Master Thesis in
Software Engineering
Thesis no: MSE-2003-27
August 2003

Development of an Adventure Game

An Evaluation of Tools, Development, and Story Writing

Erik Heinemark
Johan Persson

Department of
Software Engineering and Computer Science
Blekinge Institute of Technology
Box 520
SE – 372 25 Ronneby
Sweden

This thesis is submitted to the Department of Software Engineering and Computer Science at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Software Engineering. The thesis is equivalent to 20 weeks of full time studies.

Contact Information:

Authors:

Erik Heinemark
Studentvägen 3:13
372 40 Ronneby

E-mail: pt99ehe@student.bth.se

Johan Persson
Folkparksvägen 14:25
372 38 Ronneby

E-mail: johan@hus7.rsn.bth.se

University advisor:
Professor Lars Lundberg
Department of Software Engineering and Computer Science

Department of
Software Engineering and Computer Science
Blekinge Institute of Technology
Box 520
SE – 372 25 Ronneby
Sweden

Internet : www.bth.se/ipd
Phone : +46 457 38 50 00
Fax : +46 457 271 25

ABSTRACT

This master thesis discusses three different parts of adventure game development. The first part is about the usage of existing development environments; which one we selected and how we selected it. The second part discusses the development of the game using the selected development environment from the first part. The third and last part discusses the benefits from using skilled story writers when developing an adventure game. In this work the story writers were students from the English Department at Blekinge Institute of Technology.

Keywords: adventure games, game development, tool evaluation, story writing

ACKNOWLEDGEMENTS

We would like to thank the following persons:

Professor Lars Lundberg for supervising our work and providing valuable feedback.

Professor Danuta Fjellestad for good ideas and valuable feedback.

Sandra Brisenman, Mathias Roos, and Teri L. Schamp-Bjerede for writing the story and for working with us in the project.

Magnus Hjelmér and Johan W. Andersson for reviewing our thesis.

CONTENTS

ABSTRACT ..... **I**

ACKNOWLEDGEMENTS ..... **II**

CONTENTS ..... **III**

1 INTRODUCTION **1**

1.1 BACKGROUND AND MOTIVATION 1
1.2 RESEARCH QUESTIONS 1
1.3 RESEARCH SCOPE AND LIMITATIONS 1
1.4 OUTLINE FOR THE REMAINING CHAPTERS 2

2 BACKGROUND **3**

3 METHODOLOGY **4**

3.1 EVALUATING GAME DEVELOPMENT ENVIRONMENTS 4
3.1.1 Evaluation criteria *4*
3.1.2 Score calculation formula *5*
3.2 DEVELOPING AN ADVENTURE GAME 5
3.3 EVALUATING THE WORK DONE BY THE STORY WRITERS 6

4 EVALUATION OF GAME DEVELOPMENT ENVIRONMENTS **7**

4.1 EVALUATION 7
4.1.1 Adventure Game Studio *7*
4.1.2 AGAST *7*
4.1.3 SLUDGE *8*
4.1.4 Adventure Maker *9*
4.1.5 Wintermute Engine *9*
4.1.6 ADRIFT *10*
4.1.7 SCI Studio *11*
4.2 EVALUATION RESULTS 11
4.3 SUMMARY 12

5 DEVELOPMENT OF AN ADVENTURE GAME **13**

5.1 INTRODUCTION 13
5.2 ROLES IN THE PROJECT 14
5.2.1 The story writer *14*
5.2.2 The game designer *14*
5.2.3 The programmer *14*
5.2.4 The graphic artist *15*
5.2.5 The sound engineer *15*
5.3 THE PLANNING PHASE 15
5.4 THE IMPLEMENTATION PHASE 15
5.5 TIME MEASUREMENTS 16
5.6 SUMMARY 17

6 CREATING ADVENTURE GAMES WITH STORY WRITERS **19**

6.1 LINEAR STORY PROBLEMS 19
6.2 TECHNICAL LIMITATIONS 20
6.3 UNCLEAR GAME OBJECTIVES 20
6.4 SUMMARY 21

7 DISCUSSION **22**

8 CONCLUSIONS **23**

9 REFERENCES 24

APPENDIX A. EXTENDED BACKGROUND 25

APPENDIX B. TERMS AND ABBREVIATIONS 30

APPENDIX C. LINKS 31

1 INTRODUCTION

The purpose of this chapter is to give an introduction to this master thesis and a motivation to why it is useful. An outline of the thesis is also introduced here.

1.1 Background and motivation

Every year, hundreds of commercial computer games are released. Today many games can present close to photo-realistic graphics, which is the ultimate goal to reach for many game programmers. As we are so close to perfect graphics, the step between good and bad graphics are getting smaller and smaller, which means the significance of the same has been reduced.

To create a successful game you need something more, you need a fascinating and interesting story. Unfortunately there are not many people competent enough to write interesting stories in the gaming industry. Often stories are written by software engineers, whom generally are more interested in the programming parts of the game and see the story writing as a necessary evil.

In this project, we cooperated with three students from the English Department at Blekinge Institute of Technology. Their task was to write a story for an adventure game and our task was to turn that story into a game. The story writers had a non-technical background and knew very little, if anything at all, about game development.

Our experience of game development were also limited. Though, we were experienced in many different programming languages and our interest in computer games, both as players and developers, was very high.

The communication throughout the project was handled mostly through mail and phone, since we were located in two different cities. The most challenging part of the project was about how to structure our work in a distance working group and how to take advantage of the special skills the story writers possess. The result of our work are written down in this thesis.

1.2 Research questions

Our aim with this report was to answer these questions:

- What game development environment suits a small adventure game project best?
- How should similar projects be structured in the future?
- How can game projects benefit from using story writers?

1.3 Research scope and limitations

The master project had a very limited time budget (20 weeks), where less than half of the total project time could be spent on game development. This caused a lot of limitations in our game. We had to remove many features we planned to include in the game from the beginning.

Since we had no financial support while working with this master project, we only had the possibility to include free game development environments in our evaluation. Therefore we only tested shareware and freeware software.

1.4 Outline for the remaining chapters

Chapter 2 – Background
This chapter provides a background and motivation of why the story is useful in games.

Chapter 3 – Methodology
This chapter presents the methodology used when working with this thesis.

Chapter 4 – Evaluation of Game Development Environments
This chapter presents the results of the evaluation of different tools available for creating adventure games today.

Chapter 5 – Development of an Adventure Game
This chapter describes how we structured the work when developing an adventure game with the support of story writers.

Chapter 6 – Creating Adventure Games with Story Writers
This chapter presents our experiences of working with story writers in this project.

Chapter 7 – Discussion
This chapter discusses our results in this master thesis and the possibilities for further studies in this area of interest.

Chapter 8 – Conclusions
This chapter presents the conclusions of this master thesis.

Chapter 9 – References
References are listed in this chapter, both ordinary references and links to the evaluated tools.

Appendix A – Extended Background
This appendix extends the second chapter about the importance of stories in games, the elements of adventure games and the history of adventure games.

Appendix B – Terms and Abbreviations
This appendix contains explanations for the terms and abbreviations used in this thesis.

Appendix C – Links
This appendix contains links to the development environments and game development companies mentioned in this thesis.

