Thomas James

Student ID: 10706142

Introduction

This document contains my project plan for COMP1004 and has a focus on time management and resource control.

Within this document the Development Cycle is outlined in detail. This is followed by the Games Design Document, which will in depth discuss the aim of the project, and what the product will look like.

Software Development Lifecycle

The Software Development Lifecycle is used by professionals around the world to create a standard process of developing software. By using a variety of models, time management can be greatly improved. There are 5 stages to the SDLC.

1. Requirements

During this stage, fundamental requirements are put forward that the program must contains. This is decided by shareholders, production managers and end users. It is usually decided by deciding on what the programs is designed to do, as the program must ultimately be able to do exactly what it was programmed for. Called the concept stage within the game development sector, this is where the basic principle of the game is decided: the game loop, theme and target audience.

1. Design

This stage is where the developers decide on the programming language of the game, the platform(s) the program will be released on, and the wider overview of the requirements of the design.

1. Implementation

At this stage initial programming starts. Programmers will use the list of requirements provided to produce a program that works on the provided platform.

1. Testing

While testing is done throughout the lifecycle of program development, at this stage testers will be used to try to find weaknesses within the product that would require fixing. It also ensures that the program successfully completes all required tasks that were decided upon within the Requirements portion of the SDLC.

1. Deployment

Once the product has been tested, and has met the requirements outlined within the SDLC, it can be deployed to end users. Within the game industry, this can sometimes be done as an open alpha or beta, to allow for extra time to work on bugfixes, or allow for stress testing for multiplayer online games.

This section will touch upon the Software Development Lifecycle of the project and will describe how this has been used within the development of the product.

During the planning phase, 2 days were used to decide on the genre of the game. Many ideas were considered, but after some research it was found that most, and nearly all, had already been created by another developer. The intention was to create something new and exciting, and to make something similar was not option.

During the research into similar programs, statistics like ratings and player counts were taken into consideration to deduce what genre of game would be most beneficial to the target audience. It was decided that a game with a darker theme, that was gritty with engaging content would be best, as the program is intended to be used by players who want to learn time and resource management, which lead to a game tailored towards adults. It was also noted as to sensible game lengths, as shorter products tended to be poorly reviewed (on average scoring lower than 2/5 stars) and longer games tended to have 40% less reviews overall, and most never finished the game.

Requirements

1. Backlog

| Concepts | Work |
| --- | --- |
| Add more random events  (low priority) (coding) | Character design  (high priority) (art) |
| Add more areas  (high priority) (coding/level design) | Better sprite designs  (low priority) (art) |
| Add more mini bosses  (medium priority) (coding) | HUD design  (high priority) (coding/art) |
| Extra tools  (low priority) (coding) | Add mini games  (high priority) (coding/level design) |
| Leader boards  (low priority) (coding) | Save system  (high priority) (coding) |
| Better transitions  (low priority) (coding) | Tutorial  (low priority) (coding) |

1. User Stories

This table shows key functionality that the game will require to be functional.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | User | Statement | Why | Priority |
| US1 | Player | I want to interact with objects | I can collect objects | High |
| US2 | Player | I want to move around the world | I can play the game | High |
| US3 | Player | I want to fight enemies | So the game can be lost | High |
| US4 | Player | I want to win or lose | So the game has an end point | High |
| US5 | Player | I want to be able to restart the game | The game can be replayed | Low |
| US6 | Player | I want different location options | So that I have options | High |
| US7 | Player | I want the game to run efficiently | So that the game can be run on any device | Medium |

1. Use case diagram

|  |
| --- |
| I want to interact with objects |
| I want to move around the world |
| I want to fight enemies |
| I want to be able to restart the game |
| I want different location options |
|  |

Choose item

Press button

Open inventory

1. MDA Framework

Game development revolves around 3 basic categories that must work well together to create an experience that users can enjoy. Therefore, the MDA framework proves very helpful during the game development cycle, and is comprised of Mechanics, Dynamics, and Aesthetics.

The Mechanics:

The mechanics are the core functionality of the program, and contain the code, algorithms, data storage and retrieval, and is the part of the program that the end user should not be able to see. The code should be written in a way that is predictable to those with access to the source code, but unpredictable to the end user (within a game development standpoint) to create variety and re-playability for the user. The core functionality and the explanation of how it works will be shown later in this document, within the Games Development Document.

