggmaterial.

Till nackdel för Lagom Action är dock dess gameplay mycket simplare än det i ett brobyggarspel. I det senare kan spelaren välja form, material, och dimensioner med stor variation. I Lagom Action kan bara modernas positioner väljas, vilket är betydligt mer begränsat.

Lagom Action kan göras mer komplicerat genom att lägga till fler termer i potentialen, eller genom att lägga till fler noder. Detta är dock artificiell svårighet som inte kräver djupare förståelse, utan kan lösas med samma iterativa lösningsprocess. Lagom Action saknar därför utrymme för kreativa lösningar.

Eftersom Lagom Action är så abstrakt, är det svårt för spelaren att lära sig något om fysik genom att spela det. Igen en jämförelse med brobyggargenren: spelare kan där förstå vilka konstruktioner som fungerar, och varför broar ser ut som de gör. I Lagom Action finns ingen motsvarighet, eftersom spelet inte beskriver något konkret vardagligt system.

6.1.6 Diskussion av A Maze in Light

Utmaningen i A Maze in Light är först och främst att avgöra vilken vinkel man måste skjuta laserstrålen med för att få den att reflekteras eller brytas med rätt vinkel för att träffa de olika knapparna. Detta kan kopplas till *Challenge*-typen av underhållning i MDA-ramverket. Då vinklarna i många fall måste vara exakta ger detta både en kvalitativ och en kvantitativ förståelse för den underliggande fysiken. En annan utmaning i spelet är att utnyttja inte bara spelarens position utan även omgivningen såsom speglar, glas och gallrade väggar för att styra strålen till sitt mål. Även detta kan kopplas till *Challenge* men även till *Discovery* då spelaren kan utforska olika sätt att reflektera en laserstråle på, exempelvis med flera speglar samtidigt eller genom totalreflektion i glas.

För att få en mer verklighetstrogen visualisering av laserstrålarna valde vi att endast rita ut där lasern träffar. Detta bidrog inte enbart till ökad realism utan även till svårare gameplay vilket resulterade i en större utmaning för spelaren. Det kan även argumenteras för att det ger en ökad förståelse för den underliggande fysiken då det blir svårare att gissa sig till en vinkel utifrån hur strålen ritas ut, vilket leder till att spelaren eventuellt får tänka mer på hur och varför lasern träffar som den gör. Som resultat av användartesterna valde vi även att rita ut hela laserstrålen efter att spelaren träffat en knapp. Detta gjorde det inte enbart tydligare när man faktiskt träffat, utan genom att rita ut hela strålen kan spelaren se laserns väg och därmed få en ökad förståelse för hur strålen faktiskt bryts och reflekteras vid kontakt med de olika objekten.

Valet att införa glasobjekt tillförde en ny speldynamik som vi inte observerat i de tidigare optikspel vi analyserat. Koncept som kanske inte är välkända för gemene man, som exempelvis totalreflektion som sker då ljus inuti glaset träffar ytan med en tillräckligt stor infallsvinkel, kan ge spelaren kunskap och förståelse för en ny typ av fysik samtidigt som det ger upphov till mer utmanande pussel, vilket kan göra gameplay mer intressant.

En förenkling av fysiken gjordes i samband med glasobjekten, då vi valde att enbart låta strålen antingen brytas eller reflekteras i dess yta, istället för att dela upp den i en reflekterad och en bruten stråle med olika intensiteter. Detta gjordes med tanke på gameplay, då det skulle blivit för många strålar att hålla koll på för spelaren om de skulle delats upp, särskilt när endast prickarna ritas ut eftersom det inte skulle gå att se vilka prickar som hör till vilken stråle. Därför anser vi att detta var en rimlig anpassning för att få ett spelbart spel.

Många av användarna som testade detta spel tyckte att det var svårt, speciellt i och med designvalet att inte rita ut hela strålen. En del av dem ansåg att de fick tänka en del på vinklarna för att lösa de flesta pusslarna, medan vissa pussel kunde lösas genom att skjuta på målfå. Huruvida spelet var roligt eller inte fanns det en delad mening kring; några tyckte spelet var roligt, utmanande och underhållande medan det fanns några som tyckte att det var alldeles för "grått och tråkigt". Det senare skulle dock kunna förklaras som en konsekvens av att grafiken inte stått i fokus. Den allmänna åsikten bland speltestarna verkade dock vara att konceptet är intressant, och hade eventuellt kunnat tillföra en ny intressant spelmekanik till ett större spel.

6.1.7 Diskussion av ej implementerade spelprototyper

De idéer vi utvecklade som inte ledde till fullständiga implementationer innefattar fysikområdena kretslära, elektriska nät, elektromagnetism och fourieroptik. De tre sistnämnda sammanfattas här i elektromagnetism, då det innefattar alla dem.

Spelidén för att använda kretslära skulle förmodligen till största del resultera i ett spel för lärande om elektriska kretsar snarare än för underhållning. Underhållningsvärdena som skulle kunna finnas är *Challenge*, då spelaren utmanas i att lösa olika pussel, och *Submission*, ifall pusslen är tillräckligt enkla. Det skulle alltså kunna ge en motivation till att lära sig, snarare än vara ett intressant spel i sig. Möjligheten att använda det för intressant gameplay kan öka om man sätter in det som ett delmoment i ett större spel, då det får bidra till variation av delpussel snarare än vara huvudsyftet med spelet.

Elektromagnetism är ett brett område, och vi har därför flera idéer i underområden. Fourieroptikidén handlar om fälten då det inte finns någon laddning eller ström. Labyrintspelet och Bullethellspelet handlar om hur laddningar påverkas av krafter från fält. Elektrofall handlar om hur oladdade medium ändå kan påverkas av fält, genom polarisering. Eftersom elektromagnetismen är så bred, finns det anledning att tro att vi har missat underområden.

Spelidén om fourieroptik var att fokusera ljus på ett mål, genom att placera ut optiska komponenter. Designrymden begränsas av hur många olika komponenter som kan implementeras. Det finns en risk att spelets pussel bara får enkla lösningar, eftersom optiska komponenter har specialiserade funktioner.

Huvudfokus i labyrintspelet ligger på samspelet mellan spelarens laddning och elektriska fält i omgivningen. Konceptet att ändra på spelarens laddning är ett fenomen som inte förekommer naturligt, utan detta designförslag är ett försök till förbättrad gameplay.

Bullethellspelet kan jämföras med existerande spel. Skillnaden är mekaniken med laddning, attraktion, och repulsion. Målsökande projektiler är en snarlik mekanik som redan existerar, så det skulle inte vara helt nytt.

Elektrofall handlar om att leda fallande vatten till ett mål, genom att placera ut laddningar. Ett problem som tros kan uppstå i nivådesignen är att alla nivåer handlar om vatten som faller ned; det kan inte ske på så många sätt. Designrymden är liten eftersom det finns så få objekt att bygga nivåer av.

6.2 Diskussion av genomförande

Här presenteras en diskussion av vår metod för dokumentationen av tidigare spel och utvecklingen av minispelen, både allmänt och specifikt för de olika implementationerna.

6.2.1 Diskussion av dokumentation och analys

Vår metod för dokumentation och analys av tidigare spel fungerade bra. Att detta påbörjades tidigt i projektet gav oss en bra grund för att själva kunna utforska och implementera olika sorters fysik i våra egna minispel. En fördel som vi hade var även att de flesta medlemmar i gruppen har ett stort intresse för spel, vilket innebar att vi på förhand kände till många olika typer av spel som använder sig av fysik som en central del av gameplay. Detta resulterade i att vi till stor del kunde spara tid på att endast analysera själva fysikanvändningen i spelen snarare än att först försöka hitta dem. Spel som söktes fram var främst spel som använder sig av fysik som inte var vanligt förekommande i de spel som vi tillhandahölls av vår handledare eller kände till från början. Många stora spelbibliotek idag ger möjligheten att söka efter spel utifrån olika kategorier. Då fysik är en vanligt förekommande sådan var det relativt enkelt att hitta spel som använde sig av olika typer av fysik som central del av gameplay. Detta underlättade skapandet av vår kartläggningsmatris.

I de fall där dokumentation om den implementerade fysiken saknades har analysen baserats nästan uteslutande på att vi antingen spelat spelen själva eller tittat på gameplay-videor där andra spelat dem. De spelen vi kunde spela själva var lättare att analysera än de där vi var tvungna att kolla på gameplay-videor, då vi mer fritt kunde testa runt precis det som vi ville utforska på det sätt vi ansåg bäst lämpat för analysen. Hade vi haft en budget för att köpa in de spel vi saknade hade det kunnat ge en ökad kvalitet på analysen för just de spelen. Huruvida den implementerade fysiken var realistisk eller inte avgjordes sedan genom att jämföra om fysiken i spelet stämde överens med vad som förväntades av gruppens fysiker eller om där fanns avvikelser. Då fysik alltid är approximationer valde vi att bedöma fysiken som korrekt om den såg någorlunda korrekt ut. Detta sätt att själva utforska fysikanvändningen i de olika spelen gjorde att det tog lite längre tid än då vi analyserade spel som redan hade den underliggande fysiken dokumenterad, samt att det blev en mer ungefärlig bedömning. Däremot hade det kunnat ta ytterligare längre tid att hitta spel med de olika typerna av fysik om vi hade avgränsat oss till spel med utförlig dokumentation om dess fysik, så vi fokuserade istället på att analysera populära eller viktiga spel snarare än att välja utifrån existerande dokumentation. I de fall där dokumentation saknades hade man även kunnat försöka kontakta spelutvecklaren för mer implementationsdetaljer men i mån av tid valde vi dock istället att uppskatta realismen själva, vilket resulterade i en bredare men grövre analys.

6.2.2 Diskussion av mjukvaruutveckling

Vi anser att vår arbetsmetod och vårt upplägg av projektet fungerade bra. Att vi delade upp oss i smågrupper för att utveckla flera spelidéer parallellt visade sig vara en mycket bra idé som resulterade i många olika undersökta fysikområden med motsvarande implementationer. Detta var möjligt då vi hade flera fysiker i gruppen. Då vi endast var fem personer totalt i projektgruppen kunde vi dock inte skapa mer än två smågrupper i taget då vi inte ville att någon skulle jobba helt själv på en implementation. Hade vi istället haft en sjätte medlem hade vi förmodligen försökt dela upp oss parvis för implementationsdelen för att ytterligare maximera våra resultat, med en fysiker och en programmerare i varje grupp som den ideala uppsättningen.

Valet att använda libGDX som spelmotor anser vi var ett bra val. Inlärningskurvan skulle kunna beskrivas som exponentiell, vilket resulterade i att våra olika minispel gick betydligt fortare att implementera ju närmre projektets slut vi befann oss. Den initiala inlärningsprocessen var även den lagom i mån om tidsåtgång och svårighet. Hade vi istället valt att arbeta med någon annan mer populär spelmotor, så som Unity eller Unreal Engine, hade det tagit ytterligare tid för inläring då ingen i gruppen hade tidigare erfarenhet av att programmera i C# eller C++,

vilket hade kunnat resultera i mindre fokus på våra faktiska syften med projektarbetet till följd av tidsbrist. LibGDX gjorde det även enkelt för oss att använda Box2D i och med dess inbyggda stöd, vilket underlättade mycket vid implementeringen av fysik i de olika spelen.

Att använda Box2D underlättade arbetet på så vis att vi inte själva behövde skriva metoder för att lösa rörelsekvationer och liknande. Det kunde dock innebära problem när saker inte fungerade som tänkt och det var svårt att hitta vad som orsakade problemet när man inte skrivit metoderna själv. Dess dokumentation var även i vissa fall inte så utförlig, så det kunde vara svårt att veta vad olika metoder gör. Oftast gick informationen dock att hitta via andra källor som tutorials och forum.

Användandet av GitHub och feature branch Git workflowet fungerade smidigt och var i princip helt självgående efter uppstartsveckorna. Initialt var det lite tekniska problem för de som inte hade arbetat med Git tidigare, men med hjälp av de mer erfarna i gruppen kunde detta lösas snabbt.

Valet att göra fler implementationer gjordes en bit in på projektets gång, vilket huvudsakligen resulterade i att mer tid lades på de första implementationerna. Hade detta val gjorts tidigare hade vi kunnat gå vidare från de första spelen tidigare, vilket eventuellt hade resulterat i fler undersökta områden och implementationer. Dock gjorde det trots allt höga antalet implementationer, samt att vi valde att undersöka även några ej implementerade spelprototyper, att vi ändå kunde undersöka många olika fysikområden.

I början av projektet arbetade varje person på sina arbetsuppgifter relativt avskilt från alla andra, och man var ofta ovetandes om vad alla andra faktiskt arbetade med för tillfället. Visionen för de olika implementationerna var inte alltid tydligt definierade, vilket i vissa fall resulterade i att folk utvecklade olika delar på samma spel som visade sig vara oförenliga med varandra efter ett tag. När det gäller utvecklingen av ubåtsspelet till exempel, påbörjades implementationen medan vi hade olika idéer om slutresultatet. Detta resulterade i att stora delar av spelet behövde ändras ett par veckor in i processen då vi hade haft ett planeringsmöte. Hade detta möte istället hållits tidigare i processen hade detta kunnat undvikas och tid sparats. Detta tog vi med oss senare i projektet då fler implementationer skulle tas fram, där vi mer utförligt planerade spelutvecklingen i ett tidigare stadie. Ju längre in i projektet vi var desto bättre blev även kommunikationen mellan oss i gruppen. Alla idéer som en enskild person kom på under tiden förmedlades omgående till resterande inblandade personer via vår gruppchatt. Den öppna dialog som detta resulterade i gjorde att vi mer effektivt kunde arbeta på de olika implementationerna då vi hade bra koll på vad som arbetades på och vad som behövde göras.

6.2.3 Diskussion av avgränsningar

Att vi valde att avgränsa oss till att enbart utveckla 2D-spel medförde både för- och nackdelar. Bland fördelarna har vi först och främst att det är mycket lättare att göra, huvudsakligen ur spelutvecklingsperspektivet. Utvecklingen av 3D-spel hade med stor sannolikhet tagit lite längre tid, vilket hade resulterat i färre implementationer. Detta hade i så fall resulterat i färre antal utforskade fysikområden, vilket hade varit kontraproduktivt gentemot huvudsyftet med projektet. Den största nackdelen med denna avgränsning var dock det faktum att vi inte kunde utforska fysikanvändning i 3D-gameplay. Då den riktiga världen är tredimensionell blir även möjligheten att använda sig av realistisk fysik i spel betydligt större i ett 3D-spel. Då 3D-spel är de mest populära spelen i spelindustrin idag hade det även varit mer intressant att utforska speldynamiker för just denna typ av spel, i de fall där det blir avsevärda skillnader mellan 2D- och 3D-spel. Eftersom vi även valde att utforska användningen av olika fysikområden i spel som vi inte tänkte implementera hade vi dock möjligheten att utforska användningen av realistisk fysik i 3D-spel utan att behöva lägga tid på implementation. På så vis kunde vi ändå utforska detta trots vår avgränsning gällande implementering, om än på en mer teoretisk nivå.

6.2.4 Diskussion av utvecklingen av specifika implementationer

De implementerade spelens gameplay kan ha påverkats av metoden. I detta avsnitt diskuteras därför hur val av klasser, metoder, och arbetssätt påverkade resultaten.

Spacetimewar Skapandet av implementationen *Spacetimewar* började med en idé att utforska hur informationsfördröjning kan användas i spel. Detta ledde snabbt till att spelets miljö blev

rymden runt ett svart hål, eftersom vi ville att scenariot skulle vara någorlunda realistiskt. Detta ledde till att första spelidén blev att göra en version av det, enligt oss, fantastiska spelet *Spacewar!*, men med relativistisk fysik tillagd. Det här valet gjorde att implementeringen kom igång snabbt, eftersom vi hade en ungefärlig idé av vad vi skulle försöka uppnå för resultat. Dock kanske det inte var det bästa förslaget vi skulle ha kunnat komma fram till. Den nyimplementerade fysiken hjälpte inte spelet att bli roligare, snarare tvärtom. Kanske hade en annan sorts spel fungerat bättre med allmän relativitetsteori.

Valet av grafik till *Spacetimewar* kan verka något märkligt, då spelet använder en 2D-vy, men egentligen är byggt av 3D-modeller. Att använda 2D-grafik hade antagligen underlättat en del. Till exempel hade något som Box2D då kunnat användas för att exempelvis detektera kollisioner. Eftersom libGDX kan användas för både 2D- och 3D-grafik valdes ändå att försöka göra ett spel i 3D, för att utforska möjligheterna att använda 3D i libGDX längre fram. Vägen till slutresultatet må ha blivit något svårare, men de nyvunna kunskaperna kan anses vara värda den svårare vägen. Slutresultatet självt blev till slut en fungerande demonstration av ett spel med allmän relativitetsteori, vilket var målet med implementationen.