2 BACKGROUND

When creating a game it is important to have an interesting story, especially when creating adventure games. An adventure game without a good story can be very hard so sell nowadays. As graphics in games have became almost as good as it can be, the story gets more and more focus. Looking at it from a historical perspective, we have had one period where games were all about game play and story since they were text-based or had very limited graphics. In the next period, the presence of stunning graphics was the most vital part and the story got less attention. Nowadays we are back in a position where we have both good graphics and focus on story. If you are unfamiliar with the role of story writing in game projects, reading Appendix A may be appropriate.

3 METHODOLOGY

Our work was done in three different phases. First, we evaluated the current available game development environments and choose the most appropriate one. Second, with the selected tool we created a game based on a story written by the story writers. Third, we evaluated the outcome of the work with the story writers.

3.1 Evaluating game development environments

At first we had to choose a suitable environment that would simplify the development of an adventure game. By defining the different parts we believed were important when developing an adventure game, we obtained the necessary criteria for an evaluation. Since most environments were specialized for certain game interfaces, it was important to know what type of game that was to be created when selecting the development environment.

Next we choose a number of development environment candidates based on the chosen criteria. Since the game genre was what we call adventure, there were some specific criteria for what interface the game could use. The interface is an important part of a computer game and therefore it had much influence when deciding which environment to use.

When we selected the most suitable environment for us, we used a weight system for the different criteria. The weights for the criteria were values ranging from 1 to 5 (the selected weights for the criteria can be found in table 4.1). Then by evaluating the environments a grade for each criteria was set. The grading values ranged from 1 to 3, where 1 meant that the criteria did not match our needs, while a 3 meant that the criteria was a perfect match of what we were looking for.

3.1.1 Evaluation criteria

The development environment to be chosen, should fulfill all or most of the criteria listed below. The different criteria had different importance to us and to illustrate this in the evaluation, we assigned a weight to each criteria.

3.1.1.1 Existing games

There had to be some existing games developed with the environment and the games should preferably have commercial quality. This criteria showed if it was possible to create a complex (meaning large and high-quality graphics) game with the development environment and therefore this was the most important criteria. This criteria was assigned the weight of 5.

3.1.1.2 Development Environment

The development environment should support the development of an adventure game. With the development environment we mean the part that was used to combine the resources to a working game, for example scripting or by point-and-click. This criteria was assigned the weight of 4.

3.1.1.3 Flexibility

The development environment should have a flexible game engine and scripting language. It should have the ability to create own solutions, for example the possibility to develop an own menu bar. The criteria was assigned the weight of 3.

3.1.1.4 Utilities

The development environment should contain some components, which simplify the development of an adventure game. It could have been for example a media conversion tool, an image creation tool or an animation tool. This criteria was assigned the weight of 2.

3.1.1.5 Documentation

Documentation should exist for the environment. Complete documentation is important when developing any application. Incomplete documentation often means longer development times. This criteria was assigned the weight of 4.

3.1.1.6 Project overview

Most environments have some kind of application for managing game projects and these often contain some structural overview of the current project. We will determine how easy it is to get a clear view of the structure of the game. This criteria was assigned the weight of 2.

3.1.2 Score calculation formula

To calculate the total score, for each criteria multiply the grade and weight and then add the results from those calculations. The formula for calculating the total value for each environment is shown in equation 3.1.

$$\text{total} = \sum_{x=first}^{last} grade_x * weight_x$$

Equation 3.1 – *The equation used when calculating the total score.*

3.2 Developing an adventure game

The making of an adventure game was a big part of our work. We built it with the use of the development environment we selected in the evaluation. When developing the game we could see if the development environment worked as well as predicted.

During the project we cooperated with three story writers. They handed us a story and our task was to turn that story into a fully functional game. When starting the project we had not much of experience from earlier game development projects, especially not projects where the story had played an important role, so it was troublesome for us to figure out how to act in the beginning of the project.

When developing the game we used the communication model shown in figure 3.1. We believed it was important to have someone coordinating the project, a point in the

communication channels that could filter out irrelevant information and ideas. Therefore we decided to let all communication go through us, the programmers, in this project.

Figure 3.1 - *The communication model.*

During the development of the game we measured the total time used by the different participants, and with that data we could make a suggestion on how to build a team for a similar project in the future.

3.3 Evaluating the work done by the story writers

The questions we wanted to answer after this project were if the story writers added benefits to our project and if they counteracted the project in any way. To do this we had to look at the results of the project work with the story writers.

By answering the following questions a conclusion was made about the benefits from using story writers in the adventure game project:

- What did they produce?
- What positive effects did they have on the project?
- What negative effects did they have on the project?
- What other things did they affect?

4 EVALUATION OF GAME DEVELOPMENT ENVIRONMENTS

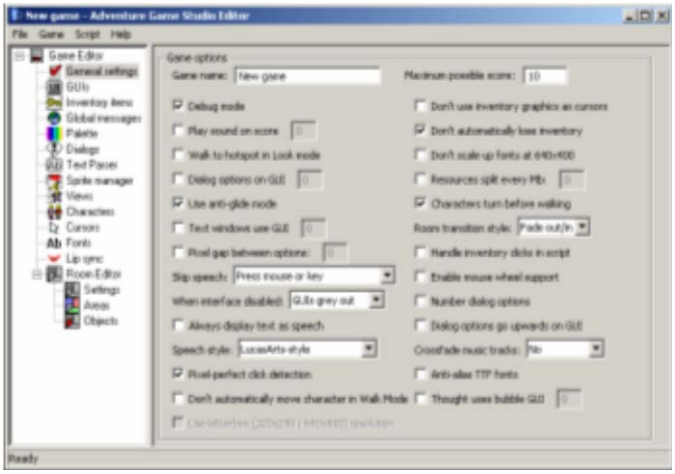
The purpose of this chapter is to present the results in the evaluation of different tools available for creating adventure games today.

4.1 Evaluation

4.1.1 Adventure Game Studio

The Adventure Game Studio tool consists of a number of different parts. These parts are a script editor, settings control for main game functions (resolution, target environment, etc.), a dialogue editor, an object editor for both placement and events triggered, point-and-click for setting up events and C-style scripting language for more control over events. The engine can handle a number of popular media formats for music and graphics encoding. It also has built-in inventory management, which is necessary for a good adventure game engine. The documentation on the homepage is thorough and explains everything step-by-step including pictures of the used parts of the tool.

Criteria	AGS
Existing games	2
Development Env.	1
Flexibility	3
Utilities	1
Documentation	3
Project overview	1



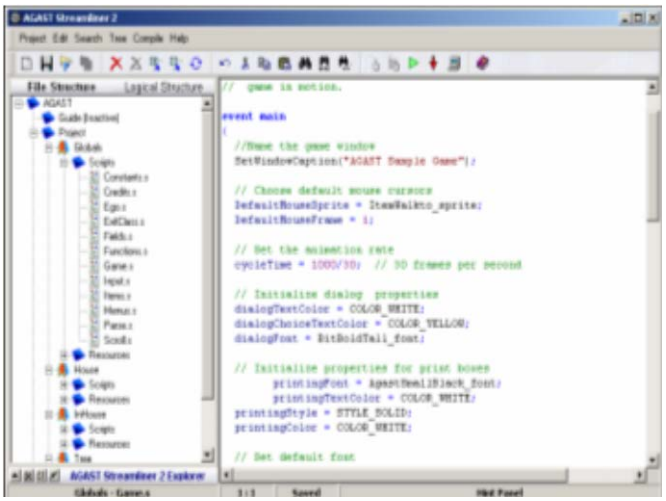
There are no way to get a good project overview with this environment, because of the application layout. No extra utilities come with the package. There are a lot of existing games made with this environment, but no commercial quality games. It is free for non-commercial use. Flexibility is no problem for this game engine, because there are a number of non-adventure games created with it, for example a platform game engine. The development environment is not very intuitive and resource control is more complicated than it should be.

