

TellStory

A Medialogy project about storytelling in handheld games



Medialogy - 10th semester

Project period:
01-02-2010 to 16-06-2010

Supervisors:
Tony Brooks & Kristoffer Jensen

Student:
David Lindholm

Abstract

This paper describes a project made to explore storytelling in a game on a hand-held platform. The application used in the test is a small game-like iPhone app, implemented using the iPhone SDK 3.2 and various other tools.

The application tells two stories using two different storytelling tools: Non-player character (NPC) dialogue and pure text. To evaluate the impact of having a character there to tell the story versus just reading a screen of text, a small group of people were tested and interviewed. The results give some insight into what factors influence storytelling in a hand-held game, as well as the understanding of the story and storytelling preferences.

David Lindholm

Reader's manual

The report is numbered with Arabic numerals, and the appendix is numbered using Roman numerals. When referencing other sections, both the section and page numbers will be listed. All figures and tables are numbered incrementally using Arabic numerals.

When reading this report, any mentions of previous or earlier projects are to be understood as previous projects and project groups I have been involved in.

Acknowledgements

Parts of the test application relies on graphics that were reused from previous projects. Additionally, as there is a small amount of overlap between this project and previous works, parts of this report contain content also used in earlier reports. For those reasons, I would like to thank my former associates Razvan Enescu, Qiong Jia, and Nicolaj Hansen, for allowing me to continue the work that we started together.

Disc contents

The report also comes with a CD with the following contents:

- The report in PDF format
- Sound used in the application
- Images used in the application
- Application code
- Compiled application

Table of Contents

Abstract.....	2
Reader's manual.....	3
1. Introduction.....	5
1.1 Project description.....	5
1.2 Problem formulation.....	6
1.3 The application idea.....	7
1.4 Motivation.....	8
2. State of the art.....	10
2.1 Interaction.....	10
2.2.1 Usability.....	10
2.2.2 Design principles.....	11
2.3 Handheld gaming.....	13
2.3.1 History of handheld gaming.....	13
2.4 Storytelling.....	16
2.4.1 History of storytelling in games.....	16
2.4.2 Types of storytelling.....	18
2.4.3 Gameplay and storytelling.....	20
3. Implementation.....	21
3.1 Software choices.....	21
3.2 Hardware choices.....	22
3.3 Story design.....	24
3.3.1 Text-based storytelling.....	24
3.3.2 NPC-based storytelling.....	25
3.3.3 Story creation.....	25
3.3.4 Sound elements.....	26
3.4 Graphical implementation.....	27
3.4.1 Character design and animation.....	28
3.5 General implementation.....	31
3.5.1 Initiation & progression methods.....	31
3.5.2 Story methods.....	34
3.5.3 Interaction methods.....	34
3.5.4 Animation methods.....	35
4. Results.....	37
4.1 Research method.....	37
4.2 Target group.....	38
4.3 Test goal.....	38
4.4 Test setup.....	38
4.5 Test process.....	41
4.6 Test results.....	43
5. Discussion.....	47
5.1 Reflection on results.....	47
5.2 Implementation limitations.....	48
5.3 Project evaluation.....	50
5.4 Further possibilities.....	51
5.5 Business plan.....	52
6. Conclusion.....	54
7. References.....	55

1. Introduction

Before the project could properly begin, it was necessary to establish the aim, the general goals, and the problem formulation. First the scope of the project was set, and then the problem was formulated based on the goals. In order to further direct the project, the motivational factors behind it were also determined.

1.1 Project description

Stories can be told through many vastly different media, and in vastly different ways even within one medium. Storytelling is a major topic in game development, and in the past many games have relied heavily on it to compensate for various shortcomings due to the technology behind the game. This was the case in many early role-playing and adventure games for the PC, such as the 1975 role-playing game *Colossal Cave*. In these games the graphics would typically be poor, and the gameplay often repetitive. As such, the main selling point of the game was an interesting story. As technology became more advanced, more game developers began to focus on things such as graphics and gameplay; often reducing the story to a minimum. This is illustrated no better than by the 1993 first person shooter *Doom*, where the entire story of the game is told in five still images with a bit of text, each of which are shown to the player once he has completed four to five levels. In the late 1990s an increasing amount of game developers began to view games less as toys and more as storytelling mediums on par with movies and books. This development has continued to the point where some games are close to being 20 hour interactive movies, such as the 2010 interactive drama *Heavy Rain*. Even games which focus on gameplay and graphics, such as the 2010 games *Bayonetta* and *Dante's Inferno*, take time to also tell an engaging story.

As this development has happened on the PC and gaming consoles, it has also affected hand-held gaming. Due to technological limitations, the earliest hand-held games had very little story; a trend that is still going within certain types of games. However, just like gaming consoles, hand-held devices also have their share of games that rely on storytelling.

Regardless of the platform a game is played on, there are numerous ways storytelling can be implemented. These ways are described by Österberg (2006), who lists eleven different tools to use in storytelling in games. These tools are:

- Pre-rendered video
- In-game cutscenes
- Scripted sequences
- Voice-over
- Ambient actions
- Character development
- Location evolvment
- Interactive dialogue
- Storytelling items
- Player journal
- Plain text

Österberg (2006) explains what each tool is and how it can be used, and research has been done to determine how some of these tools work in PC games. An example of this is a study done by Mallon and Webb (2005) regarding narrative elements in adventure and role-playing games. Others argue that gameplay and storytelling does not mesh at all as it is two different things (Juil, 1999). However, little research had been done to determine how each tool works on a hand-held platform, since storytelling in these types of games is still a fairly new idea.

In order to investigate this further, a hand-held application has been created. It is used to tell two

stories using some of the tools mentioned by Österberg (2006). Specifically, the focus is on text-based storytelling (in-game books, player journals, or plain text) versus non-player character (NPC)-based storytelling (dialogue, ambient actions, monologues, etc.).

1.2 Problem formulation

The goal of this project was to evaluate how the different storytelling tools work on a hand-held platform, and what the difference is between text-based and NPC-based storytelling. The focus will be on how well the test participant understands the story, and which type of storytelling tool he prefers.

The hypothesis, which was to be disproved by the end of the project period, is as follows:
There is no discernible difference between text- and NPC-based storytelling, on a hand-held device.

Specifically the goals to be evaluated were:

- Understanding – How well does the test participant understand the story.
- Preference – Which storytelling tool does the test participant prefer, and why.
- Difference – Which factors influence how the storytelling in a game is perceived.

1.3 The application idea

When doing any type of project alone, the process of generating ideas is important, so as to not just grab the first idea because it sounds right.

There were several rounds of idea generation required for this project, after the initial choice of storytelling as a general research topic. First the exact research topic had to be determined, then possible implementations for the test application, followed by the stories that should be told in the application, and finally the type of test to be conducted.

In order to distil the many possible ideas into one solid solution, it was necessary to go through several iterations of creative processing for each part of the project. The idea generation phase of the project was inspired by the processes used when deciding on ideas in the groups of previous projects. It was divided into four distinct parts in the following way:

1. Setting the scope – The objective was to come up with ideas that would fit the goal of this project. The phase occurred prior to choosing the iPhone as the platform for the project, so any ideas were allowed as long as they could be related to storytelling.
2. Idea consideration and delimitation – Any ideas that were impossible or ridiculously impractical were removed.
3. Distillation – The remaining ideas were organized into groups and ordered by their feasibility. To determine the feasibility, several factors were considered, such as time constraints, prior knowledge, impact on the test process and results, and how well they would work for a test with focus on storytelling.
4. Final distillation – Here the final choices were made, based on the knowledge gained from the third step. The other ideas were not discarded, but kept as alternatives if a choice should later turn out to not be possible.

The final idea was to focus on just a few storytelling tools, as there are too many to accurately implement, test, and compare in a single project period. These storytelling tools would be implemented in an iPhone application as separate stories, each told by a different tool. The stories are heavily modified versions of stories created for previous projects, to keep the development time shorter. The navigation in the stories should be handled by simply touching the screen of the device, and it should be possible to move both forwards and backwards in the stories in case the test participant missed something. This application would then be used to conduct a qualitative test about storytelling.

No game world as such is required, which reduces the development time further. Storytelling in a game is usually the result of several factors working together in the game world, but in order to properly test a single tool, it is simplified if the tool is allowed to stand on its own.

1.4 Motivation

There are two important aspects of motivation to discuss in relation to this project; The motivation behind doing the project, and what will motivate people to participate in the test.

In several previous project groups in this education, it became a priority to test something that could be used in the creation of games and/or animation. This was the result of a general interest in the topics within the group. It is also the main motivation behind this project, as storytelling is an integral component of animation and many types of games.

Major topics of the previous semesters of the Medialogy education has been storytelling, game development, and interaction, and this was a major contributing factor to the choice of research topic. Depending on the type of game, storytelling can be one of the most important things to understand as a game developer. In first-person action/horror games and role-playing games, it is common to have a strong story and a solid feeling of engagement. The immersive feeling is important as the game will otherwise not feel as scary or interesting, and the story is important in both creating the right feeling and involving the player in the game.

However, a common issue is how to properly use the story to engage the player in the game world. The way a story is told can affect the way it is perceived and understood by the player. It does not matter as much if the story is believable in relation to the real world, but rather in relation to the game world that has been created thus far. If a game is about medieval vampires, it should not suddenly have a level filled with aliens, unless a very good explanation is given as to why they are there. Similarly, the story has to be understood by the player, in order for him to be properly engaged in the game world. He does not have to know every detail, but without at least a basic understanding of the story, he may get confused or simply be unable to relate to the actions of NPCs and things happening in the environment. The tools used to tell the story play a big role in helping the user to understand. Different tools may affect the player differently, both in regards to his understanding as well as his enjoyment of the experience. It does not matter if a game has a great story, if it is presented in a way that makes the player annoyed or bored when it is told.

To research the topic further, an application was created in which players experience two stories told by two different storytelling tools. A test was carried out to determine, amongst other things, the understanding of the story and which tool the test participants preferred.

Despite having previously done multiple projects related to gaming and hand-held gaming in particular, there are still many areas to investigate. This is especially true for storytelling in games, as it is still a very new field. This will be discussed further, in the state of the art chapter, for now it is sufficient to say that it is a strong part of the motivation behind doing the project.

The second type of motivation is the matter of motivating the test subjects to participate in the test, and what would motivate the players to play the game if the storytelling tools from the application were implemented into a full game. It is important to understand what will motivate a person to participate in the test, and perhaps to recommend it to his friends, as it is then possible to improve those factors in order to get more motivated participants. If parts of the application are ever to be used in a real game, motivation will also be a major factor in who will like or dislike the game. Additionally if the test participant is motivated by factors in the application, he will most likely take the test more seriously, which in turn should lead to more accurate results.

Motivation can be viewed as a type of stimulation, in which there are two major types, which can be defined as follows:

- Intrinsic motivation, when a person engages in a activity because of interests or because it's a satisfying experience, according to Gagné & Deci (2005).
- “Extrinsic motivation [...] requires an instrumentality between the activity and some separable consequences such as tangible or verbal rewards, so satisfaction comes not from the activity itself but rather from the extrinsic consequences to which the activity leads” Gagné & Deci (2005) further adds.

Looking at the test for this project, it is mostly for the intrinsically motivated, though there is something for both groups of people.

There were several intrinsic motivational factors involved in the test. All the test participants were gamers to some extent, and therefore they may be motivated by the overlap in the topic of the test and their hobby. The subject and a bit of background theory was explained to the test participant. It was made clear what was hoped to be achieved by doing the test, and what the results could be used for. This was done to increase the interest in the subject, and motivation to take the test.

Furthermore, the NPC was designed to have animations that made him more interesting and engaging to look at, making the NPC storytelling a satisfying experience for the test participant.

The extrinsic motivational factors are fewer in the test. They are limited, to the verbal reward when the test participant is told that he has completed each story, and to when the participant is asked to give a summary of the story and is able to successfully do so. These are extrinsic factors, but they are much less significant than the intrinsic factors.

Stakeholders

There are two types of stakeholders in this project. The internal stakeholders are the easiest to identify as it is those who are directly involved in carrying out the project.

The external stakeholders are those that will benefit from the results of the test and possibly the test application. In this case, it is especially people who work with the creation of storytelling and games, and wish to investigate possible means of implementing storytelling elements in their virtual world.

Another external stakeholder is game developers in general, as the application demonstrates a way of simply implementing two different storytelling tools on the Apple iPhone without a big development team or budget.

2. State of the art

In this chapter the relevant theoretical topics related to the development of the application and test will be examined. First interaction is covered, then hand-held gaming, and lastly the various aspects of storytelling.

2.1 Interaction

Keeping the problem formulation in mind, there are three important goals the product must fulfil:

- Allow the player to experience two stories
- Allow the two stories to be told by two different storytelling tools
- Avoid interfering with the test as much as possible

These goals were the basis for both the latter part of the idea generation process, and the design process of the final idea.

The fact that the story takes place in a game-like environment means that the traditional rules of experiencing it e.g. reading it from a book, watching it on TV etc. are changed. The user is no longer purely the passive observer but rather an actor in the story itself. In the case of the test application created for this project, the user is still mostly a passive observer, but there are additional concerns to address compared to traditional storytelling.

Before starting the development of the application, two areas has to be considered specifically:

Video interface

This application uses the most common way of displaying a game, which is to simply have the user look at a single screen where everything appears. There are many alternatives, such as a head mounted display or the dual screen approach of the Nintendo DS. However, for all other hand-held platforms, the video options are limited to either the built in screen or an external projector. For the purpose of this test, it did not make sense to use a projector as it is not a common gaming device.

Interaction metaphor

Unlike the video options, the methods of interacting with a hand-held device vary greatly. Gaming devices typically have physical buttons and a joystick or joypad. Multi-purpose devices commonly use a mix between physical buttons (often a keyboard), touch screens, various sensors, and sometimes voice recognition. For the iPhone the interaction is a mix between the touch screen which all applications use, and the many built in sensors which vary in use between applications. To keep the test application simple, the most basic form of iPhone interaction was chosen: Touches on the screen. The iPhone supports multiple forms of touches; from simple touches, to swipes, or even gestures using more than one finger. The single touch was used for all the interaction in the test application, as it is the simplest. This turned out to be the correct choice, as none of the test participants had any trouble figuring out how to interact with the buttons and NPC in the application, despite some of them never having used an iPhone before.

2.2.1 Usability

Usability is one of the fundamental aspects when designing an interactive product. Designing a usable product means designing with the user's needs and expectations in mind. The application that was created for this project is supposed to be able to tell two simple stories through two storytelling tools. This means that every aspect of the product (hardware, audio/video interface, aesthetics, etc.) should strive to help the users understand its purpose, and to use it efficiently.

To help identify the objectives for the product, Sharp et al. (2007) suggest splitting them into

usability and user experience goals.

Usability goals

Usability goals are related to the capabilities of a product, in order to be effective to use, easy to learn and remember, and improve the work efficiency of the user. There are 6 usability goals:

- Effectiveness – How well is the product doing what it's supposed to be doing?
- Efficiency – How good is the product supporting the users in carrying out their tasks?
- Safety – How safe is the product?
- Utility – How appropriate are the program functions at solving the tasks at hand?
- Learnability – How easy is the system to learn how to use?
- Memorability – Once a user learns how to use a system, how difficult is it for him to remember it?

User experience goals

User experience goals are about how the user feels about the product, and they are used to describe the various feelings of the user experience when using the product. These feelings have a subjective nature and include both the positive and negative aspects of the user experience. There are many goals and not all of them can be relevant to a specific product and sometimes they can conflict with each other (Sharp et. al., 2007).