The Dynamics:

This constitutes anything that the player can directly control, interact with, or have an influence over. Within this program the hope is that the dynamics change drastically for every play through, to keep the game engaging for those who have played the game before. For example, the game will be designed for the rooms to be dynamic, shifting from location to location to confuse the player if they do not adequately map the area that they have found themselves in. Dynamics are used to change the program into something less static and create multiple possibilities for the player.

The Aesthetics:

The difference between a fun game and a great game often come down to the way the game makes the player feel while playing. If the player feels challenged, the game becomes gripping and exciting, as the player becomes involved in the game as it makes them think about their actions. Games that are too easy can becomes monotonous and boring, and players can decide that it isn’t worth their time and choose not to play the game. Games that are too hard can put of the more casual player, and significantly reduces the overall player base size, and can reduce the chances of the game selling well. As with anything, a careful balance of Aesthetics must be found, and this can only be done using feedback from players.

Aesthetics contain the feeling a game conveys and can be announced through several different mediums. The difficulty, brightness, narrative, genre, and art style are major factors in the overall game aesthetic and can really change the target audience for a program.

For a player, the first thing that they will notice will be the general aesthetic of the game. A dark, dungeon crawler will appeal to a different target audience than a brightly lit, open world game, and the aesthetic must match the intended target audience for the game. This can be done through playtesting, as the Aesthetic of a game is often one of the last parts of the game development lifecycle, and can be easily changed by using agile development, as opposed to the waterfall method. By altering the Aesthetic of the ga me to better suit the target audience, a better product can be developed.

Architecture

Below will use HTML, CSS, and JavaScript for all its functionality, and no external libraries will be used. It will be rendered using the Canvas function, which will allow for dynamic changes that can easily be worked on.

Legend

Code that handles specific code functions

Components within the webpage

Link between functions

Name

Description

Component Diagram:

Application name

Single Web Page Application, written with HTML, using JavaScript Canvas Element

Main draw function

Draws to HTML canvas

Challenge randomiser

Function for choosing current room conditions

Challenge Components

Function for handling win/lose calculations

Movement Components

Functions for Moving through areas (external buttons)

HUD Components

Functions for drawing the on screen display

JavaScript Canvas

Creates Canvas and initially shows splash screen

Draw Function:

Characters

Tutorial

Lose

Win

Conditions

Choices

Pickups

Bosses

Movement choices

HUD

Interactions

Scenery

Visuals

Draw

Inputs

This shows the game logic used within the game space. The draw function will lock the frame rate to the refresh rate of the device used and is fed from user input.

Sprint Planning

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Below | December | January | February | March | April |
| HTML Page |  |  |  |  |  |
| Graphics |  |  |  |  |  |
| Movement |  |  |  |  |  |
| HUD/Gameplay |  |  |  |  |  |
| Tutorial |  |  |  |  |  |

Sprint 1 has been completed and an alpha webpage has been developed. 3 more sprints are required to put the game in a playable state, with the tutorial sprint being optional depending on the state of the workflow. Depending on the speed of development, the tutorial sprint may be repurposed to bugfixes and optimization.

Sprint 1: 21/12/21 – 31/12/21

This sprint was completed quickly and without difficulty. The initial webpage has been developed, and the layout has been set. While no functionality has been implemented yet, the foundations have been set for future implementations.

Testing:

The program is in its infancy in its current state, and so far, has required no reasonable level of testing.

During development the program will be under constant review, and before any sprints can be completed, bugs will be fixed.

Reflection

Below has been developed with a strong project plan that as of this report has been adhered to, and the future roadmap is looking good.

Early in development, there was no project plan in place and due to this development was initially very slow. Once a clear plan was put in place, development has sped up exponentially and allowed for easier development.

Over the next few months, the project will stick to the sprint plan, and will not stop development until the project is ready for release, and meets the initial requirements indicated within this document.

Games Design Document

Product Vision

In ‘Below’, the aim of the game is time and resource management.

