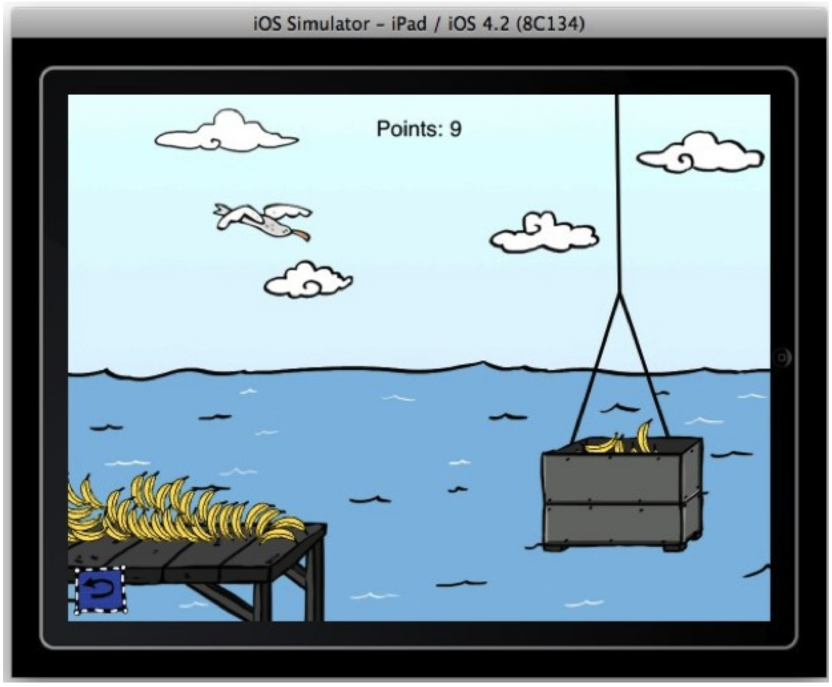
CHALMERS



Gamelets

Providing diegetic elements through injection into digital children's books

Master of Science Thesis in the programme Interaction Design (Game Design)

Marcus Antonsson

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
Göteborg, Sweden, December 2012

Gamelets
Providing diegetic elements through injection into digital children's books

Marcus Antonsson

Examiner: Staffan Björk

Chalmers University of Technology
University of Gothenburg
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Cover:
Screenshot from implementation

Department of Computer Science and Engineering
Göteborg, Sweden December 2012

Contents

Acknowledgements

I would like to thank Appmazed for the possibility to perform this thesis and I would also like to thank my supervisor Staffan Björk for great input and for helping me get this thesis done.

Abstract

Tablets and smart phones using touch interfaces are becoming more and more common. New ideas and applications are brought to life every day, making daily use of computers more ubiquitous. With this new technology comes the possibility to read not only the news, but to read literature in new ways. The possibility with hypermedia means that narratives can be told using not only the text, but also with animation and interactable computer graphics and sound.

In this thesis the idea of using *gamelets* are discussed. Gamelets are interactions and smaller games used within digital books in order to support the diegesis and add to the experience provided by a book.

Several proof of concept prototypes was designed and implemented in order to analyze and create suggestions on what to consider when designing for an already existing narrative. These suggestions that can help utilize the modern technology in order to affect (or not affect) the existing narrative.

1 Introduction

Stories have been used in most cultures as a form of entertainment among other things. In the early days, this was done orally and probably by carving onto cave walls. Later, as technologies were invented, it was made possible to be presented in book form. Further technological innovation lead to stories being presented through film at movie theaters, and later directly in peoples homes through television and video.

New technology has in the past provided new ways of performing storytelling, and still continue to provide even more possibilities. With the release and increasing popularity of touch based smart phones and tablets, more and more people are using this technology as a way of reading books and newspapers. The possibility to use this new medium creates new ways to create interactive books and worlds.

There are companies that have realized there can be big business in producing material that present a kind of narrative on new technology and Appmazed is one of these. They have seen that there are publishers that hold publishing rights for lots of material, but with no means of producing anything themselves. In turn, usually the authors of these books hold the right to create new material, something that neither publishers nor developing companies do. So the publishers have interest in re-publish existing material without changing anything in the content, but rather build a new product around the existing material that can be experienced by and sold to customers.

The basis for this thesis grew out of a meeting with the CEO of Appmazed, Edward Antov. Discussions started about what they do at *Appmazed* and what they had in mind with their future product.

1.1 Research question

Appmazed point of entry is to present a way for the publishers to re-use existing material that they already have the publishing rights for and that has been released in another form. In this case with *JerryMaya* it has previously been on hardcover paper books. Presenting it in a new media format using games and interactions present the possibility to affect the narrative, possibly even changing the way it was supposed to be experienced. Given this, the following describes the aim of the project.

> *How can smaller games and interactions be integrated into interactive children's books.*

1.2 Purpose

This thesis will focus on what to consider when designing interactions and games that should support the narrative in digital books. This will be done by gathering research from scientific articles, design several small games and interaction proof of concepts that would be candidates for this kind of integration.

These proof of concepts and prototypes will be reflected upon in order to answer the questions mentioned in section 1.1 and will also be used by Appmazed showcasing their product.

1.3 Limitations

There are several different platforms - old and new - that can provide storytelling combined with and games and interaction. It could be in a web-browser, installed on a regular computer or as an embedded solution in a physical toy to mention some. *Appmazed* wishes to develop their tool on touch interfaces and mainly on the iPad. That is why, in this thesis, the work will be limited to touch interfaces on tablets.

Appmazed is interested in presenting a way for publishers to re-use already existing material, due to this special consideration will be taken on how the already existing narrative can be affected with the addition of game modules and interactive elements in a book.

The games and interactions should be generic but will be evolved with regard to a certain genre. *Appmazed* already had mock-ups created based on the children's book series *JerryMaya*[1] and it will be used as a starting point. Since this book series aims towards children in the ages 7-9, this will be inherited in the games and interactions that will be designed/prototyped.

The number of games and interactions that will be designed is dependent on what is feasible to implement during the limited time of this thesis.

The ideas and implementations will not be user tested in order to determine anything stated in section 1.1. Given the limited amount of time for this project and regarding the time factors for pre-study, implementation and still be able to write the report it was deemed not feasible.

2 Background

The iPad is a touch based tablet computer developed and made popular by Apple following their touch based mobile phone, the iPhone. It is promoted as a product were the user can, among other things, play games and read e-books[2]. Parallel with the release of the iPad, Apple released a Software Development Kit (SDK) to encourage companies to design and implement applications for their product.

There are already several forms of narrative that have emerged over time with new technology. Now days it's very common with portable music devices, which in turn made it possible for sound books to become a popular phenomenon. A traditional book told by a professional, recorded in a format supported by most portable players so that you can just listen instead of reading. For instance while running or commuting.

Another creative way of creating a new experience that have emerged are by pop-up illustrations in books. This is when you turn a page and there are folds of hard paper, that unfold when you open a page and show an illustration, as shown in figure 1

Another narrative form are game books, as the *Lone Wolf* series by Dever and Chalk [1984]. This was written books that have branching plots with role-playing elements put into them. The player create a character that have different skills and attributes that can affect which path the story take, the player also make choices that affect the outcome and length of the story.

A related example would be digital Games that are largely story driven. For example, the children's game *ButaVX: Justice fighter*[3]. The player takes on the role as ButaVX, a piglet humanoid with a very vivid imagination. In the intro cut scene you can see ButaVX in the middle of a football game on a stadium, filled to the brim by a large audience. After attempting to score a goal the ball flies into and crashes a window, there the scenery changes and you are informed that it was all a fantasy and that ButaVX is actually playing on a field by himself. The game is designed as a classic Japanese Role Playing Game (JRPG) and continues to tell the story of how ButaVX gets his ball back through the embedded and emergent narrative in the game.

[1] http://www.lassemaja.net/

[2] http://www.apple.com/pr/library/2010/01/27Apple-Launches-iPad.html

[3] http://www.nekomuragames.com/games/cute-rpg/butavx.html

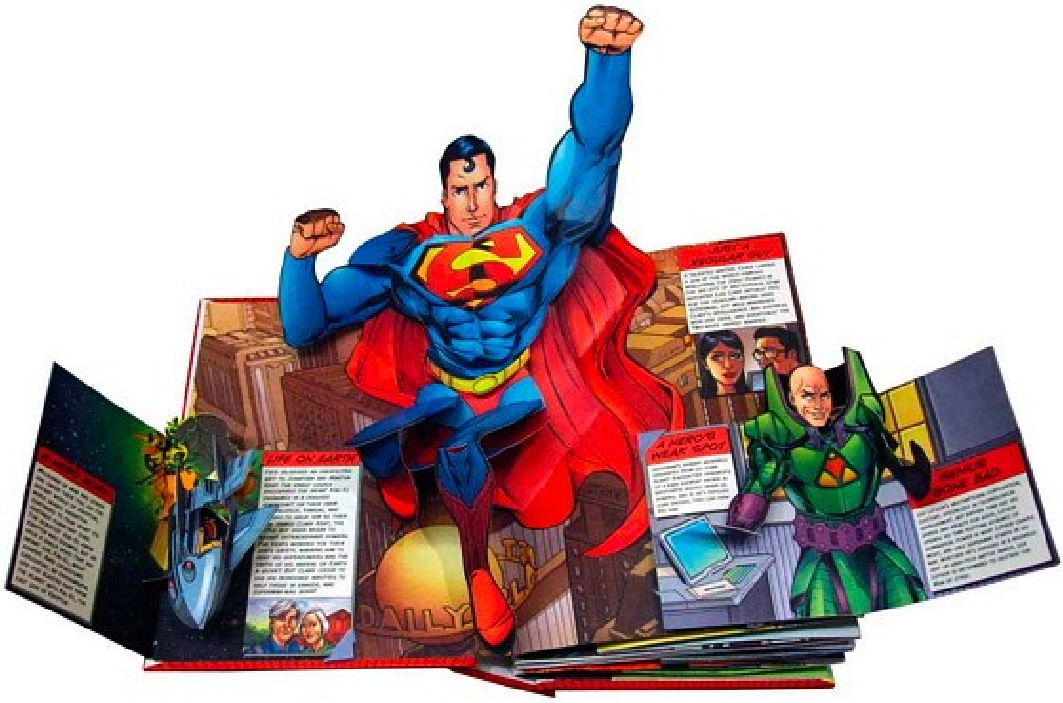


Figure 1: Pop-up superman, from DC Super-Heroes Ultimate Pop-Up Book

ButaVX: Justice fighter is a more traditional digital game experienced on a regular computer. It has a fictional world and an embedded back story that justifies why the player need to make ButaVX take certain actions and it tells a story by using cut-scenes between gameplay, environmental storytelling and dialogue. The narrative experience is more about gameplay, since it is through gameplay the story it is told.

Another example would be *What's bothering Carl?*[4], a digital book with an embedded narrative supported by interactions. The story about the Cyclops Carl takes place in a fictional world were there exists cyclops and giants. It's installed as it's own entity and by using the mouse as input the reader can click letters in the text and trigger animations in the images. The story is told by a narrator as well as in text and each page have interactive elements on them. Some key words are interactable as well and will be read out loud as feedback to that interaction. Some other more or less hidden hot spots are placed among the pages, they can easily be spotted due to the mouse turning gold in color when hovering them.

The above are related to a bit older technologies relative to the iPad, that in turn hold lots of products for distribution through their store.

One popular game based on the children's cartoon The Smurfs, is the Smurfs' Village[5] by Capcom Interactive. The player build and tend a village full of smurfs by, for instance, building houses and shrubs or by sowing and harvesting crops. The game requires the player to get back to their gaming session in intervals, and time flows in the game even if the player turns off the application.

Cut the rope[6] is a popular physics based game. The goal is to get a piece of candy into the little green creature's mouth by cutting ropes at the right time. It contains of several

[4] http://www.storyfort.com/what-is-bothering-carl/

[5] http://itunes.apple.com/us/app/smurfs-village/id399648212

[6] http://itunes.apple.com/se/app/cut-the-rope/id380293530

maps that are part of so called map packs, which usually introduce a new gameplay element into the game like for instance teleport hats.

