

Detailed Design

Contents

DESIGN	2
PART 1: THE FINAL VERSION OF THE GAME.....	2
1.1 <i>Overview</i>	2
1.2 <i>Gameplay design</i>	2
1.2.1 Game summary and main objective.....	2
1.2.2 Game Story	3
1.2.3 Map of Fondville.....	4
1.2.4 What survivors can say	4
1.2.5 Sleeping.....	4
1.2.6 Stats.....	5
1.2.7 Increasing Fear when playable character "sees" Rot.....	6
1.2.8 'Fear effects'	7
1.2.9 In-game Items	8
1.2.10 Using items and inventory.....	10
1.2.11 Being attacked.....	11
1.3 <i>Artificial Intelligence design</i>	11
1.3.1 AI of Rot.....	11
1.3.2 AI of survivors	12
1.4 <i>None gameplay screens and interface design</i>	13
1.4.1 Main Menu.....	13
1.4.2 In-game Menu.....	14
1.4.3 Game over screen	15
1.4.4 Game complete screen.....	16
1.4.5 Conversation screen.....	16
1.4.6 Credits screen.....	16
1.4.7 Saving and loading	17
1.5 <i>Level design</i>	17
1.6 <i>Media design</i>	18
1.6.1 List of pictures (sprites) needed.....	18
1.6.1 List of sound effects and music needed.....	19
PART 2: THE PROTOTYPE.....	20
PART 3: DESIRABLE FUNCTIONALITY	20
PART 4: HOW TESTING WILL BE PERFORMED	23
4.1 <i>Software testing</i>	23
4.2 <i>Game testing</i>	23
PART 5: RISK ANALYSIS.....	23
5.1 <i>Risk analysis table</i>	23
5.2 <i>Alternative development environment: RPG Maker XP [14]</i>	24
REFERENCES	25
BACKGROUND BIBLIOGRAPHY	26

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 1.2.6 on page 5. 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 1.2.10 on page 10. 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.

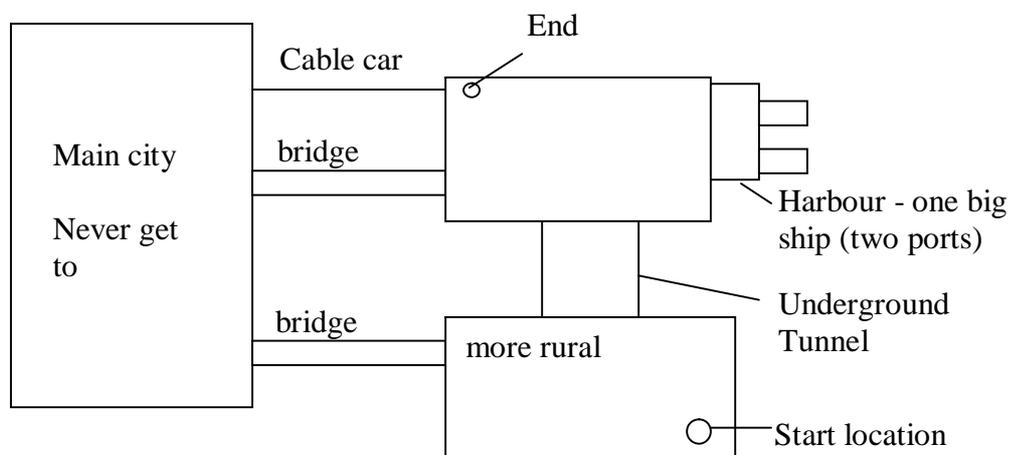
The playable character will be able to move around the game world by key presses on the keyboard. They will never be able to move out of viewable area of the game world or through solid objects.

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 Game Story

The 'Rot' where once the inhabitants of the town of "Fondville". These inhabitants have been infected by a plague that has turned them into mindless zombies. 7 years ago a volcano called "Mount Fog" erupted on the island near Fondville. It was a fairly small and "normal" eruption. However, unknown to the towns people this triggered the release of a gas. This gas was far enough away from the town not to be a problem but it did affect animals in the surrounding area. Recently, the people of Fondville learnt about a coal deposit near the volcano. Realising the opportunity, the people of Fondville quickly established a coal mine at the site. Coal was extracted and loaded onto a truck heading for the port in the North district of the island. The workers at the mine came into close contact with the gas having terrible consequences. Those workers became 'infected' by the plague and with realising brought it back to town. Since that time people have started changing into Rot. This is where the game starts.

