Literature review and system design

Contents

Literature review

This section will include a review of several computer game development books and articles. Their usefulness with respect to the game I will develop is also evaluated.

1. Better Game Characters by Design: A Psychological Approach [1, 2]

This book deals with character design within games. It discusses how real-world people react to certain events and explains how these reactions can be incorporated into games. It focuses more on realistic human behaviour and psychology than the technical side of computer game creation or Computer Science. A large part of the book is devoted to animation styles and how to portray the emotion of in-game characters, in particular characters body language and movement in relation to other characters. All techniques discussed in the book are reinforced by examples. There are even interviews with expert game character designers and researchers pushing the boundaries of the social behaviour of game characters.

In general the book seems to focus more on large-scale games for marketing and not any game I could develop in the limited time frame of this project. It may be a useful book for future work or in computer graphics and animation courses.

2. Rules of Play: Game Design Fundamentals [3]

Described as "...a monumental examination of the emerging field of games design..." by The Guardian in November 2003. The book adopts a unified model for all types of games be it board games, sports, computer games, etc. The book has been reported to have been used by some professional game creators. It defines many core features of games such as design and interactivity. It is heavily based on game design theory as an emerging field that is being seen increasingly more as a serious profession. With 672 pages and small print it is a large book that may be difficult to understand by novices to games development. It covers every aspect of computer game use from a story telling medium to a communication method. It is divided into 4 sections with a different game design document written by experts in the field, at the end of each.

The book was published in 2004 so may be starting to show its age but it is one of the only books available that focuses on game design theory without focus on practice. It is a good book to have for anyone considering games design as a profession. However, with 672 pages, it presumably goes into too much detail than the example game to be created would need.

3. Programming Believable Characters for Computer Games [4]

This book is aimed at game programmers rather than game designers but therefore does involve Computer Science more than the other books reviewed. It focuses more on enemies in games but does explain artificial intelligence (AI) well using familiar techniques like finite state machines. While not attempting to push forward any boundaries it does give programmers tried and tested ways of creating believable AI in games.

The book focuses on the step-by-step creation of 3-dimensional (3D) animated characters with relation to common AI techniques used in Computer Science. These techniques include path-finding, rule-based systems, goal-orientated planning and decision trees. These can be applied to AI in general and not just to game programming.

Since it is focused more on 3D characters it does not apply much to the game I can create in the restricted time scale of this project.

4. AI Game Programming Wisdom [5, 6]

This book contains a collection of formal articles by experienced authors concerning useful techniques that can help game developers avoid reinventing the wheel. There is a large section of the book devoted to machine learning believing that the next major development of Computer Science and computer games will be to create a machine that learns. Complex AI techniques are broken down into easily understandable sections and most are accompanied by program code or pseudo-code. It includes descriptions of A* path-finding which is often used in commercial games. Some of the examples can be hard to follow without a good programming background. As the articles are written by professionals in the game industry they often explain decisions made and problems encountered in game development projects. Many articles make use of UML diagrams and C++ but a detailed understanding of C++ syntax is not required. The book focuses a lot more on AI development for PCs and does not take the restricted resources available to consoles or handheld computers into account.

It is definitely not for beginners to computer science but when you have some experience and understanding of computer systems it becomes a good resource for taking common computer science techniques and applying them to games. From the review it goes into more detail than needed for the AI I will need to create for the game.

5. Programming Game AI by Example [7]

The book includes many AI techniques used in Computer Science and computer games. Explanations of concepts are entertaining to keep the reader interested and helpfully depicted with diagrams and illustrations. It discusses Finite State Machines for state based AI and Messaging so objects within the game can send messages to each other to co-ordinate attacks for example. Scripting, fuzzy logic, goal-oriented AI using states and path tree search are also covered. The book also covers path-finding in large detail with many descriptive diagrams. A plus side for the game I am to develop is that all the examples in the book are based upon 2-dimensional games. That said the techniques discussed can still be applied to 3-dimensional games. The book does assume the reader is familiar with C++ and STL (Standard Template Library), which I am not.

This book may be of use in the project if the AI of computer-controlled characters is developed further.

6. Game Design Complete [8]

This book understands that most games are developed under constraints, be it hardware limitations, marketing issues, time, money or sales potential. The book tries to make you think of these as more of an opportunity to create exciting and original games rather than a problem. It covers many aspects of game design including designing for licenses when a game is based on a film for example and how to design games for technology that has significant limitations for example, limited memory or limited display size. It also covers controversial design ideas such as dealing with strange and challenging environments such as Mars or the arctic. The book features many useful design techniques and modern approaches. It even includes a section on Disaster Management for when things go wrong in game development.

Overall, I think this is a very good book for game design which is a very important part of game development. After all, if a game isn't designed well it won't sell well and may be difficult to implement. It may be of use for the game I will develop.

7. The Game Maker's Apprentice: Game Development for Beginners [10]

This book is a must for all beginning game developers. It takes you through all the stages of game development from having no idea for a game to implementing state-based AI using Finite State Machines. It is written by both former computer scientists and current professionals in the games industry.

Since I will be using Game Maker to create the game this book is essential. There is a website with very useful tutorials that can be downloaded and a list of other recommended game design books.

8. Game studies: Where the action is [11]

This website contains links to many academic papers analysing computer game design. In particular "Formal models and game design" describes how games are thought of as systems which are designed using mathematical models. Many of those models are theoretical including state-transitions, mappings and logic but there are also more practical techniques related to computer science. The article also describes why formal methods are important when designing a game in the same way that they can be important for designing any computer software.

