***Maze Game Analysis***

*Project proposal - pg1*

*Project plan outline - pg2*

*Project plan (****Gantt Chart****) - pg3*

*Feasibility Study - pg4*

*End user survey design - pg7*

*End user survey analysis - pg8*

*Requirements specifications - pg13*

***Maze Game***

*Proposal*

Project Criteria

* Project must include an interface appropriate for users
* Project must include input validation
* Project must include 2D arrays
* Project must include recursion
* Project must include interfacing with files on the computer

Objective

The objective of this project is to create a game in which a small rectangular sprite can move through a maze by the player’s command. There should be at least one maze (although if there is extra time, more mazes may be added) and the ability to save and load the game. The maze will most likely be a 10x10 block maze, but may also be larger. It must be able to run on both a windows 7 and 10 OS. Due to the short amount of time available, the end product of this project will most likely only be a ‘taster’ of a larger game. I.e this game should only be comparable to the games that are currently in alpha development.

End user Group

The end user group is fairly vague. The people who will use this program include people who are interested in puzzles and/or video games. The common age/computer speed is currently unknown. More information on the end user group will be found after a survey is conducted on people who are interested in the game.

Deliverables

The game’s deliverables are as following:

* A main menu/in game menu for user interface, which can start the game, quit, load (main menu) and save (in game menu). (Input validation is achieved here, as when anything other than buttons are clicked nothing happens. This will also create an effective user interface.)
* At least one level with a rendered maze (This will be achieved through the use of 2D arrays)
* A sprite which can move left, up, right and down, as well as detect the walls of the maze and not collide with them and win when the sprite exits the maze. (The solution for moving the sprite will include recursion)
* The ability to both save and load player/maze data (This will interface with user data)
* The ability to run on a Windows 7 computer
* The ability to run on a Windows 10 computer

*Project Schedule Plan Outline*

Week 1:

* Finish analysis
* Begin Requirements specifications

Week 2:

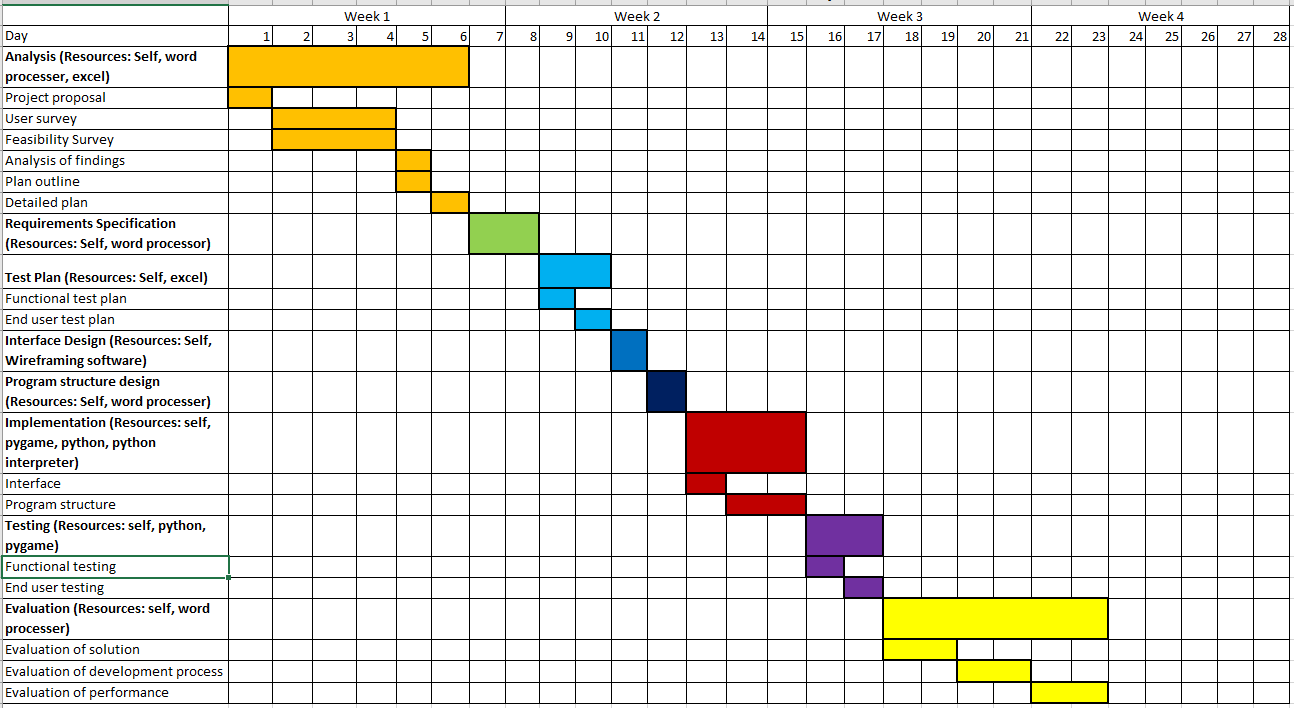
* Finish Requirements specifications, Test plan, Interface Design, Program Structure design
* Begin implementation