3 Theoretic frame of reference

Games have been a part of human culture for a long time, with Senet as one of the oldest dating back a couple of thousand years BC [Juul, 2005]. Digital games that today are a big part of the gaming culture go back to the 1960s in USA [Juul, 2005]. There simple game like artifacts where created as hobby projects and the first was probably one called *Spacewar!* created by Steve Russel. Later on in the seventies started development of the first home console, game as such were entering the homes of people more rapid than anyone thought possible.

Figure 2: An instance of *Spacewar!* running on an old PDP-1

3.1 Narrative

When you have a traditional narrative, a conflict is set in motion as soon as the protagonists face an obstacle that in some way stand in the way of them accomplishing their goalsFullerton [2008]. Fullerton [2008] provides a description of the dramatic arc, here called a narrative arc for consistency. When the conflict starts there need to be tension increase with time only to become better at the end, this forming the arc along two axes were the horizontal axis representing time and the other representing tension. This forms the arc shape from where it gets it name. At first there is the *exposition*, followed by *complication* which in turn is followed by the *climax* which then quickly drops with the *resolution*.

Salen and Zimmerman [2004] describe J. Hillis Miller's framework for narrative as a definition when they talk about narratives in games. That model consist of three elements that are used when trying to understand narrative, here described in the same way as by Salen and Zimmerman;

Situation: A narrative has an initial state, a change in that state, and insight brought about by that change. This process constitutes the *events* of a narrative.

Character: A narrative is not merely a series of events, but a *personification* of events through a medium such as language. Miller doesn't mean character in the usual sense of fictional persona, but rather the process by which "character is created out of signs." This component references narratives as not just events that take place in the world, but at represented events, events that occur via *systems of representation*.

Form: Representation is constituted by *patterning and repetition*. This is true on every level of a narrative, whether it is the material form of the story or its conceptual themes.

3.1.1 Diegetic vs. narrative

When talking about stories two words reoccur quite often: narrative and diegesis. This section is meant to define what is meant by these when used through this thesis. ***Diegesis*** is how the story is told and ***Narrative*** is the actual story told.

For instance the gamelets (excluding the highlighted words in the text) in *What's bothering Carl?* have diegetic properties, but doesn't always follow the main narrative as in figures 4 and 5. They still contribute to the experience and the telling of the story.

3.2 What is a game?

There are no clear uniformed definition of what a game is. Chris Crawford has a definition that is described by Fullerton [2008]. It's a fine grain definition that distinguishes four types of play, the one at the top is classed as the most interactive:

Games are rule-based systems in which the goal is for one player to win. They involve "opposing players who acknowledge and respond to one another's actions. The difference between games and puzzles has little to do with mechanics; we can easily turn many puzzles and athletic challenges into games and vice versa."

Puzzles are rule-based systems, like games, but the goal is to find a solution, not to beat an opponent. Unlike games, puzzles have little replay value.

Toys are manipulable, like puzzles, but there is no fixed goal.

Stories involve fantasy play, like toys, but the cannot be changed or manipulated by the player.

Another more abstract definition is provided by Salen and Zimmerman [2004], where they write that:

> "A game is a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome."

This definition still covers separating stories and toys from games and puzzles, but it also mashes games and puzzles into one category, namely games. There are of course limit cases of which some games fall outside this definition but are still considered to be games. One of these is role-playing games that [usually] lack a quantifiable outcome.

3.3 Games and narratives

Most games feature a narrative introduction and a back story, but is it the same as a narrative? Aarseth [1997] compares what he calls Cybertexts with traditional narrative work. Traditional literary work have a reader, and even though the reader can be highly engaged and involved in the narrative, can still only be considered as a spectator in the same way as a football game spectator. The spectator can shout, take sides and even shout abuse but she's not a player.

> "He [the reader] cannot have the player's pleasure of influence: "Let's se what happens when I do *this*!". The reader's pleasure is the pleasure of a voyeur. Safe, but impotent." [Aarseth, 1997]

A Cybertext reader on the other hand, have the possibility to get lost within the narrative not only by not understanding what it means, but by loosing some part of the text entirely while other parts might be revealed. In the same sense that Aarseth says that a reader is safe, he also says that a Cybertext reader is not safe, in it's choices and effort, and that she therefore cannot be regarded as a reader.

> "To claim there is no difference between games and narratives is to ignore essential qualities of both categories"[Aarseth, 1997]

Let's take a look at another view of narrative play in more detail. Salen and Zimmerman [2004] describe two structures for narrative play, *embedded* and *emergent*.

Embedded narrative is when there exist a pre-generated narrative content that exist prior to the player's interaction to the game. This narrative is experienced as a story context, providing meaning for the actions the player have to take. As an example, in the game Half-Life 2[7] the player is permanently cast as Gordon Freeman, a theoretical physicist who helps a human resistance group fight the alien oppressors that has enslaved humanity. The story about Gordon and that he's trying to free humanity from the Combine is an *embedded* narrative, experienced by player interaction but exists independently.

Emergent narrative instead arise during interaction with the complex system of the game. This means that it is depending on the outcome of the player's interaction and choices, also the player's fantasy might help create emergent narrative elements that add to the experience. To use Half-Life 2 as example again, one part where Gordon have to find Eli's lab by making way through the city controlled by the Combine. Is it a smooth trip? Will it be stressful for Gordon, damaging or traumatic or will it be like a day at the beach just dancing between bullets and almost doing cartwheels over minefields? This is where the way the player interact affect the emergent narrative.

Juul [2005, 120] also discuss this when writing about the fictional world present in most games:

> "Rules and fiction compete for the player's attention. They are complementary but not symmetrical. ... rules are designed to be objective, obligatory, unambiguous and generally above discussion. With fiction in games, we find the opposite to be true: a strong part of the attraction of fiction in games is that it is highly subjective, optional, ambiguous and generally evocative and subject to discussion. Rules and fiction are attractive for opposite reasons."

[7] http://www.valvesoftware.com/games/hl2.html

3.3.1 Environmental storytelling

Environmental storytelling is when you let the story be embedded in the environment that the player experience through exploring and interacting with the game world [Jenkins, 2004, Carson, 2000]. Carson [2000] describes several methods a designer can use to place story elements throughout the environment. One of the most successful way of designing environmental storytelling is by *cause and effect*. This is explained as staged areas where the player have to come to their own conclusions about what happened or what is about to happen. *Cause and effect* can also be used to describe change over time, for instance a familiar place that has been completely altered in a way that hints that something has happened.

4 Methodologies

The methods presented are in chronological order, following the planning and implementation section 5 and 6.

4.1 Prestudy

The prestudy consists of several main areas. Finding relevant scientific articles and literature covering aspects that can be of use while conceptualizing, analyzing and designing ideas. Studying competitors on the market to see what has been done, this to gain ideas and get a feeling for what works and what has been done before. The last area is to study technical manuals, API:s and similar in order to be able to implement for the iOS. Also searching for tools that could be used to aid in the development of the concepts.

4.1.1 Literature and article studies

Finding and reading articles and books covering the matter is an essential part of getting started properly. Searching for scientific articles in article databases, among special interest groups and studying books covering the area to find relevant studies, theories and conventions.

4.1.2 Studying existing work

Finding and testing existing similar products that can be used to inspire ideas[Fullerton, 2008, p156] and be used for examples and analysis. Products were found through systematically browsing the Internet, searching through known producers of children's entertainment, searching through application stores for tablets/smart phones and from various recommendations from stakeholders.

4.1.3 Technical research

This consists of studying the manuals and API needed to develop all from prototypes to potentially finished products.

4.2 Brainstorming

Brainstorming techniques are structured methods used to gather ideas through a formalized system of idea generation. Fullerton [2008, p150–152] Provides a few points for best practices that sum up a brainstorming session:

- **State a challenge**

  To be clear on the goal of what is to be achieved from the session and to narrow it down.

- **No criticism**

  Every idea is a good idea, save the sorting of bad ideas for later.

- **Vary the method**

  Good not to solely depend on one method for idea generation.

- **Playful environment**

  To inspire participants.

- **Put it on the wall**

  Forcing the participants to visualize their ideas, helping the other participants to embrace the idea and maybe get one of their own based on it.

- **Go for lots of ideas**

  Quantity first, quality later.

- **Don't go too long**

  Working the idea brain on full speed is tiresome, you shouldn't keep going more than 60 minutes.

4.3 Iterative methods

Iterative methods refer to when you work in a repeating cycle, fixing problems that was detected in previous iterations and/or adding features for each loop. [Sharp et al., 2007] They can also help structure projects to avoid dependency deadlock and to measure progress. [Gold, 2004]

4.3.1 Iterative design

Fullerton [2008] describe an iterative process for when designing games. First you generate ideas and after that you formalize them in order to be able to test them. Evaluate the results from the testing and if you have problems with your design then you start from step one. If you have no problems you can exit the loop.

4.3.2 Scrum

Scrum is an iterative methodology for project management, very commonly used when for software projects [Schwaber and Sutherland, 2011].

Scrum employs an iterative, incremental approach to optimize predictability and control risk. Scrum is based on the empirical model of process control, which basically means that the process is going to be inspected regularly. Each of these inspections in Scrum have three main points, called legs. The first leg is called *transparency*, the second *inspection* and the last *adaption*. In their Scrum guide, Schwaber and Sutherland [2011] describes them in the following way:

Transparency “Transparency ensures that aspects of the process that affect the outcome must be visible to those managing the outcomes. Not only must these aspects be transparent, but also what is being seen must be known. That is, when someone inspecting a process believes that something is done; it must be equivalent to their definition of done.”

Inspection “The various aspects of the process must be inspected frequently enough so that unacceptable variances in the process can be detected. The frequency of inspection has to take into consideration that all processes are changed by the act of inspection. A conundrum occurs when the required frequency of inspection exceeds the tolerance to inspection of the process. Fortunately, this doesn’t seem to be true of software development. The other factor is the skill and diligence of the people inspecting the work results.”

Adaption “If the inspector determines from the inspection that one or more aspects of the process are outside acceptable limits, and that the resulting product will be unacceptable, the inspector must adjust the process or the material being processed. The adjustment must be made as quickly as possible to minimize further deviation.”

A Scrum team consists of three roles. The *Scrum master* is responsible for and make sure that the process is understood and followed. The *Product owner* whose role it is to maximize the value of the work performed by the team. Last but not least, there have to be someone doing the work, and that’s the *Team*. In the case of software development, the team consists of developers with the skill to make sure that the*Product owner*’s requirements are turned into a releasable product.

The heart of the SCRUM method is the sprint. Within a sprint a usable product increment is developed and immediately followed by another sprint until the product is completely finished. During a sprint the following rules apply:

- No changes are made that would affect the sprint goal.
- Development team composition remains constant.
- Quality goals do not decrease.
- Scope may be clarified and re-negotiated between the product owner and development team as more is learned.

4.4 Digital Prototypes

When coding prototypes you try to build models of core elements you have questions about.[Fullerton, 2008]. Eric Todd’s (Senior development director on Spore[8]) four types of digital prototypes are described by Fullerton, mentioning they consist of only basic functionality, also underlining that they are not at all finished games. These are *Game mechanics*, *Aesthetics*, *Kinesthetics* and *Technology*.

Game mechanics Game mechanics are discreet features of the formal aspects of the game. Some of these can be tested using physical prototypes, but some of the questions you ask can be hard to answer using physical prototypes. For instance, a mechanic that will allow you to bend time in unusual ways. One more concrete example from the book is allowing the player to se into the future when playing a game of pool and how that affected gameplay.

[8] http://www.spore.com/ftl

Aesthetics In their definition, aesthetics are the visual and aural dramatic elements of the game. For instance testing how the animation will work with a combat system.

Kinesthetics The only of the four that absolutely need to be prototyped digitally, this is how the game “feels”, for instance how well the control maps onto the controlled car’s behavior.

Technology This is creating models to test the physical feasibility of a game. This could be graphics, AI or the physics.

4.5 Analysis

There exists several analytical tools created to help to try understand games and their components, differences, problems and what make them good/bad. Depending on what you want to achieve you need to choose the right tool for the right job.

4.5.1 Narrative component framework

Bizzocchi [2006] provides a “framework of limited narrative parameters” that can be used when trying to support a classic narrative with gameplay. The framework consists of *Storyworld/fictional world*, *character*, *emotion*, *narrative interface* and *micro-narrative*.