1.2.3 Map of Fondville



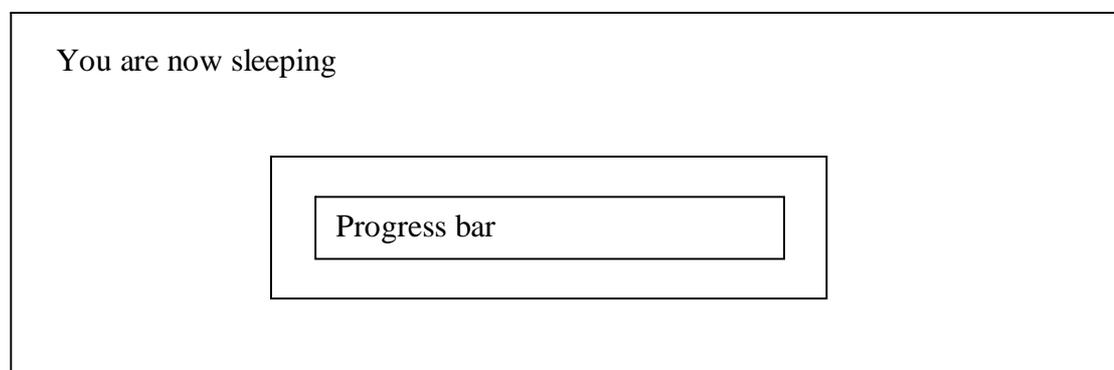
1.2.4 What survivors can say

Unless the survivor is given specific text to "say" first they choose a sentence at random from the following list.

If your hunger gets too high then you better find some food or your health will drop.
Don't forget to drink to lower your thirst.
Use wood to place a fire and get warmed up. Cold around here.
Remember to drink as often as you need or you'll end up dead!
Get some wood. Use it as a spear. Insert it into Rot. 'Nuff said.
If you start getting scared find a safe spot to sleep.
You're not superwoman you know.
Barricades can keep out the ulgies and make a building safe for sleeping.

1.2.5 Sleeping

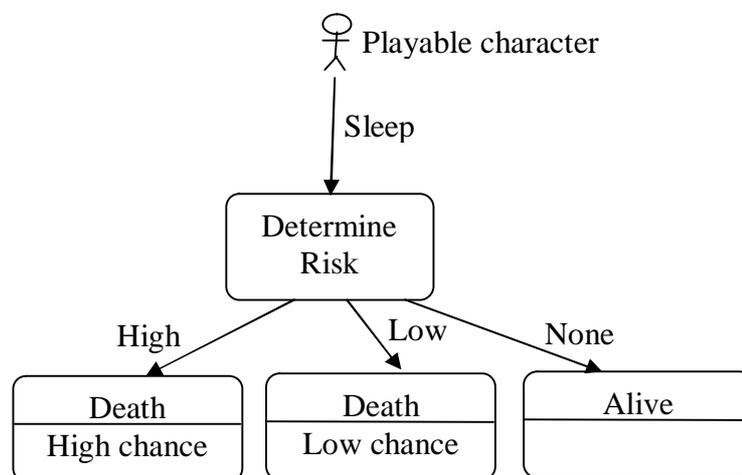
During gameplay the user can instruct the playable character to sleep. Sleeping is used to affect the "Fear" stat described in the next section. Sleeping will be indicated with a static screen (shown below) and will temporarily stop gameplay.



The progress bar indicates 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 1.2.8 on page 8.

<u>Where character sleeps</u>	<u>Risk</u>	<u>Description</u>
Outside	High	Random chance of death. Large chance.
Inside	Low	Random chance of death. Smaller chance.
Inside barricaded	None	No chance of death.



A summarized 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 life is lost. After a short time the character reappears at another position on the screen.

1.2.6 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.

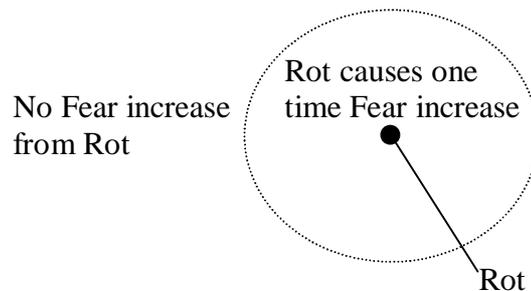
<u>Stat</u>	<u>Want to keep</u>	<u>Description</u>
Health	High	Decreases when playable character collides with Rot or if 0 of an indicated stat. Increases by using bandage or med kit item. When becomes 0 playable character dies. The playable character is removed from the game screen and 1 live is lost. 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 is reached value health slowly decreases.
Fear	Low	Increases slowly over time. Large increase if playable character "sees" Rot. 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.