Week 3:

* Finish implementation and testing
* Begin evaluation

Week 4

* Finish Evaluation
* Continuously iterate review of full project



*Feasibility Study*

Technical feasibility

The needed hardware to complete this project include:

* A windows 7 computer
* A windows 10 computer

The needed software to complete this project include:

* Python
* Pygame
* A python interpreter

Both a windows 10 computer and windows 7 computer is available to me, as is python, pygame and a python interpreter. The python interpreter used will be pycharm community edition. Therefore this project is extremely viable technically as all components needed to design, implement and test the program is available readily.

Economic feasibility

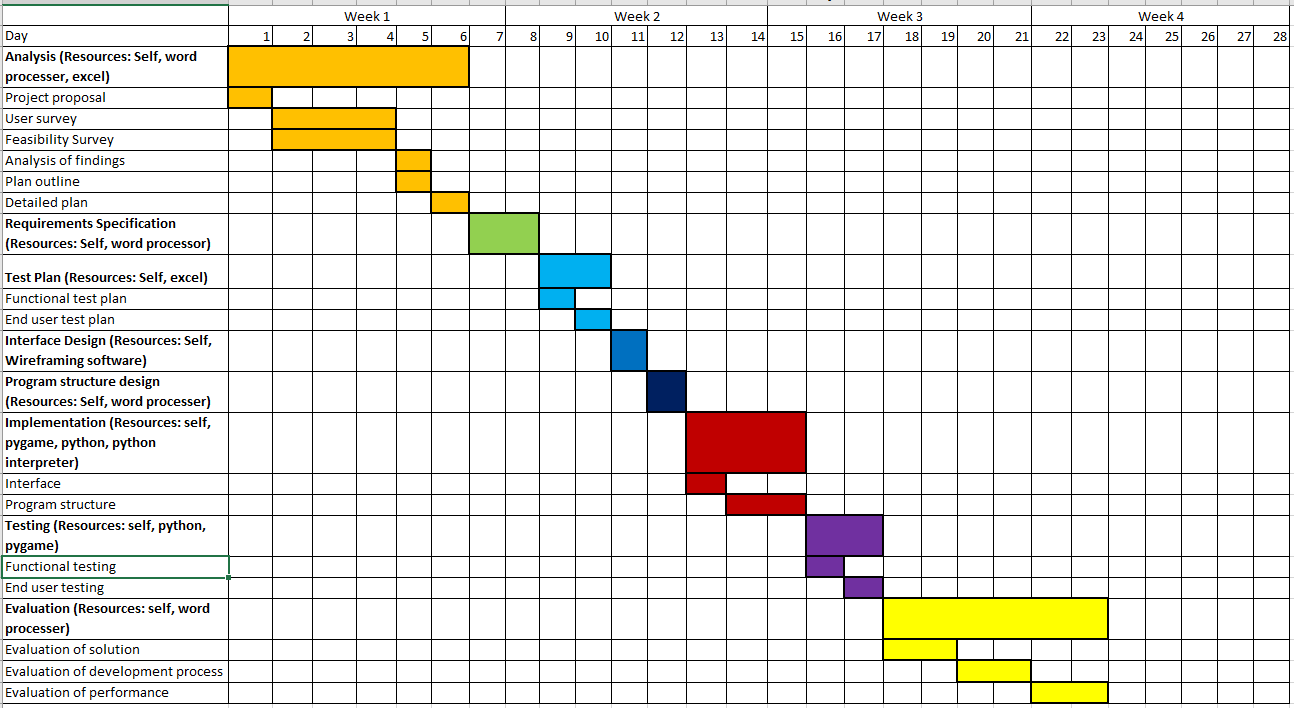
Due to all software and hardware being readily available and no part of the project being sold, there is no economic implications for this project.

Legal feasibility

There are no health and safety implications, but there are a few licensing implications as software from another company is being used. Python’s terms and conditions state python to be explicitly open source and free to use/distribute/change.(1) As long as the copyrights for python are expressed in the shell (which is built into the initialisation of the program), you may use python modules as you wish, including changing files due to it being open source. Pygame, just like python is open source and useable in any situation.(2) The edition of Pycharm used is also open source, resulting in no legal implications for changing and distributing it.(3) It also means there are no implications for any programs made using it.