Storyworld/Fictional world The storyworld is the world where the narrative/game takes place. Juul [2005, 122–133] mention that most games have a fictional world and that there are different types of worlds that games usually present. Juul goes on and describes that there are five types of games:

Abstract games does not represent any narrative or anything else other than the game itself. Tetris is a good example of this, where the pieces represent nothing else but the pieces needed to get further in the game by following the rules.

Iconic games are games where the individual parts of a game start to map to recognizable real world objects and therefore have an iconic meaning. Contrary to an *abstract game*, you can make some conclusion to what the parts might mean and their relation drawn from the real world. In a deck of cards you first have the numbers, but then you have the jack, queen and the king which can be mapped to our world and the relation between them as parts of a game could be presumed.

Incoherent world games is a game with a fictional world but the game contradict itself or some parts of the game cannot be explained by the fictional world. For instance, what in the fictional world could explain that when fighting a war in Call of Duty 4: Modern Warfare[9] you spawn at the end of a map together with your troop, hear a slight music jingle and after a brief countdown you start to fight with the other side?

Coherent world games have fictional worlds where nothing prevents from imagining them in detail. According to Juul, most adventure games fall into this category.

Staged games are a special case of an abstract/iconic game is played within a more elaborated world. An example of this could be in the first Police Quest[10] were the main character Sonny Bonds have to win a poker game (that the player have to play) in order to proceed and meet Jessie Bains.

[9] http://www.callofduty.com/codmw
[10] http://sarien.net/policequest

Character Bizzocchi goes on and focus on the term *character* from J. Hillis Millers's model for understanding narrative. Describing that the characters (protagonist, villains, NPC:s etc.) are enacting the game and story progress. [Salen and Zimmerman, 2004, 380] takes J. Hillis Miller's theory of narrative and interprets it a bit more general:

> "**Character:** A narrative is not merely a series of event, but a personification of events through a medium such as language. Miller doesn't mean character in the usual sense of fictional persona, but rather the process by which "characters is created out of signs." This component references narratives as not just events that take place in the world, but as represented events, events that occur via systems of representation" Salen and Zimmerman

Emotion Perron [2005] presents three types of emotions being elicited from players playing games, these are theories that are based on E.S Tan's theories regarding emotion in films [Tan, 1996].

Fiction emotion is when the player, through imagination, visits the fictional world portrayed by the narrative. This is not the same as if the player should actually enter the fictional world, but what Perron call *witness emotion* were the player take on an invisible observer role and observe the protagonists and experience with them. Bizzocchi [2006] map this emotion to *imaginative immersion*.

Artifact emotion is when the player is starting to become aware of artistry and manipulation. This is when the object of emotion no longer is the fictional world but the product itself. When speaking about computer games this could be when the player becomes astonished by realistic real-time rendered lighting or how the player avatar's moves maps perfectly to how you manipulate the controller.

Gameplay emotion is according to Perron the main part of the experience when playing games, even with story-driven games. These are the emotions that arise as a reaction to our actions in the game. A study by Lazzaro [2004] showed that several emotions can be elicited by gameplay and that they are separated from those elicited by the narrative. This could be playing just to beat the game (fiero) to see how good you can get or social factors (meeting friends, schadenfreude) etc.

Narrative interface is when you incorporate the narrative into the interface design. This could be the layout, graphics or even characters incorporated into the interface.

Micro-narrative Considering the narrative arc in a narrative, it's possible to identify the characters and their way to a resolution, but with gameplay there's a loss of authorial control when regarding the narrative arc and the details of the progress. Bizzocchi writes that:

> "The changing context for play is constantly set up with fresh complications and challenges, the gameplay itself is an instantiation of the narrative development phase, and intermediate successes and failures act as interim resolutions and localized climaxes."

For this he uses the term *micro-narrative*, as used by Jenkins [2004] when he describes narratives that enter games in local contexts. These *micro-narratives* are a smaller localized narrative arcs embedded in the "main plot", or main narrative arc if you will. Jenkins means that even though some games don't have a large scale narrative or plot, they may still rely on *micro-narratives* in order to create part of the experience for the player.

5 Planning

The initial week based work schedule made for planning purpose

Week	Description
1–4	Initial research, both theoretical and technical. This includes searching for relevant articles and books together with studying the manual for developing for iPad/iPhone. Also search for any tools that can be used in the technical development of the prototypes. Take care of technical dependencies and start working on getting a developers license for iOS. Set up blog structure and start writing work diary.
5–6	Start initial brainstorming and try to come up with ideas that can be used as proof of concept. Add ideas as blog post for reference material later.
6–7	Evaluate the ideas and start developing concept prototypes. Enter first sprint.
8	Re-evaluate concept prototypes and make sure everything needed for the first sprint is complete.
9	Present concept prototypes for stakeholders and supervisor, then start first iteration/re-design. Prepare for second sprint.
10–11	Implement changes and new designs.
12–13	Finish sprint two, analyze result from prototypes and prepare for a third sprint.
14	Start sprint 3 and start analysis. Plan for more sprints if required.
15–16	Start to properly working on the report, finish additional sprints if required. More theoretical work, finding proper articles and references from books. Finish the analysis and gather the results.
17–20	Writing on the report and work on opposition.

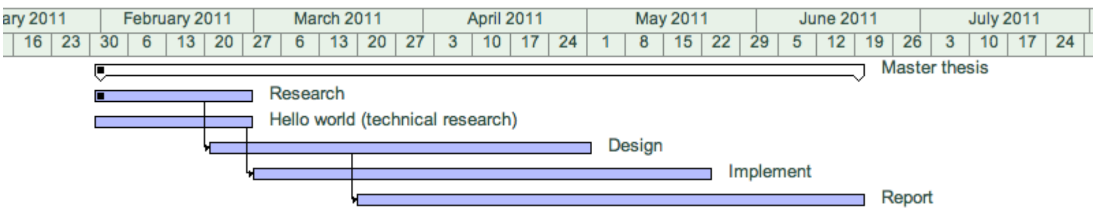


Figure 3: Rough gantt diagram from early planning

6 Realisation

This section will be presented in incremental working order, following the planned schedule presented in section 5. This to describe the working flow and iterations done on during implementation in a natural way.

6.1 Pre-study

The first four weeks were allocated for pre-study and divided into three parts. Firstly finding relevant scientific articles and literature covering aspects that can be of use while conceptualizing, analyzing and designing ideas. This was meant to provide with theoretical material on how to design good games for children. Methods, frameworks, analysis tools or even just basic guidelines that could be used as a starting point.

As mentioned in section 4.1, during these weeks, besides reading through literature and browsing databases, testing a lot of children games were a part of the initial plan, with the hope that this would also bring an idea of what has been done and exists as popular products on the market. This was considered as the second part.

The third and last part was to study technical manuals, API:s and similar in order to be able to implement for the iOS. Also searching for tools that could be used to aid in the development of the concepts. This would just provide with basic tools for implementing working prototypes.

Chalmers library provide access to several databases that where used. Also search engines and open databases were used. Initially the course books used throughout the masters program were studied and from there finding new books through references in course books and articles. Main sources for studying literature and articles were *Association for Computing Machinery*'s (ACM) digital library[11], *Digital Games Research Association* (DiGRA)[12] and *Google Scholar*[13]. ACM and Google Scholar provide a wide range of articles, where Google is more general and ACM lean toward computer science related articles. DiGRA provide a more generous number towards games and entertainment. A large amount of the allocated time was spent on finding relevant articles, starting with general search key words moving towards more narrow searches. After filtering out the relevant articles, new articles were selected from the reference list and searched for using the databases. This procedure was repeated a number of times until something applicable was found. In this case finding the framework discussed by Bizzocchi [2006] that is described in section 4.5.1. This in turn lead to articles by Lazzaro [2004] and Perron [2005] used to describe the framework.

Courses involving game design and theory at Chalmers use two books as course literature. Game Design Workshop by Fullerton [2008] covers game design and methods for this and was used as source for most of the game design related methods in section 4. Rules of Play by Salen and Zimmerman [2004] works as a general theoretical framework for games, and together with Half-Real by Juul [2005] they were used for describing the more theoretical aspects, such as the previously mentioned article by Bizzocchi and defining what makes a game.

Through this time a number of games and interactive books meant for children (and sometimes also adults) was played and explored to look for inspiration and what techniques were used. A large number of these came directy from Disney[14] and Appstore[15]. This was done as a study of what the competition was for Appmazed and as a source of what has been done. Some of the most relevant are mentioned in section 2 and was used as inspiration and as existing proof of concepts later when describing ideas for Appmazed.

A technical research conducted in learning the software developing tools needed to implement future gamelets. This included studying briefly on how to set up projects in

[11] http://portal.acm.org/
[12] http://scholar.google.se/
[13] http://www.digra.org/
[14] http://www.disney.se/disney-spel/
[15] http://www.apple.com/se/mac/app-store/

XCode[16], learning to code using Objective C which is the language used when developing for iOS (that is running on iPad and iPhone) and last but not least finding and learning a technical framework for this environment that could be used to create gamelets. After some searching online, the framework Cocos2D[17] was chosen. The choice was partially due to a co-worker at Appmazed that was skilled in using the tool and that it had good support for using Box2D, a physics engine that was known to the author.

It turned out that finding relevant articles was hard, which was to some extent disappointing since this was thought to be the greatest source for how to move on. Instead realizing what went on in existing products was the main source for inspiration on how to move on. When testing the very amusing interactive story about the giant Call in *What's bothering Carl* described in section 2 it started to form the idea that would be the main approach for the following weeks.

6.1.1 Identifying Gamelets

Lots of interactive books both from old and new technologies, use interactive hot spots embedded into their products. This could be in the title screen or instead of pictures in the text. These interactive hot spots will from here on be referred to as *gamelets*. Gamelets could consist of a game, a puzzle or an interaction. In this thesis gamelets will be limited to replace illustrations, usually where they would be in between the text. Further possibilities exist as having them as entry points, accessing what would also be a gamelet in a sense, but in a different context than just as an illustration replacement.

During supervision, a long discussion about what these should be called took place. Among the suggestions were *IxLets* and even existing terms as *Widgets* before landing on *gamelets*. This discussion actually lasted over several sessions and into the final weeks of the project.

The first reason for this definition was to make it easier to discuss this concept without having to describe it all the time. The second and main reason was that it also grew into something really useful for the progress, used to narrow the scope of this thesis and provide a focus point for when trying to find ways of designing diegetic elements.

Using gamelets as parts of interactive books – where the main point is to tell a story – there was a need to investigate how they can be designed to support the narrative. In section 4.5.1 when describing fiction emotion and the imaginative immersion that books try to invoke, this is also a part of what Aarseth [1997] mention when referring to a reader as a *voyeur*. This creates a natural conflict with games since, as mentioned in section 3.3, tries to create an active immersion, where the player need to participate in order to reveal the outcome.

6.2 Conceptualizing

Appmazed thought that focusing on gamelets as a source of adding to the existing material was something that could be of use in their product, but mostly from a technical aspect since they can be implemented individually and in short time. Also, they had potential to be used in order to affect the diegesis. It was decided that this would be explored further. Both how gamelets could work as their own entity and how they could be used as diegetic elements to support or maybe even change the narrative. The goal was to end up with ideas that could be implemented in a later phase of the project and be put in Appmazed's product.

[16]http://developer.apple.com/xcode/
[17]http://www.cocos2d-iphone.org/

Figure 4: When clicked, Carl makes a scary face and frighten the gnomes enough for them to lose their grip. The interaction with the gamelet have a diegetic property that *is* coherent with the main narrative

6.2.1 Gamelet communication

Together looking at how the experience of existing material could be enhanced came the idea of exploring something new. Instead of letting them be their own entities only related by theme, they could also be used to affect each other.

One of the first and most basic ways of letting the gamelets affect each other that came to mind was through *sharing*. Considered general enough, this was used as theme for a brainstorming session hoping it would spawn more ideas of how gamelets could affect each other. It ended with some interesting results described with examples here:

- **Equipment**

  Lets say that you play Tetris, and when achieving a tetris you get access to the 'I' piece in another gamelet where you can play Lemmings and can now use the 'I' piece to build a bridge.

- **Gestures**

  By successfully using a gesture in one gamelet you can use it in another. In Cut the rope you could successfully cut two ropes in one swipe, but manage to make a looping gesture in between ropes. This could unlock the loop gesture with a character in another gamelet that lets you play a Fable-like game.