Hydrocurve Fysiken i Hydrocurve var generellt ganska enkel att implementera, med inte så många nödvändiga ekvationer. Vid övergång från vyn från sidan till ovanifrånvyn blev det enklare, då färre krafter behövde tas hänsyn till. Speciellt gravitationen skapade en del buggar när Box2Ds inbyggda funktion för detta skulle användas, då det var svårt att modifiera kropparnas massa på rätt sätt. Valet att gå över till ovanifrånperspektiv är något vi alltså borde gjort tidigare i projektet, då det underlättade mycket samt ansågs ge en bättre spelkänsla då det visade projektilbanorna på ett, enligt oss, bättre sätt. När det gäller projektilbanorna fanns ett initialt svårt moment att få bra balans i krökningen, men med lite tid löstes det utan större problem. Vårt val att endast ha diskreta ursprungs- och krökningvinklar för projektilerna passade spelets stil då det gick snabbt för spelarna att välja projektilbana, men medförde att vi inte kunde utforska krökningarnas påverkan på gameplay i lika stor utsträckning. Att istället göra spelet turn-based hade möjliggjort större frihet i projektilvalet, då det inte längre hade varit begränsat till att gå snabbt.

Tic Tac Ket Valet av libGDX tros inte ha påverkat utvecklingen av Tic Tac Ket. Klassen *Array* användes för att de klassiska tillstånden, men det hade gått lika bra med *ArrayList* från *java.util*. Inga bibliotek andra än de grafiska användes i implementationen.

Tic Tac Ket utvecklades intensivt under en kort tid, vilket gjorde att det inte blev en iterativ utvecklingsprocess. Endast vid ett tillfälle granskade gruppen implementationen och gav kritik som implementerades. Bristen på iterativ återkoppling kan påverka implementeringen negativt. Så tros inte vara fallet med Tic Tac Ket; utvecklingen stoppades eftersom implementationen kändes klar.

Lagom Action För grafiken i Lagom Action användes metoden *ShapeRenderer.line()*. En nackdel med den är att linjer endast kan ritas en pixel breda, vilket gör partikelbanan svår att se mot vissa färger i bakgrunden. Vi insåg senare, efter implementationen var klar, att *ShapeRenderer.rectLine()* hade varit en bättre metod, eftersom den tillåter tjockare linjer.

När betyget räknas ut i Lagom Action används numeriska derivator. Deriveringens skulle kunna göras symboliskt för bättre noggrannhet, men det ger ingen märkbar skillnad med de nuvarande potentialerna. Om potentialer med väldigt höga andraderivator lades in, skulle det kunna bli märkvärt.

A Maze in Light I A Maze in Light underlättade Box2Ds ray casting-funktion väldigt mycket för att hitta laserns kollisionpunkter, vilket tillät oss att simulera laserstrålar på ett realistiskt och effektivt sätt, det vill säga helt utan märkbar fördröjning. De andra sätt vi testade att göra detta på, genom att använda kollisionsmetoden som används för kroppar så som spelarens kollisioner med väggar, var alldeles för långsamma, så ray casting-funktionen var den metod vi hittade som uppfyllde våra krav för laserstrålen. En brist med funktionen var dock att den inte ger ordningen på kollisionerna, så att hitta när strålen lämnar glaset med baklänges ray casting som då alltså skulle vara den sista kollisionen, var inte möjligt. Istället löstes det genom att ha en fixtur på varje sida om glaset, vilket innebar lite brister i noggrannheten då dessa behöver ha en tjocklek trots att

de egentligen bara representerar gränsytan mellan två material. Detta var dock inget som syntes i spel, så det fungerade ändå väl.

Något som var svårt under utvecklingen av spelet var att designa pussel som var av lagom svårighetsgrad, då vi ville att spelaren skulle behöva tänka på den underliggande fysiken och inte bara skjuta på målfå tills man träffar rätt. Användartesterna visade att en del ansåg att man behövde tänka på vinklarna och hur strålen reflekterades, medan andra tyckte att det räckte att skjuta på målfå, vilket påvisar denna svårighet i utvecklingen.

6.3 Validitet och generaliserbarhet

Spelen som analyserats var som nämnt tidigare antingen spel vi tillhandahölls från vår handledare, spel vi sökt fram via nätet alternativt spel vi redan kände till sedan innan. Då spel söktes fram på nätet fick man fram väldigt många olika spel, varav vi valde en delmängd av dessa för vår analys. Även om vi med våra val försökt att hitta olika typer av spel som behandlar fysiken olika, kan vi inte säkerställa hur väl dessa representerar de analyserade spelgenrerna och spelindustrin som helhet.

De olika implementationerna och idéerna demonstrerar exempel på hur de valda fysikaliska fenomenen kan användas i gameplay. Dessa exempel användes för att nå vårt huvudsyfte, vilket var att utforska designmöjligheter kring fysikbaserad gameplay. Huruvida de fysikaliska fenomenen vi infört i våra minispel faktiskt ger intressant gameplay är en fråga som inte enkelt kan besvaras med ett ja eller nej. Då endast småskaliga användartester utförts där användargruppen bestått av oss själva, familj och vänner är det omöjligt att dra en statistiskt signifikant slutsats utifrån dessa. Det finns även en risk att användarnas relation till oss som utvecklade spelen påverkat deras omdöme på så vis att spelen framstår som bättre än de hade ansetts vara av en främling. Med andra ord får man ta våra resultat för vad de är: subjektiva bedömningar.

Implementationerna vi gjort är främst hårdkodade, men undantag finns. I implementationen Spacetime war skrevs klasser för relativistiska fysikaliska kroppar, tensorer, metriker, och mera. Dessa skrevs medvetet generella och återanvändningsbara. En klass för att snabbt och enkelt initialisera alla kameror, renderare och så vidare som behövs för att få ett körbart spel i libGDX skapades även, vilket skulle kunna användas av alla som vill utveckla spel med just libGDX.

6.4 Etiska och samhällliga aspekter

I detta avsnitt presenteras en analys av etiska och samhällliga aspekter av realistisk fysik i videospel, metoden i vårt projekt och utvecklingen av spel.

Den största nyttan med realistisk fysik i videospel är troligtvis att ge spelaren en ökad förståelse och ett ökat intresse för olika fysikaliska fenomen. Det kan alltså bli problematiskt om fysiken i spelen är förenklad, då det kan ge en felaktig bild. Syftet med vårt projekt är att undersöka möjligheterna för realistiska användningar av fysik, så vi arbetar för att minska den problematiken. En viktig del i projektet är också att undersöka till vilken grad spelen kan vara trogna fysikens lagar utan att det påverkar gameplay negativt. Detta är en viktig aspekt, då försämrade spelupplevelse naturligtvis inte är önskvärd. Förhoppningsvis kommer vi att se mer realistiska fysikaliska fenomen i spel framöver, även om det inte blir en direkt följd av det här projektet.

När det gäller metoden i vårt projekt finns det inga moment som vi har bedömt varit direkt problematiska ur en samhälllig eller etisk synvinkel. Där det möjligen skulle kunnat komma in är vid användartesterna av våra minispel, eftersom det är något som rör andra människor. Detta har dock gjorts respektfullt, och då våra minispel inte är särskilt våldsamma eller problematiska på andra sätt, ansåg vi att detta inte innebar något problem. Det hade möjligen varit problematiskt om vi hade låtit barn testa spelen då en del av dem innehåller exempelvis skjutningar, men nivån av fysiken i spelen gör att de riktar sig mot en äldre målgrupp, så det var inte något vi ville göra.

I ett vidare perspektiv kan detta projekt ses som ett försök till förbättring av videospel, något som kan ses som både positivt och negativt. Bra spel kan innebära att fler spelar och blir mer stillasittande, men om förbättringen innebär mer realistisk fysik kan den leda till större fysikintresse och förståelse, vilket vi ser som positivt. Videospelsindustrin är stor med många aktörer som önskar driva utvecklingen framåt, vilket gör att utvecklingen lär gå framåt även om fysiken inte har någon större del i det, så om fysiken får en större del i utvecklingen är det att se som positivt. Detta

kan också ses som ett bra svar på det möjliga motargumentet att det finns viktigare forskning och datorutveckling som kan göras.

Utifrån denna analys drog vi slutsatsen att det inte finns några större etiska eller samhällsliga aspekter att ta i beaktande i projektet, varken i dess problemformulering eller dess genomförande.

6.5 Framtida arbete

En vidareutveckling av vårt projekt skulle vara att utforska fler fysikområden, till exempel genom fler implementationer. Då syftet ligger i utforskningen anser vi att det skulle vara mer värdefullt än att utveckla de spel som vi har gjort vidare. Exempelvis skulle våra ej fullständigt implementerade idéer kunna realiseras i faktiska spel, då det är fysik som sågs saknas men av olika anledningar inte blev de spel som utvecklades i projektet. Detta skulle fördjupa resonemanget kring dem djupare, då en faktisk implementation tydligare visar hur fysiken fungerar i gameplay. Dessutom skulle dokumentationen av tidigare spel och hur fysiken har använts tidigare kunna utvecklas, vilket också skulle kunna ge fler idéer till fler implementationer.

Att vidareutveckla våra implementationer eller inkludera dem i större spel är ändå en möjlighet. De skulle då kunna fylla ett större syfte i att fylla de luckor som hittades vid dokumentationen av tidigare spel, om de skulle vara mer fullständiga spel. En utvärdering kring utvecklingsmöjligheter för de olika implementationerna presenteras därför nedan.

Många av våra implementationer har bristen att de inte är fullständiga spel i form av att de saknar sådant som spelinstruktioner eller en vinstskärm som talar om när spelet är slut och vem som har vunnit i de fall det är ett multiplayer-spel. Spelinstruktioner skulle som ett första steg innefatta vilka kontroller som används i spelen. I Hydrocurve skulle även instruktioner för att förklara de olika projektilbanorna och hur de uppnås vara bra, vilket möjligen också skulle utöka förståelsen för den bakomliggande fysiken. På liknande sätt skulle reflektion i speglar samt Snells lag kunna förklaras i A Maze in Light, speciellt totalreflektion då det ansågs svårt att veta hur det skulle användas enligt användartesterna. På samma sätt skulle en inledande förklaring av fysiken vara givande i både Tic Tac Ket och Spacetimewar. Lagom Action har inte en bakomliggande fysik på samma sätt.

I både Spacetimewar och Hydrocurve kan projektilerna utvecklas. I Spacetimewar skulle detta kunna innebära olika typer av projektiler, tillhörande olika typer av rymdskepp. Fler typer av rymdskepp är också en möjlighet som skulle ge större variation och även kunna ge möjlighet till fler än två spelare. Hydrocurve skulle kunna ge spelaren större frihet att styra över projektilbanorna istället för de fixa alternativ som finns nu.

Generellt kan också grafiken utvecklas, då detta inte har varit i fokus i implementationerna. Mycket utveckling i form av balansering kan också göras, till exempel i styrning och hastigheter i både Spacetimewar och Hydrocurve. För att göra mer fullskaliga spel skulle även fler nivåer eller miljöer kunna läggas till i de flesta implementationerna.

7 Slutsatser

Syftet med detta arbete har varit att utforska designmöjligheter kring fysikbaserad gameplay i videospel. Detta genomfördes genom att först studera användningen av fysik i redan existerande spel, och sedan implementera egna småskaliga spel baserade på fysik som inte har använts mycket.

Arbetet resulterade i en dokumentation av tidigare spel och fem egna implementationer. Implementationer fokuserade på fysik från ett ämne var, och dessa ämnen var: allmän relativitetsteori, strömmingsmekanik, kvantfysik, analytisk mekanik och optik. Utöver implementationerna utvecklades många idéer till spel. De rörde fysikområdena kretslära och elektromagnetism. Eftersom implementationerna och idéerna var baserade på tidigare relativt oanvända fysikområden, gav deras skapande insikter i nya designmöjligheter. Baserat på dessa insikter dras slutsatser om möjligheten att använda följande områden:

Relativitetsteori Relativitetsteori har åtminstone tre intressanta fenomen för speldesign. Det första är kinematiken från speciell relativitet, vilket gör det svårare att uppnå höga hastigheter. Det andra är den begränsade utbredningen av information, också från speciell relativitet. En märkbar konsekvens blir att information alltid är fördröjd. Det tredje fenomenet är massans

krökning av rummet, huvudfenomenet i allmän relativitet. Krökningen av rummet tillåter intressant världsdesign, som inte skulle vara möjlig i euklidisk geometri.

Strömningsmekanik Strömningsmekanik kan användas i spel för att få mer realistiska visualiseringar av aerodynamiska och hydrodynamiska effekter på kroppar. Dock kan det vara svårt att implementera då många approximationer krävs. Den ökade realismen kan även påverka gameplay negativt då spelaren behöver ta hänsyn till fler faktorer vilket kan göra det onödigt svårspelat, men kan också påverka positivt med en ökad känsla av verklighetsanknuten design.

Kvantfysik Kvantfysik kan användas för att göra system mer komplexa. Om systemet är stort kan denna ökning i komplexitet vara för mycket för en dator, men detta kan gömmas undan genom att ha en mekanik som bestraffar spelare då systemet blir för stort. Kvantfysikaliska systems intressanta respons på observationer kan göras till en spelmekanik, främst i genrerna pusselspel eller strategispel, eftersom mekaniken handlar om spelarens intag av information.

Stråloptik Stråloptikens främsta användningsområde i spel är i olika pussel. Ofta handlar det om att reflektera laser med hjälp av speglar, men andra ytor kan införas för att göra det intressantare, så som glasobjekt. Möjligheten finns även att välja att inte rita ut hela laserstrålen, vilket ökar svårighetsgraden samtidigt som det ger ökad realism.

Lagrangeformalismen Lagrangeformalismen är ett sätt att beskriva system i analytisk mekanik, och är därför inte ett område i fysiken på samma sätt som till exempel kvantfysik är. Den har därför inte några fenomen associerade med sig, om man inte räknar minsta verkans princip, något väldigt abstrakt. Denna abstrakthet gör det svårt att säga något allmänt om möjligheten att utnyttja den som spelmekanik.

Kretslära Kretslära anser vi svårt att inkludera i spännande gameplay, då det i många fall troligen skulle resultera i spel som liknar simuleringar av kretsar, men med ett tillfört spelmoment. Detta skulle förmodligen innebära spel som används för att lära sig fysiken snarare än att det är ett roligt spel. En större användningsmöjlighet skulle kunna finnas om det läggs in som delpussel i ett större spel.

Elektromagnetism Elektromagnetismen har flera fenomen som kan bli mekaniker. Laddade partiklar kring en punktladdning eller dipol har krökta banor, vilket kan vara en intressant mekanik om målet är att påverka huruvida en måltavla träffas av partiklarna. Propagationen av elektromagnetiska vågor kan göras till en mekanik i ett pusselspel, vilket kan ge spelaren en intuition för ljusets vågnatur.

Sammanfattningsvis varierar användningsmöjligheten med fysikområdet. Lagrangeformalismen har en väldigt begränsad designrymd och dras ned av att vara abstrakt. I strömningsmekanik, elektromagnetism, och optik finns fenomen som kan bli mekaniker. Dock har dessa redan använts i tidigare spel. Störst potential ser vi i relativitetsteori och kvantfysik; där finns fenomen som inspirerar till spännande nya mekaniker. Slutsatsen är därmed att det finns designmöjligheter i att utnyttja även mindre använda fysikaliska fenomen i videospel.