4.1.2 AGAST

The main components of AGAST are the compiler and interpreter. Other components included are a text editor and some conversion utilities. Extending the main AGAST components is the utility pack, which contains a development environment called Streamliner, a dialogue editor, a media converter (converts gif to sprite), a path editor (setting up areas where it is possible to go on a background image) and a font editor.

Creating games with AGAST is done through programming in a C-style scripting language. The development environment included in the utilities pack makes it easier to overview the structure of the game and has a built-in text editor. The documentation included goes through most parts of making a game with the help of AGAST.

Criteria	AGAST
Existing games	3
Development Env.	3
Flexibility	2
Utilities	3
Documentation	2
Project overview	3

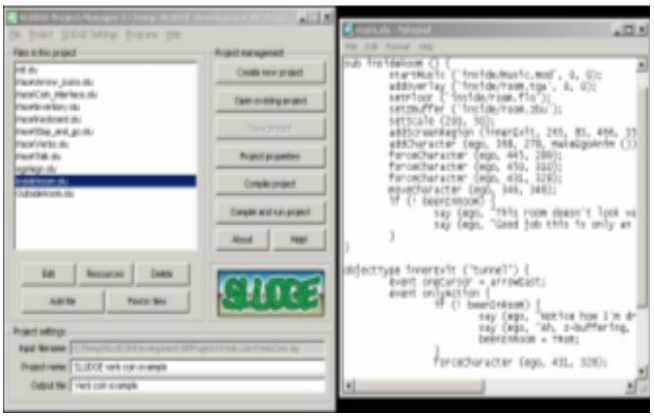


Because of the commercial game Eye of the Kraken, AGAST gets a high grade for existing games made. The development environment has its own script editor complete with code highlighting and it is easy to add images and other resources to the project. Because of the scripting used to create games it is flexible. AGAST is free for non-commercial use.

4.1.3 SLUDGE

SLUDGE is distributed as a development kit, which installs easily. It has a project manager, which handles the project files. Each file except the init-file is a part of a scene or an object and can have a number of resources, which can be listed in the project manager. The development kit also comes with extra tools like the SLUDGE Sprite Bank Editor, the SLUDGE Z-buffer Maker, the SLUDGE Floor Maker and a compiler. It uses scripts to create games, and the engine can handle most of the known media formats.

Criteria	SLUDGE
Existing games	2
Development Env.	2
Flexibility	3
Utilities	3
Documentation	2
Project overview	1



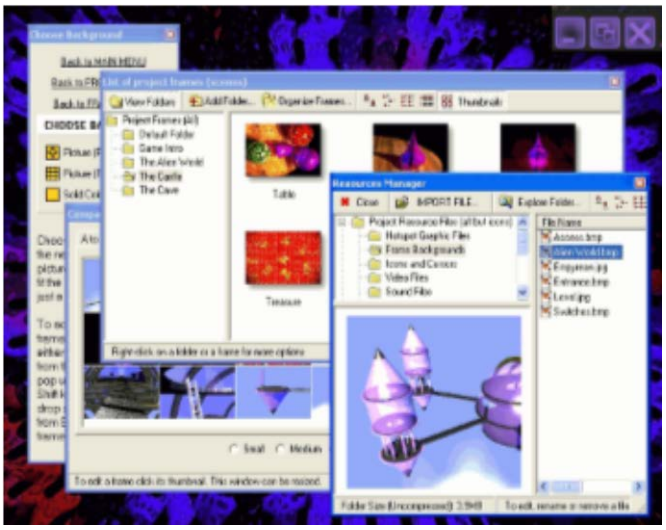
It is hard to get a good overview of the game with this environment. This becomes more obvious when creating larger projects with many files. The documentation provided in the development kit is a bit thin. We tested the shareware version of the environment, because the full version was not freely available. The differences are some missing functions and an

annoying startup message. It is a very flexible engine with, for example, the ability to create both adventure and platform games. It is easy to add different resources to the game through scripting. The default text editor of the operating system is used to edit the scripts.

4.1.4 Adventure Maker

The Adventure Maker tool is a “point-and-click” tool, which means that you do not have to do any scripting at all when you create games. It is possible to add some simple scripting (VBScript) to gain more advanced control over events. The game engine supports the most used media formats. Adventure Maker uses different manager components for developing a game, for example it has a frame manager (it organizes the different scenes of the game) and a resource manager (it handles the media files used in the game). Thorough documentation only exists for the scripting language, there is nothing on how to use the Adventure Maker tool in general.

Criteria	AM
Existing games	3
Development Env.	2
Flexibility	1
Utilities	1
Documentation	1
Project overview	2

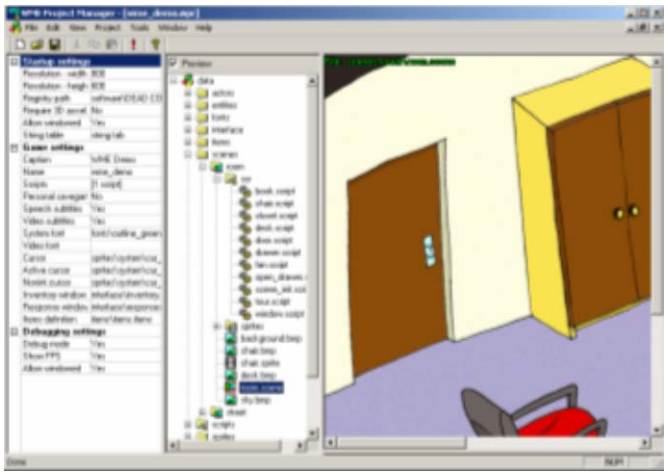


It costs money, but there is a 30-day trial version available. The only utility that comes with the package is an icon editor. The project is divided so all the resources are listed in a resource manager window and the scenes have their own manager window.

4.1.5 Wintermute Engine

The components of the Wintermute Engine are a compiler, an interpreter, the Project Manager (in essence a tool for handling the resources of the game and setting up the engine parameters), the SceneEdit tool (used to design scenes) and the SpriteEdit tool (used to create static images or animations). The engine interprets an object-oriented scripting language and uses the media resources to visualize the game. The Project Manager tool does not have a built-in editor for the scripting, instead it uses the default editor of the environment. The engine has support for hardware accelerated graphics and through this it can create better graphical effects without slowing down the environment. The Wintermute Engine package comes with a help file, but the documentation is not totally complete yet. Help also exists through the demo project accompanying the development kit.