- **Temporal self triggered events**

  In one gamelet, you fire a flare gun after 30 seconds and when in another gamelet you can se a flare in the background after 30 seconds.

- **Keys**

Figure 5: A fish jumps over the boat when clicked. That diegetic property *isn't* coherent with the main narrative

By not letting the player get further in a gamelet without first having her do something in another is indirectly sharing, but still distinct enough to get it's own mention.

- **Positioning**

  By interacting with a position in one gamelet, something interactable will happen in another at the same spot.

When the results from the brainstorm was in, sorted and pondered, another idea was spawned. It could result in interesting limitations if narrowing down the way that gamelets are allowed to share elements. As in different ways of communicating. This is of course not a complete list, more could be added or combined with others.

- **Achievements**

  The first thing that was discussed was the concept of achievements, in the same way they are used by many gaming platforms/communities like Kongregate[18] and Xbox Live[19]. With achievements, the gamelets don't directly speak to each other and are still closed systems, but you still collect points and it's the total that counts. Some people look for the easy achievements and play those games just because they want the gamerscore, not really playing the game but more the metagame of collecting points.

- **The sequel**

  Taking things with you into the sequel by letting you import your character. Even though it's an active choice from the player's side, this could let you keep old stats,

[18] http://www.kongregate.com/
[19] www.xbox.com/live

Figure 6: Results from a brainstorming session trying to sort out the good ideas

equipment, followers etc. It could also affect how the world looks and the story (as in Mass Effect) and start the sequel on a different basis. This is done chronologically, or linearly if you will. It means that you can affect the sequel but not the other way around.

- **Recursive**

  After a finished game session the game saves certain data and uses them when playing new game sessions. This is not something to be confused with high-scores, but more about elements that can affect the gameplay. In Epic Dungeon[20] you can stumble upon a tombstone where you (or any other player on the same console) died with a character. Looting it will give you one object that character was carrying, hopefully it's something usable. In Nethack[21] you can face one of your previous character's ghost, resulting in combat.

- **Ping-pong**

  Is when N games affect content in a certain order back and forth. Say that two games are built like this, it might be implemented so that you need object 1 in game 2, but that object can only be unlocked in game 1. In turn, further on you will need object 2 in game 1, that can only be unlocked in game 2 if you have object 1.

- **Crisscross**

  N games affect each other arbitrary all over, maybe making some tasks easier in others or maybe difficult. It could affect story or theme in another. Basically any gameplay element could be affected in any order back and forth. Finding the diamond in the point-and-click adventure game might give you a diamond shaped

[20] Now known as Cursed Loot http://www.eyehookgames.com/cursedloot/

[21] http://www.nethack.org/

object in the dropdown puzzle, helping you to solve tricky situations. Maybe two individual elements from two different games could be combined and unlock something in a third game.

6.2.2 Gamelet communication proof of concept

In order to show how the sharing might work, some gamelet ideas was conceptualized to be used as proof of concept. Brainstorming was once again conducted, this time using I/O-lottery, combining random elements to share between two randomly selected gamelets. A few chosen ones were then selected and different changes and possibilities discussed.

Ball Labyrinth and Detection Towers The ball labyrinth is derived from a tilting wooden ball labyrinth, where you tilt the surface try to navigate a metal ball towards the goal avoiding holes on the way.

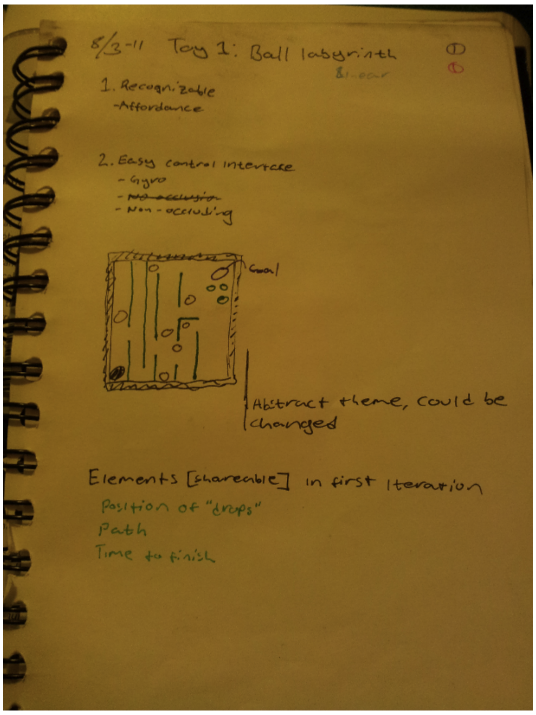


Figure 7: Picture of ball labyrinth sketch

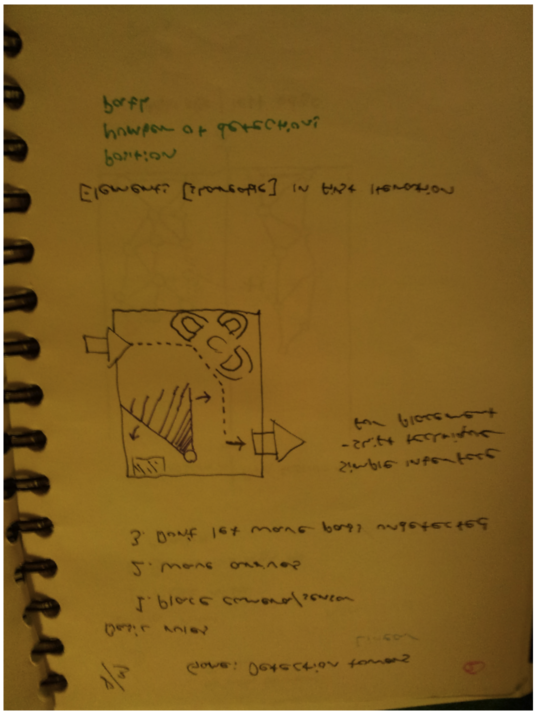


Figure 8: Picture of Detection towers

Figure 9: Text describing relationship briefly

Detection towers is a simple tower defense game where the player have to place sensors to detect waves that will traverse between two points of the map.

The initial idea was for the two to share the same amount of screen space and let the sensors appear in the same spot on the screen as where the ball drops. This would require the player to fail in Ball Labyrinth in order to succeed in Detection Towers. If you on the other hand let the gamelets share ball path and start/goal positions, so that the player have to avoid being detected this would encourage for completing both games in the gamelets.

However, this examples make one gamelet depend completely on another in order to be able to complete the game. This could be adjusted to make them to not affect crucial elements, but maybe more aiding elements.

Reminder notice fellow The reminder notice fellow gather information about all the other gamelets, and keeps telling the player about it. It could be hinting about the results from a previous game session from another gamelet or by just reminding about a gamelet that hasn't been used that much.

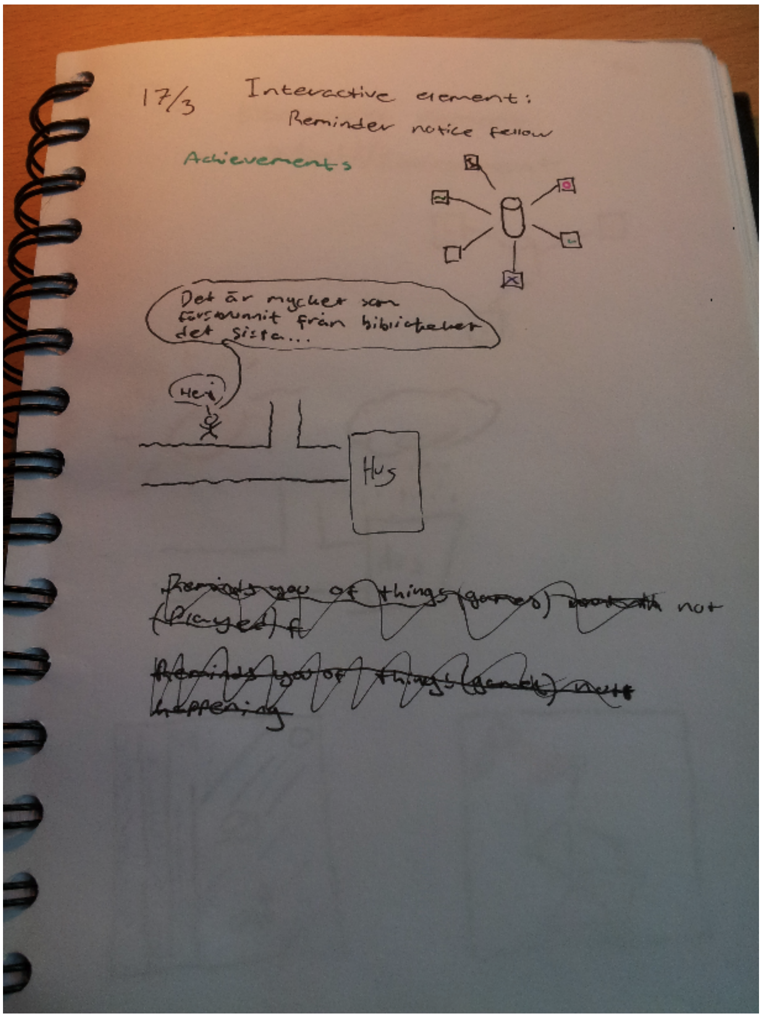


Figure 10: Reminder Notice Guy sketch

This would work as an alternative example of *achievements* communication. One that keeps track of all the other gamelet's results, states and other arbitrary data that could be relevant of telling the player.

Mini billiards This is a simple gamelet using *recursion*. The balls you can use are the ones you managed to get into the holes in previous the previous session. By flicking the balls one by one or by trying to create chain reactions of them hitting each other the player will try to get them into their corresponding holes, and then you carry those balls with you to the other session. This is a weak concept, since it is a borderline case of just being a game with levels.

Environment The idea here is that there is an overlaying environment that will affect the other gamelets. If it snows, the surface of *ball labyrinth* could become slippery, or maybe the heat could affect the *Graph Tower*.

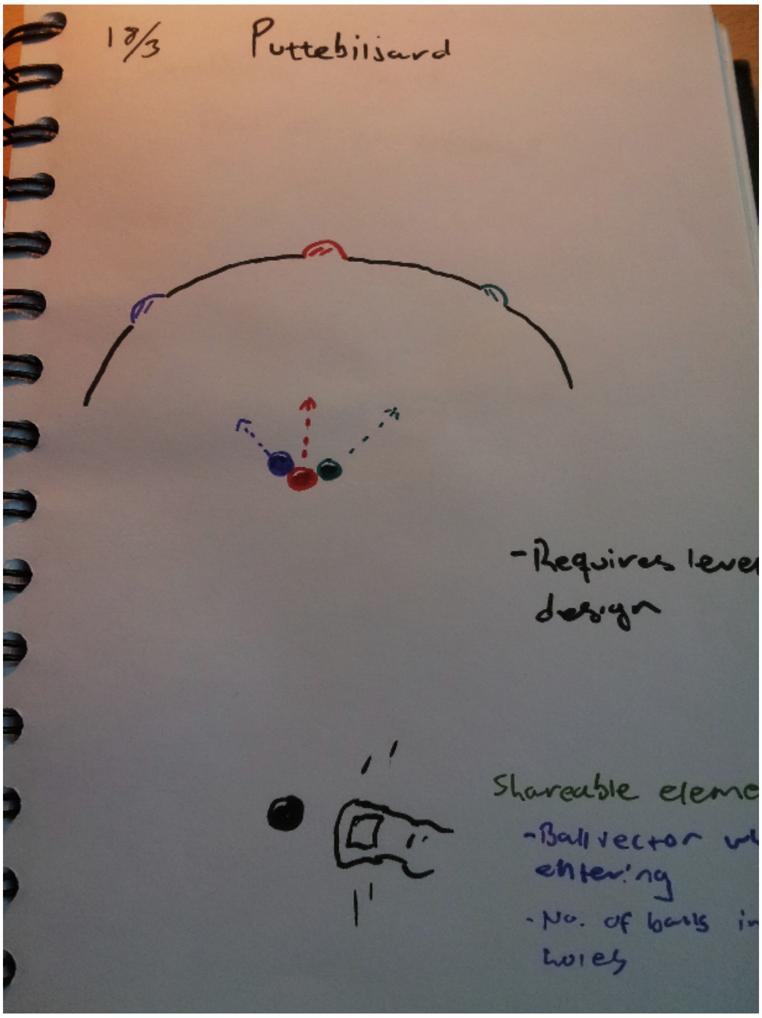


Figure 11: Mini billiards sketch

This could be considered some sort of reverse *achievement*, were one gamelet share with everybody else and affect how the rules turn out in them. This is also an examples to show that the list of communication types is not complete.