Examples of user experience goals are:

- Engaging
- Entertaining
- Helpful
- Aesthetically pleasing
- Fun

The usability and user experience goals were used when generating and deciding on ideas in the early stages of the project, to ensure that the potential ideas were thought through. They were used again in the design phase, to make the product live up to the goals stated at the beginning of this chapter. Particularly, the effectiveness, utility, efficiency, and learnability goals were considered, as these would directly influence how well the test participants would be able to perform when using the test application.

2.2.2 Design principles

Sharp et. al. (2007) further lists some design principles, that are meant to help designers when they are creating content aimed at the user experience. Amongst the most common design principles are visibility, feedback, constraints, consistency, and affordance.

Visibility

Visibility is about making the most important elements of an interactive system visible to the user. In computer games and virtual environments it is especially important to highlight the interactive objects since issues like image quality and screen resolution can make it even more difficult for the user to recognise them.

Common techniques of highlighting interactive objects in games are:

- Giving them higher luminosity than the other objects in the room (Diablo 2)
- Drawing glowing contours around them (Thief)
- Making them blink or even drawing big flashing arrows on top of them (Warcraft 3)

In the text-based storytelling mode, the virtual computer monitor have very large buttons that are clearly marked with their function. This means that when the screen loads, the user will immediately be able to see and understand how the navigation works.

Feedback

Feedback is about informing the user of the action he performed, thus allowing him to continue with the activity. Sharp et al. (2007) states that smart feedback can provide more visibility for the user interaction.

In both storytelling modes of the test application a sound is played when the user touches either a button or dialogue option. By not playing the sound elsewhere, the user knows that if he does not hear the sound when he touches the screen, then he has not touched something interactive.

Constraints

Constraints refers to the technique of limiting the user interactions at a given moment, thus preventing the user from making mistakes (Sharp et. al., 2007).

When the user is on the first screen of a story in the text-based mode, the "previous" button is disabled, and in the NPC-based mode there is no dialogue option to go back from the first screen. This ensures that the user does not accidentally go back to the status screen seen before the story begins.

Consistency

Consistency is about creating interfaces that share common functions for achieving similar tasks (Sharp et. al., 2007), thereby increasing the learnability of the system.

An example of consistency is the way the navigation works in the application. As mentioned earlier, everything is controled by simply pressing the touch screen once in the appropriate area.

Affordance

Affordance refers to the properties of an object that allow people to know how to use it (Sharp et. al., 2007). It is strongly related to how obvious the functionality of an object is, and how recognisable it is.

For example people generally agree on the representation of a GUI button and the way to use it (click on it once). As mentioned the buttons were controlled by a simple touch, and it is possible to "cancel" your touch by moving your finger away from the object before lifting it, which is a common way to "cancel" clicks in many GUIs.

2.3 Handheld gaming

Playing a game on a handheld device is different from e.g. playing it on a PC in multiple ways. It does not suffer from the restrictions in terms of location that traditional PCs, and even laptops, do. It doesn't have the same features either, and the power is usually less than that of an average PC. This leads to some interesting developments in terms of gameplay, as the developers are often required to think of new ways to do things, due to hardware restrictions.

2.3.1 History of handheld gaming

Interestingly, the origin of handheld games actually pre-dates the origin of the first traditional gaming consoles. The very first handheld game was produced in 1976 when toy manufacturer Mattel started developing a line of portable electronic sports games. In the following years, and all the way up through the 1980s, these devices continued to be produced. However, the fact that they only supported one game per device, keeps them from being considered a gaming console.

The first device to fit that description was created by Milton Bradley in 1979, and was called the Microvision (figure 1). However, the device sold poorly and only 11 games were released for it before production ended in 1981. In the next 8 years two handheld devices were released, the Entex Select-A-Game Machine, and the Japanese-only Epoch Game Pocket Computer. Both failed to become popular, only 5 games were manufactured for each of them.

In 1989 Nintendo, who had been working on the previously mentioned single-game handheld devices for a few years, decided to release a handheld console. It was called the Nintendo Game Boy, and when counting all the iterations of it, it is the best selling handheld gaming device ever to have been created. The Game Boy was sold with one of two different games, the puzzle game Tetris or the platform game Super Mario Land, and the popularity of the device may be attributed to these two games as both are some of the most popular handheld titles.

Competing game companies Atari, Sega, and NEC attempted to get a piece of the newly created handheld market by releasing their own handheld consoles; the NEC PC Engine GT/TurboExpress (1990), the Atari Lynx (1989), and the Sega Game Gear (1990). All three consoles offer better graphics, a colour screen, and more processing power than the Nintendo Game Boy, but only the Game Gear has a reasonably large selection of known franchise titles available. All three consoles also suffer from the same two problems: They are heavy and they have very short battery life. In the end, these shortcomings turns out to hurt the popularity of the devices greatly, and only one of the companies ever produce a handheld gaming console again.

In the next 8 years a number of smaller releases were made. Sega released their second and last attempt at a handheld console in 1995 as the Sega Nomad. It borrowed an idea NEC had introduced with the TurboExpress, which was to create a handheld console that played the same games as a TV-connected console. Thus the Sega Nomad could play the same games as the Sega Mega Drive/Genesis console. However, at this time Sega was maintaining 8 different and incompatible platforms, which meant less money to spend promoting the device and producing hardware and software for it. It ends up selling worse than the Sega Game Gear.



Figure 1: The Microvision - Source:

<http://www.computerbase.de/l-exikon/Microvision>

In 1997 the American toy manufacturer Tiger Electronics release a small handheld device called the Game.com (figure 2). It is inspired by personal digital assistants (PDA), and include some of the same capabilities for storing addresses and notes. It is also the first gaming device to incorporate a touch screen. Unfortunately only 20 games are released for the device, and the device is not well promoted, which leads to it failing to compete with the popularity of the now 8 year old Game Boy. In 1998 Nintendo releases the second version of their Game Boy, this time with a smaller case and a colour screen. The device is backwards compatible with all the original Game Boy games, which ensures it's popularity.



Figure 2: The Game.com - Source: <http://rubbercat.net/archive/technewsrumorinsider/>

Until this point, few games have been available for PDAs and mobile phones, and the ones that exist are usually simple games such as Tetris or Snake.

This changes in 2000 when Compaq releases the iPaq series of PDAs with the Windows Mobile operating system. These devices are much more powerful than older PDAs and have colour screens. This allows for games of much better quality to be produced. Many games are produced for the Windows Mobile platform, but gaming on these devices does not catch on for several years, as PDA-devices with no phone capabilities is a bit of a niche market.

Nintendo releases the third iteration of the Game Boy in 2001: The Game Boy Advance. As the previous version it is backwards compatible with all released Game Boy and Game Boy Color games, making it another popular device. It is released in a new version in 2003 when Nintendo creates the Nintendo Game Boy Advance SP which is lighter than the Game Boy Advance and has a backlight. Yet another Game Boy Advance is released in 2005 as the Game Boy Micro. It is the smallest cartridge-based handheld released to date, but it is incompatible with anything but Game Boy Advance games, which makes it only moderately popular.

Also released in 2001 is the GP32 created by the company Game Park. It is designed to be developed for by anybody interested, as there are no special hardware cartridges and no anti-piracy protection. Perhaps the latter is the reason why few professional developers choose to create games for the device, and it remains mostly a device for emulating older consoles.

In 2003 Nokia releases the first mobile phone designed specifically with gaming in mind, the Nokia N-Gage (figure 3). It is formed so it can be held sideways comfortably while playing games, and has important features such as a diagonal control pad. However, the device is ridiculed by the target audience for some design flaws: It has to be held in an impractical angle to be used as a phone, and the battery has to be taken out to change games. These factors make it an unpopular choice as a phone. Poor graphics, small selection of games, and the inability to turn the device on without a working SIM-card (and thus a cellular subscription), make it



Figure 3: The N-Gage - Source: http://commons.wikimedia.org/wiki/File:Nokia_N-Gage.png

unpopular as a gaming device. Most of these factors are fixed in a second edition, released in 2008, but it has yet to become more than a niche device. Nokia continues to release regular mobile phones

that are N-Gage compatible however, which makes the games somewhat popular among the owners of those devices.

Nintendo released the latest iteration of a handheld device in 2004 as the Nintendo DS. Rather than being based on the Game Boy series, this device was inspired by the games the company created in the 1980s which utilized two screens. One of the screens is touch sensitive and the device comes with a stylus to use on the screen. The device is also backwards compatible with the Game Boy Advance games, which makes it a popular upgrade for those looking for a replacement for their older handheld Nintendo device. Many of the DS games created are different than what is seen previously, and often involve doing motions on the screen to simulate an action in the game. For example, the 2006 game *Cooking Mama* has the player help a woman cook various dishes by making gestures common when preparing food (stirring, cutting, picking up items and dropping them into a pot). Three additional versions of the DS has been released to this date. The smaller and lighter Nintendo DS Lite (2006), a version compatible with downloadable titles, but no longer Game Boy Advance backwards compatible, called Nintendo DSi (2008), and a larger version aimed specifically at elderly people called the Nintendo DS XL/LL (2009).

In 2004 Sony release their first handheld gaming device, the Sony Playstation Portable. It is aimed at a different market than the Nintendo line up, with titles generally being more action oriented and similar to what can be found on the Playstation series of TV-connected consoles. The device uses a special disc format which supports both DVD-type movies and games. Sony intends for the device to be used as multi-purpose device, for playing games, music, and movies. Few movies are released for the device however. It is a more traditional gaming device than the Nintendo DS, but it still becomes popular thanks to Sony's access to popular franchise titles such as *Final Fantasy*, *Gran Turismo*, and *Metal Gear Solid*. Three additional versions of the device are released. The PSP-2000 (2007) and the PSP-3000 (2008) both have minor updates to hardware features, as well as a reduction in size and weight. In 2009 Sony announces the PSP Go, which does away with the special discs used in the other devices, making it device fully dependant on downloadable content sold in the online store.

Little development has happened in market for gaming on mobile phones since the Windows Mobile release, until 2007 when Apple releases the Apple iPhone (figure 4). It is a fully touch based device, with a powerful processor and many advanced sensors. At first there are no games available for it, and various software hacks are introduced by the community to install privately developed games on the device. In 2008 Apple releases the SDK and opens the App Store on iTunes, which opens the door for game companies to begin developing commercial games for the platform. As with the Nintendo DS, many games for the device utilize the touch screen and tilt sensors in ways not possible in other games. However, many larger companies also being porting regular mainstream titles to the platform, which is a first for a device that isn't dedicated for gaming. Apple has released two additional versions of the device to date, the iPhone 3G in 2008 and the iPhone 3GS in 2009. Each providing more sensors, more processing power, and more space for storing applications, movies, or music.



Figure 4: The iPhone -
Source:
<http://www.blog.noel.dk/k-ob-din-iphone-i-danmark-allerede-i-dag/>

The largest potential competitor to the iPhone currently is the Google Android mobile operating system (2008). It is functionally similar to the iPhone OS, but runs on many different devices. It has an online store similar to the one found on iTunes, but game development companies have yet to fully embrace the platform and develop for it to any great extent.

2.4 Storytelling

Storytelling is a vital part of most games, regardless of genre. The only exceptions being typical sports, puzzle, and multiplayer-only games, however even in those it has become increasingly common for the developer to add small story elements. Perhaps gamers have evolved to a point where they expect at least this much of games, or perhaps it is merely a tool for developers to make the game more interesting.

2.4.1 History of storytelling in games

In the short history of computer games, storytelling has progressed very far. In the earliest of games, there really was no story at all. There is no reason why the little yellow Pac-Man, from the 1980 game of the same name, has to run around in a maze eating dots while fleeing from ghosts. Similarly, there is no reason why in the 1978 game Space Invaders you are the sole defender of Earth with one cannon against an armada of invading aliens. As the player you just had to accept that things are the way things are, for no apparent reason.

However, when we consider the platform and target audience for the games, a reason for the lack of story does reveal itself. The vast majority of games in this time were made for arcade cabinets. Arcade cabinets were usually placed in large rooms with many different games, and you would go there, play a game for a while, then perhaps move on to another. An example of such a room can be seen in figure 5 to the right. Additionally each “life” in the game, or “credit” as it was usually referred to, cost a certain amount of money. Thus the games were often aimed at being easy to learn but hard to master, in order to easily capture in new people while still keep others coming back to spend more money. If the arcade games had a story at all, which many did not, it was often kept to just one or two screens of scrolling text at the start.



Figure 5: Arcade room. Picture from <http://www.cinemarcade.com/arcade84.html>

In the late seventies to early eighties, many new gaming systems came out, which made it possible for people to play in their own homes rather than going to an arcade. Amongst others, the Saba Videoplay (1976), the Atari 2600 (1977), the Philips Videopac G7000 (1978), the Interton VC4000 (1978), the Mattel Intellivision (1980), the CBS Colecovision (1982), the Commodore 64 (1982), and the Milton Bradley Vectrex (1982). However, many game developers of the time did not treat the consoles as a separate market, the way we view them today. This resulted in many of the console titles being conversions of arcade games, which still meant little to no story. The only improvement in terms of storytelling, was the fact that the games were now physical products you brought home, and which typically included a manual. Apart from instructions on how to play the game, the manuals also began to include various titbits of storytelling, such as the Pac-Man manual telling the player “The object of the game is to keep Pac-Man happy and healthy in his home of Mazeland.” (Pac-Man manual, Atari, 1982¹). Eventually this practice evolved into including several pages of story as in the 1990 game Golden Axe, which in the arcade version had almost no story,

1 http://www.atariage.com/manual_html_page.html?SoftwareLabelID=342

but in the manual for the console version included both an introduction to the story and a bit of background about each character². The practice is still used today, however not to as big an extent after in-game storytelling became common.

However, around the mid eighties something happened which changed storytelling in games. It started with the release of the Nintendo Entertainment System (1985) and the Sega Master System (1985). Both these consoles got a large array of games that included in-game storytelling elements (these elements will be discussed in the next subchapter). Games like *Dungeon Magic* (1987) and the *Zelda* games (1987 & 1988), included many storytelling elements in the game world, which combined to form a deep story. An example of these elements can be seen in figure 6. Even platform games such as the *Super Mario* series began having story elements in-game, though it wasn't until *Super Mario Bros. 3* (1990) that the story really started to evolve into more than an excuse for jumping around a level killing things.³



Figure 6: Storytelling element in *Zelda II*

Storytelling continued to develop up through the eighties and early nineties on a new generation of gaming consoles, with the release of the Sega Mega Drive (1988) and the Super Nintendo (1990). This coincided with a general shift away from large arcade complexes in much of the western world, which meant that arcade games were no longer being released in large quantities, causing more and more developers to move on to the personal platforms.

It was also around this time graphical gaming on the PC started to become common, which interestingly meant a reduction in story for many games. Up until this point, PC gaming had been largely text based, which for many games meant that the developers would have to employ storytelling devices commonly used in books, in order to make the game interesting. Multi User Dungeons (MUDs) were quite common, and so was text based adventures like the 1975 game *Colossal Cave* in which you encounter fantasy creatures like orcs and elves without ever actually seeing a picture of them.