Criteria	WME
Existing games	1
Development Env.	3
Flexibility	3
Utilities	3
Documentation	2
Project overview	3

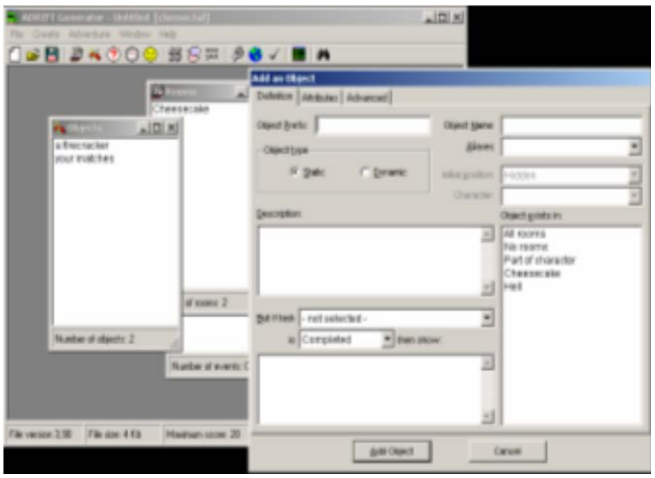


The version we evaluated was a beta version of the development kit, but we did not discover any major bugs when we tested it. The incomplete documentation makes the total score a bit lower. Another problem is the lack of games developed with the Wintermute Engine. It is free for non-commercial use.

4.1.6 ADRIFT

The ADRIFT tool has two main components; the Generator tool (GUI for creating adventures for ADRIFT) and the Runner tool (uses the files generated from the Generator to play out the game). ADRIFT is different from the other tools since it is only possible to create text-based games with it. It does not use a scripting language to create adventure games, but has an excellent documentation accompanying the package. The Generator tool has a built-in feature for showing how the rooms connect to each other.

Criteria	ADRIFT
Existing games	1
Development Env.	2
Flexibility	1
Utilities	1
Documentation	3
Project overview	2

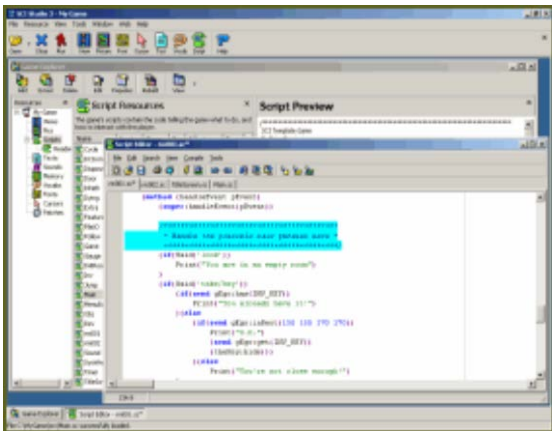


ADRIFT is a good tool to create text-based games with and the development tool has forms for everything, which makes it possible to create a game without scripting. There are a number of games developed with ADRIFT, but the lack of graphics in the games created with the environment lowers the grade. It has some limitations until it is registered.

4.1.7 SCI Studio

SCI Studio is a complete development environment. It has integrated tools for graphics, animation, code editing and it has a resource controlling view. To play a game the compiled files are run through the SCI engine. A problem with SCI Studio is the inability to change the resolution of the game, but that is an engine limitation. SCI Studio comes with extensive help documentation, which also includes pictures of the explained parts.

Criteria	SCI Studio
Existing games	2
Development Env.	3
Flexibility	1
Utilities	3
Documentation	3
Project overview	2



SCI Studio is a great development environment, but unfortunately the engine it uses for games is a bit outdated if you want to create more graphic-oriented games. All the utilities you need are incorporated into the development environment.

4.2 Evaluation results

Criteria	Weight	AGS	AGAST	SLUDGE	AM	WME	ADRIFT	SCI Studio
Existing games	**5**	2	3	2	3	1	1	2
Development Env.	**4**	1	3	2	2	3	2	3
Flexibility	**3**	3	2	3	1	3	1	1
Utilities	**2**	1	3	3	1	3	1	3
Documentation	**4**	3	2	2	1	2	3	3
Project overview	**2**	1	3	1	2	3	2	2
	Total	39	**53**	43	36	46	34	47

Table 4.1 - *The complete environment evaluation table.*

The completed evaluation is shown in table 4.1. Using the selected weights, AGAST received the most total points in this evaluation, but SCI Studio and Wintermute Engine were not far behind. Adventure Maker and ADRIFT received the least points, mostly because of the inflexibility of their game engines.

4.3 Summary

Any of the three best rated development environments could have been used in our game project, but we selected AGAST based on the ability to create different types of games easily and because of the fact that it has an integrated development environment with compiler and interpreter incorporated. For example both Sierra- and Myst-like interfaces are possible to create with AGAST. The scripting language used in AGAST is very easy to learn. Another very important part of the selection process was to look at existing games made with the environment. AGAST was used in the making of the game Eye of the Kraken, which is very good compared to the games made with the other development environments included in the evaluation.

5 DEVELOPMENT OF AN ADVENTURE GAME

The purpose of this chapter is to describe how we structured the work when developing an adventure game in cooperation with the story writers.

5.1 Introduction

> *"The flow of the dingy city landscape coupled with the murky urban environment that Sal is negotiating leads him to check the pavement so as not to step in anything... unexpected. The sidewalks are cracked and chipped, as anticipated. The gutter almost overflows with litter and the way the graffiti streaks in some artistic parody set off by drunks and lowlifes makes this short cut home not so much of a short cut. "How could I have gotten here," Sal wonders softly to himself so as not to disturb those who roam the oncoming darkness."*

You have now read the beginning of the background story for the game we call *Serendipity*. *Serendipity* is an adventure game (read more about adventure games in Appendix A), but unlike many other computer games, it is not designed to be just entertaining, it is also educating.

Figure 5.1 – *The main character has just entered the bookstore.*

Good sources of inspiration for this game have been all classic adventure games from Lucas Art, such as *Sam & Max* and *Day of the Tentacle*, but also puzzle games like *Myst* and *Riven*. Just like these games, our game is based on pre-made backgrounds where it is possible to interact with the environment.

The educating elements of the game consists mainly of a couple of hundred questions and riddles in the field of literature history. The intended audience for the game are people studying literature or people just interested in literature. But we have also added elements, like for example small puzzles, to the game so it can be interesting for other people too.

5.2 Roles in the project

In an early phase of the project we visited the game development company Massive Entertainment (located in Malmö, Sweden) to see how they work and to get inspiration for our own game. Each game title they work on at Massive Entertainment takes around 20 months to complete and more than 20 people are assigned to each project. We had approximately 3 months development time and 5 people budgeted for ours, so it was a really small project in this context.

In a big company like Massive Entertainment there are many different roles, all from producers and game designers to programmers and game testers are involved in the process of making a game. In a smaller project, like ours, most of their roles are unnecessary. We decided to adept the five most important roles into our project – the story writer, the game designer, the programmer, the graphic artist and the sound engineer. These five roles reflect the basic elements of every game – story, game play, graphics, sound and music.

As the team originally consisted of three story writers and two programmers, the roles as game designer, graphic artist and sound engineer were shared between all of us. In other words we had to create the graphics, record the sounds and compose the music needed for the game by ourselves. When we were unhappy with the quality of graphics and sounds, we consulted some of our friends for improvements.

A short description of each role in the project follows below.

5.2.1 The story writer

The story writer's main responsibility was to write the background story. But since the story was not detailed enough, it needed to be complemented with extensive descriptions of all locations, objects and characters in the game.