Graph Tower Graph tower was concepted as an example of a gamelet with recursive communication and with temporal affection. The basic rules are that you try to build a tower as high as you can using nodes in a physics based world. At times, parts of the tower, for one reason or another, will jump from your tower out to the right end of the screen. The details on the rules for why this happens was left as an abstract. The game ends when the tower topples over.

In the following game session, at the same relative moment in time as nodes jumped from your tower to the right in your previous session they will appear from the left side of the screen using the same direction vector and other relevant parameters as in the previous session. The node will attach itself to the current tower being built, changing how it's affected by gravity.

In the end of the conceptualizing procedure the result was a set of general ways of how gamelets could communicate, basically meaning putting restrictions on how they can share information. Also two ideas trying to show how *Recursion* and *The Sequel* could work. These ideas were presented to Appmazed together with the gamelet communication to make it more clear on how they could work.

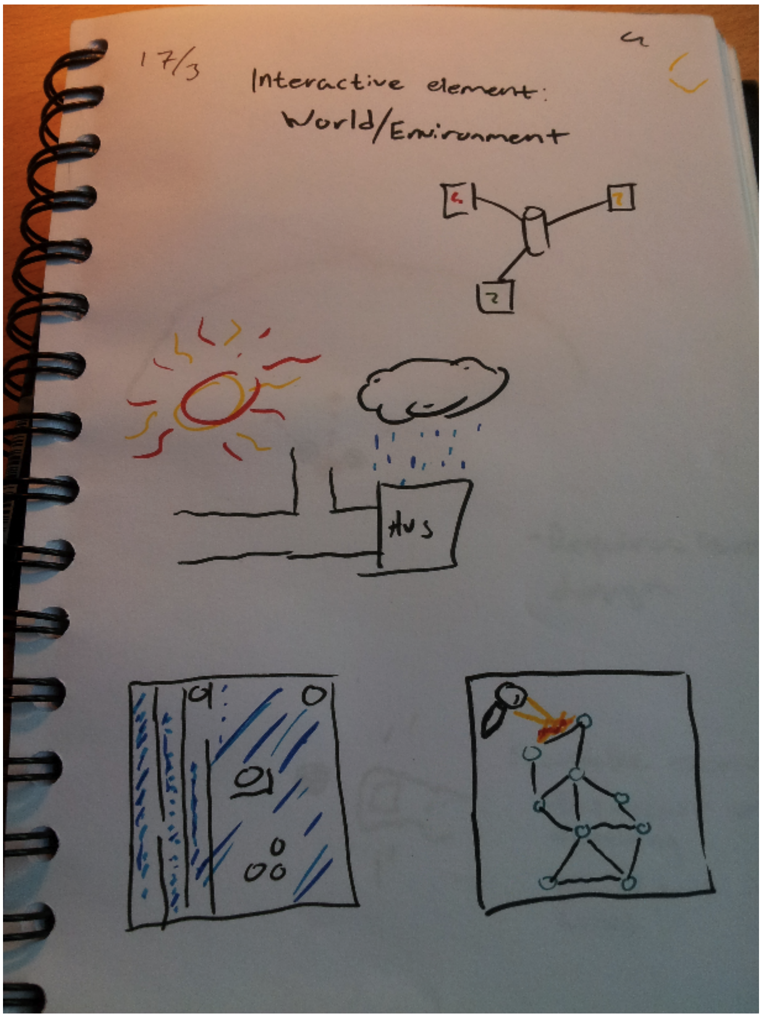


Figure 12: Environment sketch

6.3 Gamelet prototype design

The concepts for gamelet communication was deemed not feasible for implementation for two reasons. The implementation time could potentially be to long, and they are hard to apply to the given narrative of *JerryMaya*. Since Appmazed seemed quick to change their minds with regards to what should and should not be in their product, a couple of previously tossed concepts was re-worked slightly to fit into the the interactive book. The concepts were simple and to some extent trivial, but enough to work as prototypes for both testing some communication between gamelets and how they affect the diegesis. At the same time they could fit into Appmazed's demo product that they had started implementing. This seemed as a good compromise at the time, so it was decided.

6.3.1 Main book context

The first step was to design a digital book reading context in order to view the prototypes in a proper environmental context. Combining the idea of both *What's bothering Carl?* and *ButaVX: Justice fighter* explained in 6.1, both a book-like environment and a top-down more rpg-like environment was to be used. The RPG environment was already under construction by Appmazed, but the book environment was not.

The book context design was just a couple of simple design choices made to create a simple context environment for the gamelets. Using black text on white background in

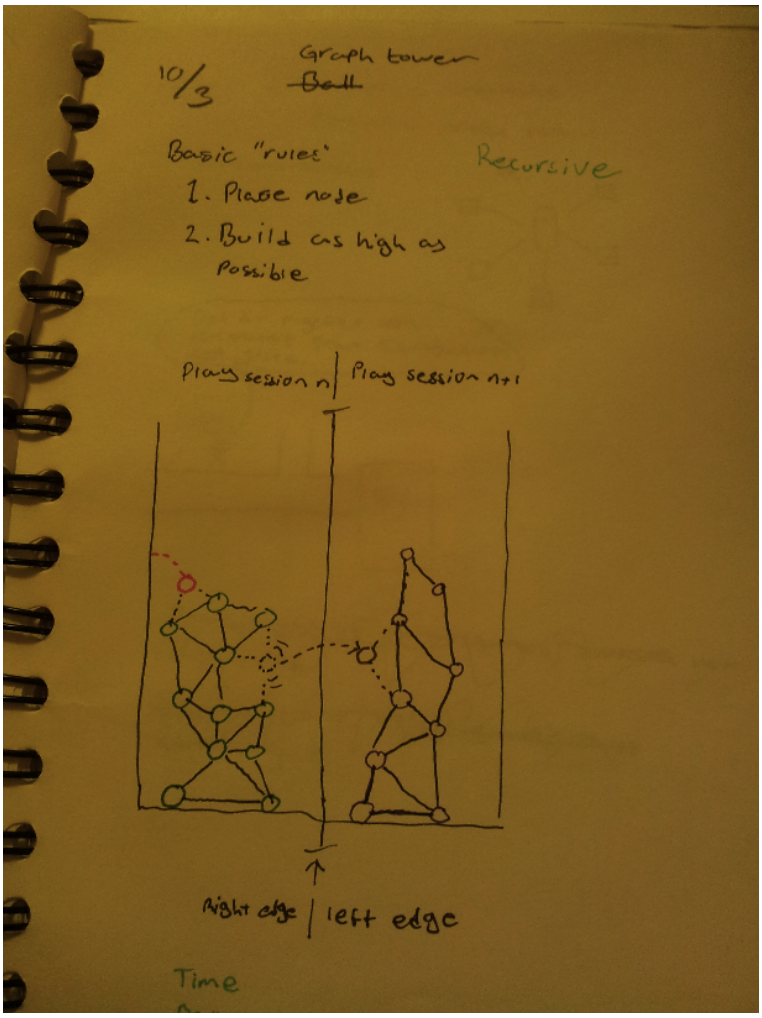


Figure 13: Graph Tower sketch

order to emulate a book with black text on white paper. A reader should be able to flip back and forward between pages and it should visually look like it's being turned. The book context should support images and gamelets.

Considering the game definitions covered in section 3.2, one can argue when puzzles become games or if games really have to have opposing players to be considered as being just games. But the important distinction for this thesis is the line drawn between stories, toys and puzzles/games since the focus lies on how to integrate games into children's books that provide a story.

6.3.2 Game: Banana Toss

The first prototype was designed to diverge from the main narrative of the book but still have diegetic properties in form of micro-narrative. In the *JerryMaya* fictional world there is a harbor on the map provided with the Diamond Mystery book. This has no part of the narrative in the book, but it's still there. It's deliberately created to bring some incoherence to the world, as discussed in section 4.5.1.

The rules are quite simple: A bunch of bananas drop onto a dock, behaving coherent with newtonian physics. You can touch flick the bananas and try to hit a boat that's leaving the dock. Each hit generates some kind arbitrary score and is also saved for high-score purposes.

6.3.3 Game: Banana Catch

Banana catch is actually the second part of banana toss in section 6.3.2 in an attempt to serve as a proof of concept for temporal self triggered events described in section 6.2.1. The idea is that the bananas will fall onto the boat at the same tempo and frequency as the player managed to get a hit in Banana Toss. Catching bananas is done by moving a container around a boat, and arbitrary points are given for each catch.

6.3.4 Puzzle: Jigsaw Puzzle

A simple jigsaw puzzle were an image from main narrative is divided into pieces and rearranged, this as an attempt to create an abstract type of gamelet as described in section 4.5.1 [Juul, 2005]. The reader/player then have to place them in their proper position in order to solve the puzzle. This particular puzzle will also change mode and enter it's own context instead of being directly accessed with interactions made in the book context. This design choice was made in contrast to Apple Poke described in section 6.3.5, just to create variation.

6.3.5 Interaction: Apple Poke

Apple poke is an interaction gamelet designed to try to have diegetic properties that follow the main narrative. By touching an apple it will start to roll down a drainpipe and roll down onto the ground. In the book narrative, the thief steals the diamonds by placing them in apples and roll them down the drain pipe from his office. This is meant to support the narrative and tell a part of it through the interaction.

6.3.6 Interaction: Diamond Pick

Throughout the pages, there will be interactive images of diamonds and when touched, they will disappear and counted. This was designed to provide some achievements to the book and also to have diegetic properties related to the narrative. By letting the reader look for diamonds and picking them, since a diamond thief is a part of the narrative.

6.4 Development

All the prototypes were developed simultaneously with a general requirement applying for all prototypes but still having individual sprint goals. The reasons for this was that it's hard to foresee implementation times, and this way it was thought it would be possible to detect which implementation could be de-prioritized in a better way than creating them in a series. Another reason was to early on be able to test gamelet communication.

With regards to section 4.4 regarding prototypes, there was two main reasons for these prototypes and that was testing the technology to see what could be done and to test narrative capabilities of the gamelets. The latter falls slightly into testing aesthetics, but might as well be in it's own sub-category.

6.4.1 Sprint 1

The implementations should cover basic functionality and be able to work together with the book context. As a general helping tool, a state holding singleton class (from now referred to as *StateHolder*) is implemented to support general communication between gamelets.

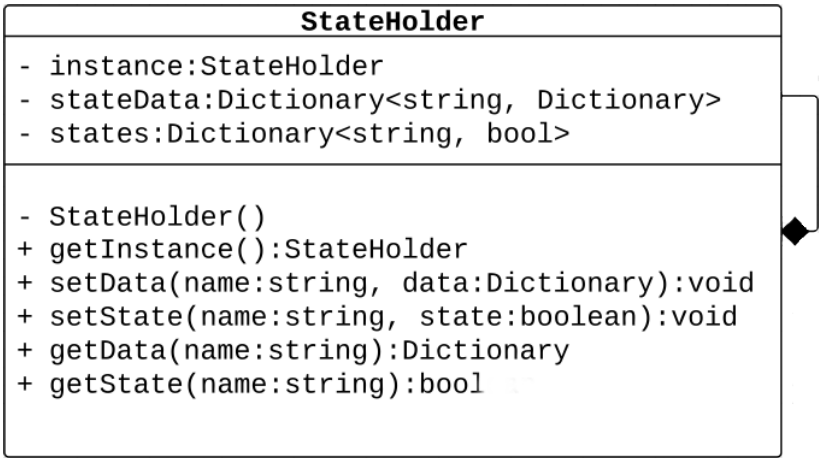


Figure 14: UML of the StateHolder class for sharing data between gamelets

Banana Toss

Starting to get the physics to work properly seemed like a natural entry point and together with making the bananas spawn and drop onto a surface as well as bouncing on each other and off the walls. Implementing a way to grab individual bananas by simple touch, moving them by dragging the finger around the screen and dropping them when finger is lifted from the screen. Since this works together with *Banana Toss* it can save the force vector on collision in the *StateHolder*.