Most of the mathematical models it gives appear far too complex for the game I will create in this project. In my opinion I would need to consider using these formal methods if I was creating a marketable game. That said they do have the disadvantage of requiring special knowledge to use in the same way that formal methods used in any other computer projects require. Also the use of formal methods in design would, most likely slow the entire project which I can not afford.

9. Gender and Computer Games: Exploring Females' Dislikes [12]

This article shows results of studies into the gender of game players and describes how these relate to game design. It provides many statistics, facts and figures. In general it shows that women play games a lot less than men and when they do they play for a shorter time. The main reason for this is the lack of meaningful social interaction and the violent and sexual content of most games. It also suggests that women gamers do not like the competitive elements of games.

I have seen many articles on the debate about the gender of players. The game I will develop will not involve much violence since this is a requirement. There may not be much competition in the game I will create, but I think competition adds to the enjoyment of a game. Many game design books support this argument.

10. Game Research - Strategy [13]

This article focuses on strategy games. These are games in which you control several characters called units. The article describes how strategy games started as turn-based games

where the player moves and gives orders to there units in one turn. The computer or other player then takes their turn, moving units and constructing buildings. Play continues in this fashion with a break between each turn. Strategy games have evolved into "real-time" games where there is no concept of turns and every action takes place almost immediately. Every player, including computer controlled players, play (for example, ordering units to move) at the same time so a useful analogy is to think of each turn in a turn-based game being joined to form one overall turn.

It would be more difficult to design a strategy game that met all the main objectives given in the "Project specification and Plan". I do, however, need to keep strategy in mind during the design stage of the project. Since the game is required to educate the player in the basic concepts of survival via the collection and use of in-game items it will need some elements of strategy and planning for success. I believe making players think for themselves and form their own strategies is a good way to educate them.

Design

In this section I will design the game that will be created. The game definition, objectives and requirements given in the "Project Specification and Plan" document will be taken into account. I will adopt a Software Engineering process for the design including the use of UML diagrams. The design will be split into five parts. The first part is the design for the final version of the game. This will endeavour to meet all the game objectives and main objectives of the project. The second part will be the design for the prototype system that will meet some but not all the stated objectives. Part 3 includes desired functionality which should be considered optional and may or may not be included in the final version of the game. Part 4 summarises how the game created will be tested. Part 5 includes a risk analysis and a brief description of an alternative game development environment.

Part 1: The final version of the game

1.1 Overview

The game will have the title Hunger and run on a PC. It is a game that focuses mainly on survival with limited reliance on fighting enemies or horror. It provides some limited educational content and aims to show the basic concepts of survival for example, the need to eat and drink. There will be enemies but they are mainly enemies to be avoided instead of fighting directly. As described in later sections the game provides the user with choices as to how to play while keeping focus on the main objective of the game as stated in section 1.2.1.

The game requires the user to think about when to use in-game items for maximum effect in order to survive. The majority of the game is 2-dimensional with a top-down perspective.

The game can be divided into five main areas of design.

1.2 Gameplay design

This is the design of the actual game that is played. In general all gameplay will be 2-dimensional and viewable only from the top down.

1.2.1 Game summary and main objective

The game will be set in a town called "Fondville" which consists of an island connected to the main city via two bridges and a cable car. The island is divided into 3 main areas as described in the "Level design" section. The two bridges are inaccessible so the player (user) must guide the playable character from the southern end of the island to the cable car in the north in order to complete the game. To make things more difficult the town has been overrun by zombies called 'Rot'. These will act as enemies and attack the playable character. They are entirely controlled by the computer. There will also be computer controlled characters called survivors which the playable character will meet at various stages throughout the game. These will provide the user with survival tips and tell them how to play the game successfully.

In order to educate the user in the basic concepts of survival, various stats will be maintained as described in section 1.2.3. The playable character will be able to move over and pick-up in-game items which are then stored in the inventory described in section 1.2.6 on page 13. These items can then be used to increase or decrease the maintained stats. The user must therefore manage the stats of the playable character while trying to progress and not be killed by the Rot.

The playable character will be able to move around the game world by key presses on the keyboard. They will never be able to move out of viewable area of the game world or through solid objects.

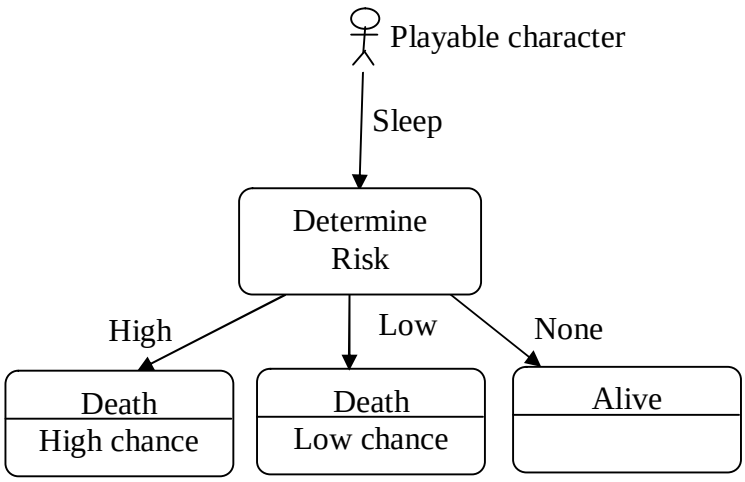
Information about the game story, the text that is or can be displayed by survivors and other details are given in the "Detailed Design" document on the project website [15]. A high-level flow control diagram of the game is given in the "Flow Control Diagram" document on the project website [15].

1.2.2 Sleeping

During gameplay the user can instruct the playable character to sleep. Sleeping is used to affect the "Fear" stat described in the next section. Sleeping will be indicated with a static screen and will temporarily stop gameplay. There is a progress bar to indicate how long until the sleep cycle ends. When the current sleep cycle ends the playable character "wakes up" and gameplay is restored to normal.