1.2.7 Increasing Fear when playable character "sees" Rot

Effectively, each Rot has an invisible circle around them. When the playable character enters that circle their Fear stat is increased, provide the Rot in the centre can "see" them. 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 becomes darker to indicate they have caused the playable character fear and cannot do

so again while the playable character is in sight. If the playable character leaves the circle for a short time or hides behind an obstacle for example, a wall, the graphic representing the Rot becomes normal again. Using this method a Rot will only cause a Fear increase once when the playable character is nearby and not every time they see the Rot. Note that the normal increase of Fear over time does not stop outside or inside this circle.

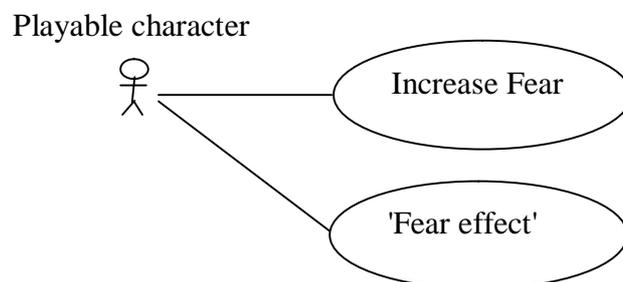


Model of Fear increase due to Rot

1.2.8 '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'

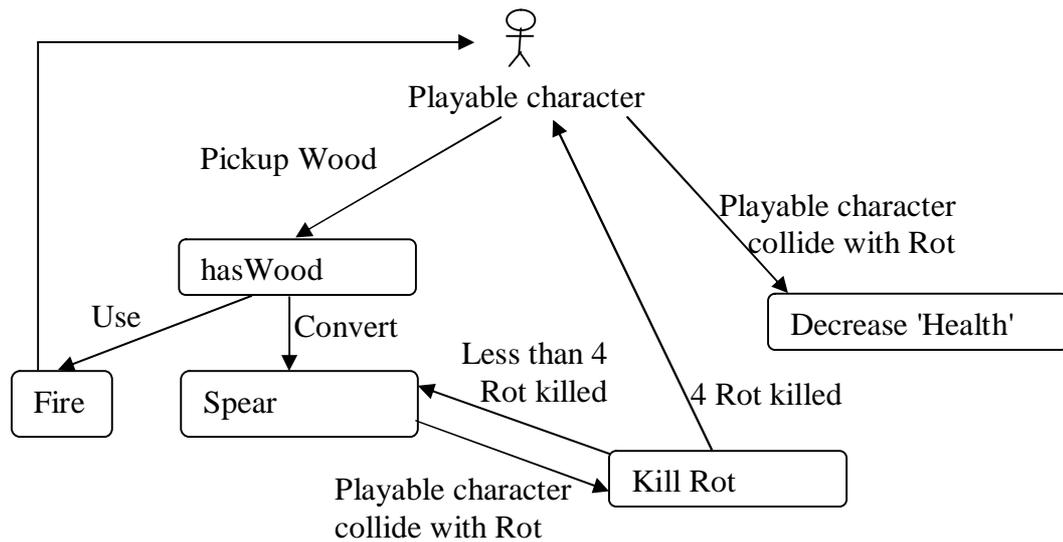
1.2.9 In-game Items

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 1.2.10 on page 10. When items are used none, one or more of the stats described in section 1.2.6 are altered.

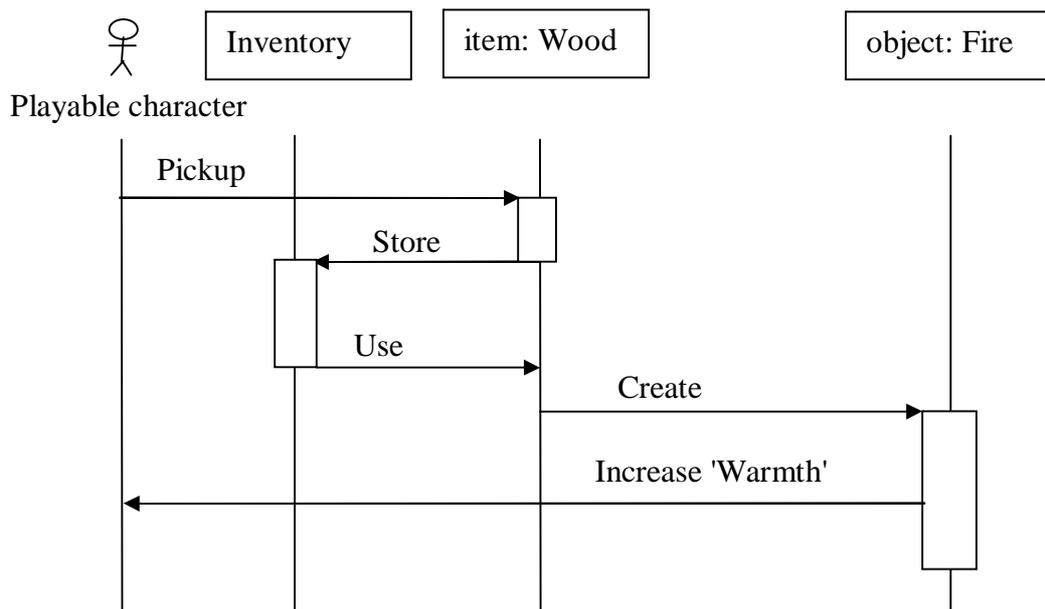
<u>Item</u>	<u>Effect when used</u>	<u>Description</u>
Food	Decrease hunger	-
Drink	Decrease thirst	-
Medkit	Increase health a large amount	-
Bandage	Increase health a small amount	-
Barricade	Stops Rot entering buildings	Use of this item will make the current building that the playable character is 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	<p>This is a dual purpose item with a different use depending on the purpose selected by the user.</p> <ul style="list-style-type: none"> • 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.