5.2.2 The game designer

The game designer is the creative mind behind the game. The responsibility of the game designer was to integrate the story into the game and to make it fun for the intended audience to play. In our project most of the work was put on making questions, riddles and puzzles.

5.2.3 The programmer

The programmer was responsible for turning the game designer's vision into a playable game. There are many different programming tasks to work with as game programmer. In our project the work consisted mainly of game logic implementation.

5.2.4 The graphic artist

The responsibility of the graphic artist was to produce the artwork required for the game. In our game the work of the graphic artist consisted of making backgrounds, a game logo, an introduction screen, a credits screen and an ending screen.

5.2.5 The sound engineer

The sound engineer was responsible for recording all the sound effects and musical soundtracks required in the game. For the sound engineer the recording of speech became the most time-consuming task.

5.3 The planning phase

Even in the beginning of the project we faced problems. In this initial phase we had no clue of what was possible for us to achieve within our limited time constraints. At the same time the story writers were anxious to start writing the story. So we decided to start with the evaluation of adventure game development environments while the story writers began to write a story for a typical adventure game.

As a result of the evaluation (see chapter 4) we decided to use the adventure game development environment, AGAST. At the same time the story writers were ready with the story, but we felt we needed more information about: locations, objects, characters and puzzles.

With locations we mean the different rooms the main character could travel between. In *Serendipity* we divided the bookstore into six different views, two at the upper floor and four at the bottom floor. The objects in the game where used as a part the different puzzles. For example, to reach the upper floor, you had to get a candle in the candle box, lit the candle in the fireplace and finally place the candle in the candle holder. The candle lit up the stairs and eventually Sal could use it to climb up to the second floor.

5.4 The implementation phase

AGAST included an Integrated Development Environment (IDE) called Streamliner. In Streamliner it was possible to create a good structure. Folders for each location were created and in them we put a background picture together with a couple of bookshelves, objects, speech, music and other things associated with that location. In this way there were no dependencies between the different locations at all, so we could work with them in the order we desired.

Everything worked just fine until the first background image was to be created. It took nearly three times longer than estimated. This forced us to reduce the number of locations from twelve to six. The removal of six locations had a great impact on other parts of the game. For example, some of the puzzles that were located in the removed rooms had to be moved to one of the remaining ones. The puzzles that were impossible to move were removed from the game, which in turn affected the storyline.

The next surprise came when it was time to draw the main character. In the game the character had to be able to walk around in many directions, to pick up objects, to push objects, to talk and more. We realized that making all the required animations were far to

complex for us or anyone we knew, so we had no other choice but to visually remove the main character from the game. The solution was quite easy, we just turned the game from a third-person adventure into a first-person adventure. Before the change, the player commanded the main character to move and accomplish tasks. After the change, the player became the main character, which meant the world now was viewed through the eyes of Sal, just like in the well-known adventure game Myst.

We experienced the same problem with the other characters in the game. But we could not just remove them from the game, so we turned them into non-moving pictures. Creating animations for the around eight characters in the game would have broken our time plan seriously.

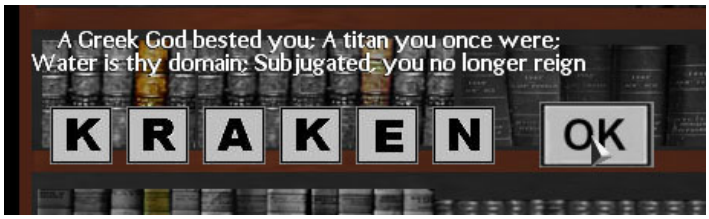


Figure 5.2 - *One of the riddles in the game.*

Adding more than hundred questions and riddles into the game was a very repetitive task. The questions were asked when the player clicked on one of the books and they could be answered with three different alternatives, the right answer gave 10 points and wrong answer removed 5 points from the total score. The riddles were answered by combining a number of dices into the right word, as you can see in figure 5.2.

5.5 Time measurements

In table 5.1 we have chosen to present the total amount of time spent on five different activities. Read section 5.2 to find out which working tasks are associated with each role.

Activity	Weeks spent	% of total work
Story writing	5	15
Game design	11	34
Programming	8	24
Drawing	6	18
Sound recording	3	9

Table 5.1 - *The time we spend during the development of Serendipity.*

We spend most time on designing, which was an expected result. The reason for this is that much time was spend on writing down questions and riddles for all the books in the bookstore. Also much time was spend on coming up with puzzles. Another possible reason is that we had three story writers in the project and we wanted to make use of them as much as possible.

When it comes to graphics, it should have taken much longer time to draw the needed artwork if we had not downscaled the game by removing a number of backgrounds and characters. If we had completed the game we planned for in the beginning, the making of graphics would have taken at least 50 % of the development time.

The time needed for programming was reduced drastically by the fact that we used a scripting language instead of building everything from scratch. It was the programming of all special cases, such as the riddle engine, that took most of our time.

We contacted the creators of the adventure game *Eye of the Kraken* to find out how the relationship between story, programming and graphics turned out in their game. The answer was that they spent around 30 % of the time on the story, 20 % on the programming and 50 % on the graphics. These values confirm that the making of graphics is the most time-consuming part of a regular adventure game project.

Based on experience we have got from our game project, the optimal team for a adventure game project with the same time constraints and the same number of resources as ours, should be composed as follows:

2 graphic artists
1 programmer
1 game designer
½ story writer
½ sound engineer

We think that a team of this size needs at least one, but preferably two skilled graphic artists. With two graphic artists in the team the graphics will never be a bottleneck in the design phase and the quality of the graphics could raise to the same level as a commercial 2D adventure game. Only one programmer is needed if an adventure game development environment with a good scripting language is used and if the game does not differentiate from a normal adventure game too much. The background story can be written by a story writer in half-time and the same goes with the sound effects.

5.6 Summary

We faced many problems during the development of Serendipity. Most of them were graphics related. Before we started the project none of us were experienced in making graphics, which fooled us to underestimate the time it would take to draw the needed artwork. Most of the character animations we had planned to include in the game were so hard to create that we had to remove them completely.

We predicted that unexpected things like this could happen and that was the reason why we added flexibility as a criteria when we searched for a suitable development environment. If we had been using a development environment lacking a flexible scripting language, it had not been possible for us to remove the main character from the game. Generally we are satisfied with the development environment we used. Besides some smaller faults in the documentation and a couple of minor bugs in the scripting language, it worked as we expected from the evaluation (see chapter 4).

A linear story can work as a background story for a game, but it is absolutely not a game design document. We solved this by structuring the game in locations, objects, characters and puzzles, a solution we were very happy with. Though, it should have been done much earlier than we did in this project. It is very important that you early create a design document which includes all design decisions and aspects of the game. Without it, all team members will not work towards the same goal.

In this project the game design took most of our time because of the great number of questions and riddles we added to the game. If we had completed the game as we thought of

it from the beginning the graphics should probably have been the most time-consuming activity. If we had the opportunity to work in a project like this one more time, we should have added one or two skilled graphic artist to the project team.

6 CREATING ADVENTURE GAMES WITH STORY WRITERS

The purpose of this chapter is to present the positive and negative aspects of working with story writers in the project.

6.1 Linear story problems

When we started the project, none of the team members had any experience in making story-based computer games. The story writers were naturally experts in writing stories, so we simply started up the project by letting them write a story for the game.