Referenser

- [1] Wheels.org. The origin of spacewar. Hämtad: 2018-01-25. Tillgänglig: <http://www.wheels.org/spacewar/creative/SpacewarOrigin.html>.
- [2] Nationalencyklopedin. Datorspel. Hämtad: 2018-01-25. Tillgänglig: <https://www.ne.se/uppslagsverk/encyklopedi/l/unhbox\voidb@x\bgroup\let\unhbox\voidb@x\setbox\@tempboxa\hbox{a\global\mathchardef\accent@spacefactor\spacefactor}\accent23a\egroup\spacefactor\accent@spacefactorng/datorspel>.
- [3] Nationalencyklopedin. Pong. Hämtad: 2018-01-25. Tillgänglig: <https://www.ne.se/uppslagsverk/encyklopedi/l/unhbox\voidb@x\bgroup\let\unhbox\voidb@x\setbox\@tempboxa\hbox{a\global\mathchardef\accent@spacefactor\spacefactor}\accent23a\egroup\spacefactor\accent@spacefactorng/pong>.
- [4] Super Data Research. SuperData Digital Games and Interactive Media Year in Review—2017. Hämtad 2018-05-11. Tillgänglig: <https://superdata-research.myshopify.com/products/year-in-review>.
- [5] Wikipedia. Pelé's Soccer, 2018. Hämtad 2018-04-01. Tillgänglig: https://en.wikipedia.org/w/index.php?title=Pe%C3%A9%27s_Soccer&oldid=827504101.
- [6] Wikipedia. Fifa 14, 2017. Hämtad 2018-04-09. Tillgänglig: https://sv.wikipedia.org/w/index.php?title=FIFA_14&oldid=40544398.
- [7] Scientific American. Getting on the ball: How the fifa 14 soccer video game finally got its physics right, 2015. Hämtad 2018-04-05. Tillgänglig: <https://www.scientificamerican.com/article/getting-on-the-ball-how-soccer-video-game-got-physics-right/>.
- [8] Wikipedia. Aerodynamik, 2015. Hämtad 2018-04-05. Tillgänglig: <https://sv.wikipedia.org/wiki/Aerodynamik>.
- [9] Wikipedia. Magnuseffekten, 2017. Hämtad 2018-04-08. Tillgänglig: <https://sv.wikipedia.org/w/index.php?title=Magnuseffekten&oldid=38822023>.
- [10] Wikipedia. Super Mario Strikers, 2018. Hämtad 2018-04-01. Tillgänglig: https://en.wikipedia.org/w/index.php?title=Super_Mario_Strikers&oldid=829560119.
- [11] Wikipedia. List of association football video games, 2018. Hämtad 2018-04-04. Tillgänglig: https://en.wikipedia.org/w/index.php?title=List_of_association_football_video_games&oldid=820019044.
- [12] Wikipedia. Half-life (video game), 2015. Hämtad 2018-04-05. Tillgänglig: [https://en.wikipedia.org/wiki/Half-Life_\(video_game\)](https://en.wikipedia.org/wiki/Half-Life_(video_game)).
- [13] A. Austin. Hämtad 2018-04-07. Tillgänglig: <https://www.bridgebuilder-game.com/bbg-info.php>.
- [14] Wikipedia. Bridge Builder, 2018. Hämtad 2018-04-05. Tillgänglig: https://en.wikipedia.org/w/index.php?title=Bridge_Builder&oldid=826634393.
- [15] Wikipedia. Crayon Physics Deluxe, 2016. Hämtad 2018-04-06. Tillgänglig: https://sv.wikipedia.org/w/index.php?title=Crayon_Physics_Deluxe&oldid=38357844.
- [16] Wikipedia. Centrifugalkraft, 2015. Hämtad 2018-04-05. Tillgänglig: <https://sv.wikipedia.org/wiki/Centrifugalkraft>.
- [17] Wikipedia. Portal, 2017. Hämtad 2018-03-28. Tillgänglig: [https://sv.wikipedia.org/w/index.php?title=Portal_\(spel\)&oldid=40761316](https://sv.wikipedia.org/w/index.php?title=Portal_(spel)&oldid=40761316).

- [18] Wikipedia. The Talos Principle, 2018. Hämtad: 2018-04-04. Tillgänglig: https://en.wikipedia.org/w/index.php?title=The_Talos_Principle&oldid=835365837.
- [19] Wikipedia. Monster strike, 2018. Hämtad: 2018-04-07. Tillgänglig: https://en.wikipedia.org/w/index.php?title=Monster_Strike&oldid=831360172.
- [20] Wikipedia. Peggle, 2018. Hämtad: 2018-04-02. Tillgänglig: <https://en.wikipedia.org/w/index.php?title=Peggle&oldid=830876785>.
- [21] Wikipedia. Wolfenstein 3d, 2018. Hämtad 2018-04-12. Tillgänglig: https://sv.wikipedia.org/wiki/Wolfenstein_3D.
- [22] Wikipedia. Newton's laws of motion, 2018. Hämtad 2018-04-12. Tillgänglig: https://en.wikipedia.org/w/index.php?title=Newton%27s_laws_of_motion&oldid=824447998.
- [23] Wikipedia. Halo: Combat evolved, 2018. Hämtad 2018-04-12. Tillgänglig: https://sv.wikipedia.org/wiki/Halo:_Combat_Evolved.
- [24] Wikipedia. Battlefield 4, 2018. Hämtad 2018-04-12. Tillgänglig: https://en.wikipedia.org/wiki/Battlefield_4.
- [25] Dice. Battlefield 4, 2015. Hämtad 2018-04-05. Tillgänglig: <http://www.dice.se/games/battlefield-4/>.
- [26] Former Army Armor Officer A. Acosta. Why do militaries issue fully automatic weapons when they almost always use them in a semi-automatic mode or fire in bursts?, 2018. Hämtad 2018-04-12. Tillgänglig: <https://www.quora.com/Why-do-militaries-issue-fully-automatic-weapons-when-they-almost-always-use-them-in-a-semi-automatic-mode-or-fire-in-bursts>.
- [27] Wikipedia. Counter-strike: Global offensive, 2018. Hämtad 2018-04-12. Tillgänglig: https://sv.wikipedia.org/w/index.php?title=Counter-Strike:_Global_Offensive&oldid=42609718.
- [28] Wikipedia. Call of duty 4: Modern warfare, 2018. Hämtad 2018-04-12. Tillgänglig: https://sv.wikipedia.org/w/index.php?title=Call_of_Duty_4:_Modern_Warfare&oldid=42693827.
- [29] Steam User: BenVin. Spray patterns and recoil compensation (side-by-side) for all cs:go weapons, 2018. Hämtad 2018-04-12. Tillgänglig: <https://steamcommunity.com/sharedfiles/filedetails/?l=swedish&id=378325692>.
- [30] Softpedia News. Cs:go gun spraying accuracy changes reverted by valve, with apologies, 2018. Hämtad 2018-04-12. Tillgänglig: <https://news.softpedia.com/news/cs-go-gun-spraying-accuracy-changes-reverted-by-valve-497840.shtml>.
- [31] Wikipedia. Space race (video game), 2018. Hämtad 2018-04-12. Tillgänglig: [https://en.wikipedia.org/wiki/Space_Race_\(video_game\)](https://en.wikipedia.org/wiki/Space_Race_(video_game)).
- [32] Wikipedia. Grand Theft Auto, 2018. Hämtad 2018-02-08. Tillgänglig: https://en.wikipedia.org/wiki/Grand_Theft_Auto.
- [33] Wikipedia. Need for speed, 2018. Hämtad 2018-04-12. Tillgänglig: https://sv.wikipedia.org/wiki/Need_for_Speed.
- [34] Wikipedia. The need for speed, 2018. Hämtad 2018-04-12. Tillgänglig: https://en.wikipedia.org/wiki/The_Need_for_Speed.
- [35] Wikipedia. Need for speed payback, 2018. Hämtad 2018-04-12. Tillgänglig: https://en.wikipedia.org/wiki/Need_for_Speed_Payback.
- [36] Wikipedia. Grand theft auto v, 2018. Hämtad 2018-04-12. Tillgänglig: https://en.wikipedia.org/wiki/Grand_Theft_Auto_V.

- [37] Wikipedia. Super mario kart, 2018. Hämtad 2018-04-12. Tillgänglig: https://sv.wikipedia.org/wiki/Super_Mario_Kart.
- [38] Wikipedia. Platform game, 2018. Hämtad 2018-02-08. Tillgänglig: https://en.wikipedia.org/wiki/Platform_game.
- [39] Wikipedia. Mekanik, 2015. Hämtad 2018-04-05. Tillgänglig: <https://sv.wikipedia.org/wiki/Mekanik>.
- [40] Wikipedia. Super mario bros., 2018. Hämtad 2018-04-12. Tillgänglig: https://sv.wikipedia.org/wiki/Super_Mario_Bros.
- [41] Gamecubicle. mario sales data, 2018. Hämtad 2018-02-08. Tillgänglig: http://www.gamecubicle.com/features-mario-units_sold_sales.htm.
- [42] IGN. IGN top 100, 2018. Hämtad 2018-02-08. Tillgänglig: <http://top100.ign.com/2005/001-010.html>.
- [43] Wikipedia. Video game crash of 1983, 2018. Hämtad 2018-02-08. Tillgänglig: https://en.wikipedia.org/wiki/Video_game_crash_of_1983.
- [44] Wikipedia. Super mario 64, 2018. Hämtad 2018-04-12. Tillgänglig: https://sv.wikipedia.org/wiki/Super_Mario_64.
- [45] Highsnobiety. Mario bros., 2018. Hämtad 2018-04-12. Tillgänglig: <https://www.highsnobiety.com/2016/06/23/super-mario-64-best-game/>.
- [46] Wikipedia. Temple run, 2018. Hämtad 2018-04-12. Tillgänglig: https://en.wikipedia.org/wiki/Temple_Run.
- [47] Wikipedia. Briad (video game), 2018. Hämtad 2018-04-12. Tillgänglig: [https://en.wikipedia.org/wiki/Braid_\(video_game\)](https://en.wikipedia.org/wiki/Braid_(video_game)).
- [48] Wikipedia. Adventure (atari 2600), 2018. Hämtad 2018-05-10. Tillgänglig: [https://en.wikipedia.org/wiki/Adventure_\(Atari_2600\)](https://en.wikipedia.org/wiki/Adventure_(Atari_2600)).
- [49] Wikipedia. Castlevania: Harmony of despair, 2018. Hämtad 2018-05-10. Tillgänglig: https://en.wikipedia.org/wiki/Castlevania:_Harmony_of_Despair.
- [50] Wikipedia. Mirror's edge, 2018. Hämtad 2018-05-10. Tillgänglig: https://en.wikipedia.org/wiki/Mirror%27s_Edge.
- [51] Wikipedia. The legend of zelda: Breath of the wild, 2018. Hämtad 2018-05-10. Tillgänglig: https://en.wikipedia.org/wiki/The_Legend_of_Zelda:_Breath_of_the_Wild.
- [52] Wikipedia. Dice 2018: Zelda: Breath of the wild wins game of the year at dice awards, 2018. Hämtad 2018-05-10. Tillgänglig: https://en.wikipedia.org/wiki/D.I.C.E._Awards.
- [53] A. Osborn. Dice 2018: Zelda: Breath of the wild wins game of the year at dice awards, 2018. Hämtad 2018-05-10. Tillgänglig: <http://nordic.ign.com/switch/10814/news/dice-2018-zelda-breath-of-the-wild-wins-game-of-the-year-at>.
- [54] Fireboy and Watergirl. Hämtad: 2018-01-25. Tillgänglig: <http://www.fireboynwatergirl.com/?p=game&id=2>.
- [55] Wikipedia. Lego Pirates of the Caribbean: The Video Game, 2018. Hämtad: 2018-01-25. Tillgänglig: https://en.wikipedia.org/wiki/Lego_Pirates_of_the_Caribbean:_The_Video_Game.
- [56] Game Guides. Smuggler's Den - walkthrough — The Curse of the Black Pearl. Hämtad: 2018-01-25. Tillgänglig: <https://guides.gamepressure.com/legopiratesofthecaribbeanthevideogame/guide.asp?ID=11460>.

- [57] Free Online PC Games. Prisme.
Hämtad: 2018-01-25. Tillgänglig: <http://www.freeonlinepcgames.net/play/prism-puzzle-games-involving-laser-beams-and-mirr/flash-game/>.
- [58] Steam. Optika. Hämtad: 2018-01-25. Tillgänglig:
<http://store.steampowered.com/app/504400/Optika/>.
- [59] Financial Post. Fifa 14 is more challenging, more realistic, less fun, 2015.
Hämtad 2018-04-05. Tillgänglig: <http://business.financialpost.com/technology/gaming/fifa-14-is-more-challenging-more-realistic-less-fun>.
- [60] Science At Home: Quantum moves. Hämtad: 2018-05-10. Tillgänglig:
<https://www.scienceathome.org/games/quantum-moves/>.
- [61] Quantum Game. Hämtad: 2018-05-10. Tillgänglig:
<http://play.quantumgame.io/>.
- [62] MIT Game Lab: A Slower Speed Of Light. Hämtad 2018-05-11. Tillgänglig:
<http://gamelab.mit.edu/games/a-slower-speed-of-light/>.
- [63] TestTubeGames: Velocity Raptor. Hämtad: 2018-05-10. Tillgänglig:
<http://www.testtubegames.com/games.html>.
- [64] TestTubeGames. Relativity 101. Hämtad 2018-05-11. Tillgänglig:
<http://www.testtubegames.com/sre1101.html>.
- [65] Wikipedia. Elektromagnetism, 2018. Hämtad 2018-02-08. Tillgänglig:
<https://sv.wikipedia.org/wiki/Elektromagnetism>.
- [66] TestTubeGames: The Electric Shocktopus. Hämtad: 2018-05-10. Tillgänglig:
<http://testtubegames.com/shocktopus.html>.
- [67] TestTubeGames. About, 2018. Hämtad 2018-02-08. Tillgänglig:
<http://www.testtubegames.com/about.html>.
- [68] TestTubeGames. Shockopus FAQ, 2018. Hämtad 2018-02-08. Tillgänglig:
http://www.testtubegames.com/shocktopus_faq.html#q1a.
- [69] Högerhandsregeln, Wikipedia. Hämtade 2018-04-12. Tillgänglig:
<https://sv.wikipedia.org/wiki/H%C3%B6gerhandsregeln>.
- [70] R. Hunicke, M. Leblanc och R. Zubek. MDA: A formal approach to game design and game research. 2004. Tillgänglig:
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.79.4561>.
- [71] M. LeBlanc. The collected game design rants of Marc "MAHK" LeBlanc. Hämtad: 2018-02-04. Tillgänglig:
<http://algorithmancy.8kindsoffun.com/>.
- [72] W. Rindler. *Relativity: Special, General, and Cosmological*. Oxford University Press, 2006.
- [73] NASA. N. Hall. The drag equation, 2015. Hämtad 2018-04-07. Tillgänglig:
<https://www.grc.nasa.gov/www/k-12/airplane/drageq.html>.
- [74] Engineering ToolBox. Reynolds Number, 2003. Hämtad: 2018-04-07. Tillgänglig:
https://www.engineeringtoolbox.com/reynolds-number-d_237.html.
- [75] University of British Columbia. William Chu. The Law of Reflection, 2001. Hämtad: 2018-04-10. Tillgänglig:
http://www.physicsinsights.org/dipole_field_1.html.
- [76] University of British Columbia. William Chu. Snell's Law - The Law of Refraction, 2001.
Hämtad: 2018-04-15. Tillgänglig:
http://www.physicsinsights.org/dipole_field_1.html.

- [77] J. D. Jackson. *Classical Electrodynamics*. 3 edition, 1999.
- [78] L. Böiers A. Persson. *Analys i flera variabler*. 2005.
- [79] C. Davidson och P. Hofvenschiöld. *Elektriska nät*. Studentlitteratur, 2 edition, 2001.
- [80] Hyperphysics. P. Nave. Magnetic forces. Hämtad: 2018-05-12. Tillgänglig: <http://hyperphysics.phy-astr.gsu.edu/hbase/magnetic/magfor.html>.
- [81] Hyperphysics. P. Nave. Electric field. Hämtad: 2018-05-12. Tillgänglig: <http://hyperphysics.phy-astr.gsu.edu/hbase/electric/elefie.html>.
- [82] Physics Insights. The Field of a Small Magnetic Dipole, 2007. Hämtad: 2018-05-12. Tillgänglig: http://www.physicsinsights.org/dipole_field_1.html.
- [83] Mathworks. ode45, 2015. Hämtad 2018-04-05. Tillgänglig: <https://www.mathworks.com/help/matlab/ref/ode45.html>.
- [84] Wikipedia. libgdx, 2018. Hämtad 2018-05-12. Tillgänglig: <https://en.wikipedia.org/wiki/LibGDX>.
- [85] Wikipedia. Unity (game engine), 2018. Hämtad 2018-05-12. Tillgänglig: [https://en.wikipedia.org/wiki/Unity_\(game_engine\)](https://en.wikipedia.org/wiki/Unity_(game_engine)).
- [86] Wikipedia. Unreal engine, 2018. Hämtad 2018-05-12. Tillgänglig: https://en.wikipedia.org/wiki/Unreal_Engine.
- [87] The Grey Studios. Why we switched to libgdx, 2011. Hämtad 2018-02-08. Tillgänglig: <http://thegreystudios.com/blog/?p=30>.
- [88] Wikipedia. Lightweight java game library, 2018. Hämtad 2018-05-12. Tillgänglig: https://en.wikipedia.org/wiki/Lightweight_Java_Game_Library.
- [89] Ninja Cave. Main page, 2018. Hämtad 2018-05-12. Tillgänglig: http://slick.ninjacave.com/wiki/index.php?title=Main_Page.
- [90] Wikipedia. jmonkeyengine, 2018. Hämtad 2018-05-12. Tillgänglig: <https://en.wikipedia.org/wiki/JMonkeyEngine>.
- [91] J. Holinaty. Box2d, 2018. Hämtad 2018-02-08. Tillgänglig: <https://github.com/libgdx/libgdx/wiki/Box2d>.
- [92] E. Catto. Faq, 2018. <https://github.com/erincatto/Box2D/wiki/FAQ>.
- [93] box2d-editor, 2011. Hämtad: 2018-04-02. Tillgänglig: <https://code.google.com/archive/p/box2d-editor/>.
- [94] Wikipedia. Paint tool sai, 2018. Hämtad 2018-05-12. Tillgänglig: https://en.wikipedia.org/wiki/Paint_Tool_SAI.
- [95] Tiled map editor, 2018. Hämtad 2018-05-12. Tillgänglig: <https://www.mapeditor.org/>.
- [96] E. W. Weisstein. Cubic spline. Hämtad 2018-02-08. Tillgänglig: <http://mathworld.wolfram.com/CubicSpline.html>.
- [97] Wikipedia. Shoot'em up, 2018. Hämtad 2018-05-12. Tillgänglig: https://sv.wikipedia.org/wiki/Shoot_%27em_up.
- [98] Wikipedia. Hållfasthetslära, 2015. Hämtad 2018-04-05. Tillgänglig: [//sv.wikipedia.org/w/index.php?title=H%C3%A5llfasthetsl%C3%A4ra&oldid=30720825](https://sv.wikipedia.org/w/index.php?title=H%C3%A5llfasthetsl%C3%A4ra&oldid=30720825).