Banana Catch

No interaction possibilities implemented, just using data saved from *Banana Toss* spawns a banana at an arbitrary point and applies an impulse force based on the one saved in *Banana Toss*. This between two pages in the book context so that a banana can be thrown to the edge on one page, and when the page is turned it continues in the same path across the pages. In this first iteration it was done by simply using a new instance of Banana Toss but spawn a banana based on information from the *StateHolder*. Applying a force vector through the physics engine making it fly in the same direction and with the same force as saved from the Banana Toss.

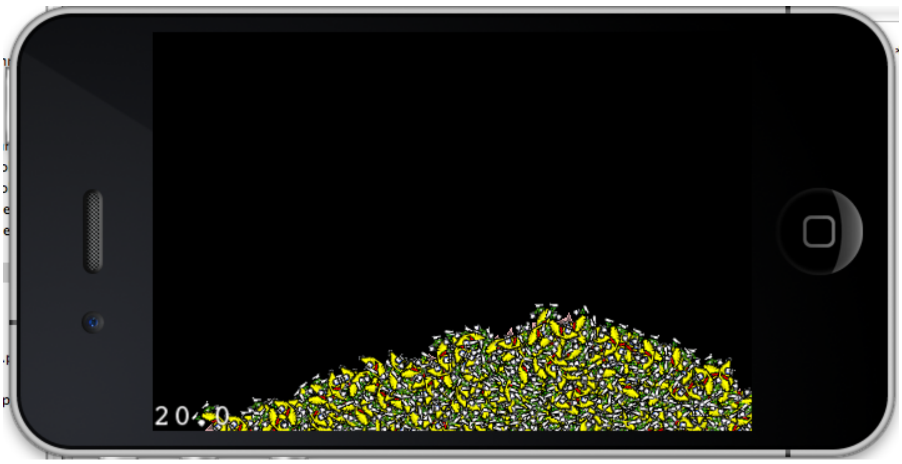


Figure 15: First prototype of Banana Toss

Jigsaw Puzzle
Fades to its own context, this was mainly just to have one gamelet that actually made a context switch. It can load any image resource and divides it into 2×2 puzzle. The resource image must have dimensions that are $n \times n$ and the four pieces are placed inside a canvas container and the locations are hard coded at this point. Each individual piece can be moved around and when placed in [roughly] the order it will mark as solved and fade back into the book context.

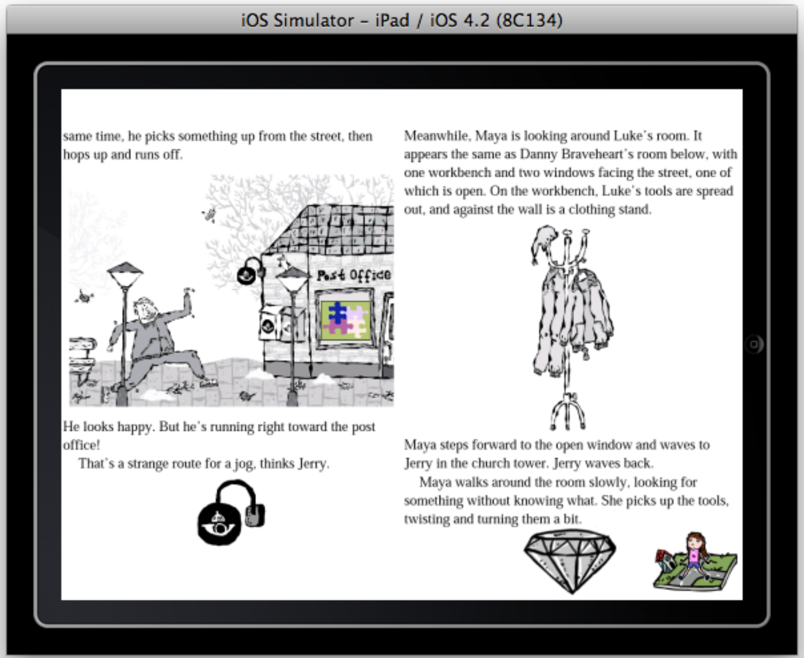


Figure 16: The jigsaw puzzle icon in the post office windows leads to the simple jigsaw puzzle

Apple Poke
A physics world was created manually by creating rigid body objects in box2D, hard coding their coordinates and sizes. They were placed on top of a plain illustration, allowing a dynamic body shaped as a circle to fall and roll on an edge and eventually off that same edge. Graphics added from JerryMaya, just as a background image to base the physical world on something.

Diamond Pick
Implemented as two images. If clicked, image two replace image one and records this in the *StateHolder*. If the reader now flips the page and lands on the same page again, it will still show the second image and nothing will happen if clicked. State will be held until a new book context instance is instantiated.

Book Context
Each page have two text objects that fill their half of the screen with placeholder text, using the built in text class in Cocos2D was selected at this time since it was close at

hand for just trying to get it working. Clicking anywhere in the white space go to the next page, and after the last page it loops back to the first.

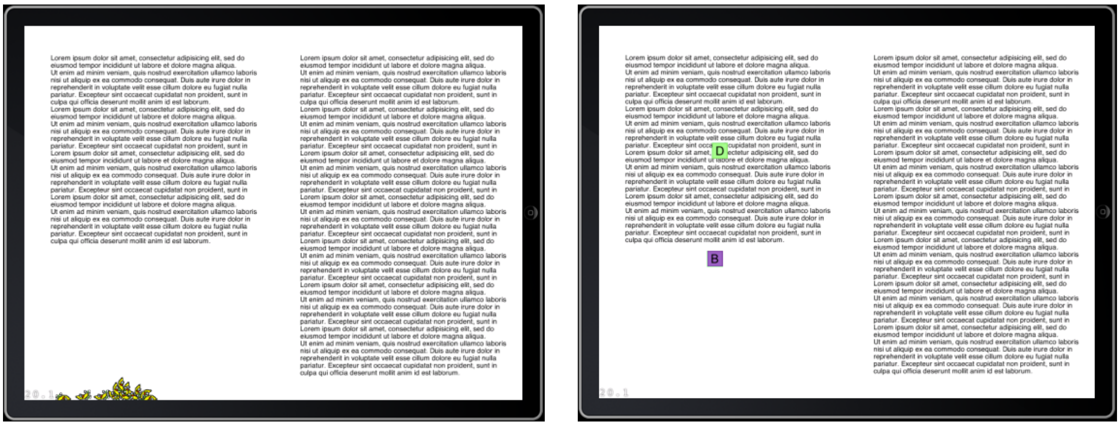


Figure 17: First early working book context with two pages. To the left the Banana Toss prototype have been inserted. On the right page D leads to the puzzle mode and B is a placeholder image for Diamond Pick

6.4.2 Sprint 2

Trough this sprint, making the gamelets function more fully with working interactions was prioritized first, this in order to get the basic functionality working properly. But most implementations are closely tied to the actual illustrations which made them vital in order to proceed with the coding and not only to provide with a *JerryMaya* theme.

Banana Toss

With Banana Toss, the implementation was built closely around the actual graphics, so here they were needed in order to finish implementing basic functionality. The graphics added was a background image of the sea with sky and clouds shown in figure 18, and on top of this a wooden jetty was placed to the bottom left. The physics world was adjusted around the jetty so that the bananas could be placed on it. The touch and drag function was changed to release itself, this resulted in a more desirable interactions were the user touch and drag become more of a flicking motion leaving the banana flying through the air depending on the flicking speed.

Banana Catch

No real functionality added in this sprint, this due to spending more time on the Banana Toss and making it work more properly. Since Banana Catch relies much on Banana Toss this was put on hold for the time being.

Jigsaw Puzzle

Implemented into the *StateHolder* so the gamelet detects if the puzzle has already been solved. This makes it not possible to enter the puzzle context when clicked. By this time, customized JerryMaya images created by Appmazed started to take form and was inserted in order to further polish the prototype.

Apple Poke

Tweaked the environment and the graphics to seem more real. Spent a large amount

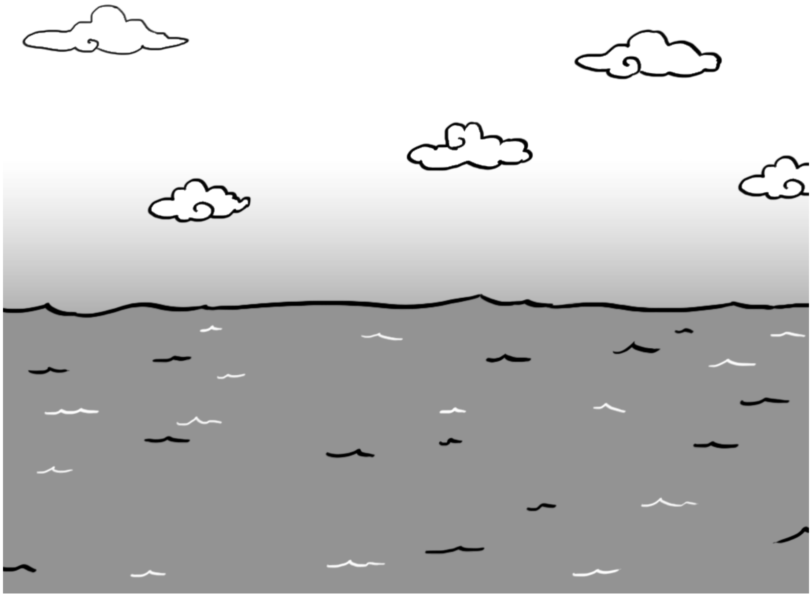


Figure 18: Background used for Banana Toss

of time masking and adding layers in so that the apple would seem to be rolling in the drainpipe. The apple is now inactive until touched, adding a simple interaction to the gamelet.

Diamond Pick

Instead of just switching image, a basic tween of the alpha value is instead implemented so the image fades out to nothing instead of switching image. This was meant to create a more feel of actually taking them, since they disappear. Custom graphics from Appmazed used instead of placeholders, as seen in figure 19.

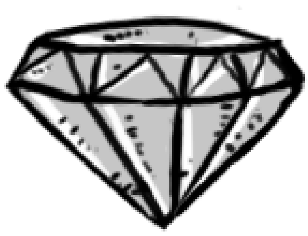


Figure 19: Diamond illustration used for Diamond Pick

Book Context

Added a visual effect that animates a page turning, so it more resembles a traditional book. Instead of having it always going to the next page when clicking, cycling to the first when reaching the end, the pages now can be clicked individually. Clicking the right page moves to the next two pages and clicking the left page moves to the previous two pages.

6.4.3 Sprint 3

In the last sprint, discussions with Appmazed resulted in more visual tweaking in order to get some gamelets more in theme with JerryMaya. General optimizing applied on all gamelets in order to prevent lagging and buggy behavior that come from the prototype style of coding. Within Appmazed the *Banana Toss* grew as the favorite of the gamelets implemented, this lead to it being prioritized when it came to implementing nice to have features.

Banana Toss

Due to the graphics Appmazed had in stock, a container was added instead of the boat. Little difference in the implementation was needed. To the container, a rope was attached and a rope like behavior was implemented, making it respond to bananas hitting the container in any way. The container was masked and divided into layers in order to allow the bananas to appear being inside the container. A hovering seagull was added as an attempt to enhance the harbor feeling, this was added as a last minute change put in from Appmazed's side.

Banana Catch

Since a lot of time was put into finishing *Banana Toss* it lead to that no time was spent further implementing *Banana Catch*. It was hoped that it would be hours later in the project that could be put into finish it, but it never happened.

Jigsaw Puzzle

No more iterations made, since it had it's core functionality and custom graphics it was flagged as finished.

Diamond Pick

No more iterations made, same as with *Jigsaw Puzzle*. The core functionality was up and running with custom graphics so this was also marked as finished at this point.

Apple Poke

Integrated the environment more into the gamelet, having the apple looking like it rolls on the ground and hitting a branch further down the path. Also added restitution making the apple bounce more realistically and adjusted the friction for a bit more realistic physics.

Book Context

The text libraries used created huge lag when rendering text covering two pages the way the book context did. The library obviously was made to display smaller amount of text strings, as in for instance speech bubbles. Instead, full pages from the book was scanned and used instead of using text strings. So for each page, there is an image displayed that was scanned from the book.