Since the game world has become inhabited by Rot obviously the playable character should not be able to sleep anywhere with no risk of being killed. There are 3 categories of place where the playable character can sleep. It is safest to sleep inside barricaded buildings. Buildings are initially unbarricaded but can be barricaded using the barricade item described in section 1.2.5 on page 11.

Where character sleeps	Risk	Description
Outside	High	Random chance of death. High chance.
Inside a unbarricaded building	Low	Random chance of death. Low chance.
Inside a barricaded building	None	No chance of death.



A summarised statechart diagram of Sleeping

When the playable character dies as a result of sleeping a dialog box is displayed informing the user of the death. When the user clicks the button labelled 'OK' the playable character is removed from the game screen and one live is lost. After a short time the character reappears at another position on the screen.

1.2.3 Stats

These will be shown on the game screen at all times when the user has control of the playable character. The stats will be represented using horizontal bars that increase or decrease with their numeric value. The following is a list of all stats.

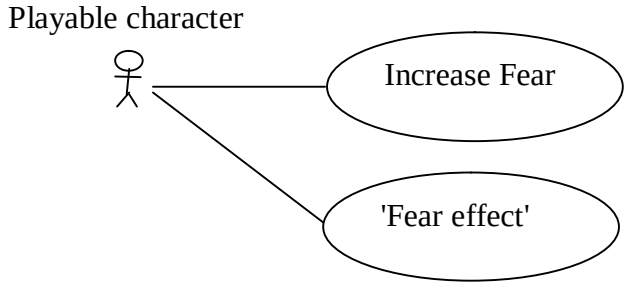
Stat	Want to keep	Description
Health	High	Decreases when playable character collides with Rot or if 0 of an indicated stat. Increases by using bandage or medkit item. When becomes 0 playable character dies. The playable character is removed from the game screen and 1 live is lost. After a short time they reappear at another position on the screen. If no more lives Game Over is displayed.
Warmth	High	Decreases over time when playable character not near fire. Increases when playable character is near a fire. When becomes 0 health slowly decreases.
Thirst	Low	Increases over time. Decreases by using drink item. When maximum value is reached health rapidly decreases.
Hunger	Low	Increases over time. Decreases by using food item. When maximum value is reached health slowly decreases.
Fear	Low	Increases slowly over time. Large increase if playable character "sees" Rot (explained in "Detailed Design" document on the project website [15]). When maximum value is reached the playable character moves around randomly at a high speed and after a short time sleeps where ever they are. The playable character needs to sleep to reset this stat to 0. Has the advantage of slightly increased movement speed at higher levels but the disadvantage of 'Fear effects' (described in next section).
Speed	High	Stays at constant level until Fear stat increases above a certain value then starts to increase over time making the playable character move faster.

To make it clear which stats the user should maintain at a high level and which should be at a low level they will be divided into two categories. Each category will be shown on a different half of the game window to provide symmetry.

1.2.4 'Fear effects'

'Fear effects' are experienced as the Fear stat increases beyond a certain value. They only occur at high-levels of Fear so as never to become annoying and predictable. There are many 'Fear effects' and when one is to occur (they occur at random past the threshold value) it will be chosen at random from the following list:

- Static Rot suddenly appear near playable character and then disappear again.
- The user sees items which are not there. When the playable character gets close to them they disappear. They also disappear after sleeping.
- The screen suddenly goes black for a few seconds and then returns to normal. While the screen is black the game is stopped.
- A large picture of a Rot is displayed on the screen for a few seconds. A sound file plays and then the screen returns to normal. While the picture is displayed the game is stopped.



A use-case diagram of 'Fear effects'

1.2.5 In-game items

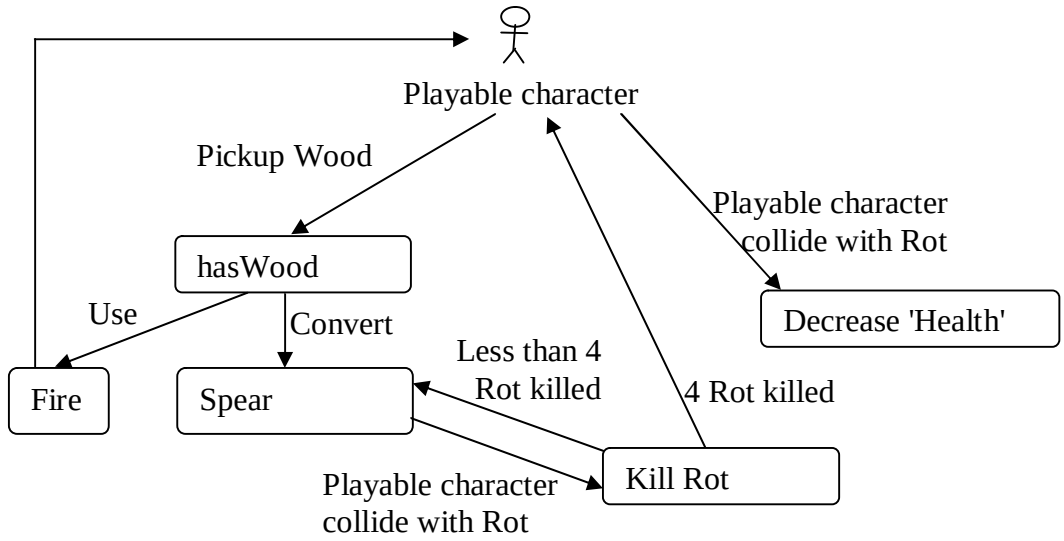
All items can be collected by the playable character in the game. To do this they simply move over of the item. Collected items can be viewed and used from the inventory designed in section 1.2.6 on page 13. When items are used none, one or more of the stats described in section 1.2.3 are altered.