- [99] Wikipedia. Anime, 2018. Hämtad 2018-04-07. Tillgänglig:
<https://sv.wikipedia.org/w/index.php?title=Anime&oldid=42435364>.
- [100] Wikipedia. The Incredible Machine (series), 2018. Hämtad: 2018-04-09. Tillgänglig:
[https://en.wikipedia.org/w/index.php?title=The_Incredible_Machine_\(series\)&oldid=830818116](https://en.wikipedia.org/w/index.php?title=The_Incredible_Machine_(series)&oldid=830818116).
- [101] Halo Nation (wiki). Covenant empire, 2018. Hämtad 2018-04-12. Tillgänglig:
http://halo.wikia.com/wiki/Covenant_Empire.
- [102] Halo Nation (wiki). John-117, 2018. Hämtad 2018-04-12. Tillgänglig:
<http://halo.wikia.com/wiki/John-117>.
- [103] Halo Nation (wiki). Spartan, 2018. Hämtad 2018-04-12. Tillgänglig:
<http://halo.wikia.com/wiki/Spartan>.
- [104] Halo Nation (wiki). Flood, 2018. Hämtad 2018-04-12. Tillgänglig:
<http://halo.wikia.com/wiki/Flood>.
- [105] Wikipedia. Call of duty (datorspelserie), 2018. Hämtad 2018-04-12. Tillgänglig:
[https://sv.wikipedia.org/wiki/Call_of_Duty_\(datorspelsserie\)](https://sv.wikipedia.org/wiki/Call_of_Duty_(datorspelsserie)).
- [106] Call Of Duty Wiki. John "soap" mactavish, 2018. Hämtad 2018-04-12. Tillgänglig:
http://callofduty.wikia.com/wiki/John_%22Soap%22_MacTavish.
- [107] Call Of Duty Wiki. Imran zakhaev, 2018. Hämtad 2018-04-12. Tillgänglig:
http://callofduty.wikia.com/wiki/Imran_Zakhaev.
- [108] Battlefield wiki. Daniel recker, 2018. Hämtad 2018-04-12. Tillgänglig:
http://battlefield.wikia.com/wiki/Daniel_Recker.
- [109] Battlefield wiki. Chang wei, 2018. Hämtad 2018-04-12. Tillgänglig:
http://battlefield.wikia.com/wiki/Chang_Wei.
- [110] Wikipedia. Frostbite (game engine), 2018. Hämtad 2018-04-12. Tillgänglig:
[https://en.wikipedia.org/wiki/Frostbite_\(game_engine\)](https://en.wikipedia.org/wiki/Frostbite_(game_engine)).
- [111] CS:GO wiki, Bomb Defusal Mode. Hämtad: 2018-04-12. Tillgänglig:
http://counterstrike.wikia.com/wiki/Bomb_Defusal.
- [112] CS:GO wiki, Deathmatch. Hämtad: 2018-04-12. Tillgänglig:
<http://counterstrike.wikia.com/wiki/Deathmatch>.
- [113] Wikipedia. Mario, 2018. Hämtad 2018-04-12. Tillgänglig:
<https://sv.wikipedia.org/wiki/Mario>.
- [114] Wikipedia. Luigi, 2018. Hämtad 2018-04-12. Tillgänglig:
<https://sv.wikipedia.org/wiki/Luigi>.
- [115] Wikipedia. Bowser, 2018. Hämtad 2018-04-12. Tillgänglig:
<https://sv.wikipedia.org/wiki/Bowser>.
- [116] GTA Wiki. Michael de santa, 2018. Hämtad 2018-04-12. Tillgänglig:
http://gta.wikia.com/wiki/Michael_De_Santa.
- [117] GTA Wiki. Trevor philips, 2018. Hämtad 2018-04-12. Tillgänglig:
http://gta.wikia.com/wiki/Trevor_Philips.
- [118] GTA Wiki. Franklin clinton, 2018. Hämtad 2018-04-12. Tillgänglig:
http://gta.wikia.com/wiki/Franklin_Clinton.
- [119] Wikipedia. Mario bros., 2018. Hämtad 2018-04-12. Tillgänglig:
https://en.wikipedia.org/wiki/Mario_Bros.

- [120] Wikipedia. Broforce, 2018. Hämtad 2018-04-12. Tillgänglig:
<https://en.wikipedia.org/wiki/Broforce>.
- [121] Broforce Wiki. Bros, 2018. Hämtad 2018-04-12. Tillgänglig:
<http://broforce.wikia.com/wiki/Bros>.
- [122] Wikipedia. Kastparabel, 2018. Hämtad 2018-04-12. Tillgänglig:
<https://sv.wikipedia.org/wiki/Kastparabel>.
- [123] Plasma Pong. Hämtad 2018-05-12. Tillgänglig:
https://en.wikipedia.org/wiki/Plasma_Pong.
- [124] Fluid Table-Tennis. Hämtad 2018-05-12. Tillgänglig:
http://anirudhjoshi.github.io/fluid_table_tennis/.
- [125] M. Francis. “Pong” gets a physics boost, by way of HTML5. Hämtad 2018-05-12. Tillgänglig: <https://arstechnica.com/science/2012/09/pong-gets-a-physics-boost-by-way-of-html5/>.

A Dokumentation över tidigare spel

A.1 Pusselspel

I följande avsnitt presenteras dokumentationen av spelgenren pusselspel.

A.1.1 Portal

Portal är ett fysikbaserat pusselspel lanserat av *Valve Corporation* år 2007 [17]. Det är ett spin-off-spel till *Half Life* och är byggt i spelmotorn *Source*. *Portal* är ett 3D-pusselspel vars utmaningar är baserade på fysikens lagar. Spelaren har möjlighet att skapa portaler med hjälp av en portalpistol. Spelaren kan som maximalt öppna två portaler som sammanlänkas. Om spelaren går in i en av portalerna teleporteras spelaren och kommer ut ur nästa portal. Detta verktyg används flitigt under spelets gång för att övervinna spelets alla utmaningar. Spelaren kan till exempel välja att placera en portal nere i en fallgrop och den andra mot en vägg för att sedan hoppa ned i portalen i gropen. Efter att spelaren hoppat ned i portalen kommer den hoppa ut ur den andra portalen med en ökad hastighet. Portalpistolen kan även transportera tunga objekt åt spelaren eller vidröra olika objekt som kan anses vara farliga att vidröra med bara händer. För att få poäng i spelet måste spelaren klara av svåra utmaningar och ta sig över hinder med hjälp av sin portalpistol. Speldesignen är fokuserad på fantasy, realism och utmaning.

A.1.2 Bridge Builder

Bridge Builder lanserades år 2000 [13] och tillhör en familj av videospel som går ut på att bygga broar [14]. Spelaren får som utmaning att bygga en bro som sedan ska tillåta en viss mängd fordon att passera. Problematiken grundas i teorin om hållfasthetslära [98] som talar om hur krafter påverkar kroppar. Om spelare väljer en allt för trivial lösning kan spelaren bli bestraffad då bron inte uppnår hållfasthetskraven och därmed kollapsar bron när fordonen försöker ta sig över. Speldesignen utgörs mer av ingenjörsprobatiken istället för arkitekturens estetik, då spelarens designval har direkta konsekvenser för de fordon som vill passera bron. Spelaren måste därmed applicera tekniker som stålbalkar och spänningslinor. Spelmekaniken i detta spel bygger alltså på realism och utmaning.

A.1.3 Crayon Physics Deluxe

Crayon Physics Deluxe är ett pusselspel skapat av *Petri Purho* som lanserades år 2009 för plattformarna Windows, IOS, Mac OS X, Linux och Android [15]. Spelet har en enkel men charmig 2D-grafik som efterliknar ett skissblock. Spelaren har möjlighet att skapa bland annat broar med hjälp av sin krita som spelaren kan rita med på ett anteckningsblock. Problematiken i spelet uppkommer då spelaren tvingas föra ett cirkulärt föremål över ett visst hinder. I slutet av hindret existerar en stjärna och om det cirkulära objektet kolliderar med stjärnan vinner spelaren och avancerar till nästa nivå. Ett exempel på ett hinder som uppkommer tidigt i spelet är då spelarens objekt befinner sig på en plattform medan stjärnan befinner sig på en annan plattform som är separerad med en fallgrop. Spelaren har möjlighet att lösa problemet genom att skapa en bro mellan plattformarna och på så sätt förflytta sig till stjärnan. Spelets problematik kan jämföras med vanlig ingenjörskonst och spelaren behöver ta hänsyn till gravitationens inverkan på sitt cirkulära föremål. Om spelaren inte tar hänsyn till krafternas inverkan kan till exempel bron kollapsa. Ett ytterligare exempel på ingenjörskonst är då spelaren tvingas att gunga föremålet på en balk för att på sådant sätt skjuta iväg föremålet och träffa stjärnan som ligger ovanför en kulle. Spelaren kan göra detta genom att placera en tung massa på andra sidan av balken och konsekvensen blir att balken faller över och det cirkulära föremålet åker iväg med en viss utfallshastighet samt vinkel.

Crayon Physics Deluxe är skapat i spelmotorn Box2D och spelet strävar efter realism och korrekthet. Fysiken i spelet är centrerad kring mekanik och har en väsentlig roll i spelmekaniken. Spelet kräver korrekt och realistisk fysik för att spelets gameplay ska fungera och för att spelaren ska ha möjlighet att lösa spelets problem.

A.1.4 Peggle

Peggle är ett pusselspel som lanserades 2007 för plattformarna Microsoft Windows, Mac OS X, Windows Mobile, Java ME, BREW, Nintendo DS, Xbox 360, PlayStation 3, iOS, Zeebo och Android [20]. Spelet centreras kring ett minfält fullt av minor av olika färger. Målet är att träffa samtliga minor av en specifik färg med hjälp av en kanon placerad ovanför minfältet. Spelet har en färgglad 2D-grafik där spelaren har en sidovy. Spelaren har möjlighet att betrakta hela minfältet men har en begränsning på ammunition till sin kanon. Spelaren behöver därmed på ett sparsamt sätt försöka träffa så många minor som möjligt. Om kulan från kanonen kolliderar med en mina av önskad färg lyses minan tillfälligt upp och försvinner efter det att kulan åkt utanför minfältet och rundan är över. Då kulan kolliderar med en godtycklig mina studsar kulan vidare medan minan kvarstår i sin nuvarande position och förblir statisk.

Peggle har en spelmekanik som bygger på klassisk mekanik. Då spelaren tillför en impuls på kulan via kanonen förändras kulans rörelsemängd och åker därmed in i minfältet. Vid kollision med minor bevaras rörelsemängden hos kulan då spelet hanterar dessa kollisioner som elastiska. Tillsammans med det faktum att minorerna vid kollision inte förflyttas leder detta till att kulan studsar med bevarad hastighet. Ett gravitationsfält är närvarande i spelet riktat nedåt, och kulan accelererar därmed konstant nedåt. *Peggle* innehåller alltså delmoment som illustrerar realistisk fysik.

A.1.5 Monster Strike

Monster Strike lanserades i Japan 2013 för iOS och Android och är ett spel anpassat för mobilanvändare[19]. Spelet presenterar ett gäng karaktärer som strävar efter att bekämpa en ondskefull makt. Spelet bygger på 2D-grafik och speldesignen är inspirerad av den japanska animeringsstilen *anime*[99]. Spelaren behöver ta sig genom ett flertal olika nivåer i olika miljöer där den placeras på ett fält tillsammans med motståndaren. Alla karaktärer i spelet representeras av cirkulära objekt. För att skada motståndaren behöver spelare skjuta iväg sina karaktärer och kollidera med motståndaren. Spelarens karaktärer skjuts iväg med en kraftig utfallshastighet och spelaren har möjlighet att välja utfallsvinkel. Vid kollision studsar spelarens karaktär vidare medan motståndaren behåller sin statiska position med en marginell förskjutning. För att spelaren ska göra maximal skada behöver den studsa sin karaktär fram och tillbaka mellan olika motståndare som för varje kollision tar en viss mängd skada. Spelaren får endast skjuta iväg sina karaktärer ett begränsat antal gånger och om spelaren inte kan besegra samtliga motståndare på fältet förlorar spelaren och behöver börja om igen på första nivån.

Monster Strike använder realistisk fysik vid kollisioner mellan olika karaktärer. Spelmekniken är baserad på klassisk mekanik. Spelaren har möjlighet att välja utgångshastighet och utfallsvinkel. Spelet hanterar kollisioner som elastiska. Då motståndaren inte förflyttas mer än marginellt efter en kollision betyder det att spelaren behåller största delen av sin hastighet efter kollisionen. Dock tillkommer friktion mellan markytan och spelarens karaktär vilket betyder att spelaren kommer sakta in och till slut tappa all sin hastighet.

A.1.6 The Talos Principle

The Talos Principle är ett fysikbaserat pusselspel som lanserades år 2014 av *Croteam*[18]. Grafiken är skapad i 3D-motorn *Serious Engine 4*. Speldesignen är en blandning av ultramodern teknologi samt en spelmiljö som efterliknar medeltiden med medeltida tempel. Problemet i spelmekaniken bygger på att klara av fysikbaserade pussel. Främst måste spelaren reflektera laserstrålar med hjälp av en kristallstav. Spelaren behöver även hantera mekaniska problem som kollision mellan klot för att ta sig vidare i spelet. Spelmekniken är baserad på fysikområdena optik och mekanik, men spelet involverar även pussel som inte är baserade på fysik som till exempel minipussel för att låsa upp en dörr. Dessutom är spelmekaniken delvis förenklad. Pussel som omfattar optik som reflektioner av laserstrålar löses genom att spelaren först placerar kristallstaven framför en laser. Spelaren väljer sedan en anknytningspunkt där den önskar att strålen skall åka mot och kopplar den till kristallstaven. Reflektionsvinkeln väljs sedan automatiskt av spelet så länge det existerar en möjlighet till reflektion vid anknytningspunkten. På så sätt behöver spelaren endast arbeta yttligt med de optikbaserade utmaningarna. Spelet strävar efter en realistisk fysik tillsammans med en ultramodern speldesign för att förmedla sin framtidsbaserade berättelse.

A.1.7 The Incredible Machine

The Incredible Machine är ett spel som lanserades första gången 1993 av *Dynamix*[100], som går ut på att spelaren ska bygga så kallade *Rube Goldberg-maskiner* för att klara olika pusselliknande nivåer. Detta innebär att olika saker placeras ut för att i en serie aktivera varandra och slutligen nå ett mål, exempelvis att ringa på en klocka. De olika sakerna kan vara allt från bollar som rullar på nästa sak till en lampa som tänds för att aktivera nästa steg. Varje delmoment i maskinen kan ses som ganska onödig, men för att klara spelnivån kan ändå inget steg hoppas över, även om vägen för att exempelvis ringa på klockan är klart onödigt komplex. De vanligaste momenten i dessa maskiner är baserade på mekanik, och innebär ofta att en sak flyttas, ofta med gravitationens hjälp i första hand, för att träffa nästa sak och på så sätt få den i rörelse, men även andra fenomen kan utnyttjas. I *The Incredible Machine* används till exempel möss som naturligtvis går dit ost finns, och på så vis kan aktivera nästa steg. Det är väl inget så fysikaliskt, men exempelvis vatten som rinner, elektriska komponenter och ljus kan också användas. För att lösa dessa nivåer krävs en viss förståelse för den relevanta fysiken, även om det ofta går att prova sig fram. Förmodligen ger det i så fall åtminstone en förståelse för fysiken.