With the introduction of graphics in PC games, many developers soon started focusing on making the game look good while the story took second place. An example of this practice can be seen in the first-person shooter (FPS) *Doom* from 1993. While the game itself is generally accepted as being a landmark within 3D gaming, the story was virtually non-existent. The only story elements used in the game was a screen of text at the end of each set of levels (5 total) that sought to describe the meaning of what you had just been doing. As in console games half a decade earlier, the story was merely used as an excuse for you to run around shooting monsters. This way of using a minimum of story became commonplace in many FPS games throughout the nineties, as well as in other genres to a lesser extent.

² <http://www.oldgames.sk/en/game/golden-axe/download/1024/>

³ http://www.gamasutra.com/features/20061103/quantum_03.shtml

In the mid nineties to early in the new millennium, console and PC gaming met up in terms of storytelling, with the release of the 5th and 6th generation of consoles. The Sega Saturn (1994), Atari Jaguar (1993), Sony Playstation (1994), and Nintendo 64 of the 5th generation were first to improve graphics to PC levels. Just a few years later, as the PC improved, the Sega Dreamcast (1998), Nintendo Gamecube (2001), Sony Playstation 2 (2000), and Microsoft Xbox (2001) formed the 6th generation of consoles. These and the current generation of consoles can be seen in figure 7.

In terms of storytelling the games for all the platforms were also very similar within each genre. At this point game genres had diverged quite far from each other as far as how much story they contained and how that story was conveyed. Sports, racing, puzzle, and FPS games generally had little to no story, while role-playing games (RPG), strategy games, and adventure games were medium to heavy on story. An example might be the 1998 real-time strategy (RTS) game Starcraft which has an extensive story spanning three races of beings, and a vast number of missions. Another example is the Final Fantasy series of games, many of which came out in this time, and all of which are very heavy on story.



Figure 7: Current and past two generations of consoles

As the first decade of the new millennium progresses, storytelling has become a bigger and bigger part of gaming. Games such as Dreamfall by Funcom have a huge and well developed story which seeks to draw you in and make you really care about the characters and the world they inhabit. Similarly, the game Jade Empire has a lot of story, despite being mostly an action game. The way you choose to act decides on the outcome of the story; a practice that has become more and more common in story-driven gameplay. While certain genres still lack behind the rest, games have gone from being a mere waste of time for a short time, to being grand stories fully on par with what you would find in movies and books.

2.4.2 Types of storytelling

Österberg (2006) describes eleven different storytelling tools used in single player action games:

Pre-rendered video

This type of storytelling is often used in the introduction of the game, or at the end to wrap up the story, and to a lesser extent as cutscenes in the gameplay itself. An example of this can be seen in the opening of the 2002 game Warcraft III: Reign of Chaos. This opening serves a triple purpose as an introduction to the major factions in the game, a general introduction to the story, and in setting the mood for the game.

In-game cutscenes

Similar to the above, in-game cutscenes are premade for the purpose of telling a chunk of the story without input from the player. The major difference is that these cutscenes are made within the game engine and not by external rendering tools. This type of cutscene can be seen at the end of the first level of the 2001 game Halo: Combat Evolved. The protagonist is narrowly escaping a

devastated spaceship on a personnel carrier, which then crash lands on the world where the rest of the game takes place. Using the in-game engine saves a lot of development time, when there is no need of the superior graphics of pre-rendered video.

Scripted sequences

These are similar to cutscenes in that they are telling a part of the story to the player, however the player retains control of his character, which means that he may miss part of the action if he moves away, but it is also a good way not to completely disconnect the player from his character as cutscenes and pre-rendered video do. This type of storytelling is quite common in action games, for example in the 1998 game Star Craft, it happens several times at the end of a mission that reinforcements arrive to fight off the rest of the enemies. They usually fly in over your base, so if your view is elsewhere you will only get a message in text that they have arrived.

Voice-over

A voice-over is simply the playing of an audio clip with some speech in it, while the player continues going about his business. It can be experienced in the beginning of the game Dreamfall, where the main character provides voice-over descriptions of some items which will be important later in the game, as well as telling you some of the backstory.

Ambient actions

These types of actions usually appear between NPCs in the game world, and seek to tell minor elements of the story by having virtual characters interact with each other or the environment. In Halo: Combat Evolved you will several times run into stranded marines who are either fighting off enemies or making small talk with each other about the enemy or the situation they are in.

Character development

This type of storytelling is commonly used to tell part of a characters past through his current look and abilities, such as it's possible with real people. In both Fable (2004) and Fable II (2008) the actions you perform in the game (good or evil) will change the look of your character, making him either glow from being just, or grow horns and get red eyes from being evil. Consequently this will change how NPCs react to you. An unlawful NPC might be hesitant to deal with someone known for being a just hero, while villagers might run away scared from an evil hero.

Location evolvment

Similar to the above, this type of storytelling is also about changing the environment over time to reflect actions of the player and things that happen in the story. In Jade Empire, early in the game you leave your village to rescue a friend. During that time you see flying machines pass overhead, and when you make it back you learn that they have bombed the village which is now mostly collapsed and on fire.

Interactive dialogue

This type of storytelling has nearly always been common in the RPG genre, and lately it is also being used more and more in other genres. In The Legend of Zelda: Majora's Mask (2000) and several other Zelda titles, NPCs will often tell you a bit about themselves or a problem they are having, followed by some questions you can respond to in different pre-written ways. Depending on your answer, the NPC might give you a quest, run away, refuse to talk to you any more, or even attack you.

Storytelling items

These items can be many different things, such as signs, a notebook, a message on an answering

machine, etc. They are used to amplify the mood and feeling of the game, and to expand the story for those interested. In *World of Warcraft* (2004) especially books are used in this manner. They are found all over the game world, in friendly places, in dungeons, and even in the capital cities of the enemy, and they tell various tales about the story of previous games, books of the franchise, and other aspects of the game lore.

Player journal

An in-game journal is a tool used for retelling parts of the story that the player has experienced. It can contain information about people he has met, conversations he has taken part in, quests he has done, and many other things. It usually will not say anything about situations happening elsewhere, but it can add extra things the character, but not the player, would have experienced, such as emotions. An example can be found in *Dreamfall* where the journal contains everything being said in the game, even when the protagonist is acting as a narrator.

Plain text

As mentioned in the previous sub-chapter, plain text is probably the storytelling tool that has been around the longest. It is simply a screen of text that conveys some part of the story. It can be found even in modern games, such as the 2008 game *Soulcalibur 4* where a short screen of scrolling text explains the backstory and motivation of a character before the actual gameplay starts.

2.4.3 Gameplay and storytelling

Mallon & Webb (2005) conducted a test on narrative elements in adventure and roleplaying games, and concluded, based on their test, “the participants' response indicates that well-designed narrative structures and devices increase engagement with role-play and adventure games.”. They go on to say “Without good narrative, the players had a poorer quality, more impoverished, experience.”. Juul (1999) contradicts this, and on the topic of games and narratives he states that “They are two separate phenomena that in many cases rule each other out.”.

Furthermore, Mallon & Webb (2005) says “Criticisms of narrative devices within a game (Talin, Adams, Costikyan etc.) seem unjustified, since we found that players wanted them”.

Chris Crawford (2005) presents yet another different view: “Interactive fiction is certainly interactive, and it's fictional in the sense of being made up, but it's certainly not storytelling.” In regards to the death of a companion robot in the 1983 game *Planetfall*, he further added “Players were overwhelmed with the moment's emotional power. There was nothing players could do to avert it.” and he sees this as a lack of interactive storytelling.

This is a strong indication that despite research in the field, there is still much disagreement on the topic. In turn this means that more testing is required, as there may be other factors involved.

Furthermore, it's also possible that the storytelling is experienced differently in an application that is not a role playing or adventure game.

3. Implementation

Once the idea had been decided upon, the work of actually developing it could begin. The development process can be divided into three separate, but mutually dependent, parts. The choice of software, the hardware, and the actual programming. These parts will be described in the following chapter.

3.1 Software choices

The choice of software is heavily dependant on several other factors in the development. Most notably, they are limited by the hardware choices. There are still several options for each part of the project though, and the purpose of this subsection is to document those decisions. The choices were based on a number of things, but especially what was possible to do in the time allotted, as well as how the finished application should appear. Prior experience partially influenced the decisions, but not to a point where a choice was made to avoid learning something new. However, knowledge of what certain pieces of software can, and cannot, do meant that some of the final choices were made easier.

Apple iPhone OS 3.2

As the Apple iPhone was chosen as the platform for the application, a choice which will be further described in the hardware choices subsection, there were few options in regards to programming language and platform. The two main options when making an application for the iPhone is to write code that runs on the iPhone operating system (OS), or create a “web app” which runs through the built-in Safari internet browser.

The advantage of using the browser is that the development is extremely easy. The graphics and functionality can be implemented by using HTML and a few pieces of Javascript. However, this comes at the cost of features, as web applications are very limited in what they can do on the iPhone. Particularly, the browser is not allocated as many resources as a normal iPhone application, so delays in animations and audio playback can occur. Similarly, a web application is not allowed much local storage, so the user will often have to wait while more graphics and audio data is downloaded over the network. These two shortcomings are the reason for choosing to develop an ordinary iPhone application, rather than a web app.

Apple iPhone SDK 4.0 beta

The Apple iPhone Software Development Kit (SDK) is a collection of tools and libraries for developing for the iPhone. To develop professional applications for the iTunes App Store it is required to use it, but otherwise it is possible to develop iPhone software without it by using one of the free tool chains available online.

The advantage to using the official SDK is access to various diagnostic tools which can help prevent bugs, access to a large library of documentation, and access to restricted application programming interfaces (API) that are not available in the free tool chains. The disadvantage is that the official SDK will only run on a Macintosh computer.

As the unofficial tool chain offers no benefits apart from running on non-Macintosh hardware, the official SDK was the better option.

Cocoa Touch

It is possible to develop for the iPhone using several different languages. However, when using the official development software, the only option is Cocoa. To use other languages it is necessary to install third party compilers and “trick” the phone into accepting the compiled binary. Using another language also severely limits the functionality you can use, without providing any substantial benefit, so Cocoa Touch is the logical choice.

Cocoa is a type of Objective-C used by Apple to develop the majority of their software for both Mac OS X and the iPhone OS. Additionally Cocoa Touch is a specialized abstraction layer used to make it easy to work with the special features of the iPhone, such as touch gestures, calendar, address book, etcetera. Cocoa Touch does not change Cocoa, but merely adds features not easily available in the latter.

Xcode

When the iPhone SDK is installed on the computer, a large amount of tools are also installed, one of those being Xcode. Xcode is a collection of free development software originally created by Apple for developing applications for Mac OS X. In addition to a traditional integrated development environment (IDE) it also includes an automatically updating documentation browser, performance tuning software, etcetera.

Many other IDEs will run and compile Cocoa code on Mac OS, for example the Eclipse IDE. However, none provide the same level of integration with the iPhone SDK that Xcode does.

The GNU Image Manipulation Program

All the imagery used in the test application needed to be edited in various ways. The changes ranged from resizing, to minor changes to the content of the image. The editing required no advanced functionality, so there were many options for graphics editing software, such as Adobe Photoshop or the GNU Image Manipulation Program (GIMP).

The GIMP was the final choice, due to past experience as well as the fact that it did not require the purchase of a software license.

Adobe Flash CS4 Professional

Part of the background imagery for the application was recycled from a previous project, but there were also several new things that needed to be created. Particularly the NPC and the animations for it, as well as the various text boxes used in the application.

Creating an NPC can be a complex task if you are inexperienced in drawing and modelling, but a tool like Adobe Flash can simplify the process to a great degree. It has various drawing aids which have been a great help in past projects, and thus it was the perfect choice for the creation of the graphics needed for this project.

Audacity

There is only one sound effect used in the test application, but before it was ready to be imported into the application it required some editing for length and volume. As these are both minor changes there were many options. In past projects, Audacity has proved to be a useful tool for these kinds of minor audio editing, and therefore it was also chosen for this project.

3.2 Hardware choices

In the project there were three major hardware decisions involved in the creation of the test application. Firstly the platform had to be decided, secondly how to display it, and finally how to control it.

Platform choice

As the purpose of this project is to test storytelling in hand-held games, that strongly influenced the choice of platform. The idea of testing hand-held games rather than games in general, stemmed from a great personal interest in hand-held gaming. Furthermore, it is an area where not much research has been made public yet. Presumably the big companies that make hand-held gaming devices, such as Nintendo and Sony, have done research into many topics in relation to them, but very little of it is available to the general public.

There are many hand-held platforms that a storytelling test could be carried out on, all with their advantages and disadvantages. The dedicated gaming platforms, such as the Nintendo DS and the Sony PSP, are usually difficult to develop for. The reason is that to develop for such a platform you need to purchase a development kit and license, which are usually only sold to game development companies. Some devices have free development kits available, but development is usually difficult, and uses many proprietary and poorly documented features of the hardware. The multi-purpose platforms, such as the iPhone and PDAs are easier to develop for as they usually have free development kits available.

As mentioned in the state of the art chapter, the iPhone has become a strong competitor to the dedicated gaming platforms, while the PDA range of devices has not been as lucky.

One of the advantages the iPhone has over PDA devices is the multitude of sensors the phone has built in. Accelerometers, touch screen, camera, microphone, luminosity detection, and proximity sensor, are just some of the features available to applications running on the iPhone. Later models include more sensors, such as a compass and a second camera. This means that by choosing the iPhone as development platform, there were many design options available for the test application. The low amount of public research available, the features available, and prior experience with the platform, made the choice of the iPhone an easy one.

Display choice

Choosing the iPhone as the platform meant that the choice of display was very limited. The iPhone has a 5,2cm by 6,4cm screen built in, but it is also possible to connect a projector to the device for a much larger image. The advantage of the projector is that it enables multiple users to watch what is happening on the phone at once, but as mentioned earlier this was not related to the research topic, so the internal screen was used alone.

Control interface choice

Due to the many sensors in the iPhone there were several possibilities for interacting with the device. A method used in several text-reading applications on the iPhone is to swipe with the finger across the screen to turn the page, as you would in a real book. This was considered for the text-based storytelling mode, but it would be an illogical way to interact with the NPC storyteller. To be consistent in the way the interaction was implemented, various touch gestures were considered as well as the tilt sensors in the phone. The choice came down to what was more logical: To tilt the phone to change page, or to tap the screen. As there are few to no applications that use tilting to flip between pages, the screen touch was the final choice.

3.3 Story design

This sub-chapter deals with the design choices made when creating the two stories for the test application.

Although thousands of video games are released every year, the genre diversity is very low. Just like in cinema and television most of the games fall rapidly under certain narrative genres like crime, fantasy, thriller or science fiction. The reason for this is that the game idea has to associate with activity or action in order to be enjoyed by the players, and the above mentioned genres are usually very action packed, (Sheldon, 2004). What generally makes games differentiate from one another is the story and the way the developers tell it through the gameplay.

The application created for this project is not a game as such, but the use of an NPC storyteller is almost exclusive to games.

When initially deciding on the storytelling elements that were to be used in this application, the previously mentioned eleven storytelling tools described by Österberg (2006) were considered. As the focus of this project is on text-based versus NPC-based storytelling, this rules out some of the tools, such as location evolvment. However, there were still multiple options for each of the two storytelling methods, and these are discussed below.

3.3.1 Text-based storytelling

The tools that are useful when developing this kind of storytelling are: Storytelling items, player journal, and plain text.

Storytelling items

These items are not necessarily text-based. They can just as well be an audio clip, video, or some other form of storytelling. For the purpose of this project though, only the text-based options were considered. There are many different ways a player could experience text in a game world, for example a book or computer screen. This type of item is simple to create and still gives the player the sense that he is not just reading a screen of text, but actually text which is inside something.