Even though the story was very well written, it was impossible for us to use it as a template for the game, because it was, as all other stories, linear. If we had build the game exactly as the story told, it had not become a game, rather a movie. Games differ from novels, books and movies in the sense of its non-linearity. In a game the player must be given some freedom to able to affect the game world. You can read more about the problems of using linear stories in games in Appendix A.

Even though we could not use the story as it was meant from the beginning, it was far from useless. From it we could gather information about the main character, and about what age and what environment the game was going to take place in. An old bookstore turned out to be the central point of the game and the main character was a middle aged man called Sal. We agreed to use the story as a background story for the game and throughout the remaining project it served as a source inspiration.

To enhance or understanding of the locations, objects and characters in the game, we handed over three different templates to the story writers. Here is the description of Frank, the owner of the bookstore:

```
Character description of Frank
---------------------------
Name: Frank
Age: 55
Attributes: -
Background: owner of the store.
Type: Non-playing character

Sex: male
Height: 1.70
Weight: 80 kg
Shape: a bit corpulent
Race: white
Hair: grey
Clothing: jeans, t-shirt, white socks, sandals
Makeup: none
Face: round face, beard and moustache, round glasses,
small nose, small mouth, blue eyes.
```

An adventure game also needs puzzles and we decided to use flow charts to be able to visualize them on paper.

6.2 Technical limitations

Just after the first meeting with the story writers, we received a mail explaining how they wanted the game to start.

> ***Game opens:*** *flows into murky city landscape, urban environment, night time, cloudy sky, smoggy, high up in the sky. Title flows around and coalesces into print – Serendipity – as the city comes closer into view.*
>
> ***View screen:*** *pans thru city, sidewalks with cracks and chips; gutter has litter and there is definite graffiti to be seen as well as drunks and lowlifes. Taken on a short trip thru the city; as this is happening there is some low background music, the weather is almost raining, and there are low rumblings of thunder in the distance.*

In other words, they wanted a fly-by camera to flow through the city. This could only be accomplished by developing the game in 3D, which we had no intention to do. Obviously we had not informed the story writers about what type of game we could build within the given time frame. After another meeting we agreed to lower the expectations of the game a bit and the intro ended up showing a picture (see figure 1) of the bookstore from the outside while a voice told a bit of the background story for the player.

Figure 6.1 - *The intro screen.*

6.3 Unclear game objectives

In the beginning we believed that we were going to develop a regular adventure game in keeping with games like Sam and Max and Myst. During the first meeting with the story writers and representatives from the English Department, though, it was indicated that the

game had to be addressed towards literature students like themselves. After leaving that topic of discussion it went at least a month before we brought it up again.

For long we believed that the fact that the game was taking part in a bookstore was reason enough to call it “a game for literature students”. But the story writers wanted to go one step further and added more than hundred literature related questions and riddles into the game. Since the story took place in a bookstore we decided to create one clickable book for each question and riddle.

Some of the books in the bookstore were special and push the story forward. These special books caused a so called encounter to happen, which triggered a scripted dialogue and the player had to answer some questions related to the characters encountered.

One could say that the game turned from a regular adventure game into an “Edutainment” game for testing literary knowledge. It is nothing wrong with that, but maybe the direction of the game could have been decided somewhat earlier.

6.4 Summary

It was very useful to have the story writers in this project, not only because of the story they wrote, but also because of all the ideas and inspiration they gave us. Apart from the story writing, they helped us with the different puzzle concepts, how the environment of the game should look like and all the questions and riddles included in the game.

Without the story writers, the background story of the game would not have been as good as it now became. But writing a linear background story is one thing and writing a non-linear storyline with different branches is a completely different thing. The person who writes the non-linear storyline for a game, must not only have a deep knowledge of game mechanics – but also needs an understanding of what is and is not possible to do with the technology of today. The story writers in our project did not possess this understanding fully. We could see this as much of their work had to be rejected because of technical difficulties. An experienced designer could have seen this before the work was handed over to the programmers.

Most of the problems during the project were related to insufficient communication between the programming group and the story writing group. Since we were working from two different locations, most of the communication had to be handled through e-mail or phone. Meetings where booked when someone had something important to discuss, but there were not more than five or six meetings in total throughout the whole project. As the story writers only worked half-time with this project, they could sometimes be hard to reach.

If we had been sitting in the same project room and worked together the whole time, we would probably have reached a better result. The next best thing would have been to arrange more meetings, especially in the beginning of the project.

7 DISCUSSION

An important part of the evaluation is the assessment of the different criteria. In our evaluation we did not specify the exact requirements for a certain grade and criteria. Instead we tested through all the different environments and tried to make a fair judgment for each criteria when considering all the different tools. This makes the evaluation process longer as we had to handle all the environments at the same time. Instead we should have specified the grades of all the criteria and then assessed one environment at a time.

The most important part is to select the right development environments to evaluate. With the best ones missing the evaluation gives a misleading result no matter how good the weights and the assessment of the grades are. We did an extensive search and we think we found the best ones for this task.

Regarding the questions and riddles added into the game, we think it is possible to use the game as a learning tool in many different fields. Now we decided to add questions about literature, but it could have been any type of questions. For example we could have added questions about computers, sports or nature instead. We have realized that instead of hard coding every question into the game, we should have put them into a text file. Then it would have been much easier for us and other people to add new questions into the game and in that way use the game as a learning tool for different fields in school for example.

8 CONCLUSIONS

Selecting the right development environment is important, we spent much time evaluating the available environments and it was time well spent. The game development environment best suited for our project turned out to be AGAST. Even after we had used the tool to create the adventure game, we believe it was a good choice and that our evaluation worked well. We discovered some minor problems with the documentation and scripting language in the environment, but nothing that extended the development time considerably. Because of the flexibility criteria we did not have any problems when we had to change from third person to first person view in the game.

When developing the game we became aware of the difficulties in creating graphics. It took three times longer to produce the graphics than estimated and therefore we had to remove some features from the game. For example the main character was too complex to animate, so we had to change the third person view to a first person view and because of this we could remove him from the game completely. In future projects we will definitely make sure to have one or more graphic artists in the project teams.

It is obvious that the background story is greatly improved by using skilled story writers when developing games, but during the project we discovered problems with using a linear story as a template for a game. We had to find another way of designing the game and come up with four basic concepts: locations, objects, characters and puzzles. This worked great and in future projects we will make sure that a game design document will be created before programming and making of graphics and sounds starts.

Another conclusion regarding the story writers is that they should either keep to writing the background story and let other people, more knowledgeable about game development, take care of the game design, or else they should try to learn more about game mechanics and development. A good story does not mean good game play.

9 REFERENCES

[1] Advent,
http://www.lysator.liu.se/hackdict/split2/advent.html

[2] Tim Anderson, The History of Zork - First in a Series, July 1995,
http://www.csd.uwo.ca/Infocom/Articles/NZT/zorkhist.html

[3] Hector Briceno, Down From the Top of Its Game - The Story of Infocom Inc, December 2000, ftp://ftp.gmd.de/if-archive/infocom/info/infocom-paper.pdf

[4] History of Dungeons and Dragons,
http://bayrps.com:8000/history_of_dungeons_and_dragons.htm

[5] Greg Costikyan, Where Stories End and Games Begin, Game Developer Magazine, September 2000, pages 44-53.