A.1.8 The Quantum Game

The Quantum Game är ett pusselspel baserat på kvantfysik från Juni år 2016[61]. Spelaren ska placera ut olika objekt för att leda en foton från en foton-källa till ett varierande antal ljusdetektorer, men samtidigt se till att det inte finns någon risk för fotonen att träffa farliga minor. Objekten som placeras ut är bland annat speglar som helt enkelt reflekterar fotonen åt någon riktning och små glasskivor som har en lika stor chans att släppa igenom fotonen som att reflektera den. För att klara en bana behöver fotonen inte alltid träffa en detektor, utan det räcker på vissa banor att det finns en chans att den träffar en av de många detektorerna, så länge det inte finns någon risk att en mina träffas. Hur stor chansen att fotonen träffar en detektor ska vara varierar från bana till bana.

Kvantfysiken i *The Quantum Game* är just att man aldrig kan vara säker på hur de olika objekten faktiskt kommer påverka fotonen. Varje gång fotonen träffar en glasskiva i spelet är det 50% chans att fotonen bara åker igenom glaset, och 50% chans att den reflekteras. Detta representeras i spelet med att den enstaka fotonen delas upp i två då den träffar glasskivan. Då alla möjliga fotoner träffat något objekt visar spelet hur stor chans det är för den faktiska fotonen att träffa varje objekt som den har en chans att träffa. Detta ger viss kvalitativ förståelse av detta område, men inte mycket mer än så. Fysiken påverkar inte heller gameplay i någon större grad. Gameplayen är fortfarande att placera ut objekt för att leda fotonen, vilket hade kunnat göras till samma effekt utan att använda kvantfysik.

A.2 Fotbollsspel

I följande avsnitt av presenteras dokumentationen av spelgenren fotbollsspel.

A.2.1 Pele's Soccer

Pele's Soccer är en av de första tilläggen av videospel inom spelgenren fotbollsspel och lanserades år 1981 för spelkonsolen *Atari 2600* [5]. Spelet hade för sin tid revolutionerande 2D-grafik där spelaren hade en vy ovanifrån fotbollsplanen. På fotbollsplanen har spelaren två backar och en anfallare samt en datorstyrd målvakt. Backarna och anfallaren har formationen av en triangel och bibehåller formationen då spelaren förflyttar dem. Om målvakten räddade bollen får spelaren en temporär möjlighet att styra målvakten för att sparka ut bollen på fotbollsplanen. Spelreglerna är vanliga fotbollsregler, där målet är att sparka fotbollen in i motståndarens bur. Det finns dock inte en domare med i spelet för att döma frispark till spelaren vid en tackling. Spelet fokuserar på realism för att försöka ge spelaren en så verklig upplevelse som möjligt. Spelaren har dock inte möjlighet att styra varje spelare individuellt utan kan endast styra spelarna som en formation, och har möjlighet att förflytta bollen via en kollision med någon av sina spelbara karaktärer.

A.2.2 Super Mario Strikers

Super Mario Strikers, känt som *Mario Smash football* i Europa och Australien, är ett fotbollsspel skapat av *Next Level Games* åt spelkonsolen *GameCube* [10]. Spelaren har möjlighet att välja bland kända Nintendo-karaktärer som till exempel *Mario*, *Luigi* och *Donkey Kong*. Efter att spelaren har valt sin karaktär har den möjlighet att välja till ytterligare tre medspelare, som till exempel *Koopa* eller *Toad*. Spelet är helt i 3D. Spelaren har en snett-ovanifrån-vy som efterliknar den vyn man ofta ser från riktiga fotbollsmatcher sända i TV. Spelaren har även en målvakt som är datorstyrd, förutom då målvakten har bollen, då spelaren har möjlighet att kontrollera karaktären. Spelreglerna efterliknar vanliga fotbollsregler där man spelar fem mot fem och målet är att förflytta fotbollen in i motståndarens bur. Spelet saknar dock en aktiv domare och spelaren har även möjlighet att tackla motståndaren utan att få varningar eller utvisningar. Bollen kan inte heller gå utanför spelplanen, vilket i fotboll leder till ett inkast eller hörna. Spelaren har möjlighet att styra alla karaktärer individuellt, förutom målvakten. Spelet är designat sådant att spelaren direkt via ett knapptryck kan passa en medspelare eller skjuta direkt mot motståndarens bur. Spelet är väldigt automatiserat vilket leder till enkel kontroll och manövrering då spelaren inte själv behöver sikta. Spelaren har möjligheten att bestämma starthastigheten hos bollen då den blir sparkad. Detta sker via en impuls som spelaren tillför till bollen för att ändra dess rörelsemängd. Om spelaren tillför en tillräckligt stor impuls på motståndarens halvsida kan den genomföra en *Super Strike*. En *Super Strike* är en inbyggd funktion i spelet där spelarens karaktär hoppar högt upp i luften och samlar in sin styrka för att sedan skjuta iväg ett extremt kraftfullt skott på mål. Speldesignen är mer inriktad mot fantasy än mot realism vilket noteras via spelfunktioner så som *Super Strike*.

A.2.3 FIFA 14

FIFA 14 är ett intressant tillägg i spelserien *FIFA*, skapat av *EA Canada* och som lanserades 2013 för plattformarna Xbox, PC, Playstation, Android, iOS, Nintendo 3DS och Wii. [6]. Speldesignen strävar efter realism och spelet är i 3D. Spelaren har möjlighet att välja bland spelkaraktärer som efterliknar riktiga fotbollsspelare med olika skicklighetsnivåer. Spelreglerna är samma som vanliga fotbollsregler där man spelar elva mot elva och målet är att förflytta fotbollen in i motståndarens bur. Spelet har med en aktiv domare som dömer vid exempelvis hårda tacklingar, då bollen går utanför spelplanen eller vid offside. Spelaren har möjlighet att styra samtliga spelare på planen och har möjlighet att passa bollen mellan olika spelare. Spelaren har även möjlighet att dribbla bort motståndaren för att föra fram bollen närmare mot målet. Då spelaren sparkar bollen tillför spelaren en impuls på bollen vilket ändrar bollens rörelsemängd. Spelaren har kontroll över denna impuls vilket gör det möjligt för spelaren att bestämma hastigheten hos bollen. Speldesignen strävar efter realism för att ge spelaren en så verklig upplevelse som möjligt.

A.3 First Person Shooters (FPS)

I följande avsnitt av presenteras dokumentationen av spelgenren First Person Shooters.

A.3.1 Wolfenstein 3D

Wolfenstein 3D lanserades år 1992 av *id Software* och är känt som det första FPS-spelet[21]. Spelet utspelar sig under andra världskriget och spelet använder sig av en tidig 3D-grafik. *Wolfenstein 3D* var det spelet som gav liv till spelgenren FPS och sägs vara det första spelet som helt utspelar sig ur ett förstahandsperspektiv. Spelaren följer soldaten B.J. Blazkowicz som försöker fly från nazisterna ur ett slott vid namn Wolfenstein. Spelaren har möjlighet att plocka poäng som ligger utspridda runt om slottet och även fylla på sin hälsa samt fylla på med ammunition. Spelaren har också en uppsättning liv som försvinner om spelaren dör från en attack av en motståndare. Spelaren förlorar då alla extraliven är konsumerade och spelaren tillslut dör i strid.

Wolfenstein 3D saknar en realistisk fysikmotor. Till exempel påverkas spelaren inte av någon rekyl då spelare avlossar en kula från sitt gevär. *Wolfenstein 3D* visar dock på en tidig användning av 3D-grafik samt första fullständiga användning av ett förstahandsperspektiv.

A.3.2 Halo: Combat Evolved (Halo CE)

Halo: Combat Evolved lanserades år 2001 av spelföretaget *Bungie* för spelkonsolen Xbox[23]. Halo CE är ett 3D-spel av spelgenren FPS-Action som utspelar sig i ett ultramodernt 2552. Människan har lyckats kolonisera andra planeter men i sin jakt på nya kolonier kom människan i konflikt med en utomjordisk allians vid namn *The Covenant*[101]. Människan tvingas tillbaka mot jorden efter att ha förlorat ett flertal kolonier till *The Covenant*. Spelaren styr en karaktär vid namn *Master Chief*[102] som tillhör en grupp supersoldater vid namn *Spartans*[103]. Vid reträtt träffar spelaren på en artificiell planet i form av en ring. Spelaren landar på den artificiella planeten som visade sig vara ett supervapen vid namn *Halo*. *The Covenant* visar sig vara ett religiöst samfund som hyllar supervapnet *Halo* och tror att *Halo* är nyckeln till frälsning. Efter att spelaren stridit hårt om kontroll över *Halo* dyker ytterligare en ondskefull fiende upp vid namn *The Flood*[104] som visar sig vara ett virus som sprider sig över galaxen.

Halo CE är det första i *Halo*-serien och spelmekanik bygger delvis på realistisk fysik. Ett exempel på realistisk fysik är kurvningen hos granater då de kastas. Granaten påverkas konstant av ett gravitationsfält som accelererar granaten nedåt mot markytan. Detta skapar en kurvad bana som granaten följer då den kastas iväg. Ett exempel som visar på en icke-realistisk fysik är rekyl hos gevären. Spelaren har möjligheten att välja bland en mängd olika vapen som vid skottlossning visar på en minimal visuell rekyl som inte påverkar spelarens precision. En realistisk rekyl kan beskrivas av Newtons tredje lag[22] som säger att kraften som kulan påverkas av vid skottlossning kommer ge upphov till en kraft som agerar i motsatt riktning till kulans. Denna kraft borde bidra till en försämring hos spelarens precision. Detta är dock inte fallet för *Halo CE* där spelaren har möjligheten att skjuta med helautomatiska vapen utan att behöva hantera en rekyl.

A.3.3 Call of Duty 4: Modern Warfare (CoD4)

Call of Duty 4: Modern Warfare, även känt som *CoD4*, lanserades år 2007 av *Activision* för plattformarna Windows, Mac OS X, Xbox 360 och Playstation 3[28]. Spelet är utvecklat av *Infinity Ward*. *CoD4* är ett tillskott i spelserien *Call of Duty* vars spelmiljö efterliknar modern krigföring medan dess föregångare efterliknade andra världskriget[105]. Spelet har två huvudspellägen vid namn *Campaign* och *Multiplayer*. I *Campaign* får spelaren köra genom spelets berättelse där spelaren tar rollen bland annat som SAS-soldaten *John "Soap" MacTavish*[106], som måste bekämpa den ryska ultranationalistiska rörelsen ledd av *Imran Zakhaev*[107]. I *Multiplayer* kan spelaren bekämpa andra spelare online i ett flertal olika spellägen som *Team Deathmatch* och *Search and Destroy*.

CoD4s spelmekanik bygger helt på fysikområdets mekanik. Spelaren rör sig i en 3D-värld och är ständigt påverkad av en gravitationskraft som drar spelaren mot marken. Ett exempel på klassisk mekanik är då spelaren avlossar en kula från sitt vapen och bemöts av rekyl. Rekyl kan beskrivas av Newtons tredje lag[22] som den kraft som uppkommer från spelarens vapen vid skottlossning och som verkar i motsatt riktning mot den avlossade kulans riktning. Om spelaren använder ett automatisk gevär och avlossar flera kulor hastigt kommer spelaren att bestraffas med en stark rekyl som försämrar spelarens precision.

A.3.4 Battlefield 4 (BF4)

Battlefield 4, även känt som *BF4*, lanserades år 2011 av *EA* och är skapat av det svenska företaget *DICE* för plattformarna Windows, Xbox 360, Xbox One, Playstation 3 och Playstation 4[24]. *BF4* är uppdelat i två spellägen, spela genom en berättelse eller spela online själv eller i lag mot andra spelare. I spelets berättelse följer man marinsoldaten *Sgt. Daniel "Reck" Recker*[108] som bekämpar den ondskefulla *Admiral Chang*[109]. *Admiral Chang* har som motiv att ta över makten från den kinesiska regeringen för att tillsammans med Ryssland starta ett globalt krig med USA. Spelaren följer med i flera av *Sgt. Reckers* olika uppdrag för att försöka hindra *Admiral Chang* från att ta över makten i Kina.

BF4 drivs av spelmotorn *Frostbite 3*[110] som är skapat av *EA DICE* och som är speciellt byggd för att driva alla spel i *Battlefield* spelserien. Denna motor har välfungerande fysik som omfattar till exempel optiskt ljus, vågor i vatten, aerodynamiska effekter från roderbladen hos spelets helikoptrar, granater som påverkas av gravitation vid kast och rekyl hos gevär vid skottlossning. Fysikområdet sträcker sig längre än klassisk mekanik. Det finns även fysikaliska fenomen i

spelet som omfattar optik och aerodynamik. Spelet tycks ha som vision att skapa en realistisk krigsmiljö för spelaren som prövar spelarens kunskaper, samarbete och skicklighet. Ett exempel på kunskap är om spelaren väljer att försöka fälla en motståndare på långt avstånd. Om spelaren väljer att skjuta rakt på motståndaren via sitt kikarsikte kommer spelaren tyvärr att missa då kulan konstant påverkas av gravitation och därmed kommer att avböja nedåt. Spelaren måste alltså ta denna effekt i åtanke vid långdistanskonflikt.

A.3.5 Counter Strike: Global Offensive (CS:GO)

Counter Strike: Global offensive, även känt som *CS:GO*, lanserades 2012 av *Valve Corporation* och *Hidden Path Entertainment* för plattformarna Windows, Mac OS X, Playstation 3 och Xbox 360[27]. *CS:GO* är ett *First Person Shooter*-spel som är det senaste tillägget i serien *Counter Strike*. Spelmekaniken går ut på att spelaren försöker ensam eller tillsammans med ett lag om fem skjuta ned motståndare för att samla poäng. Spelet har flera olika spellägen, däribland spelläget *Bomb Defusal Mode*[111]. *Bomb Defusal Mode* går ut på att spelaren tar rollen som terrorist eller anti-terrorist, där terroristerna har möjligheten att plantera en bomb på två utvalda platser på kartan medan anti-terroristerna skall försvara dessa platser eller desarmera bomben. Spelaren får poäng för varje motståndare som spelaren kan skjuta ned eller om spelaren planterar bomben som terrorist eller desarmerar bomben som anti-terrorist. Rundan är över då alla spelare i ett lag har fallit i strid, då bomben har exploderat, då bomben har desarmerats, eller då en timer når noll och bomben inte ännu planterats. Vanligtvis består varje lag av fem spelare och man kör bäst av 30 rundor. Efter halvtid (15 rundor) byter lagen roll som terrorist eller anti-terrorist. För att lyckas vinna en runda i *Bomb Defusal Mode* krävs, utöver en förmåga att sikta väl, mycket lagarbete och strategi, varför *CS:GO* kan klassas som en *Tactical Shooter*. Å andra sidan finns det även spellägen såsom *Deathmatch*[112], där det enda målet är att under tio minuter döda så många andra spelare i andra laget som möjligt för att få högst poäng. I detta spelläge kan *CS:GO* ses som mer av en *Arcade Shooter*.

CS:GO's grafik efterliknar modern krigsföring och är i 3D. Spelet strävar efter en realistisk men balanserad gameplay. Fysiken i spelet består helt av mekanik. Ett tydligt exempel på detta är då spelaren har möjlighet att kasta en handgranat. Handgranaten följer en typisk newtonsk kurva då den konstant är påverkad av en gravitation nedåt. Med newtonsk kurva menas att ett föremål faller med rörelsen $-1/2gt^2$, där g är gravitationskonstanten och t beskriver tiden. Eventuellt vindmotstånd och avvikelser på grund av blåst är försummade för att spelaren ska få en större kontroll och precision i sina kast. Ett annat exempel på fysik är rekyl som uppstår då spelaren avfyrar ett vapen. Enligt Newtons tredje lag[22] kommer vapnet vid skottlossning ge en impuls i motsatt riktning som kulan går. Detta motstånd kallas ofta för vapenrekyl. Om spelaren beslutar för att avlossa flera kulor samtidigt kommer spelaren uppleva en stark rekyl som försämrar spelarens precision.

A.4 Racingspel

I följande avsnitt av presenteras dokumentationen av spelgenren racingspel.

A.4.1 Space Race

Space Race lanserades år 1973 av Atari för arkadplattformen[31]. *Space Race* var bland de första racingspel som uppkom på 70-talet. Spelmekaniken går ut på att flyga ett rymdskepp genom ett asteroidfält. Spelets grafik är väldigt enkel och allting existerar i 2D med en vy ovanifrån. Spelaren har möjlighet att förflytta sitt rymdskepp vertikalt och om spelaren lyckas ta sig förbi asteroidfältet får spelaren ett poäng. Asteroidfältet är inte statiskt utan åker med konstant hastighet horisontellt. Spelaren behöver därför koordinera sig genom asteroidfältet genom att åka upp och ned. Om spelaren kolliderar med en asteroid börjar spelaren om från start.