Player journal

As mentioned in the State of the art chapter, player journals is mostly used for retelling what the player has already experienced. The primary purpose is to act as a reference if the player forgets something that happened previously. In current games it is usually implemented as pure text, but due to the nature of the journal it would not work well for this project.

Plain text

This is perhaps the most simple of all methods of storytelling. The exact implementation varies between games, but the result is usually a screen or text field with the story inside. The difference between this and storytelling items, is that this plain text does not pretend to be inside a book or any other in-game item. It simply appears on the screen for the player to read.

For the application created for this project, the end result became a combination between storytelling items, and plain text. The screen displays a computer monitor which has a back and forward button to flip between pages of text. This is only partially a storytelling item, as the test participant never experiences the computer before or after it is used to display text. It is not seen in an environment, nor is it closely related to the stories it displays.

The text on the monitor is in the form of a journal written by an unseen person. This is not a player journal, as the player has not experienced the events himself, it is merely a method to make the stories easier to convert between pure text and NPC dialogue.

3.3.2 NPC-based storytelling

For this type of storytelling the relevant tools are: Pre-rendered video, in-game cutscenes, scripted sequences, ambient actions, character development, and interactive dialogue.

Pre-rendered video, in-game cutscenes & scripted sequences

These three tools are typically used for the same thing, but implemented in different ways. They all show a pre-made sequence of events that the player is supposed to watch.

For the purpose of NPC-based storytelling, they could be used to show one or more NPCs doing something that tells the viewer a story. The downside to two of the three tools is that they lack interaction with the NPCs. Only scripted sequences allow for the player to maintain control of his character, but not to interact with the NPCs during the sequence. Pre-rendered video and in-game cutscenes are even more problematic as they take control completely away from the player.

If the NPC storytelling in the test application was to be interactive at all, then these three tools were not the optimal choice.

Ambient actions

This tool is related to the three mentioned above, but are mostly used as a secondary storytelling tool to assist a more prominent tool. For example, NPCs could randomly make small talk with each other to further flesh out a story that is told through interactive dialogue.

Character development

Development of the character is a tool used to change the look or abilities of the character the player controls. It does not have to be related to the story, but for the purpose of this project it is only that part which is relevant. In the test application it could be implemented together with interactive dialogue to change the possible replies depending on what type of replies the player has chosen earlier. If his replies have been friendly, the options could become friendlier and friendlier, and vice versa. Like ambient actions, it is best used as a secondary storytelling tool.

Interactive dialogue

This is one of the most widely used storytelling tools in game genres such as role-playing games. It is important to distinguish between this tool and the other text-based tools. Interactive dialogue is typically implemented as a conversation between the player and an NPC, in which the player has different possible responses. The next statement made by the NPC can be influenced by the last reply given by the player, or all replies can lead to the same result as an illusion of interaction.

For this test application the interactive dialogue seemed as the best choice. It is the most interactive form of NPC-based storytelling, so it is ideal to compare to the non-interactive text-based storytelling. While it could be combined with ambient actions or character development, it was decided to avoid doing so to keep the implementation simpler in both creation and experience for the test participant.

3.3.3 Story creation

After the methods of storytelling had been determined, it was time to develop the actual stories the application was going to tell the user. It was decided that the application should tell two different stories, so the test participant was not presented with the same story twice in different forms. Which story is displayed with type of storytelling is decided by a random function in the application, and the order in which the two methods of storytelling appears is also random.

When deciding on the type of story a number of factors were considered. The two stories had to be similar in length and complexity, while still having significantly different content. Both stories also had to work in both pure text form, and when presented as NPC dialogue.

The first story created is a heavily modified version of a story called “Base escape”, which was created for a previous project. The story in that project was not experienced in text or dialogue form, so the general idea was rewritten from a first person perspective for this project. In short, it is the story of an unspecified person who has to escape an abandoned military base. Along the way he discovers that some otherworldly experiments have taken place in the building, and some of the creatures are still around. He makes contact with a naval captain on a nearby ship who leads him around the building to discover what has happened. At the end it turns out the experiments somehow got loose and killed everyone. As more experiments are close to escaping their basement prison cells, the captain decides to launch a rocket attack on the base and the main character narrowly escapes. The title of the modified version of story is simply “Ghost story”.

The second story was based on an older project titled “Space exploration game”. The project was not about storytelling, so the story was only hinted at through the gameplay of that test application. For this project the general ideas from the story were maintained, while more content was added to make the end result comparable to the ghost story. The story is titled “Space story”, and it is the tale of an untitled person who goes on a long journey through space as the engineer of a space ship. The entire crew is put into suspended animation while the ship travels to a new planet to establish a colony, but along the way the main character wakes up alone. He discovers that the ship has been damaged and is now in orbit around a star. To fix the ship he has to retrieve metals from a nearby asteroid and a new antenna for the navigation system. After collecting the materials in a shuttle he goes back inside the ship and discovers that because of the damage the ship has taken far longer to reach its destination and he has been in suspended animation for nearly 200 years.

The full text for both stories in both their NPC-based and text-based forms can be found in the appendix on page I.

3.3.4 Sound elements

In line with the limited gameplay implemented in the application, the amount of sound effects is also limited.

Early in the development a choice had to be made, as to whether or not the NPC should have a voice. Many games use a voice actor to let the NPCs speak their dialogue while it also appears in text. This is the case with the 2009 role-playing game (RPG) *Dragon Age: Origins*, and the 2010 action RPG “*Mass Effect 2*”. However, it is equally common to either have the NPCs say generic audio phrases while the dialogue appears as text, or to simply forego NPC voices altogether. This is how it is implemented in the massively multiplayer online RPG “*Warhammer Online*” (2008-present), “*World of Warcraft*” (2005-present), and the action RPG “*Numen: Contest of Heroes*”. Earlier projects showed that voice acting for NPCs can affect the experience of the player to a degree, but it was not tested how much influence it has. Because the focus of this project is storytelling with and without an NPC, it was decided to forego NPC speech and leave it as a topic for further testing in the future.

As mentioned in the interaction sub-chapter, feedback can help the user better understand what he is doing and how the application works. For this purpose a generic “beep” sound was added to the application. It is used in the text-based part of the application when the user clicks one of the two buttons to go back or forward in the story. It is also used in the NPC-based part when the user selects one of the dialogue options. The purpose is to provide feedback to the user, so he knows when he has clicked on something that is interactive.

3.4 Graphical implementation

There are a number of different ways of implementing graphics in a game, both in terms of gameplay as well as storytelling. In the application created for this project, there was no need for advanced gameplay-like graphics, as the focus was on the storytelling. This meant that though the iPhone supports OpenGL ES, a framework for creating 3D graphics, the choice fell on the simpler 2D graphics.

Initially it was considered to reuse parts of an application from a previous project, which has a military base the player can walk around in. In this environment he could then find the two things that initiate the storytelling segments of the application. However, a lesson learned in that project was that people often had trouble navigating the building, knowing which items were usable, and often ended up going in circles to find what they needed. As this project is not about any of those game mechanics, it was felt that it would be mostly pointless to force the test participant to play through it. Instead, the implementation focuses solely on the storytelling segments.

As the application loads, the test participant is presented with a welcome screen, and then he immediately is presented with the first story. After the story, a screen is shown to indicate that there is going to be a break between the two stories. Finally the second story is shown, and followed by a screen thanking him for his participation.

The reduction of elements unrelated to the storytelling, means that the focus of application is on exactly what is necessary for this test.

Because the goal of the test is to compare NPC-related storytelling to storytelling that does not use NPCs, it was necessary to develop two ways stories could be told in the application. As mentioned earlier, the two stories were to be told through a dialogue with an NPC and by reading simple text on an on-screen monitor. Examples of these two storytelling implementations can be seen in figure 8 to the right.



Figure 8: The two storytelling modes

The first storytelling method to be implemented was the simple on-screen monitor. The graphic itself was originally created for a previous project and later redesigned to fit the iPhone. In the prior projects, the test participants understood that the image was supposed to represent a monitor they were looking at in a virtual world, which is why it was chosen for this project.

The image was further modified to include the two large "previous" and "continue" buttons that can be seen in figure 8, and to remove a number of smaller purely decorative buttons. By only having two clearly labelled buttons on the screen, there is no risk of confusion in regards to their functionality.

After entering this screen the test participant is immediately presented with the first part of the

story. The stories are split up into 8-9 pieces, each of which are accessible by simply clicking the "continue" button. The participant can use the "previous" button to return to any prior point in the story at any time.

The text on the screen is written in size 14 and the font is Arial. This is a common size and font for iPhone applications, and hand-held applications in general. A decrease in font size was considered, to reduce the amount of pages the test participant had to go through, but this resulted in the text being hard to read without straining the eyes.

The second, more complicated, storytelling method was the NPC-based one. The first choice that had to be made was the design of the NPC itself. As mentioned in the previous chapter the two stories are very dissimilar, so either two different NPCs were required, or one with a neutral look so it could be used for both stories. This led to the character design process described in the next subsection.

3.4.1 Character design and animation

In a previous project, some years ago, a character was created that fit the neutral look description. The story was about a Wile E. Coyote-like character who got himself into trouble while trying to escape a wrongfully imposed prison sentence, and so the character of "P" (short for "protagonist") was born (final design of P from that project, can be seen in figure 9). This character had a neutral appearance, and as such would be useful as a storyteller for either story.

The P used in the previous project was mostly seen from a distance, and thus did not need any facial animations. The only animation the character had was a simple walk with swinging arms. For this project the character did not need to walk anywhere, but he did need to have realistic facial animations (for a cartoon character).

In "The Illusion of Life" Thomas et. al. (1981) describe 12 basic principles of animation based on the work of leading Disney animators from 1930-1980. Not all the principles can be applied to the application created for this project, as they describe techniques used for other types of animation. However, there are five which apply: Staging, secondary action, exaggeration, solid drawing, and appeal.



Figure 9: Original design of P

Staging

The principle of staging is commonly used in many kinds of visual media, and computer games is no exception. It is the way in which a scene is created so that the viewer can immediately understand what the idea is. In the application created for this project it is important to distinguish between the staging of the story and the staging of the NPC interaction. The NPC himself does not visually tell much of the story, which makes staging less important. However, staging is very important in the way that the test participant will see the NPC and the dialogue box and immediately understand that he is talking to the NPC and that the NPC is "alive" because of his animations.

Secondary action

Secondary actions are things that happen in relation to what the viewer is expected to focus on. For example, if one wants to animate a helicopter landing that would be a primary action. The

secondary action could be rocks and dust being thrown into the air, people moving their heads to follow the motion of the helicopter.

In this test application there are relatively few animations, but there are some minor secondary actions. When the NPC dialogue changes, the NPC has a talking animation, which purely serves to illustrate that the NPC is saying the words that appear in the dialogue box.

Exaggeration

This is a principle used in cartoons to make the characters and situations appear less realistic and dull. The amount of exaggeration applied depends on the desired effect. The style of the original P was reminiscent of the Japanese anime style known as “super deformed”. This worked well for a comedy character, but it is a little too exaggerated for a less humorous application.

The new version of P was given a body and mouth of a more realistic size. However, to preserve his cartoon-like style, he still had disproportional eyes, nose, and hair.



Figure 10: Final version of P

Solid drawing

This principle is mainly about the animator understanding how to put together a character.

What the anatomy of the character should be like, understanding the proportions of the figure, etcetera.

As P was an already existing character, this principle was less important to this project, but it was still something to keep in mind when making the necessary changes to the character. Particularly it is important to not make the character too symmetrical as this can make it look odd and lifeless.

Appeal

The principle of appeal is what makes the character seem real and keeps the attention of the viewer. If a character is not appealing to the viewer it can create problems for the context in which he is used, as the viewer may begin to focus on other things that are unimportant.

In the test application for this project, the appeal of the character is closely related to the exaggerations mentioned above. By having a character that doesn't look quite human, it makes him interesting for the participant to look at and interact with.

Character images

The original version of P had a large moustache to make him appear more comical. It also made his mouth rather hard to see though, which is why it was removed for his new appearance. Similarly, his body was redrawn to be more proportional to the size of his head, and to replace his prison clothes with a simple blue shirt.

The mouth, eyes, and eyelids of P were removed temporarily, as these were the parts that needed animation. The new basic NPC model was placed on a simple background recycled from a previous project, and the end result can be seen in figure 10.

The facial animations are handled in three different ways, which will be described in more detail in the next chapter. The eyes move around randomly between six different positions, while the eyelids is a sequence of four images (figure 11). The mouth animation has two states. The first state is a random change between three images, which is used when the NPC is idle. The second state is a semi-random change between three images, which is used when the NPC is “talking” (figure 12).



Figure 11: Eyelids animation frames

The images all have transparent backgrounds so they can be superimposed on the basic NPC model inside the application. The eyelids image is on top of the eyes image, so the latter won't have to be hidden by the application every time the eyes close.

The purpose of randomizing the eyes and the mouth when idle, is to mimic some of the behaviour a real person might exhibit when standing around waiting for a reply (which is the way the story is implemented). An example of the completed NPC can be seen in figure 8 on page 27.

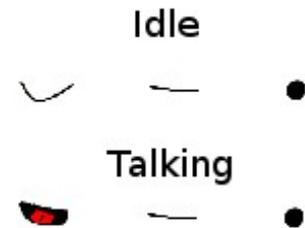


Figure 12: Mouth animations

3.5 General implementation

The test application for this project is implemented as a view-based application, which means that the interface is created as a series of "views", handled by other classes.

In a standard view-based application there will normally be multiple views which each represent one screen or area of a screen. These files are used to handle everything that happens in a view, such as moving objects around, changing pictures, or handling touches. Each view will normally only control one screen of information or objects, though it can contain methods to replace its own content.

The views are controlled by a number of view controllers. The purpose of the view controller is to manage the views in a specific area of the screen, which includes replacing one view with another when necessary. Each view controller can control any number of views, and there can be many controllers in an application.

The implementation consists of two types of files: The .m and .h files which contain the code of the application, and the .xib files which contain the graphical functionality and layout. The `tellstoryAppDelegate.h`, `tellstoryAppDelegate.m`, and `MainWindow.xib` are all general application files which contain functionality that should be executed when the application launches or shuts down. `MainWindow.xib` handles the application window in general, and it is responsible for loading the .xib which contains the screen positions of the 4 images used in the application, as well as their layered order. This file is called `tellstoryViewController.xib`. Both .xib files are created through the Interface Builder tool from the iPhone SDK package.

The appdelegate file normally contains methods that should be executed when the application launches or quits, which the test application does not require. It is also the place to declare things that need to be available all the other classes, for example if data needs to be saved before the program shuts down. In the test application it only handles the creation of the view controller reference, and the destruction of it when the application shuts down. The view controller reference needs to be accessible from the other classes, as it is the connection between the code and the .xib files.

The application also contains a `main.m` file that is used to create an autorelease pool, which is a way of storing objects which should be destroyed in the near future. It is not used in this test application, but it is a required function in an iPhone application, as it is may be used "behind the scenes" by objects that inherit functionality from Apple classes.

In the test application created for this project there is only one view and one view controller. This simplifies the development as all the code can be stored inside the `tellstoryViewController.m` and `tellstoryViewController.h` files. Normally it would be stored in `view.m` and `view.h` which would be handled by the view controller, but since the controller never needs to switch views, there is no need to expand the code outside of the view controller itself.