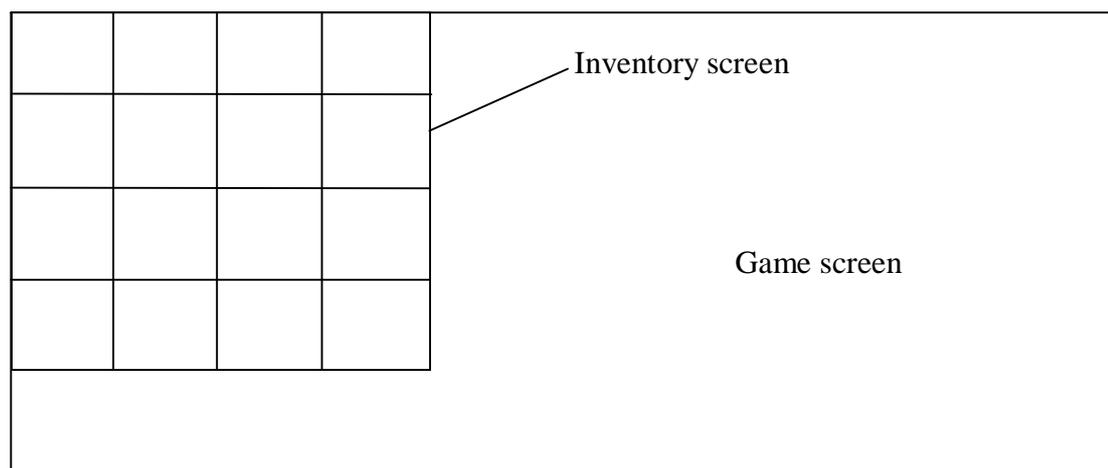


An expansion of the path leading to the Fire state as a sequence diagram:



1.2.10 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 given below.



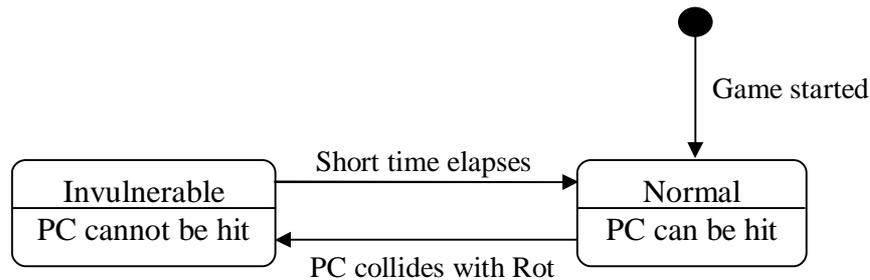
Items in the inventory can be used or dropped by selecting options from an action menu. An action menu opens when the user clicks on a store item. Action menus are different for each item and have 3 or 4 options listed in the below table.

<u>Item</u>	<u>Option 1</u>	<u>Option 2</u>	<u>Option 3</u>	<u>Option 4</u>
Food	Eat	Drop	Close	
Drink	Drink	Drop	Close	
Medkit	Use	Drop	Close	
Bandage	Use	Drop	Close	
Barricade	Place	Drop	Close	
Wood	Place Fire	Drop	Use as spear	Close

When items have been used (by selecting option 1) 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. The "Close" option simply closes the action menu for a specific item. Due to the duel purpose Wood item Wood can be used by either select option 1 or option 3. Only one action menu can be open at a time.

1.2.11 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 summarised in the below statechart diagram.