Item	Effect when used	Description
Food	Decrease hunger	-
Drink	Decrease thirst	-
Medkit	Increase health a large amount	-
Bandage	Increase health a small amount	-

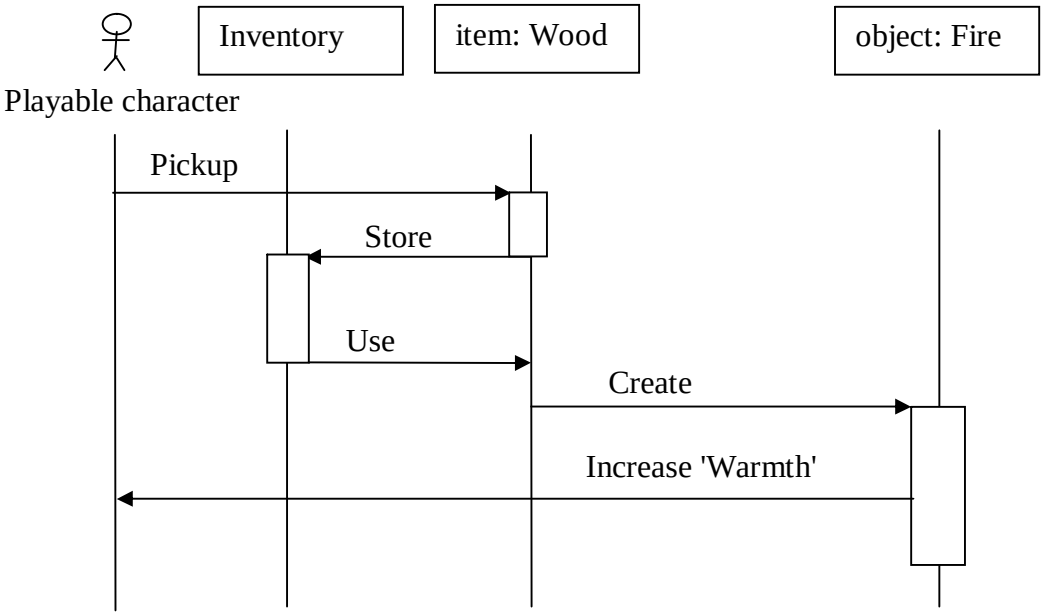
Barricade	Stops Rot entering buildings	Use of this item will make the current building that the playable character is in inaccessible to the Rot and therefore safe. Safe areas are desirable for sleeping. The playable character is not affected by used barricades and can move through them. Used barricades will be removed from the game after 4 sleeps.
Wood	Variable	This is a dual purpose item with a different use depending on the purpose selected by the user. • In its default state wood can be used to make a fire near to the playable character. This will increase their Warmth stat provided the playable character stays close to the fire. The playable character, enemies and other characters will be able to move through fires. Used fires will be destroyed after a set period of time. • If the user decides to convert the wood into a spear they can use it as a weapon to kill Rot. When the user uses a spear the graphic representing the playable character will change and they can then simply move into Rot to kill them. Once killed, Rot are removed from the game. After 4 Rot have been hit by the spear the spear will break and the playable character will revert to normal with the spear removed. Once the wood has been converted to a spear it cannot be converted back to use as wood.

In addition to wood being used to create fires and increase the playable characters “Warmth” stat there are also fire barrels. Like fires made with wood, fire barrels only increase the Warmth stat if the playable character is within a limited radius of them. Barrels are not moveable or collectable game objects.

Below is a state diagram for the more complex wood item.



An expansion of the path leading to the Fire state as a sequence diagram:



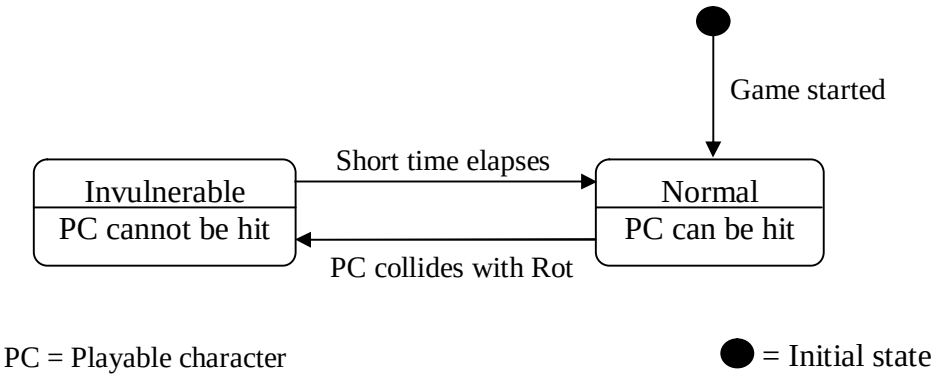
1.2.6 Using items and inventory

The game will allow the playable character to collect and store items. Collected items are viewable from the inventory screen. The inventory can be viewed at anytime while the user has control of the playable character. Gameplay will not be stopped while the inventory is open and the inventory screen will not occupy the entire game window. A diagram showing the layout of the inventory screen is shown in the "Detailed Design" document on the project website [15].

Items in the inventory can be used or dropped by selecting options from an action menu specific to the type of item clicked on. The "Detailed Design" document gives a table of the options in each items action menu. When items have been used they are removed from the inventory and cannot be used again. Dropped items are removed from inventory and placed at a random location near the playable character. All dropped items can be collected again and used as normal.