In the `tellstoryViewController.m` there are eleven methods which fall into the following four categories: Initiation & progression, interaction, animation, and story. Each of these categories will be discussed in detail below.

3.5.1 Initiation & progression methods

When the application has finished loading the `viewDidLoad` method is automatically called by the operating system. This method is the ideal location to initialize any variables that are used within the view that is being displayed. The advantage of using this method over an `init`-method is that this method will not be run if the application fails to load properly, which could happen if the iPhone was to run out of memory during the load.

	Story 0	Story 1
Type 0	NPC ghost story	NPC space story
Type 1	Text ghost story	Text space story

Figure 13: Type-Story relations

```
currentStory = (arc4random() % 2);
currentType = (arc4random() % 2);
```

Figure 14: Random number generation

Inside the method all important global variables are set to their default values. A few other operations are also handled here, most notably the random number generation (figure 14). This code creates two random integer values between 0 and 1 (both included) and assigns each of them to a variable. The `currentType` variable determines if the test participant will get the text-based or NPC-based storytelling method first, and the `currentStory` variable determines if the ghost or space story will be first. The possible options can be seen in figure 13 above. Once a combination has been displayed, the second story will be the story and type not used yet.

The `viewDidLoad` method is also where the "beep" audio clip is loaded. It is necessary to load an audio file before it can be played, and doing so can take a few seconds, so it is a good idea to do it before it is needed.

The AVFoundation framework is a set of functionality to playback audio in iPhone applications. This application uses the `AVAudioPlayer` from the framework to handle the audio file. An example of the statements used to set up the sound can be seen in figure 15.

```
filePath = [[NSBundle mainBundle] pathForResource:@"arrp" ofType:@"wav"];
fileURL = [[NSURL alloc] initWithFileURLWithPath:filePath];
beep = [[AVAudioPlayer alloc] initWithContentsOfURL:fileURL error:nil];
beep.volume = 1;
[beep prepareToPlay];
```

Figure 15: Handling of sound

The first two statements are used to create an address of the file that is meant to be played. This address is then handed to an `AVAudioPlayer` object called `beep`. The volume variable is a way to control the output levels of the individual sounds, without having to edit the audio file. It is not required if the sound is to be played at full volume (value = 1), but it is considered good programming style to include it anyway. Finally the `prepareToPlay` method on the `AVAudioPlayer` will ask it to load the file and be ready to start playing it when asked to.

At the end of the `viewDidLoad` method a second method called `changePic` is activated. This method handles all things related to the progression of the events in the app. It contains two switch-case statements. The first is based on

the flow of the app: Welcome screen – First story – Pause screen – Second story – End screen. When exiting a screen the `appProgress` variable is incremented by one. The switch-case method uses this variable to determine which step is the current. Case 0, 2, and 4 are nearly identical. The code for case 2 can be seen in figure 16.

```
case 2: //Status between stories
    sceneStatus = 1;
    screenLabel.text = @"Time for a break before the next story";

    //Switch to other type and story
    if (currentStory == 0) { currentStory = 1; }
    else { currentStory = 0; }

    if (currentType == 0) { currentType = 1; }
    else { currentType = 0; }

    dialogueStep = 1;
    break;
```

Figure 16: First switch-case in `changePic` – Case 2

The `screenStatus` variable is used by the second switch-case statement in the method to set the background image. In these three cases it is being set to a solid black colour. The `screenLabel.text` variable is a text box that appears in the middle of the screen and is used to send messages such as “welcome” and “thank you” to the test participant.

Unique to case 2 are the two if-else structures. As mentioned earlier, the `currentStory` and `currentType` variables determine which combination of story and storytelling method should be displayed, and in this case the participant has been presented with the first story, so the values are being changed to their opposites.

Case 1 and 3 are also nearly identical. They are used when a story should be displayed on the screen. Case 1 can be seen in figure 17.

First the `screenLabel` variable is set to nothing to clear the text from the screen. Depending on the type of story to be shown, the `sceneStatus` variable is set and a method is called. Each story type also has a unique method that is called to control what text appear on the screen. These methods are called `setNPCDialogue` and `setScreenText`.

The second switch-case structure in the method is used to handle the graphics on the screen by using the `sceneStatus` variable. There are three cases which handle the status screens, the text-based story, and the NPC-based story. Figure 18 shows case 2 and 3 of the switch-case structure.

Case 2 is used when the NPC-storyteller is going to be used. First the background image is set to be `scene.png`, which is an image of the NPC with no eyes, eyelids, mouth, or text. The alpha of the three other images on the screen is set to 1; making them visible. The `canBlink` and `canEyesRandom` variables are used by the animation methods to determine if animations are necessary on the images currently being displayed. This is implemented to save program resources when the text-based story or the status screens are being displayed.

Finally the animations are started by calling the `characterBlink` and `characterEyesRandom` methods. The

random mouth animations are not activated at this time, as text will be displayed very shortly after this code is executed and a talk animation is played at that time.

Case 1 and 3 are very similar, in that they both deactivate all NPC animations and set the alpha of eyes, eyelids, and mouth to 0 to hide them. The difference between the two cases is in the background image that is loaded. Case 1 merely loads a black image, while case 3 loads the

```
case 1: //First story
    screenLabel.text = @"";
    if (currentType == 0) {
        sceneStatus = 2;
        [self setNPCDialogue];
    }
    else {
        sceneStatus = 3;
        [self setScreenText];
    }
    break;
```

Figure 17: First switch-case in *changepic* - Case 1

```
case 2: //NPC story
    bgpic = [UIImage imageNamed:@"scene.png"];
    eyelids.alpha = 1;
    eyes.alpha = 1;
    mouth.alpha = 1;
    canBlink = 1;
    canEyesRandom = 1;
    [self characterBlink];
    [self characterEyesRandom];
    break;
case 3: //Text story
    canBlink = 0;
    eyelids.alpha = 0;
    eyes.alpha = 0;
    mouth.alpha = 0;
    canEyesRandom = 0;
    canMouthRandom = 0;
    screenLabel.textAlignment = NSTextAlignmentLeft;
    bgpic = [UIImage imageNamed:@"scene_screen.png"];
    break;
```

Figure 18: Second switch-case in *changepic*

background for the text-based story.

3.5.2 Story methods

As mentioned above, there are two similar methods that control what text is displayed on the screen. The `setNPCDialogue` method is used in the NPC-based storytelling. This method is implemented as a switch-case structure with a counter that is either incremented or decremented as the user navigates through the dialogue options. An example of the last two cases of the structure can be seen in figure 19.

The counter starts at 0 and on most screens the user has the option to continue with the story or go back. If he decides to continue then the counter is incremented, the `setNPCDialogue` method is called again, and the text on the screen is replaced. The `NPCLabel` variable is a text field that is placed on top of the dark green area next to the NPC, and this is where everything the NPC says, and the possible responses are shown.

When the counter has been incremented to 12 or 13, depending on the story, it will reach a special case. This case removes the text from the screen, increments the `appProgress` counter and calls the `changePic` method. As this happens, the story is over and `changePic` will replace the screen contents with the status screen.

```
case 12:
    NPCLabel.text = @"Eventually the ship was
repaired and we managed to finish the journey without
any more problems. We got to our destination and
started a colony, but we have yet to get into contact with
Earth.\n\n\n\n\n\n- Thanks for the story\n\n- Please
repeat";
    break;
case 13:
    NPCLabel.text = @"";
    appProgress++;
    [self changePic];
    break;
```

Figure 19: Part of the `setNPCDialogue` method

The `setScreenText` method, for the text-based storytelling, works very similarly to `setNPCDialogue` with a couple of important differences. The text for each story is distributed in fewer chunks, as the text area is a little larger and there is no need to have room for dialogue options. It is not a big difference though, as the stories should not feel significantly different in length between the two storytelling methods.

A more important difference is the fact that this method uses a different label to display the text on the screen. This was necessary since the label used for NPC dialogue is off to the side and would make the story text appear strange when in the text-based storytelling mode. For this reason, two methods were created, so the different functionality would not be mixed up by accident during the development.

3.5.3 Interaction methods

The control interface is handled by three methods: `touchesBegan`, `touchesMoved`, and `touchesEnded`. These methods are called automatically by the operating system, when a touch (on the screen) is detected while the application is active. The iPhone registers touches as a continuous series of events. At the first event the `touchesBegan` method is called, and at the last the `touchesEnded` is called. During the touches, if the coordinates of two touches is significantly different, the `touchesMoved` method is called.

This application only requires the detection of single taps on the screen and no swipes, but by implementing the `touchesMoved` method anyway, it makes it possible to cancel a selection if you keep your finger on the screen and move it away from the thing you were about to click on. This is common functionality across many applications for the iPhone as well as regular PC applications. The `touchesBegan` method is purely used to store the coordinates of where the user touched the screen in the variable `gestureStartPoint`.

In the `touchesEnded` method the results of the touch are handled. The method consists of several

nested if-else and switch-case statements. First a check is made to determine what image is currently being displayed. If it is the first or second status message, the appProgress counter will simply be increased and changepic is called.

However, if it is a story image, a second check is made to determine which story and type it is. As the dialogue options are different in location and length for most of the NPC dialogue, each piece of dialogue has a set of unique coordinates. When a touch is detected on an NPC dialogue is detected, it will be ignored unless the touch was within one of two sets of coordinates. The first set of coordinates represent the option to continue the story, while the second is the option to go back. This check can be seen in figure 20.

If the touch is within the bounds, the dialogueStep counter is changed as appropriate. The beep sound is played, and setNPCDialogue is called to change the text.

Touches in the text-based story mode is handled later in the same method, in a similar way. Since this screen always has the “previous” and “continue” buttons in the same location, the same coordinate check can be used regardless which part of which story is displayed currently. The beep sound is still played, and the setText method is called.

```

else if ( appProgress == 1 || appProgress == 3) {
    float clickX = gestureStartPoint.x;
    float clickY = gestureStartPoint.y;
    if (currentType == 0) {
        if (currentStory == 0) {
            switch (dialogueStep) {
                [...]
            case 3:
                if (
                    clickX <= 74 &&
                    clickX >= 43 &&
                    clickY <= 331 &&
                    clickY >= 245) {
                    [beep play];
                    dialogueStep++;
                    [self setNPCDialogue];
                }
                else if ( clickX <= 41 &&
                    clickX >= 0 &&
                    clickY <= 450 &&
                    clickY >= 245) {
                    [beep play];
                    dialogueStep--;
                    [self setNPCDialogue];
                }
            }
        }
    }
}

```

Figure 20: Touch handling

3.5.4 Animation methods

The animations for the NPC is handled in four methods called characterBlink, characterEyesRandom, characterTalk, and characterMouthRandom.

In the characterBlink method, the display of the eyelid images is controlled. As mentioned earlier, there are four different eyelid pictures that are displayed in sequence. Two switch-case statements handle the images and the time between the changes. An example of this can be seen in figure 21.

The state the eyelids are changed to depends on the state they are currently in. In the example the eyes are completely open. First the alpha is set to 1 to make sure the eyelids are visible. Then the eyelid picture is changed to the next in the

```

switch (eyelidStatus) {
    case 0: //Eyes are open
        eyelids.alpha = 1;
        eyelidsPic = [UIImage imageNamed:@"scene_eyelids_closing25.png"];
        eyelidStatus = 1;
        break;
    [...]
    switch (eyelidStatus) {
        case 0:
            [self performSelector:@selector(characterBlink) withObject:nil afterDelay:4];
            break;
    }
}

```

Figure 21: Blinking animation code

sequence, which in this case is the eyelids being 25% closed. In the second switch-case statement a special method call is made. When a method is called in this manner it is possible to insert a delay

before the call happens. In the example that delay is 4 seconds, which means that when the character opens his eyes he will keep them open for 4 seconds before blinking. The time between the images depend on which image is currently displayed. The eyes stay open for 4 seconds, then each animation frame of the closing eyes is displayed for 0.08 seconds, followed by the eyes being completely closed for 0.3 seconds.

The method will continue to call itself as long as the `canBlink` variable is set to 1. This variable is set to 0 when the screen is changed to something that is not the NPC storyteller.

The `characterEyesRandom` method is very similar to the `characterBlink` method. This method randomly changes the direction the character is looking in. A random integer value between 0 and 5 is generated, and the eyes image is changed to one of 6 possible. The image is changed every 5 seconds, but it is possible that the same image will be chosen twice in a row.

Finally the `characterTalk` and `characterMouthRandom` are closely related. The `characterMouthRandom` method works exactly the same as `characterEyesRandom`, except that only three images are possible, and the change occurs every 10 seconds.

The `characterTalk` method is somewhat more complicated. First any random mouth changes are stopped by setting the `canMouthRandom` variable to 0. Then a random number is generated, and a check is done to make sure it is different from what was last generated. This number is then used to decide between three images which represent the mouth of the character when he is talking. The method then calls itself with a delay of 0.25 seconds; making the image changes much faster than in `characterMouthRandom`. A counter is used to prevent the method from calling itself more than 30 times, making the talking animation 7,5-8 seconds long (depending on the execution time of the code). When the method is done looping, it will set `canMouthRandom` back to 1 and call the `characterMouthRandom` method to resume random mouth changes. The `characterTalk` method can be seen in figure 22 below.

```
- (void) characterTalk {
    canMouthRandom = 0;
    talkMouthAnimation = (arc4random() % 3);
    while (talkMouthAnimation == previousTalkMouthAnimation) {
        talkMouthAnimation = (arc4random() % 3);
    }
    switch (talkMouthAnimation) {
        case 0:
            mouthPic = [UIImage imageNamed:@"scene_mouth_open.png"];
            break;
        case 1:
            mouthPic = [UIImage imageNamed:@"scene_mouth_open2.png"];
            break;
        case 2:
            mouthPic = [UIImage imageNamed:@"scene_mouth_neutral.png"];
            break;
        default:
            break;
    }
    mouth.image = mouthPic;
    if (characterTalkCounter < 30) {
        characterTalkCounter++;
        [self performSelector:@selector(characterTalk) withObject:nil afterDelay:0.25];
    }
    else {
        canMouthRandom = 1;
        [self characterMouthRandom];
    }
}
```

Figure 22: The `characterTalk` method

4. Results

This chapter describes what research method was used for the project, as well as how the testing was conducted. At the end of this chapter the results gathered in the test will be presented.

4.1 Research method

In previous projects both qualitative and quantitative research approaches have been used. The qualitative approach makes it possible to delve deeper into the subjective opinions and experiences of the test participants, in a way that a typical quantitative approach would not. By spending longer on interviewing a small sample of people, it is possible to get to facts that would have been concealed in quantitative questionnaires given to hundreds of people. In similar tests for previous projects, the qualitative approach worked very well, which further inspired confidence that it would also be the correct choice for this project.

The research topic was the experience of different storytelling tools in a game. The test participant was to experience two of these tools: Plain text, and NPC dialogue.

The goal was for the participant to experience both stories and both storytelling tools, but in a random order to prevent any erroneous conclusions due to the tools always being experienced in a certain order. The application automatically picks a random set of story and tool at launch, and picks the opposite combination for the second story. This ensures that the test participant experiences both stories and tools, so he is able to comment on the difference between them.

The test sample was four people, all of whom regularly play video games. Two estimated they spent an average of 15 hours a week playing games, one estimated 10 hours, and the last said he spent an average of 2 hours per week. This test sample provided a decent spectrum of time spent on gaming, which helps make the results more generally applicable amongst gamers.