PC = Playable character

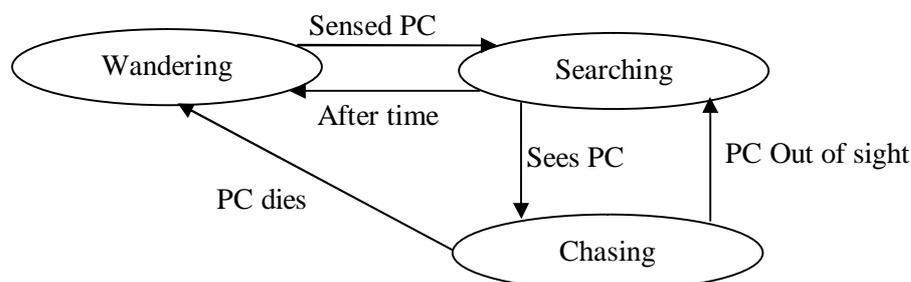
● = Initial state

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

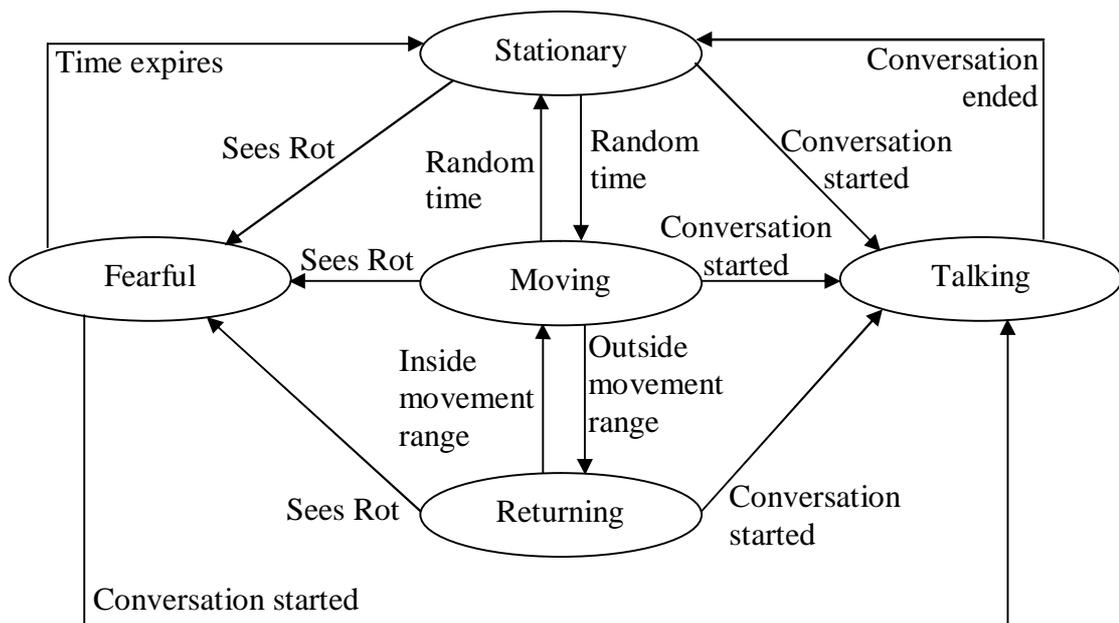
there movement speed. If a Rot in the Searching state then "sees" the playable character they will switch to the Chasing state and move towards the playable character. They will receive another slight increase to their movement speed. To "see" the playable character a line drawn from the centre of the Rot to the centre of the playable character must not be broken by solid objects, including other Rot.

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 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.

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 16. 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.