[6] The History of Sierra Entertainment,
http://www.geocities.com/petter_holmberg/history1.html

[7] The History of LucasArts,
http://www.lucasarts.com/20th/

[8] What's happening to adventure games?,
http://www.mobygames.com/featured_article/feature,13/section,57/

[9] Andrew Rollings and Dave Morris, Game Architecture and Design, The Coriolis Group, 2000.

[10] John Scott Lewinski, The Role of Story Bibles in Games, Game Developer Magazine, February 1998, pages 42-49.

[11] Toby Gard, Building Character, Game Developer Magazine, May 2000, pages 28-37.

[12] Mark DeLoura, Telling Stories, February 2001, Game Developer Magazine, February 2001, page 4.

[13] Warren Spector, Remodeling RPGs for the New Millennium, Game Developer Magazine, February 1999, pages 37-46.

APPENDIX A. EXTENDED BACKGROUND

This appendix extends the background about the importance of stories in games, the elements of adventure games and the history of adventure games.

The importance of stories in games

> *"At some point the environments look the same to the player no matter how many extra polygons or texture passes you add. We've reached that point – now we must turn our focus to story. A good story is much more accessible to people than snazzy technology."* [12]

Since the use of 3D graphics became common in computer games it has been possible to sell computer games by just adding some fancy graphics into them. Today it still is, just watch the upcoming first-person shooter Doom3 by id software, it will most likely sell millions of copies just because of its absolutely magnificent, close-to-photo-realistic graphics. But it is not that easy anymore, nowadays it is only a few, if any, titles every year that can present so original and stunning graphics that people buy it just because of the visual experience.

One of the reasons why the importance of graphics in games has been reduced is that many games of today use the same third-party 3D engines and therefore look the same. Another reason is that we are already close to photo-realistic graphics, which is the ultimate goal to reach for every graphics programmer. As we are so close to the perfect graphics, the step between good and bad graphics are getting smaller and smaller. The last and probably most important reason is that people are getting tired of all those brainless shoot 'em up games and now wants something more.

More can mean better game play but also better and more interesting storytelling. People want a reason for killing monster, a reason to build or destroy things, a reason to score goals. In many games it is not necessary to tell a story for making the players understand the meaning of the game. For example, in a soccer or hockey game, everybody knows that the goal is to score as many goals as possible and win the game. In Tetris the goal is to stay alive as long as possible and getting much points. But in many other types of games, such as adventure games and role-playing games, the storytelling is the biggest reason to play the game.

When you read a book or watch a movie you expect it to tell a good and fascinating story. But when it comes to computer games it is not that obvious. So many games with extremely bad stories have been released that the reputation of computer games as a storytelling medium has been very low. But things were about to change. When the first-person shooter Half-Life was released in November 1998 it became known world-wide because of its extremely captivating story. The success of Half-Life opened the eyes of game developers all over the world as they saw that good stories actually can sell games. Since then we have seen a major improvement in the quality of the storyline in computer games.

The difference with storytelling in computer games compared to other storytelling mediums is the non-linearity. Books, theater and movies are all linear, which means that events occur in the same order and in same way each time you read, watch or listen to them. A story is a controlled experience where the author carefully choose events, in a certain order, to create a story with maximum impact. In games things do not always happen in a predefined order.

> *"To the degree that you make a game more like a story – a controlled, predetermined experience with events occurring as the author wishes – you make it a less effective game. To the degree that you make a story more like a game – with alternative paths and outcomes – you make it a less effective story."* [5]

As Greg Costikyan points out, the challenge is to find a balance between game play and story. The most common way to solve this problem is to give the player freedom to move about a constrained space and let him or her solve puzzles or complete tasks in that area in arbitrary order. When all the tasks in the area are solved, something that drives the storyline forward happens and then access is given to a new area of the game world.

The elements of an adventure game

It is not crystal clear how to define the adventure game genre, but most of the games we refer to as adventure games include a storyline, some kind of puzzle solving and the possibility to explore worlds. In "Grim Fandango", for example, you travel through the Land Of The Dead – a world taken from the Mexican folk art and film noir. And in "The Secret Of Monkey Island" you are thrown into a pirate world where you take the role as a pirate called Guybrush Threepwood. These are only two of hundreds of adventure games that let you explore worlds you could ever imagine existed.

A major difference between games and other mediums are that games are nonlinear, while novels, books, theater and movies in opposite to games are linear. A story is linear because all events occur in the same order each time you read, watch or listen to it. In games things do not always occur in a pre-defined order. At least things must feel like they are not pre-defined, because otherwise players are not active engaged and they will probably stop playing the game.

Classic adventure games from Infocom, Sierra or LucasArts are all built upon five different building blocks. These are locations, objects, characters, tasks and events. An explanation of these building blocks follows:

Locations

A location (or a place) can be a room, a garden, a road or anywhere in a game where the course of action is taking place. A location is never meaningless; there is always a character, an object or something else that can be used to the advance in the game.

Objects

Objects are important elements in most kind of adventure games. You can get objects from characters or you can find them in hidden places. Another possible way of getting an object is by combining two different objects with each other so it becomes one. The objects can in turn be used to open up chests, doors, secret passages or you can use them as gifts to characters. A possible scenario can be that you have a bottle of wine and give it to a man who is watching a door you want to go through. When he gets the bottle, he drinks the wine, get drunk and falls asleep. Now it is safe to open the door and continue the game.

Characters

Many games are built upon character conversation. In a conversation with an NPC (non-playing character) clues about puzzles in the game might be revealed. The usual way to handle conversation in adventure games is to present three or four different choices for the player to answer questions from the NPC and to ask new ones to him.

Tasks

Tasks are all the things you can command the main character to do in the game. Often in adventure games there are pre-defined types of commands you can perform on characters, things and objects. Commands like “use”, “push”, “get”, “look” and “talk” are the most commonly used in graphical adventures, but in many games “kick”, “punch”, “jump”, “run” and more can exist. Text adventures often do not specify what kind of verbs you can use, so you have to guess which of them the game engine understand.

Events

Events respond to tasks. When a task is completed it is often followed by an event. An event can be as simple as a text message describing an object or when a main character a saying something. If you for example have a football and choose to look at it, the main character may respond with a statement like: “Hmm, a football. I’m wondering what is inside it”. This is for an experience adventure gamer an obvious clue that tells you that something is inside the ball. And the obvious solution is to use a knife to cut it open and get access to the thing inside it. When you order the main character to use the knife, it is a task, but the execution of that command is an event.

The history of adventure games

The era of adventure games started when William Crowther created ADVENT in 1972. The game is better known as “Adventure” or “Colossal Cave”, but the operating system it was written for permitted only six-letter filenames and uppercases [1]. In 1976 Don Woods at Stanford expanded “Adventure” into a puzzle-oriented game. “Adventure” was entirely text-based (the genre is also known as “interactive fiction”) and was about solving puzzles, examining different locations and picking up and using various items. The game was controlled by text input, for example if you wrote “Go north”, the main character might move north, or the game might respond "There is no way to go in that direction." If you had discovered a bucket of water in a house and wanted to pick it up, you could write “Take bucket,” the dungeon master could then answer “The bucket of water is too heavy”. It was all up to the designer of the game to decide what was going to happen in different situations.