The participants were all male, and between the ages of 25 to 28. Unfortunately it was not possible to find any female participants. Previously conducted tests have not shown any significant difference between male and female in regards to experiencing a story in a game environment, which leads to the expectation that it would not have made a major difference for this project either. The people were tested and interviewed individually, and did not get a chance to interact with each other during the tests and interviews. They were also not informed of any detail of the application and test before they were alone.

During the test the participants were encouraged to make comments about their experience with the application and stories, but they were not allowed to ask the observer questions of any kind.

A semi-structured interview was conducted with each participant after he had finished the first story, and again after the second story. The interview questions were formed to cover three main areas: How well the participants could understand the stories. What their experience of interacting with an NPC or an on-screen computer monitor was, and finally how well they could interact with the application itself. As Davies (2007) suggests, you should always avoid asking the research question directly. Instead the interview was conducted as a dialogue about their experience with the application. If an area was not covered by the test participant of his own accord, the prepared questions were used to guide him towards the relevant topics.

The questions themselves were based on questions used for another qualitative test, in a previous project titled "Base Escape!", which were originally based on a presence questionnaires made by Witmer and Singer (1998). However, those questions were merely used for inspiration, as this test is on a different topic. From the Witmer and Singer (1998) questionnaire the control factors were used as inspiration for some of the interaction questions for this test, while the Base Escape questions were used as inspiration for the questions related to storytelling.

During each test and interview, notes were taken on noticeable things the test participant did within the application, as well as his responses to the questions. At the end of the testing phase of the project, these notes were analysed and the result of this analysis can be found later in this chapter.

4.2 Target group

When doing research it is important to select a target group, or test sample, that is appropriate for the research subject. In the case of this test, a sample was selected amongst university students and those of similar age. As a qualitative test approach was chosen, a small sample of people were tested. The criteria for choosing the sample, was based on criteria used in previous projects to test other aspects of gaming. The people should have experience in using computers, and gaming in general. It was not a requirement that the amount of experience be great, but they should at least know some of the basics. The reason for this criteria is that the test is focused on storytelling in games, and people with gaming experience will be able to provide more feedback. The objective was to observe how these test participants understand and like or dislike stories told by the two different methods.

To determine gaming experience, several factors were considered: The amount of games and gaming devices (consoles, computers, etc.) owned, and the number of hours per day/week spent playing games or searching for game related information (e.g. on forums or gaming news websites). General computer knowledge refers to experience in how a graphical interface works, operating systems, input/output devices, mobile devices, different hardware/software, programming skills, and computer related education.

It was not necessary that the test participants had experience in all the areas; the factors were merely used to attempt to determine the skill of the participant.

A total of four participants were chosen, all of which had a moderate to high level of experience with computers, and from a low to a high level of experience with gaming in particular. Two of the participants had used an iPhone before, and a third had experience with other hand-held touch based devices. Two also had experience with other hand-held gaming platforms, such as the Nintendo GameBoy and the Sony PSP. This provided a decent spectrum of gaming experience. The purpose of having people with different experiences was not to draw conclusions on the individuals, since one person can't be representative for all people with the same level of experience. Instead, the reactions and the opinions of the individuals were observed, after which they were analysed as a group.

4.3 Test goal

The purpose of the test that was conducted, was to examine how different storytelling tools could affect a player's understanding of the story in a hand-held game application, as well as which type of tool they prefer. Additionally, the test application should be evaluated to make sure the stories and interface worked as intended.

The main questions were:

- Can the test participants understand the two stories?
- Is there a difference in the level of understanding between the two storytelling methods?
- Does the NPC storyteller make the test participant feel differently about reading the story?
- Does the controls and stories work similarly for both storytelling methods?

4.4 Test setup

The test application itself only requires an iPhone, and therefore it is very simple to set up. The software can run on any iPhone, iPad, or iPod Touch. However, it has only been tested with a first generation iPhone. A potential problem with the iPod Touch may be a lack of CPU power, which could lead to slowdowns when running the application. Additionally, the iPad has a much larger screen than the iPhone, which may affect the quality of the images and fonts. It would be necessary to run the application multiple times before attempting to perform the test on a non-iPhone device. Later models of the iPhone have more CPU power, more memory, and slightly higher screen resolutions, none of which should present a problem for the test application.

The application has been developed for, and tested with, iPhone OS 3.2. It has also been tested with iPhone OS 3.0 in the iPhone Simulator, and both versions of the OS ran the app with no discernable difference. Presumably the same will be the case for iPhone OS 3.1, but there have been significant changes to the OS in version 4.0, which may require the application to be modified if it is to be used on such a platform.

The compiled application provided on the accompanying CD, will only run in the iPhone Simulator. In order to run the application on a real iPhone, there are two solutions. The first possibility is to use a device which has been modified by a process known as "jailbreaking". This allows the installation of a program called AppSync which, when combined with the jailbreaking, will allow the phone to receive and execute unsigned code.

An easier approach is to register for the Apple Developer program. Once complete, it is possible to set up a number of "provisioning" devices for which you can compile applications. This is the approach that was used to conduct the testing for this project. The provisioning devices are identified specifically by their serial numbers, which is why it has not been possible to create a generic compiled provisioning version to include on the CD.

Once the application is installed, the phone should be placed in "Airplane Mode" to block any incoming calls or SMS messages from interfering with the test. If the test is run on OS 4.0 or on a jailbroken device with the "Backgrounder" application installed, it is important to shut down all applications running in the background. If other applications are allowed to run in the background during the test, it could result in the test participant experiencing slow changes between images, slow animations, delayed sounds, or unresponsive controls.

The mute button on the side of the phone needs to be turned off for sound to play in the application, and the volume buttons control the application volume. Headphones can be plugged into the iPhone if the test is performed in a noisy environment, but they were not used for any of the tests performed for this project.

The iPhone has an illuminated screen, so it does not suffer from performance issues when there is a reduced amount of light. However, too much light can be an issue. The tests were performed on sunny days, so care was taken to block out bright light as much as possible, and to place the test participant in a position where lamps and outside light would not be reflected in the blank screen.

Before the start of the test phase, various ways of recording the test on video were examined.



Figure 23: View from the camera

In previous projects, recording a handheld device has proven problematic. The main difficulty is in how to capture the screen. It proved too difficult to mount the camera above the test participant to capture the screen, as the camera would either be unable to capture the small screen properly, or require the user to sit very still to keep the screen within the field of view of a camera zoomed in. In the end, the test was set up with a camera filming the test participant, and an observer next to the participant taking notes. The observer was seated slightly behind the test participant, so as to not disturb the test. It was the task of the observer to note any problems with the interface, as well as the general behaviour of the test participant. An illustration of the test setup can be seen in figure 24 to the right, and figure 23 on the previous page shows the view from the camera.

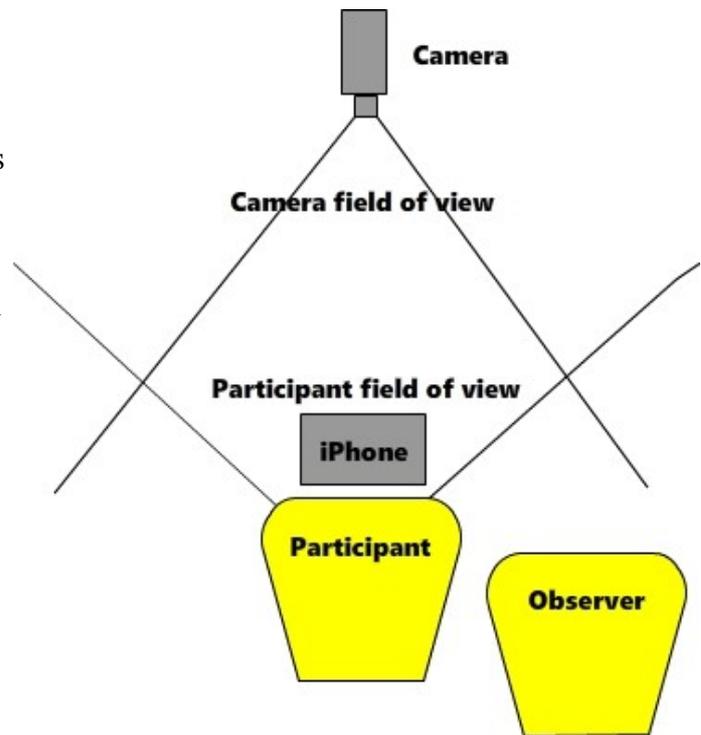


Figure 24: The test setup

4.5 Test process

Before the test started, the participant was given an explanation of the purpose of the test, but no mention was made as to what to expect or what positive and negative effects each mode of storytelling might have. He was told that the application was controlled by touch, and that he would be able to move forwards and backwards in the story. However, he had to discover how this was implemented on his own.

After this he was given the iPhone with the application started and left on the welcome screen. After making himself comfortable, the first play through would begin. As mentioned previously, the application randomly chooses a story and story mode, so the exact combination was unknown even to the observer until the test participant began the first test.

It was then up to the participant to read the story through to the end, either by flipping through the pages of the virtual computer monitor or by picking the right dialogue options when talking to the NPC. He was allowed to go backwards and forwards as many times as he liked, until he felt he was done reading the story. On the virtual computer monitor the buttons that control the story are clearly labelled with "continue" and "previous", but this is not the case for the NPC storyteller. In that mode, the navigation function is slightly obscured by being controlled by dialogue options. There are always two options, one of which makes the story go forwards and the other backwards, but the text on the options change from screen to screen. The test participant was not informed about any of this apart from the fact that it was possible to go back and forth as he wanted to. Instead he had to realize what each dialogue choice did, and pick the one appropriate to what he wanted to do. Once the end of the first story was reached, the test participant was interviewed regarding his understanding of the plot, and feelings about the interaction.

Following this short break, the second play through would begin. This time with the storytelling mode and story that were not used in the first test. This play through was conducted exactly like the first. When done with the second story, the participant was interviewed again. He was asked questions similar to those asked after the first test, and some additional questions about which storytelling method, control method, and story he preferred.

The structure of the interview was based on a structure used in previous semesters, which was originally adapted from both Witmer & Singer (1998), and Brockmeyer, Fox, Curtiss, Mcbroom, Burkhardt, and Pidruzny (2009). These are both questionnaires and thus not originally intended to be asked in an interview. The focus of the questionnaires is also somewhat different. Brockmeyer et al. (2009) is a work on producing a Game Engagement Questionnaire to evaluate how engaged users are in computer games. Witmer & Singer (1998) is a work on measuring presence in a virtual environment. Storytelling and the experience of playing a game is a part of both, but it is not the main topic. This is part of the reason why the questions were not taken directly from the questionnaires, but rather they were used for inspiration when creating the questions for the interviews conducted for this project.

The interview structure can be seen in figure 25 on the next page. It is important to note that not all of these questions were asked directly, and the interview was not limited to the questions. The test participant was allowed to speak freely about anything related to the application, story, and test. The questions were used as a reference to make sure the necessary topics were covered by the interview. The questions were not shown to the participant at any point before or after the test.

NPC-based storytelling:

- Understanding of the story
 - Could you briefly explain the story
 - What characters were in the story?
 - Who was the main character?
 - Was the main character able to successfully complete his goals?
 - (ghost) What do you think the storyteller was doing in the base before the attack?
 - (space) What do you think had happened to the ship?

- Interacting with the NPC
 - What was your opinion of the NPC?
 - How did you feel about him as a storyteller?
 - Did it seem believable that he would be the main character of the story he told?
 - Did you feel you were having a conversation with him?
 - Did you like talking to him?

- Application interaction
 - How easy/hard was it to use the dialogue options?
 - Were you satisfied with the possible dialogue choices?
 - Did the interaction with the NPC seem meaningful, or more like a tool?

Text-based storytelling:

- Understanding of the story
 - Could you briefly explain the story
 - What characters were in the story?
 - Who was the main character?
 - Was the main character able to successfully complete his goals?
 - (ghost) What do you think the storyteller was doing in the base before the attack?
 - (space) What do you think had happened to the ship?

- Interacting with the on-screen computer monitor
 - What was your opinion of the virtual computer monitor for reading a story?
 - Did it seem believable that it was a computer screen you were looking at?
 - Did it seem believable that the story would be on a computer screen?
 - How did you feel about reading a story in this manner?
 - Did you like reading the story in this way?

- Application interaction
 - How easy/hard was it to use the buttons to navigate the story?
 - Did the interaction seem meaningful?
 - Were you satisfied with the interface?

After both stories:

- Which of the two storytelling methods did you prefer?
- Did you prefer one story over the other?
- Were the controls easier or harder to use in one of the storytelling methods?

Figure 25: The interview questions

4.6 Test results

This sub-section presents the results and observations made during the test. As the application randomly picks a story and story mode upon start-up, the order and combinations was not the same for all participants. Figure 26 shows a chart with the combinations and the time spent on each story.

	Participant 1	Participant 2	Participant 3	Participant 4
First story type	Text	Text	NPC	NPC
First story content	Ghost	Ghost	Space	Ghost
First story time	2min 51sec	3min 56sec	3min 33sec	3min 42sec
Second story type	NPC	NPC	Text	Text
Second story content	Space	Space	Ghost	Space
Second story time	3min 28sec	4min 20sec	2min 35sec	3min 4sec

Figure 26: Story order and time spent

The text of the space story was one paragraph longer than the ghost story. This resulted in both the NPC and text implementations of the story being one screen longer, which in turn resulted in the average time spent on the space story to be 20 seconds more than the ghost story.

Both modes of the same story contain exactly the same amount of text though, so one could suppose that the time spent would be roughly the same, but that is not so. Interestingly, the participants spent an average of 225,75 seconds on the NPC story, but only 186,5 seconds on the text-based story. Three of the four participants got the space story from the NPC, which could explain 20 of the extra seconds, but that still leaves an additional 20 seconds spent on the NPC. The participant who did not get the space story from the NPC, still spent a full 38 seconds more on the NPC, despite the story being shorter.

The following is a list of the topics discussed during the interview, as well as any observations made during the play through.

The first set of topics are about the understanding of the story. These topics were discussed after each of the two storytelling modes.

Could you briefly explain the story?

There was no great difference in the understanding of the story between the NPC- and text-based storytelling modes. In both cases, the participants were able to give detailed descriptions of the stories they had read. Of the four participants, two had some omissions in the NPC-based story. One person said that he did not understand the part about where the space ship was going, and the other didn't mention anything about the problems the ship had experienced. Generally the summaries of the text-based stories were more concise, but none of the participants omitted any major parts of the stories or made factual errors.

What characters were in the story?

All the participants correctly identified the major characters in both story modes. Only one person mentioned the security chief as a character in the ghost story, but as he is only mentioned briefly, he is not of great importance to the story. Interestingly, in three of the four tests, the captain in the ghost story was mentioned before the guy telling the story. This seemed unrelated to the use of NPCs though, as the person who did mention the storyteller first experienced the text version, while

one person who mentioned the captain first had the NPC version.

Out of the four test participants, two remembered the name of the captain, while three remembered the name of the doctor.

Who was the main character?

The participants all identified the main character of the space story as the chief engineer of the ship. For the ghost story, it is not explained who he is, but the participants speculated that he was probably a soldier. One simply identified him as "the guy who woke up in the barracks".

Was the main character able to successfully complete his goals?

This question allows for the test participant to interpret the story according to what he remembers. Goals are not mentioned directly in the stories, but there are various tasks that can be interpreted as a goal. In the space story the ship has to be fixed, and in the ghost story the captain asks the main character to find out what has happened.