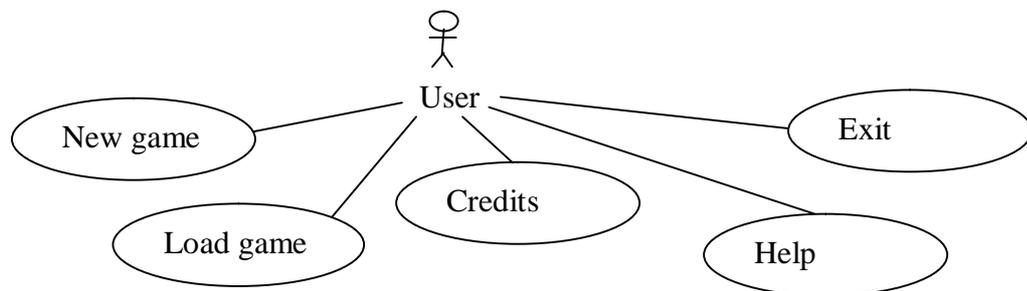
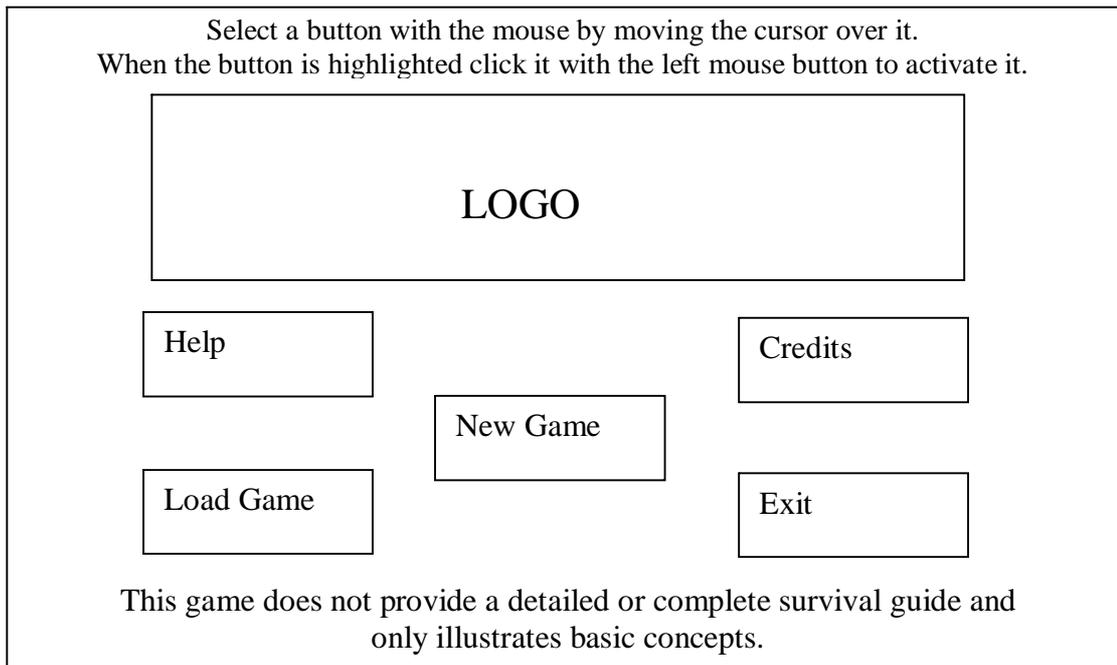
1.4.1 Main Menu

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:

<u>Button name</u>	<u>Action</u>
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.

The layout of the main menu:



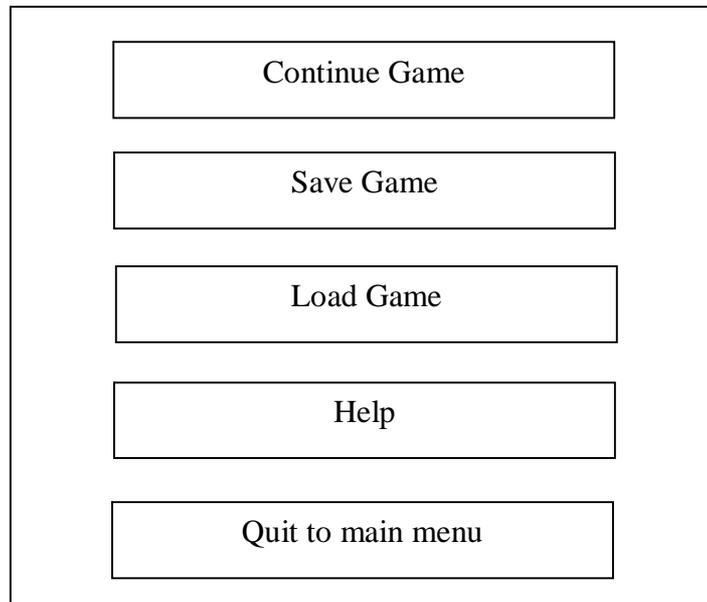
Use-case diagram of the main menu

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:

<u>Button name</u>	<u>Action</u>
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.

The layout of the In-game menu:



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:

<u>Button name</u>	<u>Action</u>
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.

The layout of the Game over screen:



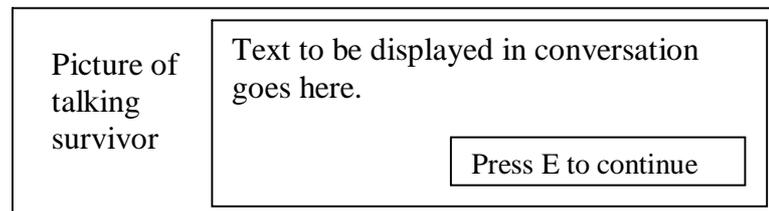
1.4.4 Game complete screen

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.