7 Result

Since the thesis was very loosely defined by Appmazed, it initially started as exploring the design space for interactive children's books on tablets trying to find new ways of creating interaction and integrating smaller gameplay aspects into the book. This lead to trying to find a framework that could be applied on the prototypes and used as discussion point when it comes to how one could affect the diegesis. This was based on that they were working with existing material, and nothing new would be created or put into the existing narrative.

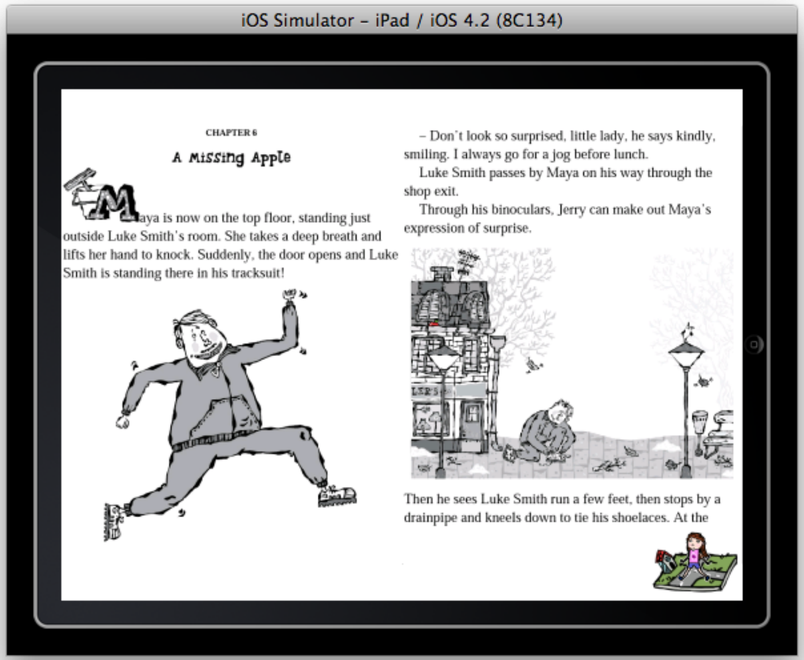


Figure 20: Apple Poke within the book context before interaction

7.1 Gamelets

When conducting studies on existing material, gamelets was defined as a game, an interaction or a puzzle that can be used as a replacement for illustrations used when presenting a narrative with text, thus using gamelets as a way of affecting the diegesis. This was not only to create an umbrella term, but also because it was identified as a powerful tool in order to create interaction and adding to the experience within already existing content. A number of more or less complete gamelet designs was created through brainstorming and rough prototype sketches.

Gamelets was explored even further by looking at how they could interact with each other and not only exist as their own entities. This this resulted in a list of ways gamelets can share information and together with its definition it provides the embryo of a framework that could be developed into a more evolved tool. The framework is supported with examples of how this communication could look in an implementation, and also from existing implementations such as *Achievements* on Kongregate, *Recursive* as in Epic Dungeon, *The sequel* that will allow the player to bring elements from one gamelet to another but not back the other way, *Ping-pong* that adds to *The sequel* so that elements can be shared either way or by letting it loose entirely with *Crisscross*.

With the framework as base, a couple of concept gamelets was designed in order to serve as proof of concept for how this might be implemented into a book context and to serve as early demonstration for Appmazed.

Ball Labyrinth and *Detection Towers* was designed as a way of showing how *The Sequel* kind of communication could work. *Detection Towers* work as a tower defense game but the player have to place the sentries (towers) in another gamelet, namely the *Ball Labyrinth*. This would need some more iterations in order to work properly, since you are totally dependent on *Ball Labyrinth* in order to even try playing *Detection Towers*. In their current form they form one game, not two gamelets.

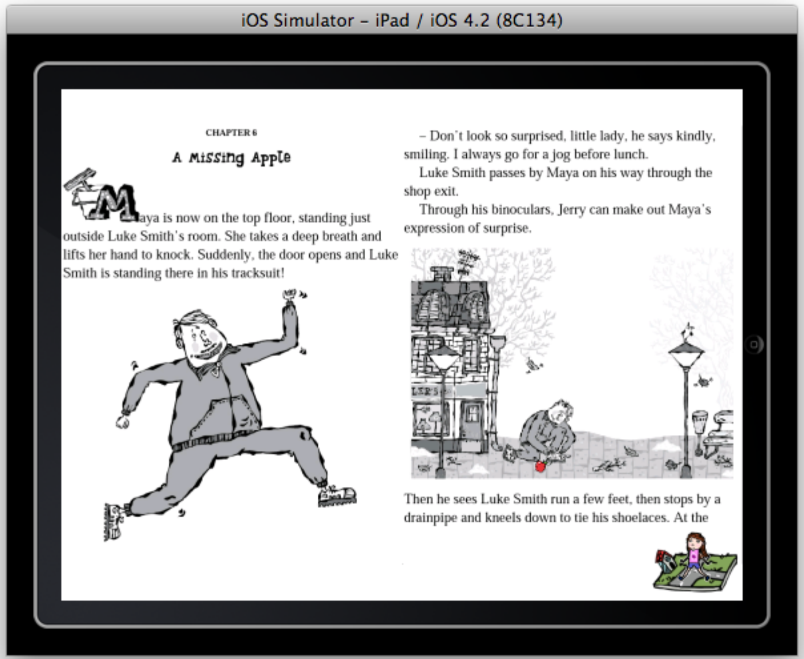


Figure 21: Apple Poke within the book context after interaction. The apple have rolled down, hit the branch and stopped

The *Reminder notice fellow* works as part of an alternative *Achievements* system. It will by itself keep track of other gamelets and keep the player updated on the progress (or non-progress) in any other potential gamelets.

Using *Recursive* communication, *Mini billiards* try to clearly show how you could bring elements to the next sessions. This share the same problem as *Ball Labyrinth* and *Detection Towers* since it borderlines a game with several maps. The *Environment* gamelet in itself don't fit as a replacement for illustrations, but it shows a reversed *Achievements* communication. It could be re-designed easily to show *Crisscross* by letting the environment in some gamelets affect other gamelets in an arbitrary fashion.

Since the narrative was central and already existed in what Appmazed wanted to present, several prototypes were designed in order to see how it could be affect the diegesis and that at the same time could serve as proof of concept for Appmazed when trying to further realize their product. This was done by using the framework presented by Bizzocchi [2006] described in section 4.5.1.

7.2 Implemented gamelets

Through brainstorming and group discussion within Appmazed, four prototypes was selected from a larger set to be implemented digitally. In order to be able to test these concepts properly, a digital book application for use as context was also considered.

A *book context* was implemented in order to have somewhere to place the other gamelets. This ended up being two scanned images placed side by side. The pages could also be turned in order to create the feeling of a book.

Banana Toss is implemented as a game that in the beginning was thought to show *The Sequel* kind of communication, but since *Banana Catch* was lost during sprints it fell short and became it's own entity. Taking place in the harbor, the player have to try to get bananas into a container by flicking them from a jetty into the container which is hanging by rope.

Jigsaw Puzzle is a simple puzzle which is solved by placing pieces of an image in the correct order. This is a simple display of *Recursive* communication, and will only allow the player to play once.

Apple Poke is implemented as a physics simulation of an apple rolling down a drain pipe and hitting a twig on the ground. The simulation starts when the user interacts with the apple and together with graphics from JerryMaya try to hint about who culprit is.

Diamond pick is a simple *Achievement* gamelet that allows the player to throughout the entire book pick and collect diamonds. This gamelet tries to create focus around the diamonds and that the reader is a part of taking them.

These concept prototypes and the book context was implemented for iOS, and optimized for iPad. Looking on how they affect the narrative components in the book, several suggestions on how the narrative components can be affected within this type of digital product were discussed and presented.

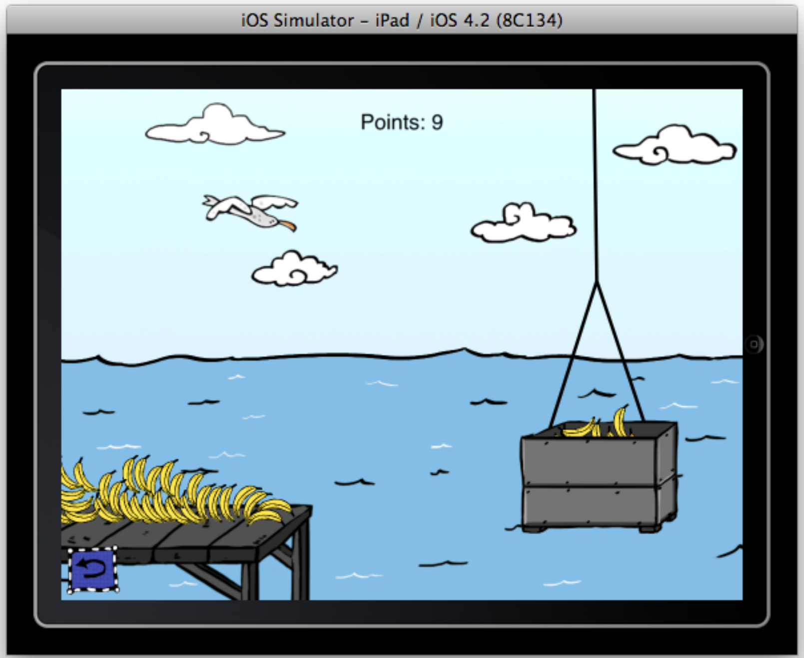


Figure 22: Finished Banana Toss prototype

This also resulted in a simple class (see figure 14 in section 6.4.1)that could be used for general communication purpose between the gamelets in addition to holding states. That class would be used further on with the digital prototypes developed later through the process.

The framework can work as a tool for future designers at Appmazed when trying to design gamelets for use as support in interactive storytelling.

The framework suggested together with the implemented gamelets working as proof of concepts was used together with other parts of Appmazed product to showcase and give some idea to new ways of presenting already existing material. Appmazed showed the prototypes to potential customers, investors and other potential stakeholders in order to show the potential in this kind of products.

8 Discussion

Answering the question on what to do in order to add to a narrative experience using existing material was not easily answered. Gamelets was brought forward as a central part in what to focus on, since trying to answer it generally a too wide question. So much can be done when designing an interactive experience for children, and it is what tablets provide with their powerful hardware and comprehensive API. Consider a book for adults, they are mainly just text. But consider a book for children filled with static illustrations, as in this case with *JerryMaya*. They can at the most provide with *cause and effect* [Carson, 2000] environmental storytelling as described in section 3.3.1, at least in its current form being just illustrations. The difference from that point of view is the potential of what you can do with gamelets that are at least interactable.

8.1 Reflection on gamelets and digesis

As soon as the prototypes was placed within the book context it was noted that the placement can have a huge impact on the reading experience. Placing the *Apple Poke* at an unfortunate place in the book context might hint pre-maturely who the perpetrator is before the text have given any clue what so ever. Another problem with regard to the reading experience is that designing immersion in a gamelet, could lead to readers spending more time with a gamelet than on the main narrative. The prototypes was analyzed using the narrative components framework [Bizzocchi, 2006] described in section 4.5.1. To isolate the different narrative parameters from each other they were set up in a matrix. One axis with the parameters, and one with the prototypes. The notes in the matrices are the initial notes taken when trying out the prototypes for the first time since they were finished. Then further thought was put into the designs on how the narrative parameters can be affected.

8.1.1 Fictional world

In *Banana Toss*, there is a weak coherent $\leftrightarrow$ incoherent bond considering the bananas going out from a small town in Sweden. If this should be incoherent can be discussed further, but small changes to rules would give a much more clear incoherence. Designing the gamelet so that you have one banana and three tries might provide a greater incoherence to the fictional world since the reason for having it can only be described by referring to the rules of Banana toss.

In the more abstract gamelets – like the *Jigsaw Puzzle* – there is a problem making sense and provide diegetic elements. In the puzzle gamelet, the only thing we can do is using images that might give a clue to the story progress. Other abstract gamelets might have problems, unless adding iconic symbols and/or using the thief on each image puzzled together as a hint of who that person can be.

Using scenery from the fictional world, the player might get to know about it better. Learning how the houses look in detail or maybe even get to know of others not currently used in the main narrative. This can make the reader more aware of what actually exist is the fictional world not currently used in the narrative, this could be connected with a micro narrative.