All test participants said that the main character of the space story completed his goal of fixing the ship. For the ghost story, two of the people who got the text version mentioned that the main character got interrupted before he could complete his goal of finding out what had happened. One of the participants who got the text version, and the participant who had the NPC version both said he completed his goal of getting away from the island.

What do you think the storyteller was doing in the base before the attack? (Ghost story) & What do you think had happened to the ship? (Space story)

The answer to these questions is nowhere to be found in the stories, but rather the test participant has to speculate based on the story he experienced.

Those that had the text-based version of the ghost story all agreed that the main character was probably a soldier or guard, based on the fact that he woke up in the barracks and was taking orders from the captain. One participant further speculated that he may have been undercover in the base since he had to keep using a radio to get orders, rather than getting them directly.

The test participant who had the NPC-based version of the story, said that the storyteller was probably somebody working in the base, maybe a technician or caretaker of some sort. He based this on lack of a uniform or typical army clothes on the NPC.

Generally the people participating in the test did not have much of an idea what could have happened to the ship. One mentioned an asteroid while another speculated that it might have used more power than it was supposed to and run out.

The second set of topics are about the interaction with the NPC, and the on-screen computer monitor. The questions for the NPC mode each had a counterpart for the text mode.

What was your opinion of the NPC? (NPC mode) & What was your opinion of the virtual computer monitor? (Text mode)

All four participants had a generally positive opinion of the NPC. Two people mentioned that he was fun and likeable, and seemed like a nice guy. The two that started with the text-based story, said that the NPC changed the mood. Because he is a cartoon character, the story seemed to have a lighter tone than the story on the virtual monitor.

Generally the participants were less enthusiastic about the virtual computer monitor. The two people who started out with the NPC, went so far as to say that the monitor was boring and just a thing to flip through. The two people who started out with the text said that the monitor was fine, but a bit small. This was further supported by the video which showed one of them positioning himself in a bent forward position, much closer to the screen than intended. Interestingly, the text is the same size in both text modes, but he did not feel the need to position the screen so close to his face in the

NPC-based mode. This may be a result of there being less text on the screen when talking to the NPC.

**How did you feel about him as a storyteller? (NPC mode)
& How did you feel about reading a story in this manner? (Text mode)**

Three of the test participants said that the NPC seemed very talkative and conveyed the story fine, but that he seemed like the kind of person who just talks and talks while the listener can't get a word in. These were the same three people who got the space story from the NPC. The also said that the story seemed a bit long with too many details that weren't necessary.

All four participants were less happy about the virtual computer monitor however. Two people directly said that they disliked reading stories in this way, unless it is a part of something and important that they read it. One participant said that it depends on the story and the abilities of the author. He added that in some games he has experienced it done well, such as the Resident Evil series.

**Did it seem believable that he would be the main character of the story he told? (NPC mode)
& Did it seem believable that the story would be on a computer screen? (Text mode)**

Generally the participants felt that the NPC as main character seemed believable, though one person mentioned that he did not look like the type of person that would be a chief engineer.

Similarly, it was felt that the virtual computer monitor was believable, as it looked like the sort of screen that is seen in games and movies. One person mentioned that the colours and sounds made it seem even more like an in-game monitor.

**Did you feel you were having a conversation with him? (NPC mode)
& Did it seem believable that it was a computer screen you were looking at? (Text mode)**

The four people all agreed that the NPC storytelling felt more like he was talking to you, than an actual dialogue. Some of them suggested that if the NPC had more reply options which could take the dialogue in different directions, it would have felt more like a conversations. It was also commented that it was a bit too obvious that the replies were just versions of "previous" and "continue" written in various ways. A participant commented that when a developer is doing dialogue it has to be a trade off between writing dialogue for a million different situations which is realistic, and writing only a few options that seem less realistic.

As mentioned above, the participants were generally happy with the virtual computer screen. Two people theorized that it could be the kind of log made by the main character, like it is often seen in Star Trek and other sci-fi stories.

**Did you like talking to him? (NPC mode)
& Did you like reading the story in this way? (Text mode)**

Those that started with the text-based storytelling mode were generally more positive about the NPC. The test participants said that they liked talking to the NPC, but it would have been better with more of a dialogue and more possible response options.

Those that started out with the virtual monitor said that it was fine but a bit boring, especially for a longer story. The two people who started out with the NPC, said that the monitor was a bit boring. One of them commented that it seemed just like reading a book, but because it's on an iPhone you expect more from it than you do from a book.

The third set of topics are about the interaction with the application itself, and the implementation of the storytelling modes. As with the previous category, each NPC mode question had a text mode counterpart.

How easy/hard was it to use the dialogue options? (NPC mode)**& How easy/hard was it to use the buttons to navigate the story? (Text mode)**

All four participants felt that it was very easy to navigate the story, regardless of the mode it was presented in. Two commented that it can't get any easier than with the virtual monitor where there are only two buttons and they are labelled with their function. Two other participants said that even though the responses to the NPC dialogue are written in different ways, it was immediately obvious what the purpose of each response was. One of them further added, that a negative side effect of adding more dialogue options could be, that it would become less obvious which choice would cause the story to move forward.

Did the interaction with the NPC seem meaningful? (NPC mode)**& Did the interaction seem meaningful? (Text mode)**

The participants all felt that the virtual monitor interaction was meaningful and easy to understand. For the NPC the opinions were mixed. One person said that he seemed like the type of character you often meet very early in an RPG, who has to tell you a lot of things to quickly get you up to speed with the story behind the game. Another said that the NPC did seem meaningful, but it would have been better if he was more emotionally involved in the story he was telling. As mentioned earlier, the test participants also felt that it would have been better if the NPC had more dialogue options, so he was more interactive.

Were you satisfied with the possible dialogue choices? (NPC mode)**& Were you satisfied with the interface? (Text mode)**

As mentioned before, all the test participants would have liked more dialogue options for the NPC. They did not have trouble using the ones that were implemented and understanding them, so the desire was to have options that would make the NPC dialogue more broad with different answers and responses for different situations.

All four test participants were also satisfied with the virtual computer interface, but some felt it was perhaps too simple. One commented that it was a bit boring due to a lack of visual elements, and another stated that the text might have been a bit small.

The final group of topics were discussed after the test participant had completed both stories. They focus on his opinion of the difference between the two.

Which of the two storytelling methods did you prefer?

All four participants said that they clearly preferred the NPC for the stories implemented in the application. A comment was that he made the experience more engaging and provided something to look at. Another stated that even with just the dialogue options being “backwards” and “forwards” in disguise, the story seemed more interesting when it was told to you directly by someone.

Two of the participants commented that for storytelling in general, it depends a lot on the length of the story and the purpose for telling it. One said that an NPC is best for shorter pieces of storytelling and when the text can be made interactive as a conversation. He added that pure text is better for long static pieces of text.

Did you prefer one story over the other?

Two of the test participants preferred the ghost story, and said that it was more interesting in both plot and wording. They also commented that the space story seemed a bit too much like a long list of facts. The two other participants preferred the space story because of the sci-fi plot, and said that they found that kind of story more interesting.

Were the controls easier or harder to use in one of the storytelling methods?

The four participants agreed that the controls were easy to figure out in both storytelling modes. One commented that it was nice that the replies changed when in the NPC mode.

5. Discussion

In this chapter the meaning of the test results will be discussed, the project will be evaluated, and further possibilities for the project will be considered.

5.1 Reflection on results

After the tests were complete and all the results had been examined, it was possible to reflect on what the results were, and what possible explanations there could be for them. The results can roughly be divided into three groups. Those that concern the implementation and understanding of the story. Those that concern the technical implementation. And most importantly, those that concern the difference in text-based and NPC-based storytelling.

There was no real difference between the understanding of the story in each of the two storytelling modes. As mentioned in the results, two of the test participants did omit some of the details in their descriptions of the NPC story, and this did not occur in the text-based storytelling. However, the stories are quite detailed, so it is perhaps not surprising that it's possible to miss some details. The fact that it was not consistent across all four participants indicates that it was most likely not a result of the NPC storyteller, but rather a coincidence.

The other questions related to the content of the stories shows a similar lack of difference between the two storytelling methods. However, when asked about the main character of the ghost story, an interesting response did come up. A test participant used the look of the NPC to conclude that the character was probably not a soldier as he did not wear a uniform. Similarly, another participant who had the NPC tell the space story, commented that the NPC did not look like a chief engineer on a space ship. This did not happen in the text-based storytelling mode, where the test participants were perfectly happy reading either story on the virtual computer monitor, and did not relate the monitor to the stories they were reading.

This tells us that though there may not be a difference in understanding of the stories, there is a difference in the interpretation. The fact that there is an NPC there at all, is a secondary storytelling tool, in addition to the dialogue.

A point that was made clear in the testing, was that taking a story that is pure text and loading it into an NPC storyteller, is not necessarily a good idea. Several of the test participants commented that the NPC was just talking to them and they did not feel they were having a conversation with him. This is due to the fact that the story is implemented in much the same way in both storytelling modes: Two navigation “buttons” flip between the pages of a story.

However, when asked about it, the participants revealed that the truth is a bit more complicated. It was mentioned that the choice of storytelling mode should depend on the length of the story. They would prefer text-based storytelling for longer more factual stories, and NPCs for shorter pieces of storytelling where you can have an actual dialogue with multiple response options. That being said, nearly all the test participants also found the virtual computer monitor boring or at least less interesting than the NPC. Based on this, one could wonder if it is best to completely avoid longer pieces of text in a game, and instead tell that kind of story by using multiple storytelling tools.

The technical implementation of both storytelling modes did not seem to bother the participants, as nobody made any particularly negative comments. As mentioned, they would have preferred more interaction with the NPC through additional options in the dialogue, and for the stories to be shorter when told by an NPC. Overall, the test participants spent more time on the NPC storytelling mode. The reason for this extra time could be attributed to several different factors. Unlike in the text mode, the participant had to read the dialogue options and make sure he was pressing the right one. There were also more screens with text, though the amount of text itself remained the same, which would increase the time slightly each time the participant had to stop reading and make a choice. Finally, the NPC himself may have been distracting or interesting enough to make the person spend more time.

5.2 Implementation limitations

During the development of the test application, certain limitations were made for different reasons. These limitations are described below.

Use of storytelling tools

As mentioned previously, Österberg (2006) describes eleven different tools; of those, only two are tested in this project. It would have been desirable to test all eleven tools for the purpose of comparing them to each other. However, as mentioned previously some of the tools are not meant to stand on their own, but are merely secondary tools. Some of the tools would also put additional constraints on the stories, but it may have been possible to work around these constraints. The major obstacle in implementing a test of all eleven tools was time constraints. Some of the tools require a significant amount of work to implement, such as pre-rendered video, and it would have made the project of a magnitude beyond what is possible to do for one person in a single semester. The tools that were chosen are some of the oldest and most fundamental storytelling tools used in gaming, but more tools would have made for a broader range of results to conclude on.

Gameplay

The test is about storytelling in games, but the gaming aspect of the application is very limited. The two storytelling modes are implemented as they would appear in a typical game, but there is nothing leading up to it, and nothing after. In a typical game, a significant story would not be told purely by one of these storytelling tools. For example, if the story is about a man who had his house burned down, images of the house would be shown, possibly with accompanying sounds. There are examples of smaller stories being told purely through an NPC. This is a common approach used in MMORPGs such as Warhammer Online, Star Trek Online, or World of Warcraft. There are hundreds of small side-stories in these games, and to reduce the development time, some of them are purely told through an in-game story item or an NPC dialogue.

To fully test all the aspects of a storytelling tool, it would be necessary to also combine it with other tools to see how it works with them. This would require extensive testing though, as each combination of tools could produce different results, and it would be necessary to rule out that any potential results were caused by the other tools interfering.

Sound

The sound in the application is limited to one sound used as feedback, when the user touches interactive elements on the screen.

Apart from the tools described by Österberg, there are additional storytelling devices that can be used. Sound is one such device, as it can be a powerful secondary storytelling tool. Even minor sound effects also give a lot of hints that are consciously or subconsciously picked up by the player. For example in a first person game, if the sound of footsteps can be heard when walking down a hallway, it may provide the player with additional hints as to who he is controlling. A loud clacking sound might make somebody think of a woman wearing high heels, while a softer creaking could indicate the sneakers a technician would wear.

Background music is another powerful way sound can be used to tell part of a story by putting the player in the right mood. It is similar to how music is used in movies which Prince (2007) describes as: "Movie music emphasizes emotional effects most often by direct symbolization: The music embodies and symbolizes an emotion appropriate to the screen action".

In many modern games music is used in the manner described by Prince (2007). For example all three games in the Halo: Combat Evolved series use a soft melodic music when you are exploring outside in beautiful areas, but quickly switches to a more militant faster paced tune when you approach enemies. Not only does this set the mood for the fighting parts of the game, but it also

serves as an indication of when you can expect to run into enemies.

Music and more sound effects were not implemented in this project due to time constraints, and to prevent them interfering with the test of the NPC and text-only tools.

Quality of graphics and animations

Parts of the graphical elements were reused from previous projects, and modified to work with this testing application. The NPC has a cartoonish style which works well with lower quality graphics, but he does not have many animations. For a more realistic NPC, further movement of the body could have been added, so that his arms, head, and shoulders moved as well. The virtual computer monitor is recognizable and works for what it was intended, but it is also made from a rather low quality picture.

The graphics are not poor, but they are also not on par with what a regular 3D engine can create. However, for the purpose of testing, it was felt that these graphics were sufficient. If any part of the application is ever to be used in an actual game, improvements will have to be made though.

Source of the imagery and sounds used in the application

As mentioned, the basis for the NPC and the graphics are reused from previously conducted projects. The sound used for feedback is a modified clip acquired from a free sound repository online.

It would have been preferable to use imagery and sounds that were created specifically for this project, as this would enhance the quality as well as the possibilities in terms of look and feel. However, as they don't interfere with the test in any way, it was decided that the currently used materials would be good enough. Creating new materials for this project would only have increased the development time, which is why it was not implemented.

Code optimization

As it is, the application code is not very efficient. It runs fine for the purpose of the current test, but there are some improvements which would lower the battery and memory usage, and improve the speed. The latter is especially important if additional storytelling tools were to be added to the application, or even if more sounds should be played. Lower battery life would also mean that more testing could be done before recharging the device.

The way animations are implemented is especially taxing for the CPU, as it is three methods that repeatedly call themselves, even when the user is not doing anything. The iPhone supports various animation tools that could be used to lessen the amount of method calls, for example OpenGL ES. However, in the development phase of the project, a test was done where the phone was left with the application running for an hour, to test the impact on the battery. The result was that the looping methods did not show a drain on the battery that was significant enough to warrant changing all the animation methods. This is especially the case since the methods only run during the NPC storytelling mode, which lasts about 5 minutes per test participant.

Gender difference and character design

All the test participants were male, and as mentioned earlier, previously conducted projects showed no great difference in the results between male and female when playing games. However, there is an area where a difference is likely to occur, and that is in the perception of the NPC storyteller. Kafai (1996) conducted a test on the difference between boys and girls when choosing and designing game characters: “what kind of games would girls choose to design? The results indicate significant gender differences in game character development and game feedback”.

The NPC in the test application is designed to be a slightly comical and friendly character, and this was the way he was perceived by the participants. However, it is possible that there would be a difference in the way that he is perceived if some of the test participants had been female. Unfortunately, that topic requires further testing before a definite conclusion can be made.