1.2.7 Being attacked

As mentioned before the enemies in the game are Rot and when the playable character collides with a Rot their health decreases. Upon this collision the playable character is also moved back slightly and made invulnerable for a short time. While invulnerable the playable character cannot be attacked and the graphic representing them is made partially transparent. This is summarized in the below statechart diagram.

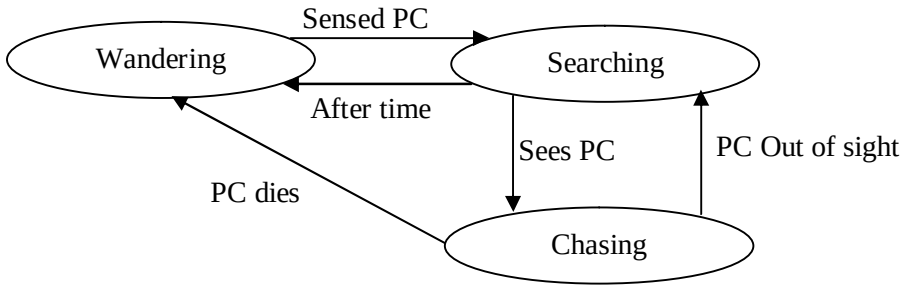


1.3 Artificial Intelligence design

As mentioned in the "Gameplay design" section there will be two types of computer controlled entities: Rot and survivors. Neither will be able to move through solid objects such as walls. This rule will not be represented on diagrams to avoid clutter.

1.3.1 AI of Rot

Rot will act as enemies within the game and attack the playable character. The following finite-state machine describes their behaviour.



PC = Playable character

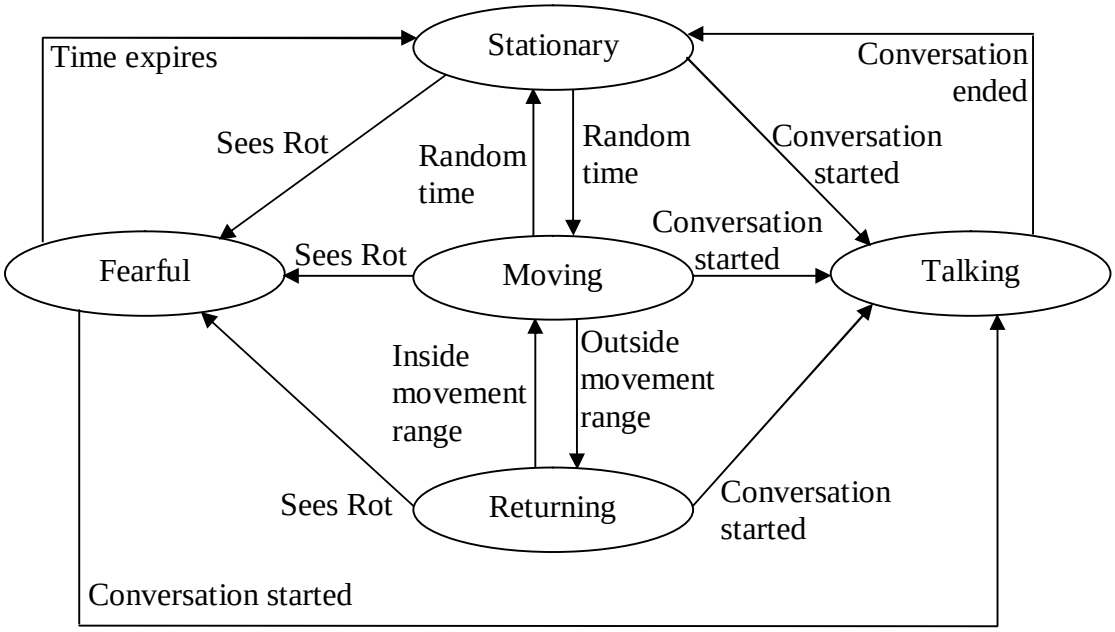
Initially Rot are in the Wandering state. Here they will move around at random with a slow movement speed. If the playable character moves too close to a Rot, that Rot will "sense" them and switch to the Searching state. Here they will move in the general direction of the playable character but not necessarily towards them. They will receive a slight increase to there movement speed. If a Rot in the Searching state then "sees" the playable character they will switch to the Chasing state and move towards the playable character. They will receive another slight increase to their movement speed. To "see" the playable character a line drawn from the centre of the Rot to the centre of the playable character must not be broken by solid objects, including other Rot. The graphic representing the Rot will be shaded red to indicate they are in the Chasing state.

If the playable character hides behind an obstacle and can no longer be seen by the Rot the Rot will switch back to the Searching state. Also if the playable character moves far away from the Rot (out of their visible range) the Rot will switch back to the Searching state. This will remove the red shading and start a timer. Once that timer has reached 0 the Rot will switch to the Wandering state as long as the playable character has not been seen by the Rot.

If the playable character dies while a Rot is in the Chasing state that Rot will revert back to the Wandering state. The red shading will be removed from the Rot.

1.3.2 AI of survivors

Survivors will act as friendly characters and advisors to the playable character. The following finite-state machine describes their behaviour.



The initial state for all survivors is Stationary. In this state they will not normally move. After a random time they will switch to the Moving state. In this state the survivor will walk in a random direction. When the survivor has walked far from their start location (original position in game world) they will switch to the Returning state and move towards their start location. When the survivor is close to their start location (within their normal movement range) they revert back to the Moving state. In this way they will never move far from their start location. After a second longer random time they will switch back to the Stationary state.