Space Race har en väldigt förenklad spelmekanik. Därmed är ämnet fysik inte relevant i sammanhanget då spelaren inte upplever en större anknytning med spelvärlden. *Space Race* var bland de första racingspelen och hade en väldigt begränsad spelmekanik.

A.4.2 Super Mario Kart

Super Mario Kart lanserades år 1992 av spelföretaget *Nintendo* för sin spelkonsol SNES[37]. Spelaren har möjlighet att välja bland kända Nintendo karaktärer som *Mario*[113], *Luigi*[114] och *Bowser*[115] som alla har unika skickligheter och förmågor. Spelaren har möjlighet att spela ensam eller tillsammans med ytterligare en spelare mot datorstyrda motståndare. Spelaren tävlar i gokart och måste besegra datorn för att låsa upp nya karaktärer och nya banor. Man har möjlighet att ställa in svårighetsgrad på motståndarna. Spelaren har möjlighet att under sin resa plocka upp specialvapen som bland annat bananskal, sköldpaddsskal och svampar. Dessa specialvapen kan användas för att bland annat öka sin topphastighet, göra sig immun mot skada och för att beskjuta motståndaren. Om spelaren inte lyckas vinna loppet kan spelaren börja om igen men kommer inte att låsa upp nya karaktärer eller nya banor.

Spelmekaniken är byggd med en icke-realistisk fysik. Vid kollision tar spelarens gokart ingen visuell skada. Vid kollision mellan två gokarter sker alltid en elastisk stöt. Spelmekaniken har delvis korrekt fysik som gravitation men förenklar delar av fysiken för att förenkla gameplay. Spelet strävar mer mot fantasy än mot realism. Spelet är i 2D och är projicerat med en 3D-effekt.

A.4.3 Grand Theft Auto V (GTA V)

Grand Theft Auto V (GTA V) lanserades år 2013 av *Rockstar Games* för plattformarna Playstation 3, Playstation 4, Xbox 360, Xbox One och Microsoft Windows[36]. *GTA V* introducerar spelaren till en verklighetsbaserad 3D-värld som grundas i amerikansk samhällskritik. *GTA V* utspelar sig på en ö vid namn *Los Santos* där man följer tre karaktärer vid namn *Michael Townley*[116], *Trevor Philips*[117] och *Franklin Clinton*[118]. Historian utspelar sig i modern tid (tidigt 2000-tal) och omfattar bankrån, biljakter och kidnappningar. För att framhäva sin verklighetstroga 3D-värld använder sig *GTA V* av delvis realistisk fysik.

*GTA V*s gameplay omfattar en mängd olika delmoment som bygger på en realistiskt baserad fysik men fokus för denna dokumentation centreras kring fordon och biljakter. Det som mest kännetecknar *GTA V*s bilfysik är hanteringen av kollisioner samt de eventuella skador som fordon tar efter en kollision. I verkligheten är kollisioner mellan två fordon oftast väldigt oelastisk. Med en oelastisk stöt menas att föremål efter kollision förblir statiska jämfört med en fullständigt elastisk stöt där föremål istället åker iväg med en eventuellt förändrad hastighet. I *GTA V* strävar man efter att efterlikna realistiska krockar genom att simulera relativt oelastiska stötar, dock inte fullständigt oelastiska. I samband med detta orsakas en del skador på fordon och andra föremål som exempelvis ljusstolpar. Eventuella skador på fordon omfattar exempelvis spruckna rutor, en intryckt motorhuv och förstörda strålkastare. Utöver detta kan föraren i bilen även kastas ut genom rutan om föraren kolliderar med ett föremål i en tillräckligt hög hastighet. Kort kan fenomenet förklaras som att fordonet bromsas in och övergår till ett statiskt läge medan föraren fortgår med bilens ursprungliga hastighet. Därav kastas föraren ut genom rutor vid en kraftig inbromsning.

A.4.4 Need for Speed (NFS)

Need for Speed är en spelserie som totalt omfattar 28 racingspel och samtliga är lanserade av *Electronic Arts (EA)*[33]. Det första tillägget i spelserien heter *Road & Track Presents: The Need for Speed* och lanserades år 1994, och det senaste heter *Need for Speed: Payback* och lanserades år 2017. I samtliga spel i spelserien är huvudsyftet att skapa ett racingspel där spelare har möjlighet att tävla mot datorn (och andra spelare online i de senare tilläggen) för att låsa upp nya fordon och nya banor att tävla på. I de senare tilläggen finns även andra typer av uppgifter som spelaren kan ta sig an. Dessa typer av uppgifter omfattar bland annat biljakter där spelaren måste hinna ikapp ett fordon.

I jämförelse mellan det första och senaste tillägget i spelserien kan man observera en tydlig utveckling av grafik, gameplay och en allt mer realistisk fysik. I *Road & Track Presents: The Need for Speed* observeras en väldigt elastisk hantering av kollisioner och en helt avsaknad av eventuella skador hos fordon och föremål efter en kollision medan i *Need for Speed: Payback* observeras en allt mer oelastisk hantering av krockar och fordon samt föremål tar skada efter kollisioner i form av exempelvis krossade rutor och bucklor på fordon. Man kan även notera en utvidgning hos hantering av friktion mellan däck och markyta där man i det senaste tillägget har en dynamisk friktion som beror på om spelaren exempelvis kör på asfalt eller på grus.

A.5 Plattformsspel

I följande avsnitt presenteras dokumentationen av spelgenren plattformsspel.

A.5.1 Mario Bros., Super Mario Bros. och Super Mario 64

Mario Bros. lanserades år 1983 för plattformen *Arcade* av spelföretaget *Nintendo*[119]. *Mario Bros.* tillhör en subgenre som kallas *Comical action game* och som karaktäriseras av att spelaren endast kan röra sig inom en spelram. Om spelaren går utanför spelramen åt höger i *Mario Bros.* teleporteras spelarens karaktär och kommer ut från vänster direkt. Detta gäller dock endast om spelaren går utanför spelramen i horisontell riktning. Om spelaren istället går utanför spelramen i vertikal riktning sker inte denna momentana teleportering. Spelaren följer bröderna *Mario* och *Luigi* som måste besegra olika typer av varelser som lever nere i avloppen. Spelmekniken är väldigt simpel och bygger på att spelaren måste ta sig mellan olika plattformar för att undvika varelserna som kryper ut genom avloppen. Spelaren tvingas även att besegra varelserna för att nå nästa nivå, och för att få poäng. Detta görs genom att spelaren hoppar underifrån på golvet varelserna går på för att få varelserna att snubbla och ramla omkull. När de väl ramlat måste spelaren snabbt ta sig till den nu hjälplösa varelsen och tackla den, innan denne hinner ställa sig upp igen. Tacklingen eliminerar varelsen och ger spelaren poäng. Spelfysiken var väldigt enkel och bestod endast av gravitation och dämpade elastiska stötar.

Efter sin succé med *Mario Bros.* utvecklade spelföretaget *Nintendo* uppföljaren *Super Mario Bros.*[40] för sin spelkonsol *Nintendo Entertainment System (NES)*. Spelet hade byggts vidare med en berättelse, en övergång från fast spelram till en rörlig spelram (ej automatisk) och en antagonist vid namn *Bowser*. Spelaren följer *Mario* som försöker rädda den bortrövade prinsessan *Peach* från den onde *Bowser*. *Super Mario Bros.* sålde över 40 miljoner kopior världen över och sägs vara den stora revolutionen inom plattformsspel. Dock grävdes ingen ny spelfysik fram jämfört med företrädaren *Mario Bros.*. Spelfysiken omfattar fortfarande enkel mekanik. En gravitationskraft och dämpade elastiska stötar.

Spelvärlden tog nya tag år 1996 då spelföretaget *Nintendo* lanserade sitt senaste *Super Mario*-spel, *Super Mario 64*[44] för spelkonsolen *Nintendo 64 (N64)*. *Super Mario 64* omfattade nu en fullständig 3D-värld som spelaren kunde utforska. Spelets berättelse efterliknade berättelsen som företrädaren *Super Mario Bros.* utspelade. Protagonisten *Mario* måste bekämpa antagonisten *Bowser* för att rädda den bortrövade prinsessan *Peach*. *Super Mario 64* utvecklade dock spelfysiken jämfört med företrädaren genom att presentera både vridmoment och icke-inertiella referensramar. All mekanik från tidigare företrädare lämnades kvar så som gravitation och dämpade elastiska stötar. Dock innehöll *Super Mario 64* ofysikaliska delmoment som exempelvis icke-kontinuerliga kraftmoment. Detta ofysikaliska fenomen uppstår då spelaren vid ett tillfälle ska kasta iväg den onde *Bowser*. Detta kast efterliknar ett släggkast där *Mario* roterar i en cirkel med *Bowser* för att accelerera. Genom att utnyttja centrifugalkraften kastas *Bowser* iväg efter att en tillräckligt högt vridmoment har skapats. Dock kastas *Bowser* iväg ortogonalt från *Marios* kropp som momentat bromsas in efter att ha kastat iväg *Bowser*. I verkligheten skulle *Marios* kropp fortsätta rotera en liten stund och *Bowser* skulle istället inte heller åka spikrakt iväg ortogonalt från *Mario*. Ur ett etiskt perspektiv kan detta vara missvisande och leda till en felaktig förståelse av fysik. Dock kan detta ofysikaliska fenomen ha skapats för att förbättra gameplay och ge spelaren ett enklare delmoment att klara av.

A.5.2 Braid

Braid[47] lanserades år 2008 av *Number None* för spelkonsolen Xbox 360. *Braid* ingår i subgenren *pusselbaserade plattformsspel* och som namnet antyder bygger spelmekniken på att lösa pussel. Pusselbaserade plattformsspel efterliknar vanliga plattformsspel där spelaren hoppar från plattform till plattform för att ta sig vidare men skillnaden är att spelaren måste klara av fysikbaserade pussel för att avancera till nästkommande nivå. Det som gör *Braid* unikt är dess hantering av tid. I *Braid* följer man protagonisten Tim som försöker rädda en prinsessa från ett monster. Tim har en unik förmåga att kunna manipulera tiden och gå tillbaka i tiden för att ändra på det förflutna. Denna typ av spelmekanik kan anses vara ofysikalisk men speciell och allmän relativitetsteori talar om möjligheten att gå tillbaka i tiden. Om man går med ljusets hastighet "stannar" klockan och om man skulle åka snabbare än ljusets hastighet skulle man gå tillbaka i tiden. Enligt hypoteser kan

det vara möjligt att åka snabbare än ljusets hastighet men allmänt kan en massa aldrig röra sig lika snabbt eller snabbare än ljusets hastighet. Så att helt utesluta möjligheten att resa runt i tiden kan inte förkastas med de ekvationer som bygger upp speciell och allmän relativitetsteori. Utöver tidsfenomenet påverkas spelaren av klassisk mekanik. Spelaren måste lösa mekanikbaserade pussel för att avancera vidare till nästkommande nivå. Realistisk fysik har alltså en väsentlig roll i att leverera trovärdig och välfungerade gameplay då spelmekanik just är baserade på mekanikbaserade pussel med inslag av tidsresor.

A.5.3 Broforce

Broforce[120] lanserades för Microsoft Windows år 2015 och senare för Playstation 4 år 2016. Spelet är skapat av *Free Lives* och är publicerat av *Devolver Digital*. *Broforce* tillhör subgenren *Run-and-gun platform game* som karaktäriseras av blandningen mellan ett skjutarspel och ett plattformsspel. *Broforce* har ett flertal valbara karaktärer som spelaren låser upp under spelets gång och vissa karaktärer är unika för olika nivåer. *Broforce* tar ett samhällskritiskt perspektiv på USAs maktposition runt om i världen och blandar den med humor. Resultatet är att spelaren har möjlighet att spränga byggnader, "liberate bad guys" och rädda dagen genom att spränga saker i bitar. Bland de valbara karaktärerna finns *Rambo*, *Bro Hard* och *Bro Lee* som alla är baserade på hjältar inom den amerikanska action-filmbranschen[121]. *Broforce* bygger på en relativt realistisk fysik med surrealistiska tillägg av komisk effekt. Fysiken är centrerad kring klassisk mekanik med inslag av gravitation, rörelsemängd och dämpade elastiska stötar. Fysiken i sig spelar inte en central roll i gameplay utan gameplay är mer centrerad kring explosioner och vapenhantering. Spelet hanterar rekyl som en animation istället för en fysikalisk kraft som försämrar precision. Granater hanteras som newtonska kastparabler[122]. *Broforce* är ett typiskt exempel på en *Run-and-gun platform*-spel och visar på hur man förenklar delar av vapenhantering för att förbättra gameplay. Förenklingar som kraftigt dämpad rekyl är väldigt vanligt inom subgenren.

A.5.4 Temple Run

Temple Run lanserades år 2011 av *Imangi Studios* för plattformarna iOS, Android och Windows Phone 8[46]. *Temple Run* är utvecklat av *Imangi Studios*. Spelet tillhör en subgenre vid namn *Endless runner* som karaktäriseras av att det i huvudsak är ett plattformsspel vars spelram är oändlig längs färdriktningen. *Temple Run* har inget förbestämt slut utan spelet avslutas då spelaren exempelvis kolliderar med ett hinder. *Temple Run* är byggt på 3D-grafik och spelmiljön är en kombination av träsk, exotisk djungel och tempelbyggnader. Spelets huvudsakliga spelmekanik bygger på att spelaren ska försöka springa så långt som möjligt. Längs vägen finns det hinder som spelaren måste manövrera sig förbi och det existerar även guldpenningar och *power ups* som spelaren kan samla på. Spelaren blir bedömd på hur långt spelaren kan springa i spelets egna längdenheter.

Temple Run har en speldesign som är uppbyggt av klassisk mekanik. Spelet anpassar gravitationen efter balansering istället för att skapa en realistisk representation av gravitation. Detta på grund av att spelet försöker anpassa sina kastparabler efter sina hinder, troligtvis för att förenkla gameplay för spelaren. Spelet hanterar även kollisioner mellan spelarens karaktär och de hinder som är utplacerade längs spelplanen.

A.5.5 Defy Gravity

Defy Gravity är ett plattformsspel som lanserades i mars år 2016 av *Fish Factory Games*. Spelet går ut på att komma till slutet av varje bana genom att, förutom att springa, hoppa, dubbelhoppa (med hjälp av en Jetpack) och studsa på toppen av annars dödliga bollar skjuta ut så kallade *wells* ur sin *Gravity Gun*, där blå *grav-wells* attraherar objekt, och orangea *anti-wells* repellerar. Dessa *wells* beskrivs som fickor av gravitation respektive *anti-gravitation*, vilket är ett fenomen som antagligen inte finns - åtminstone har det inte bevisats. Med dessa *wells* kan spelaren öka längden och höjden på sina hopp genom repelleringen från *anti-wells*, eller hänga mitt i luften med de attraherande *grav-wells*, för att manövrera sig förbi de många hindren på varje bana. Spelaren kan också aktivera en gravitations-sköld för att skydda sig från effekten av sina *wells*. Även vissa, guldfärgade, objekt påverkas av *wells*. Spelaren kan alltså även använda *wells* för att flytta på vissa plattformar eller för att flytta farliga föremål ur vägen för spelaren. De flesta objekt är dock grå, och blir inte påverkade av *wells*.

Defy Gravity är något inkonsekvent i att inga av de objekt som agerar som plattformar eller hinder i spelet påverkas av någon gravitation, förutom den som skapas av *wells*. Detta skulle kunna förklaras med att spelet utspelar sig i rymden, men konstigt nog dras spelaren själv ned mot marken av vad som åtminstone verkar vara gravitation. En annan konstig grej har med spelarens *Gravity Gun* och dess *wells* att göra. Det är att endast guldfärgade objekt och spelaren som berörs av *gravitationen* från *wells*, medan grå objekt (då spelaren har sin *Gravity Shield* aktiverad) är opåverkade av *wells* gravitation. Hade det faktiskt varit så att *wells* hade skapat gravitation och *anti-gravitation*, borde de påverkat alla objekt, inte bara vissa utvalda. Detta får det att verka som att *wells* egentligen har mer med elektromagnetism att göra, då magneter både kan attrahera och repellera magnetiska objekt.