8.1.2 Character

Changing setting and slightly changing the rules of *Banana Toss* might get another result. Initially it doesn't drive the story forward since it's not connected to the narrative, but say throwing apples into a drainpipe from an office characterizes events coherent to the main narrative of *JerryMaya*. Also here the more abstract gamelets have a problem since it's in their nature to be non-narrative driven, thus making it hard to have them characterizing events in the narrative. But instead using more concrete interactions as the *Apple Poke* provides with a more clear characterization of events. Placing of a gamelet can have huge impact on temporality where things happen prematurely or to close to the end, making any possible surprise too obvious and ruined.

8.1.3 Emotion

The 4 Piece puzzle provide the emotion you get from solving a puzzle. This is trivial, but still important to note. The abstract gamelets have a hard time providing fictional emotion, but they can be tweaked hard to provide the desired gameplay emotion.

Banana Toss doesn't really create any gameplay emotion that leans towards what the book is creating in fictional emotion. If on the other hand it is changed so that it's a two player co-op effort it could give the same friendship/co-operation emotion the narrative portrays.

8.1.4 Micro-narrative

Coherent and incoherent games prove very good at telling a micro-narrative, just have to make it more narrative-driven. Add dialogue between deckhand and sailor or more cause and effect environmental storytelling by placing things in the background. One could easily imply many micro-narratives with this kind of gamelets that don't need to rely on any previous material. The potential for a micro-narrative to become incoherent with future releases created by the original author is a possibility. An abstract game doesn't really tell a micro-narrative unless heavily interpreted by the reader.

8.2 Reflection of overarching methods

The end result of what Appmazed really wanted was not clear from the start, this resulted in the focus being moved towards a more theoretical analysis than initially stated. The result was initially thought to be a bit more technical, providing an API and a schematic on how this kind of gamelet system could be implemented and used together with their tool for creating content. This together with a couple of gamelets that was designed using the given API.

Switching to a more theoretical approach, the result became more of design suggestions that can be used when designing gamelets for an interactive book.

In retrospect, a more clear and finished vision of what was going to be investigated/produced would be to prefer. A lot of the articles read in the prestudy was not used at all, even if they covered studies and/or theories that could be very useful for these particular designs, including: Play and learning, children using gestures, how children understand gestures when they are described in text, game design in general with regards to children etc. This resulted in wasted time with regard to productivity towards the final result. Appmazed also changed their minds quite often on what to prioritize. This in the end, let to poorly optimized work flow. Some prototypes was discarded mid sprint, and even goals was changed during sprints. In the end, I feel that more of the interesting prototypes could have been implemented instead of just being ideas.

If done again, focus would still be gamelets and how to use them for diegesis. But implementation would be done differently as mentioned above. For instance implementing the Banana catch (section 6.3.3) or Graph tower (section 6.2.2) properly to get more working implementations showing the gamelet communication in action.

8.3 Reflections on specific methods

Using brainstorming when working alone seems redundant in retrospect. It turned more into just sketching on ideas and posting them in a book.

Working with Agile methodology, in this case Scrum i think would have worked fine. What didn't work was that Appmazed was not used to working with this kind of model and the specifications were changed in the middle of sprints which is not Scrum development at all. This made it hard to focus on tasks and it hurt some of the prototypes, taking the development to a halt rather than having them evolve.

The framework for analyzing what narrative components could be affected was of great help. Dividing into several groups and in some cases sub-groups (emotions especially) created a greater insight in what happens to the diegesis when the reader uses a gamelet when reading the book.

Prototype coding worked well. None of the base framework from Appmazed was really done, so creating a system for the gamelets would most probably be a waste of time since it was not known at the time how the base system was supposed to be designed. This also provided a quick way of testing the concepts and getting results very early. One drawback was that modifications during later sprints sometime took a bit more time than desirable.

8.4 Generalization

The implementations are more or less designed to work with JerryMaya, and are also coded using a prototype coding style so re-use of the implementations is not recommended. It would probably be more beneficial to re-implement them using a framework designed to fit gamelets into their desired environment. For instance, a more technically mature version of the product that Appmazed wish to create would be a good entry point.

Gamelets are ideal to use as design framework in virtually any [digital] context already using images in any way, the exact effects of this cannot be guessed and would of course have to be analyzed separately. Gamelets are not limited to digital books, but could also be extended to different contexts. One existing example of this could be the Google's so called doodles. As one example they decided to tribute Pac-Man's 30th anniversary by making their logo on their main site a re-make of the original game with one complete and playable map[22]. In this case effects could be that they increased general awareness around a piece of game history, and might even have introduced it to some people who actually never played it. Or it could also be part of a well designed scheme to increase the general view towards them as a company. Both of these suggested effects can classify as a para-functional, which shows that gamelets carry general potential for achieving different results. This shows one other application, wonder what would happen if Google would let their gamelets/doodles communicate.

8.5 Future work

To properly implement communication between gamelets by actually implementing more mature versions of the proof of concepts described in section 6.2.2 would be a desirable continuation for this work. Some of the gamelets presented show weakness when it comes to being defined as gamelets and would have to be iterated over a couple of times in order to fix these problems. The gamelets are also quite small and would be relatively easy to implement. This could open up for testing on target groups in order to study how different gamelets affect how they read. Maybe they will jump back and forward

[22] http://www.google.com/doodles/30th-anniversary-of-pac-man

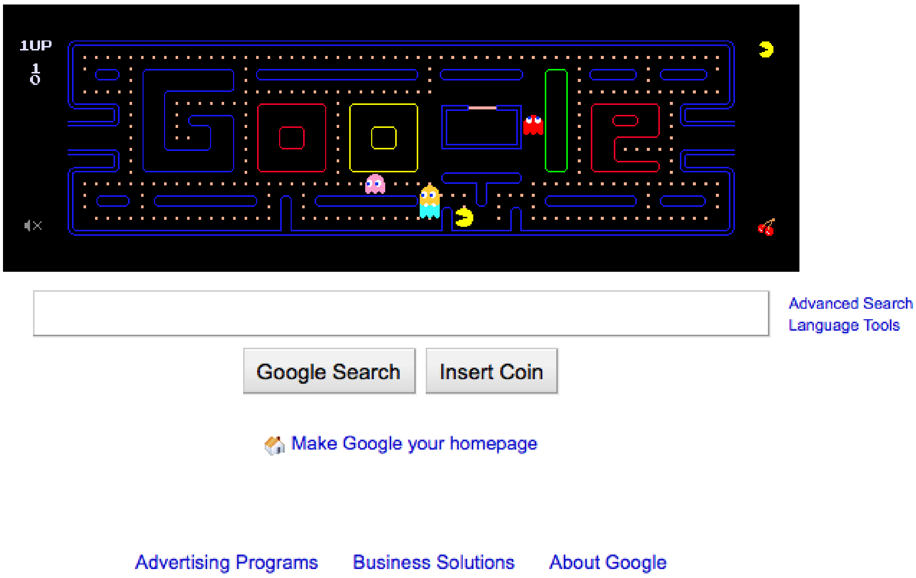


Figure 23: Google logo remade as a Pac-Man tribute

between pages instead of going linearly or maybe they stop to read completely. Would this be desirable and what could be the causing factors for this. When describing Eric Todd's prototype types Fullerton [2008], they lack to mention that aesthetics also could be testing what people feel and how they appreciate certain elements of a game, and in this case it would really be ideal to implement them digitally since the main reason was to mix something already created (paper books) with something new (iPad) by inserting gamelets. Testing this digitally would be the ideal way to move on and elaborate the framework since it at it's current state have not been the subject for any user testing, only by me and Appmazed which by definition is a bad group to even begin testing on since the JerryMaya demographic is age 7–9.

To start exploring gamelets in another context, and in the end also completely general and broaden the way they are discussed in this theses would also be a nice addition to the framework. This way, the full potential with using gamelets as a framework for design could be explored and tested in a more satisfying matter.

9 Conclusion

The work described in this thesis started with trying to figure out how to design games that could be used to support the next big narrative form on the popular iPad. The iPad had recently had gained a lot of ground on the market and several products released for Apples touch based system, iPad and iPhone alike, were harvesting great commercial success. This is something still true today.

While – during the pre-study – trying to understand what products exist today and how they work with these kind of designs, the idea of working with what would be defined as gamelets came to life. This was something that was impossible to guess beforehand due to the loosely defined project, but rather came to life as a central part through discussion.

In order to explore the subject, brainstorming sessions was conducted to try to gain ideas on how to move forward with this concept. The brainstorming spawned an idea to let the gamelets communicate in different ways with each other while sharing information and in this way explore the possibilities of how they can be designed and also how it could affect the narrative. The main gamelet communication types identified for this was named after one example of application, but not necessarily locked to that particular application. They consist of *Achievements*, *The sequel*, *Recursive*, *Ping-pong* and *Crisscross*. This list is probably not complete, but is meant to work as a starting point in how to develop this further and to show only some of the possibilities with using *gamelets*.

Several prototypes was conceptualized as proof of concepts, which in turn was presented to Appmazed together with the general idea. After discussion with Appmazed, a subset of these were selected for implementation. These implementations would work as parts of a demo, both in order to see what happened when the implemented gamelets were put into a context and further on showcasing their idea for a product. The prototypes all had a theme that would (more or less) fit into the *JerryMaya* fictional world.

The more abstract *gamelets* like *Diamond pick* and *Jigsaw puzzle* provided the most safe way to maintain the main narrative and still add some aspects of diegesis. Although, they still could affect the actual story and not only how it was told. In this case that power would belong more to the images used, but they were still a part of the gamelet. The non-abstract gamelets *Banana toss* and *Apple poke* could definitely provide more of a narrative of their own, which means those types of *gamelets* would have to be more carefully designed with regards to this. But even with a design very coherent to the main narrative, these types of *gamelets* could change the reading experience immensely. One example would be the that placement of *Apple poke* could prematurely give too much information about the plot and therefore changing the way it's supposed to be read, giving away who the culprit is way to early. This of course could be considered as both good and bad, depending on what the goal of the design is.

Gamelets are usually inserted as replacement for pictures, making them a central part of the diegesis by injecting them into where the reader keeps attention. This way, gamelets can affect the narrative experience significantly if not designed carefully, both with its content (gameplay, interaction or graphics) and where it's placed. Letting them affect each other is an addition that can be used to try and achieve more depth in the usage and also try to help the designers create diegetic elements, invoke emotion and come up with new creative ideas.

Bibliography

E. Aarseth. *Cybertext: Perspectives on Ergodic Literature*. John Hopkins University Press, Baltimore, 1997.

J. Bizzocchi. Games and narrative: An analytical framework. 2006.

D. Carson. Environmental storytelling: Creating immersive 3d worlds using lessons learned from the theme park industry, March 2000. URL http://www.gamasutra.com/view/feature/3186/environmental_storytelling_.php.

J. Dever and G. Chalk. *Lone Wolf: Flight from the Dark*. Berkley Books, New York, USA, 1984.

T. Fullerton. *Game Design Workshop*. Elsevier, Burlington, 2nd edition, 2008.

J. Gold. *Object-oriented Game Development*. Pearson Education Ltd. Harlow, Essex, England, 2004.

J. Jenkins. Game design as narrative architecture. In N. Wardrip-Fruin and P. Harrigan, editors, *First Person:New Media as Story, Performance, and Game*. The MIT Press, Cambridge, MA, 2004.

J. Juul. *Half-Real: Video games between real rules and fictional worlds*. The MIT Press, Cambridge, Massachusetts, London, England, 2005.

N. Lazzaro. Why we play games: Four keys to more emotion without story. 2004.

B. Perron. A cognitive psychological approach to gameplay emotions. 2005.

K. Salen and E. Zimmerman. *Rules of Play*. The MIT Press, Massachusetts, 2004.

K. Schwaber and J. Sutherland. Scrum guide, March 2011. URL http://www.scrum.org/storage/scrumguides/Scrum%20Guide.pdf.

H. Sharp, Y. Rogers, and J. Preece. *Interaction Design: Beyond human-computer interaction*. John Wiley & Sons Ltd, The Atrium, Southern, 2nd edition, 2007.

Ed S. Tan. *Emotion and the structure of narrative film: Film as an emotion machine.* Hillsdale, NJ, England: Lawrence Erlbaum Associates, Inc., 1996.