If the playable character is within a short distance of a survivor in any state except the Talking state, and the user presses a particular key on the keyboard it will start a conversation. This immediately transfers the survivor to the Talking state. Here they will not move but will turn to face the playable character. They will then choose a random statement from a list of possible statements to display in the "Conversation screen" designed in section 1.4.5 on page 18. If this was the first time that survivor was talked to they will display a special statement if they have one. Once a conversation has ended the survivor will switch to the Stationary state. The user can end a conversation by pressing the "End conversation" key on the keyboard.

If the survivor sees a Rot nearby the survivor will switch to the Fearful state. Here they move away from the Rot at an increased speed. This could occur in any state but is ignored in the Talking state. After a time, the survivor will stop and return to the Stationary state. If they can still see the Rot they will return to the Fearful state and move again. If they can't see the Rot

they will move back to their start location. When they are at that position they return to the Moving state.

1.4 None gameplay screens and interface design

This section covers the design of screens where the user does not control the playable character in the game for example, the main menu and end game screens. All none gameplay screens except the game complete screen use the mouse to select and activate buttons.

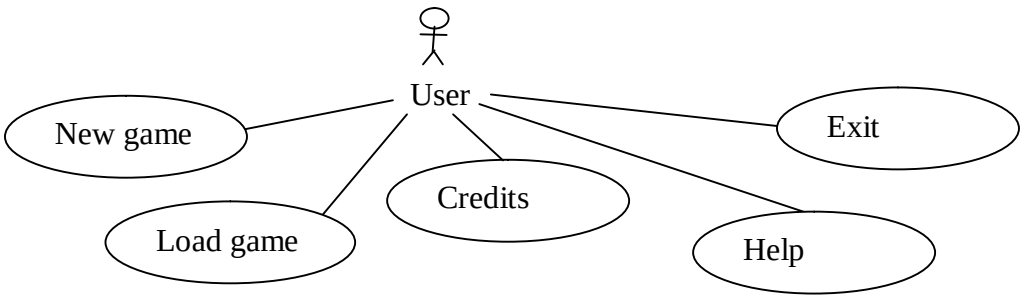
1.4.1 Main Menu

This will allow users access to the actual game and initiate gameplay. It will contain several buttons that can be clicked by the user to perform various actions. The buttons and associated actions are:

Button name	Action
New game	Begin a new game.
Load game	Load a previously saved game.
Credits	View the names of people involved in the project.
Help	Display the help information for the game.
Exit	Exit the game software and return to the operating system.

There will be a logo showing the title of the game and some text stating that the game does not provide a detailed or complete survival guide and only illustrates basic concepts.

A diagram showing the layout of the main menu is shown in the "Detailed Design" document on the project website.



Use-case diagram of the main menu

1.4.2 In-game Menu

There will also be a sub-menu that can be accessed while the game is being played. It will contain the buttons:

Button name	Action
Continue game	Close this menu and return to the game being played.
Save game	Save the current game.
Load game	Load a previously saved game.
Help	Display the help information for the game.
Quit to main menu	Quit the current game and return to the main menu.

A diagram showing the layout of the sub-menu is shown in the "Detailed Design" document on the project website [15]. Gameplay must be temporarily stopped while this menu is viewed and continued when the "Continue game" button is pressed. If development time for this part is short the in-game menu can be implemented as the main menu with some disabled buttons.

1.4.3 Game over screen

This will be displayed when the number of lives is equal to 0. It will contain the words “Game Over” in large text and provide 3 buttons:

Button name	Action
Return to main menu	Redisplay the main menu.
Load game	Load a previously saved game.
Exit	Exit the game software and return to the operating system.

A diagram showing the layout of the game over is shown in the "Detailed Design" document on the project website [15].

1.4.4 Game complete screen

This will be displayed when the playable character reaches the cable car at the end of the game. After 1 or 2 minutes this screen disappears and the main menu is displayed.

1.4.5 Conversation screen

This will be displayed when the playable character starts a conversation with a survivor. This screen will therefore be viewable while the game is being played. It will not occupy the whole of the game screen. The game will stop while this screen is being viewed and resume when the user presses the "End Conversation" key on the keyboard. If the conversation is larger than will fit on one conversation screen the user will have to press a key on the keyboard to view the remaining text on another screen. A diagram showing the layout of the conversation screen is shown in the "Detailed Design" document on the project website [15]. If time is short the conversation screens can be replaced by default Game Maker message boxes.

1.4.6 Credits screen

This will display the names of all the people involved in the project and their role within it. The names will scroll upwards. The screen will also include a button to return to and redisplay the main menu.

1.4.7 Saving and loading

At many points in the game the user can save their progress through it by clicking the save game button. The user is prompted for a name for the save. All game progress is saved in **<name>.sav** files where **<name>** is the name the user input. If **<name>** is the empty string or if it contains any of `< > \ / ? " * : |` it is said to be invalid and a new name is asked for.

If a file with **<name>** already exists a warning is displayed asking the user if they wish to overwrite that save game. If they do not a new name is asked for. If they do the game is saved and a conformation message displayed.

Loading of save games is done in a similar way. When a load game button is clicked the user is prompted for the name of the save game to load. In the same way as saving the game, this name cannot be invalid. If a saved game with the input name does not exist an error is given and no save game is loaded.

When saving the game or loading a save game, if the input name is CANCEL the prompt asking for a name disappears and no game is saved or loaded.

1.5 Level design

The game will be divided into 3 levels or "areas". Each area will usually be too big to fit in a single window. Views (provided by Game Maker) will be used on each area that follows the playable character as they move about that area. In each area there will also be a view that represents a smaller map or mini-map of that area. The mini-map will show the location of the playable character, location of obstacles, location of enemies, location of survivors and the location and type of in-game items.