Förutom dessa skumma saker är fysiken i *Defy Gravity* ganska förtroendeingivande. När det kommer till mekaniken, vilket nog främst handlar om spelarens hopp och hur objekt studsar, studsar objekten som de antagligen hade gjort i rymden. De kan dock inte studsa på varandra, så det de studsar på är lod- eller vinkelräta väggar. Dessutom känns spelarens hopp som något som skulle kunna hända på en plats med svagare gravitation än jorden. *Wells* är helt okej i vissa aspekter. Till exempel påverkar *wells* objekt starkare ju närmre centrum de är, vilket både magneter och gravitation gör. Men eftersom *anti-wells* skjuter ifrån sig objekt, och allt inte blir påverkat av varken *grav-* eller *anti-wells*, låter det mer som att *wells* är magneter med bara en pol än vad spelet beskriver dem som, alltså fickor av gravitation. I det fallet fungerar de någorlunda realistiskt i att även dessa påverkar objekt som är nära starkare än långt borta, även om de elektromagnetiska krafterna avtar långsammare än vad de hade gjort för en riktig magnet.

A.5.6 The Electric Shocktopus

The Electric Shocktopus, släppt av *TestTubeGames* i Januari år 2015 för Mac OS, Windows och Linux, är ett plattformsspel där man spelar som en bläckfisk som kan byta sin elektriska laddning. På banorna finns, förutom vanliga plattformar och en massa hinder såsom taggar och kanoner, elektriskt laddade block som skapar magnetfält som *The Electric Shocktopus* kan använda för att ta sig fram då han ändrat sin laddning. Spelet introducerar saker som Coulombs lag och högerhands-regeln, med flera, och *The Electric Shocktopus* måste använda sig av dessa för att ta sig vidare i spelet. Dessa lagar används flitigt genom spelets gång för att ta sig vidare genom spelet.

A.6 Action-Adventure

I följande avsnitt presenteras dokumentationen av spelgenren Action-Adventure.

A.6.1 Adventure

Adventure släpptes 1979-1980 till spelkonsolen Atari 2600 och anses vara det första spelet inom Action-Adventure genren [48]. Den spelbara karaktären är en simpel kvadrat och målet är att återfinna en magisk bägare som stulits av en ond trollkarl och gömmts ute i världen. Under spelets gång kommer man träffa på olika fiender i form av drakar, samt olika pussel och labyrinter som var och en måste besegras för att kunna ta sig vidare. För att besegra drakarna måste spelaren hitta olika föremål som är gömda runt om i spelet. Ett av dessa föremål är en magnet som används för att attrahera nycklar placerade på ställen som spelaren annars inte kommer åt, vilket är en kvantitativ användning av fysik. Spelarens karaktär kan bli uppäten och dödad av en drake om denne hamnar inom dess räckhåll. I detta scenario ges spelaren ett alternativ att återuppliva karaktären och placera denna vid slottet istället för att starta om spelet från början. Detta spel anses vara det första som introducerade detta koncept, som har blivit en standard i moderna spel.

Spelet består av tre nivåer där varje nivå är svårare än den föregående. En fladdermus åker runt i världen och flyttar runt på föremål samt drakar, varav föremålen även kan stjälas från spelaren. Fladdermusen går dock att jaga ikapp vilket resulterar i att föremålet kan tas av spelaren. Spelaren kan endast hålla ett föremål i taget. Spelet är inte särskilt realistiskt då det innehåller drakar och magi, samt att spelaren utan större problem lyckas lyfta stora objekt som exempelvis broar.

A.6.2 Castlevania: Harmony of despair

Castlevania: Harmony of despair (HoD) är ett 2D-action-adventure-spel tillhörande subgenren *platform-adventure* släppt 2010 [49]. Spelet är uppbyggt av sex olika nivåer, alla bestående av olika fiender och slutligen en boss. Målet är att hitta och besegra bossen innan tiden tar slut. Sex olika spelbara karaktärer finns, alla med olika förmågor. En av dessa förmågor är att med hjälp av magneter dra sig till olika metalliska ytor, vilket är en kvantitativ men orealistisk användning av magnetism. Hinder i form av pendlar finns även på några av nivåerna i spelet, dessa är dock helt opåverkade av fenomen som luftmotstånd och kollisioner och använder sig alltså inte av realistisk fysik. Vid början av en nivå är karaktären svag vilket gör det svårt för oerfarna spelare att klara den. Detta gör att man kan behöva leta efter nya vapen och ny utrustning för att uppgradera karaktären först.

A.6.3 Mirror's Edge

Mirror's Edge är ett 3D-action-adventure-spel tillhörande subgenren *platform-adventure* som släpptes 2008 [50]. Spelet spelas ur ett förstapersonsperspektiv, och går ut på att navigera sig igenom en stad med hjälp av rörelser som är kraftigt inspirerade av parkour och freerunning. Pussel i form av olika hinder förekommer som gör att spelaren måste utnyttja kombinerad av rörelser för att nå sitt mål. Fiender i form av vakter som försöker döda dig träffas på runt om i spelet; spelaren kan besegra dessa med hjälp av närstrid eller skjutvapen.

Fysik har använts bland annat för att skapa realistiska visuella effekter i spelet. Exempel på detta är hur glas betar sig när fönster krossas, hur vind påverkar löv i omgivningen samt hur olika plastdraperier påverkas av att någon springer igenom dem. Annan fysik i spelet är hur spelaren rör sig samt tar skada vid kollisioner och fall. Mycket kompromisser mellan realism och bra gameplay har dock gjorts. Exempel på dessa är hur snabbt man återhämtar sig efter fall samt att vissa av rörelserna är osannolika att de skulle kunna utföras av en människa på riktigt.

A.6.4 Zelda: Breath of the wild

Zelda: Breath of the wild släpptes i Mars 2017 av Nintendo och är ett action-adventure-spel som bland annat använder sig av pussel för att ta sig vidare i spelet [51]. *Breath of the wild* hyllades snabbt av både kritiker och spelare för sin kreativa användning av fysik i de olika delarna av spelet. Spelet har en kraftig fysikmotor som tillåter spelaren att experimentera med olika typer av fysik för att lösa pussel, besegra fiender och ta sig framåt i spelet. De har använt sig av både icke-realistisk fysik för att passa spelets art-style samt mer verklighetstrogen fysik; målet ska ha varit att skapa en fysikmotor med bra balans mellan realism och positiv påverkan på gameplay. Exempel på implementerad inkorrekt fysik i spelet är ragdollfysik för animeringar då en varelse dör, dynamisk väderfysik samt användandet av magneter för att flytta föremål. Bland användandet av korrekt fysik så finns "gungbräde-fysik" då exempelvis en stor sten släpps på en gungbräda med spelaren ståendes i andra änden, hur vind påverkar objekt samt spridning av eld.

Spelet är det första i spelserien som har en öppen värld i motsats till uppdelade nivåer och ställen. Detta tillåter spelaren att fritt utforska världen och exempelvis göra uppdrag som finns i spelet för nöje snarare än som ett sätt att ta sig vidare i spelet. Dessa kan dock resultera i att spelaren får mer pengar, liv eller andra fördelar i spelet. Mindre tempel bestående av fiender och pussel måste alla klaras av för att ta sig vidare i spelet.

A.7 Spel som är svåra att dela in i genrer

Alla spel är inte lätta att dela in i en viss genre. Detta stycke listar några sådana spel. Dessa räknas in under "Övriga spel" i kartläggningen i avsnitt 5.3.

A.7.1 Velocity Raptor

Velocity Raptor släpptes i December år 2011 av TestTubeGames på deras hemsida[63]. I *Velocity Raptor* är man en väldigt snabb velociraptor, som ska ta sig igenom sin nemesiss, *Professor Rex*, gömställe som är fullt av fällor såsom kanoner, farliga dörrar, och hålor med vatten. Marken är dessutom helt av is vilket gör att man lätt halkar ned i vattnet. De första tre eller så banorna är

helt normala; man bara går från början av banan till slutet, medan man undviker kanonskott och att falla i vatten. Men snart drar Professor Rex i en spak, som gör att ljusets hastighet minskas till endast 3 miles per timma. Detta gör att, då spelaren närmar sig ljusets hastighet, vilket nu går väldigt snabbt, sker en massa relativistiska effekter på världen utifrån spelarens perspektiv. Det första spelaren introduceras till är längdkontraktion - när velociraptorn springer verkar han bli mindre. Detta kan användas för att göra det lättare att undvika kanonskott, då en mindre velociraptor har svårare att bli träffad av kanonskotten. Även färgen på vissa utvalda objekt ändras beroende på hur snabbt man åker mot eller ifrån dem, vilket spelet använder som ett pussel - man ska matcha så att alla olikfärgade block "får" samma färg, genom att man helt enkelt springer bort eller från dem. Längre fram introduceras även tidsdilation, alltså att tiden, ur omvärldens perspektiv, passerar långsammare för objekt som rör sig i hastigheter nära ljusets hastighet.

A.7.2 Quantum Moves

Quantum Moves är ett spel baserat på kvantfysik som släpptes år 2012 till plattformarna Windows, Linux och Mac OS och är en del av projektet ScienceAtHome[60]. Spelet går ut på att flytta en väldigt nedkyld atom från ett ställe till ett annat, utan att tappa bort dess position. Atomen representeras av en våg. Mittan av vågen representerar positionen där atomen är mest sannolik att vara, medan resten av vågen är ställen där atomen är något mindre sannolik att befinna sig, men där den ändå kan vara. För att flytta atomen används en i spelet osynlig laser, som gör en energidal som atomen kan trilla ned i. Lasern, och därmed dalen, kan sedan flyttas på. I denna dal verkar atom-vågen bete sig ungefär som en vätska i att den skvalpar runt. Utmaningen i spelet ligger alltså i att försiktigt föra över så mycket av atom-vågen som möjligt till den angivna positionen. Är man inte försiktig skvalpar vågen över och det blir väldigt osäkert var atomen faktiskt befinner sig, vilket resulterar i låga poäng. *Quantum Moves* må vara baserat på kvantfysik, men själva fysiken har ingen riktig inverkan på gameplay. Det är väldigt svårt att förstå vad spelet egentligen handlar om utan att gå in på ScienceAtHomes hemsida och läsa om det. Detta spel är gjort för att bygga en kvantdator. Datan på hur man rör musen sparas och skickas iväg för att hjälpa forskningen i detta område, eftersom höga poäng innebär att man använde en bra väg för att flytta atomen till slutpositionen.

A.7.3 A Slower Speed Of Light

A Slower Speed Of Light av MIT Game Lab släpptes år 2012 för plattformarna Windows, Linux och Mac OS[62]. I *A Slower Speed Of Light* spelar man som ett barn som nyligen dött, och som försöker nå ljuset. Ljuset är dock, som bekant, väldigt snabbt. För att sakta ned och kunna hinna ikapp ljuset ska spelaren därför samla på sig sfärer som ligger utspridda i denna tredimensionella värld. Dessa sfärer saktar ned ljusets hastighet. Att ljusets hastighet blir långsammare gör att spelaren kommer närmre ljusets hastighet då den springer, vilket ger upphov till en mängd relativistiska fenomen. Bland annat får spelaren, då den nått en hastighet tillräckligt nära ljusets, se infraröda färger, som vanligtvis inte kan ses med det mänskliga ögat. Ju fler sfärer spelaren samlar, desto långsammare blir ljuset, och därmed blir de relativistiska effekterna mer uppenbara. Detta försvårar också spelet. Det finns dock inga egentliga hinder som kan skada spelaren. Svårigheten ligger helt i att de relativistiska fenomenen gör det svårare för spelaren att se vad den håller på med, och att kontrollera sin position.

A Slower Speed Of Light är gjort för att på ett förståeligt sätt introducera relativistisk fysik, och ger en kvalitativ förståelse för hur världen skulle se ut om man vore nära ljusets hastighet. Den ger också viss kvantitativ förståelse om fysiken, eftersom spelaren gradvis, i en långsam takt, kommer närmare ljusets hastighet.

A.7.4 Plasma Pong och Fluid Table-Tennis

Plasma Pong var ett Indie-spel skapat 2007 för plattformarna Windows och Mac OS X.[123] *Plasma Pong* spelade precis som den gamla klassikern *Pong*, men med en Twist: Spelet utspelade sig nämligen i en vätska. Spelarnas racketar kunde få denna vätska att röra sig bort från racketen, vilket gjorde att den nu snabbt rörande vätskan knuffade bollen, vilket gjorde det svårare att förutse hur bollen skulle röra sig. Racketarna kunde även suga upp bollen, för att fånga den en begränsad tid. Detta påverkade dock inte vätskan. Vätskan, som fick en viss färg beroende på

vilken spelare som fick den att röra på sig, krockade med resten av vätskan, vilket gjorde att två strömmar av vätska som möttes i mitten krockade med varandra för att skapa vackra mönster och tydligt visa fysiken spelet använde: Fluidmekanik.

Plasma Pong dog tyvärr snabbt, på grund av ATARI som äger PONG-märket. Spelet *Fluid Table-Tennis*[124], en HTML5-kopia av *Plasma Pong*, gör dock att spelet lever vidare.[125]

A.8 Övriga spel

Följande tabell, Tabell 3, listar de spel där fysiken analyserats, men ingen längre analys av spelet som helhet har gjorts. Vilken av de genrer som beskrivits i föregående avsnitt i denna bilaga varje spel tillhör nämns också. I kartläggningen, kapitel 5.3, placeras dessa spel i tillhörande genre. I det fall ett spel kan placeras i fler än en genre, kartläggs det spelet som den i tabellen förstnämnda genren.

Tabell 3: Övriga spel

Spel	Genre
Angry Birds	Pussel
Bars and Balance	Pussel
Prisme	Pussel
Cut the Rope	Pussel
Fireboy and Watergirl	Pussel/Plattform
Forza Motorsport 7	Racing
Hammerfight	Övrigt
Kerbal Space Program	Övrigt
LEGO Pirates of the Caribbean	Action-Adventure
Max & the Magic Marker	Plattform
Optika	Pussel
Oxygen not Included	Övrigt
Pixeljunk Shooter	Övrigt
Polyball	Plattform
Quantum Conundrum	Plattform/Pussel
Rocket League	Fotboll/Racing
Space Race	Racing
Where's my Water	Pussel

A.9 Sammanställning

Tabellen i detta avsnitt, Tabell 4, sammanställer samtliga spel som använts för kartläggning och analys i detta arbete, samt vilken sorts fysik de använder. Det bör noteras att tabellen inte tar hänsyn till hur realistisk fysiken i spelet är.

Tabell 4: Samtliga spel som användes för kartläggning och analys

Spel	Fysik	Mekanik	Optik	Elektro- magnetism	Relativitets- teori	Kvant- fysik	Strömnings- mekanik	Termo- dynamik
Adventure				X				
Angry Birds		X						
A Slower Speed Of Light					X			
Bars and Balance		X						
Battlefield 4		X					X	
Braid		X						
Bridge Builder		X						
Broforce		X						
Call of Duty 4		X						
Castlevania: HoD		X		X				
Counter Strike: GO		X						
Crayon Physics Deluxe		X						
Cut the Rope		X						
Defy Gravity		X		X				
The Electric Shocktopus		X		X				
FIFA 14		X					X	
Fireboy and Watergirl		X	X					
Forza Motorsport 7		X						
Grand Theft Auto V		X						
Halo: Combat Evolved		X						
Hammerfight		X						
The Incredible Machine		X						
Kerbal Space Program		X					X	
Lego Pirates of the Caribbean		X	X					
Mario Bros.		X						
Max & the Magic Marker		X						
Mirror's Edge		X						
Monster Strike		X						
Optika			X					
Oxygen Not Included		X						X
Peggle		X						
Pele's Soccer		X						
Pixeljunk Shooter		X					X	
Plasma Pong		X					X	
Polyball		X						
Portal		X						
Prisme			X					
Quantum Conondrum		X						
The Quantum Game						X		
Quantum Moves						X		
Rocket League		X						
Super Mario Bros		X						
Super Mario Kart		X						
Super Mario Strikers		X						
Super Mario 64		X						
Space Race								
The Talos Principle		X	X					
Temple Run		X						
Velocity Raptor		X			X			
Where's my Water							X	
Wolfenstein 3D								
World of Goo		X						
Zelda: Breath of the Wild		X		X				

Bidragsrapport

Dessa sidor sammanfattar vad som har gjorts av vem. Vi skrev alla vår egen sektion, med insyn från resten av gruppen.

Åke Andersson

Denna sektion vad jag, Åke, har bidragit med till projektet.

Ansvarsområden

- Jag tog ansvar för uppstarten genom att förbereda en dagordning inför första dagen gruppen träffades.
- Jag har varit ansvarig för fysiken i *Spacetimewar*.
- Jag har varit ansvarig för implementationerna *Tic Tac Ket* och *Lagom Action*.
- Jag har varit ansvarig för filhiarkin i L^AT_EX-källkoden.
- Jag var med i slutredovisningen och förberedde mitt innehåll till den.