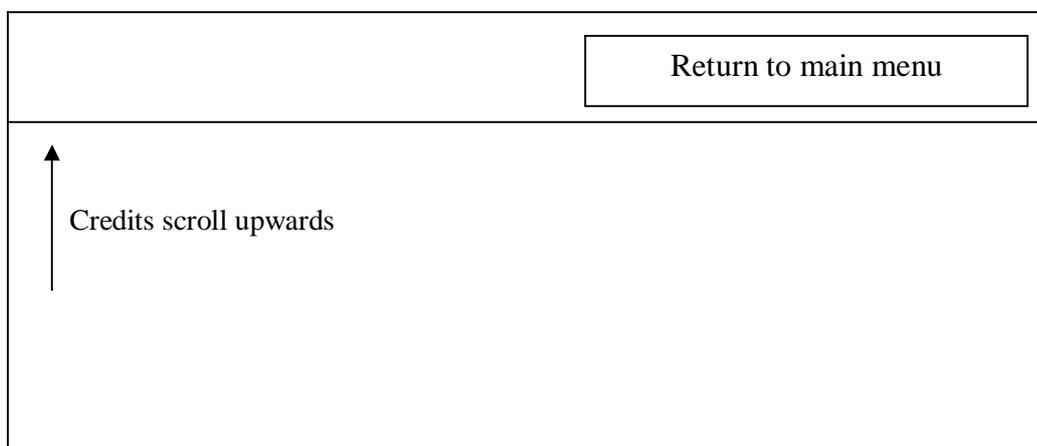
A diagram showing the layout of the conversation screen is shown below.



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. If time is short some of these may be omitted or simplified.

1.6.1 List of pictures (sprites) needed

The sprites needed for required functionality are listed and briefly described. A @ symbol indicates that the sprite has been drawn and implemented in the game.

- @ Character - a skateboarding paper girl. Carries bag over shoulder (you must have seen paper boys before). Should have helmet or something to clear so skating from top-down. No animation required. File type is gif.
- @ Character with wooden spear pointing forwards. No animation required. File type is gif.
- @ Tiles - grass, road and concrete. 32x32 pixels. File type is gif.
- @ Buttons - Non-highlighted and highlighted versions.
- @ Walls - no bigger than 32x32 pixels.
- @ Survivor - a civilian. Person in casual clothes. Only 1 type. Does not need much animation (only legs out frame and standing frame). File type is gif.
- @ Zombie - like civilian. These were miners before 'zombification' so maybe mining hat and broken pick axe sticking out of back. Does not need much animation (only legs out frame and standing frame). File type is gif.
- @ Items -small 32 x 32 pictures. File type: gif.
- @ Food - Apple
- @ Drink - Glass of water
- @ Wood - Plank
- @ Barricade - Red hexagon with no entry symbol on it
- @ Medkit - Box with smiley face on it. Top-down.
- @ Bandage - cloth thing. Can be side on view or frontal or top-down.
- @ Placed Barricade - 32x32 yellow and blacked striped square. Top-down.
- @ Placed fire - Wooden camp fire from top down.

- @ Image to show where barricade can be placed - sandbags
- @ Logo - The game title is Hunger. Not huge picture but not tiny. bmp probably best but can have jpeg, png, gif and most others.
- @ Game Icon - it should be no bigger than 32 x 32 and should be an icon file. It should be based on the logo but a lot smaller so probably just the hand and the H.
- @ Fire drum - 32 x 32 picture of drum/steel barrel on fire. No need animation. File type: gif.
- @ Big close up zombie face. Should not be animated. bmp probably best but can have jpeg, png, gif and most others. Scary! (Not too gruesome)
- @ Sleep image. Does not need animating. Maybe girl in bed with Z's floating around. bmp probably best but can have jpeg, png, gif and most others. Size: relatively big less than about 256 pixels in height.
- @ Cable car icon. 32x32 pixels wide gif file. Small side on picture of cable car but ignore small details like nails. Put the word EXIT on it in capitals.
- @ Game Over sign - Words "Game Over" in same style as Logo. bmp probably best but can have jpeg, png, gif and most others.
- @ Small camera - 32x32 pixel gif. Should not have detail as will only be displayed for a very short time. Will be used to indicate a screenshot has been taken.
- @ Crashed cars - 4 in different directions. Gif files. Not much detail and must be obvious that not usable.

1.6.1 List of sound effects and music needed

The sound effects, which must be .wav files, and music, which must be .midi files, needed are listed and briefly described. A @ symbol indicates that that sound or music has been implemented in the game.