Each area will depict a different part of the virtual town where the game is located. The area the playable character starts in (level 1) will represent the sub-urban village area of the game. That will connect to the thinner tunnel area (level 2). Finally, there will be a large area representing a more urban landscape (level 3). This last area will contain more enemies and

obstacles than the first two to make the game progressively more difficult as the user advances through it. The third area will contain a stylized picture of a cable car which the playable character can move into to complete the game.

Diagrams of level layouts can be downloaded from the project webpage [15]. A diagram of level 1 is given in the file "Level1.bmp". A diagram of level 2 is given in the file "Level2.bmp". A diagram of level 3 is given in the file "Level3.bmp". The designs shown in these files are **not** an exact match to level layouts used in the final version of the game.

1.6 Media design

This section covers the pictures used to represent in-game objects and any sound effects required by the game. Since the actual graphics and sound are not an important part of this project I will only list and briefly describe them in the “Detailed Design” document available on the project website [15]. If time is short some of these may be omitted or simplified.

Part 2: The prototype

This will:

- Allow the user to control the playable character on the screen via key presses. This character will not be able to move outside the viewable area of the game world.
- Contain **only** a single level which the character can move around in and collect items.
- Include all stats specified above that change dynamically.
- Store collected items and allow all those items to be used as specified above. Note this does **not** include 'Fear effects'.
- Include a main menu screen with all buttons. Many buttons will **not** yet have any functionality.

Note that any pictures and sounds used in the prototype may not accurately represent the final version of the game.

Part 3: Desirable functionality

These are extra functions that could be performed in order of most desirable. They are not required and therefore may not be implemented.

1. The inclusion of the newspaper item:

Item	Effect when used	Description
Newspaper	Discover game story	Use of this item will show a part of the games story. There will be a maximum of 4 newspapers and they will not be removed from inventory when used. Since there are only 4 in total there is not the problem of them taking up too much inventory space. The text they will show when used is given in the "Detailed Design" document on the project website [15].

2. Via the in-game menu the user can access a map of the town to check their progress through the game. This will require an extra button:

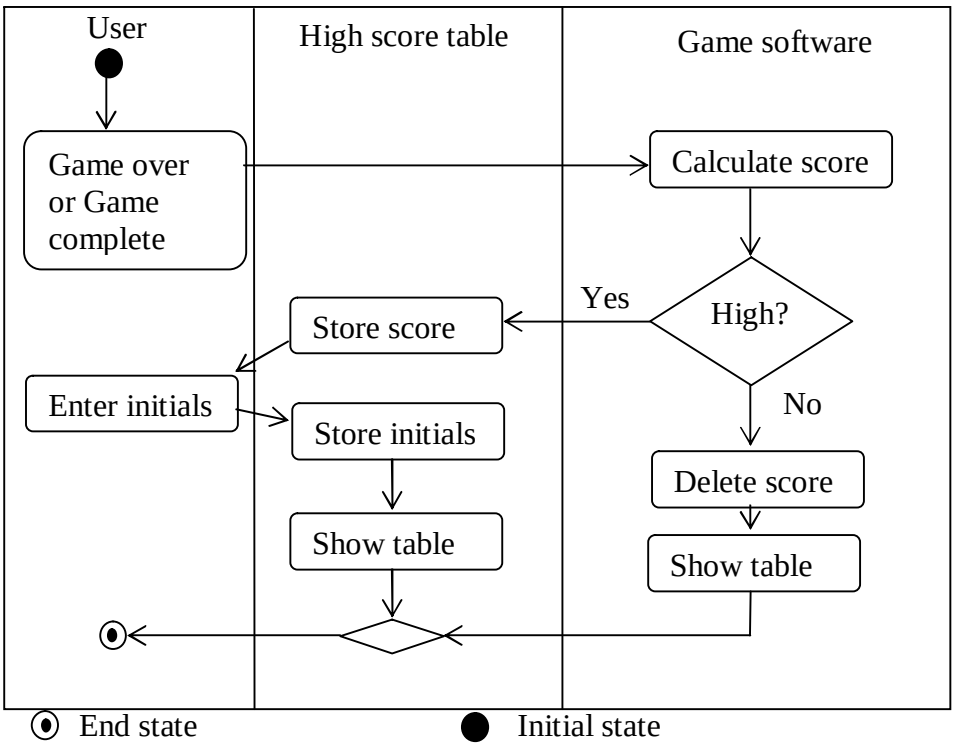
Button name	Action
View Map	Display the map of "Fondville" and highlight the area the playable character is in.

3. A procedure to automatically detect the resolution of the monitor displaying the game and resize the game window accordingly. This would be done between the activation of the game software and the main menu being displayed.

4. Giving the playable character a score that is 0 at the beginning of a new game. It would increase by collecting newspapers and completing tasks set by survivors. The score would be maintained when the game is saved and loaded. At the Game Over and Game Complete screens the score would be automatically compared to other scores and stored if high. If a score was stored the user is able to input their initials. There would also need to be an extra button on the main menu:

Button name	Action
View High Scores	Display the top 15 highest scores and the initials entered by the user who made them. Scores are ranked from 1-15 with 1 being the highest.

An activity diagram of this



5. The inclusion of the following additional 'Fear effects'.
   - All Rot in the current game view move towards the playable character independent of obstacles for a short time. For example, Rot move through walls while effect is activated.
   - Game window goes blurry but eventually becomes clear again.
   - All graphics become greyscale for a limited time before returning to normal.
6. An Options screen that can be accessed by clicking a button on the main menu. This would contain a slider to set the game difficulty and menus that allowed the game controls to be changed. If desirable function 3 is not met then there could also be a control to manually set the resolution of the game.
7. Allow survivors to be killed by Rot.