Bidrag i form av idéer

- Jag kom på tillsammans med gruppen, huvudsakligen Thomas, att det fanns en designrymd av krökta kurvor. Detta blev senare implementationen *Hydrocurve*.
- Jag kom själv på hur vi kunde göra ett spel av allmän relativitet: *Spacetimewar*.
- Jag kom tillsammans med Thomas på idéer till spel, som ej blev fullständigt implementerade. Dessa var ett konstruktionsspel om termodynamik, en tornbyggarspel om elektrostatik, och ett bullethellspel om elektromagnetism. Det är samma bullethellspelidé som Razmus senare oberoende(?) kom på.
- Jag kom tillsammans med Thomas på idéer som blev implementationerna *Tic Tac Ket* och *Lagom Action*.
- Jag diskuterade med Thomas hans idé om fallande vatten under påverkan av elektrostatik.
- Jag kom på en idé om ett spel där antalet geometriska dimensioner kan manipuleras, efter jag hade hört Staffan (vår handledare) berätta om en sci-fi-roman.

Bidrag till implementationerna

- Jag skrev själv under första veckan en minimal implementation av bullethellspelet i swing, med syfte att visa hur magnetiska dipoler kan användas för att deflektera laddade partiklar.
- Jag skrev själv under första veckan en minimal implementation av allmän relativitet i swing, med syfte att demonstrera informationsfördröjning.
- Jag hjälpte gruppen som arbetade med *Hydrocurve* med hur fysik implementeras, Runge-Kutta-metoden. Den implementeringsdetaljen blev senare utbytt mot funktioner från Box2D.
- Jag skrev fysiken i implementationen *Spacetimewar*, Algot skrev resten. Av de 62 commits som gjordes var 48 mina.
- Jag skrev själv implementationen *Tic Tac Ket*.
- Jag skrev själv implementationen *Lagom Action*.

Bidrag till planeringsrapporten

- Jag skrev själv sektionen *Fysik*.
- Jag gjorde huvuddelen av tidsplaneringen. Hela tidsplaneringen gjordes tillsammans med resten av gruppen.
- Jag korrekturläste diverse sektioner.

Bidrag till slutrapporten

Jag var huvudförfattare till:

- Inledningen till teorikapitlet 3.
- Teorin för newtonsk mekanik 3.2.
- Teorin för speciell relativitet 3.3.1.
- Teorin för allmän relativitet 3.3.2.
- Teorin för kvantfysik 3.5.
- Teorin för Lagranges ekvationer 3.6.
- Teorin för fourieroptik 3.7.2.
- Teorin för elektromagnetism (Thomas skrev första stycket, sedan fyllde jag på) 3.9.
- Beskrivandet av hur vi kom på idéer 4.4.
- Utvecklandet av *Tic Tac Ket* 4.5.3.
- Utvecklandet av *Lagom Action* 4.5.4.
- Resultatdelen om *Tic Tac Ket* 5.1.3.
- Resultatdelen om *Lagom Action* 5.1.4.
- Idéen om fourieroptik 5.2.4.
- Möjligheten att använda relativitet 5.4.1.
- Möjligheten att använda kvantfysik 5.4.3.
- Möjligheten att använda lagrangeformalismen 5.4.4.
- Resultatdiskussion om *Tic Tac Ket* 6.1.4.
- Resultatdiskussion om *Lagom Action* 6.1.5.
- Resultatdiskussion om elektromagnetism 6.1.7.
- Metoddiskussion om *Tic Tac Ket* 6.2.4.
- Metoddiskussion om *Lagom Action* 6.2.4.
- De ej ämnesspecifika delarna i slutsatsen 7.
- Slutsatsen om relativitetsteori 7.
- Slutsatsen om kvantfysik 7.
- Slutsatsen om lagrangeformalismen 7.
- Slutsatsen om elektromagnetism 7.

Jag skrev enstaka stycken i:

- Utvecklandet av *Spacetimewar* 4.5.1.
- Resultatdelen om *Spacetimewar* 5.1.1.
- Spelidén om laddade partiklar 5.2.2.
- Validitet och generaliserbarhet 6.3.

Bidrag till bidragsrapporten

- Jag skrev den inledande texten.
- Jag skrev denna sektion om mig själv.

Jag rättade även språkliga fel i andras sektioner. Jag och Algot fixade tillsammans layouten i rapporten.

Thomas Jonsson Damgaard

Bidrag i form av idéer

- Jag har kommit på grundidén till Hydrocurve (tillsammans med Åke).
- Jag har kommit på grundidén till Spacetimewar (tillsammans med Åke).
- Jag har kommit på grundidén till Lagom Action (tillsammans med Åke).
- Jag har kommit på grundidén till Tic Tac Ket (tillsammans med Åke).
- Jag har kommit på grundidén till Elektrofäll (tillsammans med Åke).
- Jag har kommit på grundidén till ett termspel (tillsammans med Åke).
- Jag har kommit på grundidén till tower-spelet (tillsammans med Åke).
- Jag har kommit på grundidén till bullet hell-spelet (tillsammans med Åke).

Bidrag till implementationerna

- Jag har hjälpt till att implementera ubåtspelet

Bidrag till slutrapporten

- Jag har tagit det huvudsakliga ansvaret bakom dokumentationen av existerande spel, analysen av existerande spel inom olika spelgenrer samt olika fysikområden med en resulterande kartläggning, diskussion, slutsats, med bidrag från: Åke - hjälpt till att hitta spel för kartläggning, Anna - Hjälpt till att skriva om fysikområdet optik, korrekturläst, Algot - Hjälpt till med kartläggning och har tagit ansvar för att skriva om denna kartläggning i rapporten, hitta spel med exotisk fysik och analysera dessa, korrekturläst, Razmus - skrivit om Action-Adventure Spel, har skrivit metoddiskussion helt själv, hjälpt till att skriva resultatdiskussion och slutsats.
- Anna har ritat en digitalbild av min skiss på Elektrofäll som jag har kommit på tillsammans med Åke
- Jag har formulerat syftet i rapporten och avgränsningar
- Förberett halvtidsredovisning med Algot
- Jag har skrivit om ej implementerade spel (Elektrofäll EM) tillsammans med Åke vars grundidé har framtagits tillsammans med Åke.
- Skrivit om elektromagnetism i teoridelen (3.9) med bidrag från Åke.

Utöver detta har jag:

- Förberett och hållit i halvtidsredovisningen tillsammans med Algot
- Tagit ansvar för att alla i gruppen har fått varit delaktiga att bestämma syftet och fått arbeta med något som varit av intresse. Detta gjordes främst genom att fråga runt och anteckna.
- Fokuserat mindre på att implementera minispiel (utöver Hydrocurve) och istället tagit huvudsakliga ansvaret för dokumentationen.
- Likt alla andra har även jag korrekturläst och kommenterat/rättat till olika delar i rapporten.
- Genomfört den muntliga opponeringen tillsammans med Algot

Algot Nilsson

I planeringsrapporten skrev jag:

- Introduktionen
- Bakgrunden
- Var delaktig i skapandet av tidsplaneringen, tillsammans med resten av gruppen.

I slutrapporten var huvudsaklig författare till kapitlen:

- *1 Introduktion*
- *4.5.1 Utvecklingen av Spacetimewar*
- *5.1.1 Implementationen Spacetimewar*
- *5.3 Kartläggning*, basically sammanställningen av dokumentationen
- *6.1.2 Diskussion av Spacetimewar*
- *6.2.4 Spacetimewar i Diskussion av utvecklingen av specifika implementationer*,
- Ett avsnitt om framtida arbete av Spacetimewar. Denna kom ej med i rapporten, men Anna la in mycket av det i *6.5, framtida arbeten*.

(Åke hjälpte till att kolla så att det jag skrev om fysikaliska fenomen faktiskt var korrekt, tacktack) Dessutom fixade Jag och Åke tillsammans layouten för rapporten.

Bland dokumentationen (I Bilaga A) skrev jag:

- *A.5.5 Defy Gravity*
- *A.5.6 The Electric Shocktopus*
- Alla spel i *A.7, Spel som är svåra att dela in i genrer*
- *A.8, Övriga spel*
- *A.9, Sammanställning*, (den där sjuka tabellen där allt är i bokstavsordning.)

(Thomas var väldigt bra och stöttande vid all Bilage-skrivning!)

Jag var dessutom delaktig i implementationerna

- *Spacetimewar*, med Åke
- Grafiken i *A Maze in Light*.

Utöver detta har jag:

- Förberett och hållit i halvtidsredovisningen tillsammans med Thomas.
- Varit delaktig i dokumentation av spel, och analysen av dessa.
- Bokat de flesta grupprum till grupp- och handledarmöten, och ansvarat för att meddela vår handledare om dessa.
- Självklart har jag korrekturläst och kommenterat/rättat till rapporten!
- Genomfört den muntliga opponeringen tillsammans med Thomas.

Anna Romeborn

Ansvarsområden

- Bidrog mycket till strukturen för hur vi la upp arbetet från början, med mötesprotokoll m.m.
- Huvudsakligt redaktionellt ansvar för både rapporten och planeringsrapporten. Både angående struktur och språk. Korrekturläste i extra stor utsträckning. Skapade t.ex. ett dokument där våra “regler” för strukturen skrevs upp (och skrev många av dem)
- Deltagit i slutredovisningen och förberett mina delar till den

Bidrag i form av idéer

- Har kommit med flera idéer till implementationer, varav en var den som ledde till A Maze in Light och en som blev den ej implementerade idén om elnät.

Implementationer

- Hydrocurve: Huvudsaklig ansvarig för fysiken, bidragit till implementationen
- A Maze in Light: Huvudsaklig ansvarig för fysiken, mycket idéer, mycket implementering

Huvudsaklig författare till följande kapitel i planeringsrapporten

- MDA
- Etiska och samhällsliga aspekter

Bidrag till slutrapporten

Huvudsaklig författare:

- Tack till
- 2.7.2 Bakgrund: Optik i videospel (med hjälp av Thomas)
- 3.1 Teori: MDA
- 3.4 Teori: Strömningsmekanik
- 3.7.1 Teori: Stråloptik
- 3.8 Teori: Kretslära
- 4.1 Genomförande: Tidsplanering
- 4.5.2 Genomförande: Hydrocurve (med hjälp av Razmus och Thomas) (även figur)
- 4.5.5 Genomförande: A Maze in Light (med Razmus) (även figurer)
- 5.1.5 Resultat: A Maze in Light (med Razmus) (även figurer)
- 5.2.1 Resultat: Ej implementerad idé om kretslära
- 5.4.2 Resultat: Möjlighet att använda strömningsmekanik i gameplay
- 5.4.5 Resultat: Möjlighet att använda stråloptik i gameplay
- 6.1.7 Diskussion: Ej implementerad idé om kretslära
- 6.2.4 Metoddiskussion: A Maze in Light (med Razmus)
- 6.2.4 Metoddiskussion: Hydrocurve
- 6.4 Diskussion: Etiska och samhällsliga aspekter

- 6.5 Diskussion: Framtida arbete (med hjälp av Algot)
 - A.1.7 Dokumentation av spel: Incredible Machine
- Delförfattare: (bidragit med några stycken)
- 5.1.2 Resultat: Hydrocurve (fixade även figurerna)
 - 6.1.3 Diskussion: Hydrocurve
 - 6.1.6 Diskussion: A Maze in Light
 - 7 Slutsats (skrivit om strömningsmekanik och kretslära)

Övrigt:

- Skrivit många korta mellanstycken mellan rubriker
- Hittade mallen som används för de första sidorna och modifierade den för att få den att fungera för vår rapport
- Gjorde bilden till framsidan
- Renritade bilden till Elektrofall (utifrån Thomas skiss)

Rasmus Strandell

Ansvarsområden

- Skapade vårt Git repo
- Lärde folk i gruppen att använda Git samt hjälpte till att fixa problem som uppstod
- Presenterade den agila arbetsmetoden för mjukvaruutveckling vi använde under projektets gång
- Ansvarig för val av spelmotor
- Deltagit i slutredovisningen och förberett mina delar till den

Bidrag i form av idéer

- Kommit på och utvecklat de ej implementerade spelförslagen inom Elektriska fält (varav det andra av de förslagen baserats på en idé från Åke).

Bidrag till implementationerna

- Gjort grafik och implementering för Hydrocurve
- Implementering för A Maze in Light

Bidrag till planeringsrapporten

- Skrev hela metodiksektionen.
- Skrev hela planeringssektionen (gjorde ej tidsplanen men bidrog till den)
- Korrekturläst och rättat på olika ställen

Bidrag till slutrapporten Som huvudsaklig författare:

- Skrivit abstracten/sammandraget
- 2.6 Gjort analys på action-adventure genren, samt skrivit om detta
- 4.2 Skrivit metod för dokumentation av spel.
- 4.3 Skrivit metod för mjukvaruutveckling
- 5.1.2 Skrivit om resultatet av Hydrocurve.
- 5.1.5 Skrivit resultatet av A Maze in Light med Anna.
- 5.2.2 Skrivit om spelförslagen som tillhör elektriska laddningar, samt gjort skisserna tillhörande dessa.
- 6.1.1 Skrivit resultatdiskussionen kring analysen (med Thomas)
- 6.1.3 Skrivit om diskussionen av Hydrocurve.
- 6.1.6 Skrivit om diskussionen av A Maze in Light.
- 6.2 Skrivit Diskussion av metod, genomförande och avgränsningar
- 6.2.4 Skrivit om diskussionen av metod av A Maze in Light.
- 6.3 Skrivit Validitet och generaliserbarhet
- A.6 Dokumenterat de spel som analyserats inom action-adventure genren

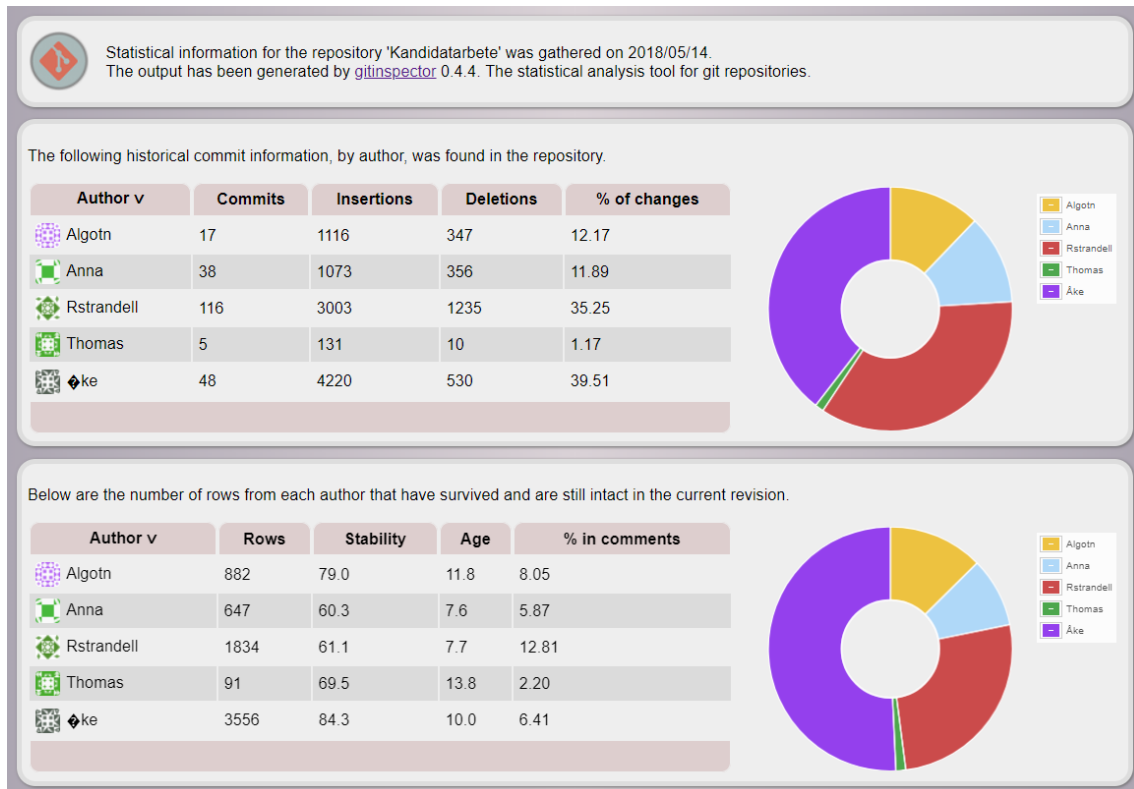
Som delförfattare:

- 4.5.2 Skrivit lite (något/några stycke(n)) om genomförande av ubåt
- 4.5.5 Skrivit lite (något/några stycke(n)) om genomförande av optik
- 6.1.7 Skrivit diskussion av labyrint/laddningsspelet.
- 7 Skrivit stråloptikdelen i slutsatsen

Allmänt:

- Korrekturläst och rättat/ändrat i rapporten

Git inspector



Figur 28: Statistik över vårt implementerande. Gjord med gitinspector för masterbranchen.