The probably most known text adventure, “Zork", was created between 1977 and 1979 by Tim Anderson, Marc Blank, Bruce Daniels and Dave Lebling from the Dynamic Modeling Group in the MIT Artificial Intelligence Lab [2]. Here follows a simple example of the interplay between player input and game input in Zork:

```
West of House
You are standing in an open field west of a white
house, with a boarded front door. There is a small
mailbox here.
```

```
>OPEN THE MAILBOX
Opening the small mailbox reveals a leaflet.

>TAKE THE LEAFLET
Taken.

>READ IT
"WELCOME TO ZORK!"

ZORK is a game of adventure, danger, and low cunning.
In it you will explore some of the most amazing
territory ever seen by mortals. No computer should be
without one!
```

When “Zork” was completed the people behind the game decided to start a company named Infocom. Infocom was founded in 1979 and a year after that, it released a commercial version of “Zork” for the TRS-80 Model I, beginning its entry into the software entertainment industry [3]. After “Zork” followed a row of successful text adventures from Infocom.

Both “Adventure” and “Zork” where inspired by “Dungeons and Dragons”, a fantasy simulation game (not computer based) invented by Dave Arneson and Gary Gygax in 1972 [4]. D&D was a role-playing game, which is not included in the adventure game genre. But there existed other non-computer games that where close to real adventure games. The which-way book (also called “game book” or “choose-your-own-ending” book) became popular in the mid-1980s [5]. A which-way book starts like a normal book by telling a story, but after a couple of pages you are faced with a decision. Depending on what you decide, you are directed to another page. For example, if the question can be answered with a “yes” or a “no”, the first option may lead to page 11 and the second to page 18. If you choose the wrong path you will most likely die and are therefore forced to start over again. When you have found the right answer for every quest you will ultimately reach the last page and the “game” is over. The most advanced type of which-way books, which have more similarities to a real game, include some kind of combat system where you can fight against daemons and other kind of evil creatures to decide the path you’re going.

Later on computers became more and more powerful and in 1980 the first graphical adventure game entered the market, namely “Mystery House”. It was created by “Sierra On-Line”, a company founded by Ken and Roberta Williams [6]. The game itself was much like the text-based adventure games, but illustrated the different locations with simple uncolored graphics instead of text. In 1994 Sierra released a game called “King’s Quest: Quest for the Crown”, which was the first game that had similarities with the adventure games of today. In that game you were able to move your character freely in a graphical world, but still you had to write text input to perform other actions than the walking. Other popular game series from Sierra were “Space Quest” (1986) “Leisure Suit Larry” (1987) and “Police Quest” (1987).

The next step in the evolution of adventure games was taken by a company called “Lucasfilm Games” (today known as “LucasArts Entertainment”) when their programmer Ron Gilbert developed a point-and-click interface called SCUMM (Script Creation Utility for Maniac Mansion) [7]. With this new system, the need of a keyboard was removed, now everything could be done with the mouse only. The first game released with this system were “Maniac Mansion” (1987) followed by “Zak McKracken” (1988), but they were both outshined by Lucas Arts’ maybe greatest game ever - “The Secret of Money Island” (1990). Monkey Island was followed by a line of successful games, such as “Indiana Jones and the Fate of Atlantis“ (1992), “Day of The Tentacle” (1993) and “Sam & Max Hit the Road” (1993).

When the CD-ROM entered the market, new possibilities were brought to the game developers. A single CD-ROM stored 650 MB of data, which meant it was now possible to store hundreds of high-resolution pictures or movies in an easy way. One of the first games that took advantage of the new media was Trilobyte's "The 7th Guest". It was released in 1993 and offered high resolution SVGA graphics along with fully pre-rendered Full Motion Videos (FMV). The game play was basically about solving a series of logic puzzles and if you succeeded, a movie clip was shown and advanced the plot. The following year the most popular and profitable adventure game ever, "Myst", was released. "Myst" was created by Rand and Robyn Miller and it was actually a very simple game [8]. "Myst" was, as "The 7th Guest", a puzzle-solving adventure game. There was no combat, one reason for that might be that the main character was the only living soul in the game. The strength of the game was instead the excellent graphics, which consisted of a number of pre-rendered high-resolution images, and the extremely difficult puzzles that most of them were of the same kind as the ones in "The 7th Guest".

"Under a Killing Moon" (1994), "The 11th Hour" (1995) and "Phantasmagoria" (1995) are also well-known games that were based on FMVs. "Phantasmagoria" came on 7 CDs, consuming over 3 GB. It was a lot at that time and it is still the largest game ever. These titles were often referred to as interactive movies rather than games. Interactive movies may be beautiful and may create a good atmosphere, but they are not so interactive as a game should be. Players cannot interfere in dialogs, as many of them are in full-motion video. And the players do not have control when videos are played.

1995 was certainly the year of glory for interactive movies. Even LucasArts released a game that was close to an interactive movie, namely Full Throttle. It was shorter and easier then previous released games by LucasArts and included a lot of cinematic cut-scenes. There was also some real-time action in the game, for example in one scene you had to beat up a rival biker on an abandoned mining road. Action scenes were not until this point in time anything you associated to the adventure game genre, but from here on then action-adventure genre became more and more accepted as traditional adventure games.

In 1996 Core released "Tomb Raider", an action-adventure in a 3-D world where puzzles were solved by fighting enemies, open secret passages etc, not by character dialogue and combining objects as in the classic adventures [8]. The action-adventure genre started with games such as "Prince of Persia" (1989), "Another World" (1991), "Alone in the Dark" (1992) etc., but are today among many people incorrectly referred to as pure adventure games.

Even though action-adventures became the most popular games, a number of good classic adventures were still to come. LucasArts released another two game based on Monkey Island – "The Curse of Monkey Island" (1997) and "Escape from Monkey Island" (2000). Between them they also released "Grim Fandango" (1998). It was their first game with 3D rendered characters and it became a huge success and is even today referred to as one of the best adventure games ever by many adventure fans.

APPENDIX B. TERMS AND ABBREVIATIONS

This appendix contains explanations for the terms and abbreviations used in this thesis.

GIF	Stands for Graphics Interchange Format and it is a format for storing pictures
Sprite	Format used to store animated pictures
First person view	Seeing from the perspective of the main character of the game
Third person view	Term used when describing seeing an object, for example the main character, from a distance when played in a game.
Env.	abbreviation for "Environment"
Point-and-click	Expression used to describe, for example, development without any need to use scripting
Background story	A text used as inspiration for the game designer
Game engine	Software, which provides functions for easier game creation
Linear story	A non-interactive story, which makes it impossible to affect the outcome, for example reading a book

APPENDIX C. LINKS

This appendix contains links to the development environments and game development companies mentioned in this thesis.

Development environments

Adventure Game Studio	http://www.adventuregamestudio.co.uk
AGAST	http://allitis.com
SLUDGE	http://www.hungrysoftware.com/#/tools/sludge
Adventure Maker	http://www.adventuremaker.com
Wintermute Engine	http://www.dead-code.org
ADRIFT	http://www.adrift.org.uk
SCI Studio	http://www.bripro.com/scistudio

Game development companies

LucasArts	http://www.sierra.com
Sierra Entertainment	http://www.sierra.com
Massive Entertainment	http://www.massive.se
id Software	http://www.idsoftware.com
Broderbund	http://www.broderbund.com

Other

Eye of the Kraken	http://kraken.absurdus.org