Part 4: How testing will be performed

The created game will be tested in 2 different ways.

4.1 Software testing

The game software will be tested to ensure it meets all the requirements listed in the "Project Specification and Plan" document. This will be done by testing components in the sections designed above. For example, once the stats have been implemented they will be tested before the implementation of the in-game items is started. Sections already tested will need to be

tested again when other sections that affect them are implemented. For example, it will be tested that all in-game items correctly affecting only the stats they are designed to. When the final version of the game is complete it will be tested for any errors that were not detected and fixed in earlier stages.

4.2 Game testing

As briefly mentioned in the "Project Specification and Plan" the game created will be given to game testers at various points during development. There will only be a small number of game testers. They will provide feedback on the development of the game and complete a questionnaire, created by the game designer to assess a range of people's reactions to the game. All versions of this questionnaire used in the project can be found on the project website [15]. The feedback and answers to the questions will be taken into consideration when continuing development of the game. If the results of game testing indicate any new features should be added to the game these will be considered desirable functionality and may or may not be incorporated in the final version of the game. Game testing will not affect the prototype of the game.

A summary of the testing performed and feedback from game testers will be produced.

Part 5: Risk Analysis

This section analyses the risks involved in the project and gives a brief description of an alternative game development environment which can be used to create the game if needed.

5.1 Risk analysis table

Risk	Risk Probability	Recovery Plan
Loss of time due to university coursework or other reasons.	High	Implement only core functionality. Delay or cancel releasing of prototype game software to testers.
Planned stages taking longer than estimated.	Medium	Implement only core functionality.
Unreliable code.	Low	Perform frequent testing and rewrite code if needed.
Game Maker development environment not capable of implementing all requirements of the game.	Low	Re-implement game using different development environment and re-draft documentation.

5.2 Alternative development environment: RPG Maker XP [14]

RPG Maker XP is another game development environment which, like Game Maker, only creates games to be run on the Windows operating system. Also like Game Maker it is event-driven, registering objects with events and performing some action when the event occurs. The main difference between this and Game Maker is that it only creates RPGs (Role-Playing Games). This is okay for the game I will create as it is similar to a short RPG. More information can be found at reference [14]. I now list the advantages and disadvantages of RPG Maker XP in relation to Game Maker.

Advantages	Disadvantages
• Many professional sprites, sounds and images provided with environment. • Isometric sprites and layering of objects gives created games a 3-dimensional look. • Pre-programmed way of entering and exiting buildings with separate interiors. • Support for encryption and decryption of game files to increase security. • Easy to implement conversations with computer controlled characters.	• Limited to 30-day trail version and then costs $60 for full version. Game Maker is free for non-commercial use. • No official tutorials on software use or game design. • Stats for character creation are overcomplicated for game I will create. For example, it will not make use of a characters strength or dexterity. • Unclear how to add new functionality for example, an inventory. • No built-in debug feature.

References

[1] Title: Book Excerpt - Better Game Characters By Design
Website URL: http://www.gamasutra.com/features/20060602/isbister_01.shtml

[2] Title: Better Game Characters By Design
Website URL: http://friendlymedia.sbrl.rpi.edu/bettergamecharacters/

[3] Title: Rules of Play - Game Design Fundamentals
Website URL: http://www.amazon.co.uk/Rules-Play-Game-Design-Fundamentals/dp/0262240459/sr=1-1/qid=1161366626/ref=sr_1_1/026-6976647-0885206?ie=UTF8&s=books

[4] Title: Drama princess - Tale of tales develops an autonomous character for real-time 3D - Programming Beliveable Characters For Computer Games
Website URL: http://www.tale-of-tales.com/DramaPrincess/wp/?p=17

[5] Title: Review of AI Game Programming Wisdom
Website URL: http://www.omlettesoft.com/document.php3?did=108

[6] Title: IGDA Forums
Website URL: http://www.igda.org/Forums/showthread.php?threadid=1534

[7] Title: GMAN-GAMES - Programming game AI by example
Website URL: http://greggman.com/edit/editheadlines/2005-02-12.htm

[8] Title: Paraglyph Press - Game Design Complete
Website URL: http://www.paraglyphpress.com/pr12062005.php

[9] Title: ibs.it Internet Bookshop - Game Design Complete
Website URL:
http://www.internetbookshop.it/ame/ser/serdsp.asp?shop=2293&e=1933097000

[10] Title: The Game Makers Apprentice - Game Development for Beginners
Authors: Jacob Habgood and Mark Overmars
Publisher: Apress
Publication Year: 2006

[11] Title: Game Studies - Where the action is
Website URL: http://www.gamestudies.org/0501/

[12] Title: Gender and Computer Games - Exploring Females' Dislikes
Website URL: http://jcmc.indiana.edu/vol11/issue4/hartmann.html

[13] Title: Game Research - Strategy
Website URL: http://game-research.com/?page_id=33

[14] Title: RPG Maker XP
Website URL: http://www.enterbrain.co.jp/tkool/RPG_XP/eng/index.html

[15] Title: Hunger - Project Website
Website URL: http://uk.geocities.com/tdc1@btinternet.com/Hunger/Hunger.html

Background Bibliography

1. Title: CS4099 (major) / CS4098 (minor) Software Project
Author: James McKinna, University of St. Andrews.
Publication Year: 2006

2. Title: Tutorial - What is a good game?
Author: Mark Overmars
Publisher: Mark Overmars
Publication Year: 2004.

3. Title: Software Engineering - Seventh Edition
   Author: Ian Summerville
   Publisher: PEARSON - Addison Wesley
   Publication Year: 2004
4. Title: Practical UML - A hands on introduction for developers
   Website URL: http://dn.codegear.com/article/31863