5.3 Project evaluation

At the end of the project period, near the completion of the report, it's time to sit down and think about how the project went. There were several challenges to overcome this semester, such as the process of learning about storytelling tools and what can affect them when they are put into an application. Despite experience from a prior iPhone project and from running an iPhone application development business, there were still new things to learn and consider for this project. As the iPhone operating system is still a young platform and frequently receive updates, a lot of the documentation is also not always as good as it could be. There are many functions that are not exemplified much, which means that you often have to hope the documentation can help you understand how to do what it is you want to do. The documentation created by Apple is very good, but it is at times obvious that it is written by advanced developers, when a vital piece of information is left out because it is in some way implied by other information in a non-obvious way. This sometimes leads to taking development down one path, only to discover that it is not possible to do what you thought, and you have to go back and start over from an earlier point. This is minor details compared to some previous projects though, as the iPhone still has more than adequate documentation for the most common functionality. There are also many fora online where additional help can be found, if you want to do something that has not been properly documented by Apple yet.

Several things were achieved during the project period. As experienced in the project last semester, working solo is very different from working as a group. When working as a group you learn cooperative skills, but when working alone you quickly have to learn a level of self-reliance not necessary in a group situation. The danger when working alone is that you may begin to assume that your subjective opinions are automatically correct, since you have no group members to discuss them with. That was avoided in this project by using friends, co-workers, and other acquaintances, as stand-in group members to discuss ideas with or to get advice from. What friends cannot help with is the actual execution of the project, and in that part there are also several important changes when doing a solo project. An advantage to working alone is that you can plan your time more freely. This is important when you are running a business on the side while studying. It means that you have to be strict with your time planning, and that a lot of evenings, weekends, and vacations need to be spent on the project, as they were in this project period. Another disadvantage to working alone is that you have to be a jack of all trades, since you cannot assign any part of the development or writing to the person who is most experienced in that particular area. This is only a disadvantage in terms of time required though, as it also forces you to learn more about areas you are not strong in, which is a good thing.

In the end the project was completed successfully, indicating that the solo project work was done right. Some lessons were learned along the way, such as how test participants would have preferred more dialogue options in the NPC-based storytelling. Apart from minor issues, the application fulfilled the requirements for a test application for this project. For example, to allow the user to experience two stories by different means, and to avoid interfering with the test.

Finally, the results from the four conducted tests were useful and made it possible to draw some conclusions about storytelling in hand-held games. Overall, the result is that the work in this project period has been successful.

5.4 Further possibilities

There are many ways in which it is possible to further expand the test done in this project period, as well as the application. Unfortunately it is only possible to pursue a limited amount of topics when you have limited time, which is the reason this is a test of only two storytelling tools.

The most obvious way to expand on the test, would be to include more storytelling tools. Initially just other text tools, and then later the more different tools like pre-rendered video. It would also be prudent to investigate how these tools work together, as that is how they are most often presented in a game. Some tools are secondary and will work poorly on their own, so it will be necessary to test these in the proper context. It is also important to remember that there are more storytelling tools than the eleven found by Österberg (2006). During the test for this project, the look of the NPC was a secondary storytelling tool which helped the test participant to interpret the story he was reading. How these tools work independently and in combination is a very large topic, and it would require several different tests to uncover all of it.

The results found in this project were done in a game-like setting, and the test participants were all gamers. However, the lack of a game outside the storytelling may have had an unknown impact on the results. A way to ensure that this is not the case, would be to expand the application. Two 3D environments could be created where the player can walk around and experience the story. In one the story would have to be told purely by text, such as in-game storytelling items and screens with pure text. The other mode should then have several NPCs telling the story. The problem with this implementation is that as the NPC in the test for this project revealed some of the story, so would a 3D environment but on a much larger scale as the environment would be bigger and more detailed. However, if implemented in the correct manner, it should be possible to conduct this test as part of a much bigger game application.

In addition to other possible storytelling tools, there are areas which could be interesting to expand the research to. One such example could be how people react to different NPCs. The test participants found the one in this test nice and fun, but what if he had looked like a more unpleasant type? Would that affect their perception of the story, their interest, or their ability to remember it? The look and clothes of an NPC may affect the player's opinion of what he is saying, and possibly cause them to mistrust him if the story has parts that are unclear.

Furthermore, the results from the test done by Kafai (1996) indicate that gender plays a role in the perception of an NPC. This is another detail to take into account if a test was to be conducted on NPC storytellers.

The style and quality of the NPC graphics may also make a difference. The one created for this project is a cartoon character, but a more realistic character may be taken more seriously by the player.

Finally, as it was discovered during this test that the content of the stories matter a great deal, this is yet another further possibility for research. Two different sub-topics were mentioned by the test participants. Firstly there is the length of the story. According to the participants, they would prefer shorter stories by NPCs and longer stories as pure text. Rather than putting the two against each other again, a test could be conducted where only the length of one of the two tools was tested. Several stories of different lengths could be created and tested to see which the test participants think are too long.

Secondly, the participants indicated that they would like more dialogue options for the NPCs. This could potentially be tested with the application created for this project. The NPC mode could be used alone to tell two different stories, one with many dialogue options and one with few. The results would be able to tell something about the understanding of the content, when presented as a multiple choice conversation.

5.5 Business plan

The process of creating something for the purpose of research is not always the same as creating something for the purpose of business. There is some overlap between the two areas, but both also have parts that are incompatible with the other. For example in a research project it may be okay to have a test application that is not very pretty, as it is the test that is the important part. When creating a business, that line of thinking will not work with a product that has to be sold to end users, unless perhaps if it is sold to people who want to conduct a test.

There are a number of topics that are important in a business plan as noted by Harvard Business School Press (HBS 2007), and an adapted version of these serve as the section titles in this subchapter.

Executive summary

The topic of this project can be divided into two different business ideas. One being based on the results of the test, which can be used in the process of creating games that involve storytelling in some way. There is no one specific product associated with these results, and they could apply any game that uses storytelling in some way.

The second area of interest is the test application, which in itself is not a product, but parts of it could be used in the creation of a storytelling product. It could be a game, but it could also be a storyteller who tell a story for children, or a story about a company for advertisement purposes.

Business description

A company developing iPhone applications, for the purpose of advertising a company by telling the story of the company. The story is told by an NPC storyteller who can be customized in physical appearance, body language, and speech to suit the company in question. The story can be as complex as desired, and the NPC can have various dialogue options which enables the user to get further information on a subject.

Business environment analysis

An analysis known as the PEST analysis looks at external business environment (AWARE n.d.). It is used to examine political, economic, sociocultural, and technological factors in the business idea. A potential political factor in the business could be the kind of information the application is used to make public. It would be necessary to go through the information to make sure that it adheres to the current laws concerning data protection and advertising. Normally iPhone applications are distributed via the App Store on iTunes, and thus all information would be available to the general public. It would probably also be necessary to evaluate the potential customer on a case by case basis, to avoid that the application be misused for a criminal purpose.

The main economical factor to consider would be the current economic downturn. The type of application suggested is mostly meant to advertise a business, and when a person or company has less money available in general, he may be less inclined to spend it on an unproven way of advertising. The sociocultural factors are similar to the political ones. In addition to laws, it would also be prudent to investigate if the information in the application could have an impact on other businesses, individuals, or social, political, and religious groups. It is also important that it does not contain things that could be found to be offensive or misunderstood in a way that would harm the business trying to advertise themselves.

The technological factors are limited as the application is mostly functional in the current state. If the advertisement idea catches on, it may be a good idea to create templates for NPC storytellers, to reduce development time. More animations could also be added to the storyteller to make him seem more lively. Additionally the fact that the current application only runs on the Apple iPhone may prove to be a concern if the Google Android mobile operating system increases in popularity and companies wish to have their advertisement on both platforms.

Industry background

Familiarity with the processes and technologies involved in creating the applications, due to a year of experience running a business developing for the iPhone. Developing the applications suggested in this business plan will not be substantially different from the test application created for this project. However, it will potentially be necessary to hire a professional writer to make sure the stories are exactly as the companies want them, unless they have a premade text they want to use. A lack of experience in selling advertisement could be an issue in finding customers, but not as much in the development itself.

Competitive analysis

In the research for this project, it was not possible to find any company doing precisely what this business idea suggests. However, companies that do advertisement through storytelling do exist, and some utilize mobile platforms. This means that there is some overlap between this idea and existing ideas, but not enough to say that there is a direct competitor.

Market analysis

The concept of advertising on the iPhone is not new. In the two years that have passed since the release of the App Store, countless applications have been released for the purpose of advertising something. An example is the application "Lightsaber" which is a simple piece of software that makes lightsaber (of Star Wars fame) sounds when the phone is moved around. The advertisement exists in the form of a full screen ad for the game "Star Wars – The Force Unleashed" every time the application is launched. Companies like Coca Cola have many different applications that advertise their products on the App Store as well.

Marketing plan

The best way of marketing the product would be if it was possible to become associated with an advertisement company specializing in other forms of ads. This business could then supplement theirs, by offering advertisement in the form of applications on the iPhone.

Alternatively, program demonstrations could be created to attempt to sell the idea to companies through promotional material. However, this approach would likely be harder as the business would initially not have any customers to show as reference material for other interested companies.

Operations plan

Once an application is created it would require little maintenance. As it is self-contained and hosted on the Apple servers, the only problems would occur when a new version of the iPhone operating system is released. When that happens, it would be necessary to test the platform with the new operating system and recompile applications for it, as well as change the code if necessary.

However, as the changes to the code would be the same for all applications, it would not take much time.

Management summary

The business would not need to be very large, unless a truly huge application was to be created. One to two people should easily be able to produce the code and write the story material required to create one of the applications suggested.

Financial plan

The financing will be difficult initially due to not having any customers. However, as iPhone applications do not require much work to be sold on the App Store, a possible way of financing would be to first create some unrelated applications which can be sold to people in general. They could then generate a steady income for the business, while customers are located.

6. Conclusion

The purpose of this project was to explore storytelling in a game on a hand-held platform, by using stories told through two different storytelling tools. The overall success of the project depends on how well the goals from the problem formulation were answered during the project, and if the hypothesis was disproved. In the problem formulation the following three objectives were defined:

- Understanding – How well does the test participant understand the story.
- Preference – Which storytelling tool does the test participant prefer, and why.
- Difference – Which factors influence how the storytelling in a game is perceived.

The hypothesis was: There is no discernible difference between text- and NPC-based storytelling, on a hand-held device.

This hypothesis was clearly disproved by the results. No discernible difference was found when it comes to understanding the story. However, the test participants found the NPC more interesting and preferred being told a story by him than by the virtual computer monitor. The NPC also proved an additional storytelling tool in himself, by allowing the viewer to interpret the story through him, as one did when they used his clothes to determine his role in the military base.

Of course this does not mean that the hypothesis is disproved for all situations. Indeed, a point brought up by several of the participants was that the preference of one of the storytelling tools depended on the length and type of story. It was further supported by complaints that the NPC was talking to you, more than he was having a conversation. Nonetheless, in the case of the stories created for this project, told by an NPC and a virtual monitor, and presented on a hand-held platform, there was clearly a difference between the two storytelling modes.

In regards to the understanding objective, there is little to say. Both storytelling modes let the participants understand the story. As mentioned, the NPC provided an extra factor in the story for some people, but the lack of an NPC in text-based mode did not negatively impact the story. As mentioned above, the overall preference of the test participants was the NPC mode. Though, for longer more factual stories, some people said that the NPC would not be as great, as it turns into too much of a monologue. For shorter stories and pieces of information the NPC was the preferred. However, for the NPC dialogues several participants complained about a lack of response options. This means that to be successful, NPC dialogue should strive to not just be “back” and “forward” in disguise, but actually provide responses that will change what the NPC says in return. This leads to the theory that up to a certain point, the more interactivity the NPC dialogue has, the better it will be perceived to be compared to the virtual monitor. This theory would require additional testing to be certain though.

This leads to the third objective, which is the factors that influence how the storytelling in a game is perceived. At the start of this project, it was assumed that it would only be the presence or lack of an NPC storyteller. During the project period it became clear that there are many other factors than the eleven tools described by Österberg (2006). As Prince (2007) stated, music can directly affect the emotional state of the viewer in movies. Music is used in similar ways in games, and while the test application did not have music, there is no reason to assume that it would not have had an effect. The test showed that the way the NPC looks can influence the viewers perception of him and details of the story. It also showed that the way dialogue is implemented can influence the way the viewer feels about the NPC, as we saw when some of the participants felt that the NPC was the kind of person who just talks and talks without letting you get a word in.

In conclusion, while trying to disprove the project hypothesis, it became clear that there are actually many more factors involved than originally assumed. It is clear that there is much room for further research in this field.

7. References

- AWARE (n.d.). *Business Environmental (PEST) Analysis. Marketing & Competitive Intelligence FAQ*. Retrieved Jan 12, 2010, from <http://www.marketing-intelligence.co.uk/help/Q&A/question14.htm>
- Brockmyer, J. H., Fox, C. M., Curtiss, K. A., Mcbroom, E., Burkhart, K. M., and Pidruzny, J. N. (2009). The development of the game engagement questionnaire: A measure of engagement in video game-playing. *Journal of Experimental Social Psychology*.
- Crawford, C. (2005). *Chris Crawford on interactive storytelling*. Boston, MA: New Riders. ISBN: 978-0321278906
- Davies, M.B. (2007). *Doing a successful research project*. Houndsmills, Basingstoke, Hampshire: Palgrave. ISBN: 9781403993793.
- Gagné, M. and Deci, E. L. (2005): Self-Determination Theory and Work Motivation. *Journal of Organizational Behavior*, vol. 26. ISSN: 0894-3796.
- HBS (2007). *Creating a Business Plan: Pocket Mentor Series*. Harvard Business School Press. ISBN: 978-1422118856
- Juul, J. (1999). A clash between game and narrative: A thesis on computer games and interactive fiction. Unpublished master's thesis, University of Copenhagen. Retrieved Jan 12, 2010, from <http://www.jesperjuul.dk/thesis>
- Kafai, Y. B. (1996). Gender Difference in Children's Construction of Video Games. In P. M. Greenfield and R. R. Cocking, *Interacting with Video (Advances in Applied Developmental Psychology, Vol 11)* (pp. 39-66). Praeger. ISBN: 978-1567501520
- Mallon, B. and Webb, B. (2005). Stand up and take your place: identifying narrative elements in narrative adventure and role-play games. *Comput. Entertain.*, 3(1):6.
- Prince, S. (2007). *Movies and Meaning – An Introduction To Film (Fourth Edition)*. Pearson Education. ISBN: 0-205-48077-2
- Österberg, W. (2006). *Storytelling in Single Player Action Computer Games - The tools of the trade*. Unpublished master's thesis, Royal Institute of Technology, Stockholm, Sweden.
- Sharp, H., Rogers, Y., and Preece, J. (2007). *Interaction Design: Beyond Human Computer Interaction*. Wiley. ISBN: 978-0-470-01866-8
- Sheldon, L. (2004). *Character Development and Storytelling for Games (Game Development Series)*. Muska & Lipman/Premier-Trade. ISBN: 978-1592003532
- Thomas, F. and Johnston, O. (1981). *The Illusion of Life: Disney Animation*. Disney Editions. ISBN: 978-0786860708
- Witmer, B.G., & Singer, M.J. (1998). Measuring Presence in Virtual Environments: A Presence Questionnaire. *Presence, Vol. 7, No. 3, June 1998* pp. 225–240. ISSN: 1054-7460.