Working on a deserted world, the player will overcome various hazards and make amazing discoveries within the tunnels of a disused mineshaft once used by an alien race. The player will be expected to collect items that can be placed in their inventory, that can be used to traverse into new areas of the map. Items will be craft-able, like swords and armour, that will increase their chances of survival. However, resource management will be key here, as some resources can be used for multiple things, and the player can lose the game if they waste their resources on unnecessary things. The player will have limited supplies of batteries, and will have to manage their time wisely, or risk being stranded in the deep dark caves. This will force the player to make decisions on their feet, which increases the difficulty while making the player think.

Below in detail

Genre- Hypertext Crawler

Target audience- Adults

Player Experience and visuals-

The game will be designed to be dark and claustrophobic, to put the player in the shoes of their character. This will help the game make an impact on the player and should increase the engagement level. The player will never know the name of their character and will never hear spoken words or readable text in the world. This is to compound the fact that the setting is on a barren world with no humans on the planet. They are alone, deep below.

Game loop

|  |  |  |
| --- | --- | --- |
| Enter new room | Survey room | Fight enemies |
| Use map | Game loop | Collect loot |
| Craft items | Check inventory | Repair/clear mineshaft to allow entry to new areas |

Main items

Map:

Will allow the player to track what rooms they have accessed in the past, and what rooms are adjacent to them. Maps will have to be updated over time, as the player will get lost and doors will start to lead to places that they didn’t initially lead to as the game advances.

Tools:

Sword:

Will increase the damage dealt to enemies, allowing for further advancement in the game, as some enemies will be too strong without upgrading the damage output of the player.

Pickaxe:

Will allow for new passageways to be opened by the player but will consume durability. This allows the player to skip certain bosses or enemies but will be required for some parts of the storyline. Some hidden chests can be found with this tool.

Health:

Health will be very limited in Below and will be treated like a resource. Some enemies will require the player to use their hearts to advance the game. Run out of lives, and the game is over.

Torch:

Torches allow the player to see within the tunnels and will have a battery level. If the torch runs out the player will need to replace the batteries (consumable item). If they run out of batteries, will become lost in the mineshaft, and they will lose all their hearts.

Legal Issues

There are many laws in the UK that I will ensure that the Below strictly adheres to.

The Data Protection Act 2018 is the UK’s law pertaining the General Data Protection Regulation (GDPR) and is a law that must be upheld by any developer worldwide that collects data on any EU citizen. While the UK is no longer part of the EU, this act is still used within the UK, and must be followed. This act protects user from their data being processed unnecessarily and must allow for a user to opt-out of their data being collected at all. This applies to any data that could be used to identify a person, whether directly or indirectly. This can be names, dates of birth, locational data, IP addresses, or online identifiers.

“The Data Protection Act 2018 controls how your personal information is used by organisations, businesses or the government.

Everyone responsible for using personal data has to follow strict rules called ‘data protection principles’. They must make sure the information is:

used fairly, lawfully and transparently

used for specified, explicit purposes

used in a way that is adequate, relevant and limited to only what is necessary

accurate and, where necessary, kept up to date

kept for no longer than is necessary

handled in a way that ensures appropriate security, including protection against unlawful or unauthorised processing, access, loss, destruction or damage

There is stronger legal protection for more sensitive information, such as:

race

ethnic background

political opinions

religious beliefs

trade union membership

genetics

biometrics (where used for identification)

health

sex life or orientation

There are separate safeguards for personal data relating to criminal convictions and offences.”

(UK Gov, 2018)

The Copyright, Designs and Patents Act 1988 is the UK copyright law that protects content creators from unlawful usage of their materials. It allows the creator to choose how their work is used, or distributed, and means their work cannot be copied or used without explicit consent.

Below will not collect any user data throughout the game to ensure that The Data Protection Act is adhered to and will ensure that no additional features added in the future collects additional data. If the game requires an age rating, then the age of users will need to be verified but will not be collected. Only copyright free or original designs will be used within the game to ensure that no Copyright infringements are met.

Ethical Issues

This game will touch upon issues such as injury and death and may be found inappropriate to some age ranges. To combat this, I intend to implement an age verification, and if a user is below a certain age, certain features will be disabled (such as blood, death, and demons). The game will aim to be politically correct, having no ties to the real world to reduce the risk of offense to users.

References:

UK Government, 2018, 10/12/2021, <https://www.gov.uk/data-protection>