- @ Main Menu music
- Playable character collides with Rot sound - struggle sound.
- @ Pick up item sound - Gulp sound.
- Use item sound
- @ Cannot drop item/ place fire sound
- Playable character dies sound
- @ Game Over sound
- Game Complete sound
- @ Random Rot sounds - Groan, "Hunger!"
- @ Rot enter chasing state sound
- Rot dies sound

- @ Button click sound
- @ Inventory full sound
- @ Spear break sound
- @ Survivor sees rot - low volume scream. Can happen often so must not be too loud.
- Fear effect over sound - woman's scream.
- @ Maximum fear reached sound - woman's scream.
- @ Static rot placed in 'Fear Effect' sound - thud
- @ Barricade collapsed sound
- @ Spear breaks sound - Wood snap

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:

<u>Item</u>	<u>Effect when used</u>	<u>Description</u>
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.

Text displayed when Newspapers are read in order in which they are found

1) The future is under our feet!

A coal mine has been established at the nearby Mount Fog volcano. This could... no, *will* bring great wealth to our humble town. No more will the town of Fondville be a mere dot on the map. Rejoice! Our salvation has come!

2) Mount Fog history:

This volcano last erupted 7 years ago. We asked several leading scientists if they thought the coal mining team, or for that matter, the town itself was in any danger of future eruptions. They all came to the conclusion that Mount Fog is dead and extinct. The last eruption was so small in size it was hardly felt by anyone. Rest easy in your beds citizens, no harm can possibly come from there.

3) Miners make discovery!

Yesterday the miners at Mount Fog discovered a gas venting from the volcano. Scientists on the scene say it was the result of the last eruption and is perfectly normal. Many question the lack of animals in the area but this can easily be explained by the animal's fear of the hot gas.

4) The future draws near.

The first cargo of coal has been loaded onto a truck and is heading to the North district. Our port has been expanded and already we have a buyer willing to pay for our treasured cargo. This is *our* cargo, our gift to the world. No one knows what the future will bring but I do know this: It is a great day for mankind!

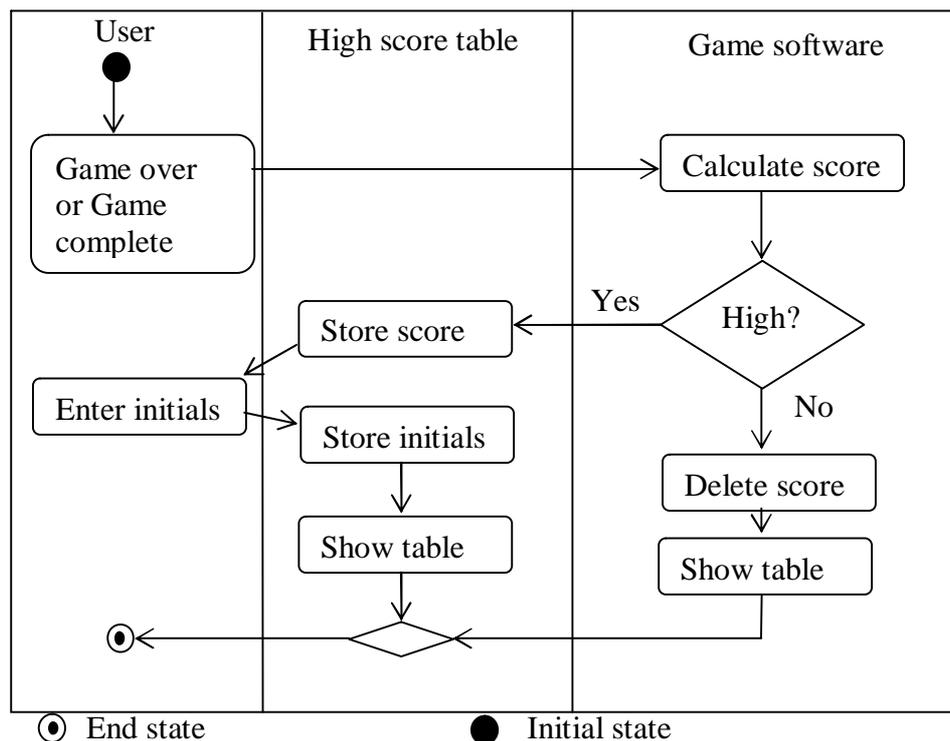
2. Via the in-game menu the user can access a map of the town to check their progress through the game. This will require an extra button:

<u>Button name</u>	<u>Action</u>
View Map	Display the map of "Fondville" and highlight the area the player 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

<u>Risk</u>	<u>Risk Probability</u>	<u>Recovery Plan</u>
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.

<u>Advantages</u>	<u>Disadvantages</u>
<ul style="list-style-type: none"> • 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. 	<ul style="list-style-type: none"> • Limited to 30-day trial 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 Believable 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 Sommerville
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>