Schedule feasibility

The game will take approximately 3-4 weeks to develop and will most likely be ready in that time. Even if it is not, due to the need for a project to hand into the SQA at submission date, and lack of economic/legal risk, the project must be at least partially completed by this time. Hopefully, the project will be finished by a week before submission date, but later is acceptable.



1. <https://docs.python.org/3/license.html> - python terms of use

“Subject to the terms and conditions of this License Agreement, PSF hereby

grants Licensee a nonexclusive, royalty-free, world-wide license to reproduce,

analyze, test, perform and/or display publicly, prepare derivative works,

distribute, and otherwise use Python 3.7.3rc1 alone or in any derivative

version, provided, however, that PSF's License Agreement and PSF's notice of

copyright, i.e., "Copyright © 2001-2019 Python Software Foundation; All Rights

Reserved" are retained in Python 3.7.3rc1 alone or in any derivative version

prepared by Licensee.“

1. <https://www.pygame.org/docs/> - Pygame Documents

“LGPL License

This is the license Pygame is distributed under. It provides for Pygame to be distributed with open source and commercial software. Generally, if Pygame is not changed, it can be used with any type of program.”

1. <https://blog.jetbrains.com/pycharm/2017/09/pycharm-community-edition-and-professional-edition-explained-licenses-and-more/> - Pycharm Community Edition Licensing Explained

“Let’s go to the [LICENSE.txt in the root of the GitHub repo](https://github.com/JetBrains/intellij-community/blob/master/LICENSE.txt). JetBrains’ open source projects are generally licensed under the Apache 2.0 License. This means that you can use it anywhere you’d like to, and modify it freely. There are some restrictions, which we’ll look into below.”

*End User Survey Design Questions*

***What do you want to find out?***

* What is our main target age group?
* What performance level computers are we designing for?
* What are users past experience with maze games?
* What would users like to see from a future maze game?

***Group of people to target?***

* Any age
* Low-High performance computer owners
* Those who have never played/want to try a maze game.

***Number of people to target?***

50+ people

***How can a survey help me collect information for this problem?***

This will help collect information on end user expectations and what features end-users may want in this program.

***Questions to ask?***

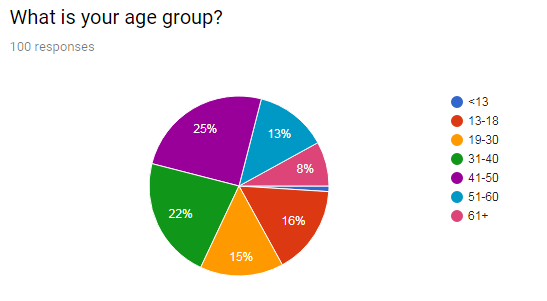
* Age group?
* Computer performance/speed?
* What maze games have you played before?
* Summary of previous maze game experience?
* Desired features in a maze game?

***Survey form***

The survey will be in digital form, as this makes it easier to hand out in the time frame that I have. The survey will be published online to Facebook and Snapchat in order to ensure both the largest and most diverse sample. It will also have a consent form, containing how the data will be used.

*Total Responses: 100*

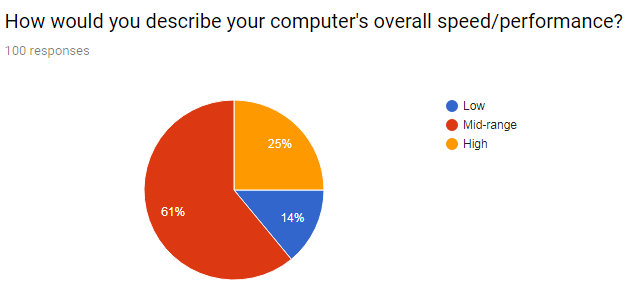
*Question 1:*

**

This shows that there is a fairly equal balance between age groups that are to be using this game (excluding people <13). Therefore, the end user group seems to be more generalized.

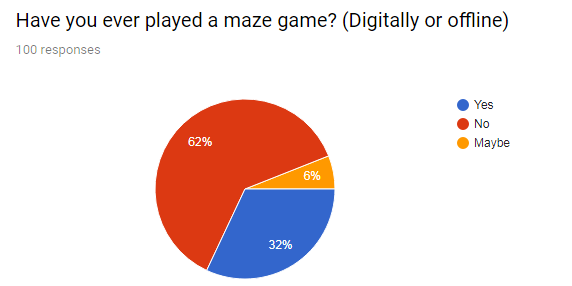
As a result, useability testing must take into account both the youngest and oldest of users. Game installation and use must be simple enough for those who are not experienced with technology and also be enjoyable for all age groups.

*Question 2:*

**

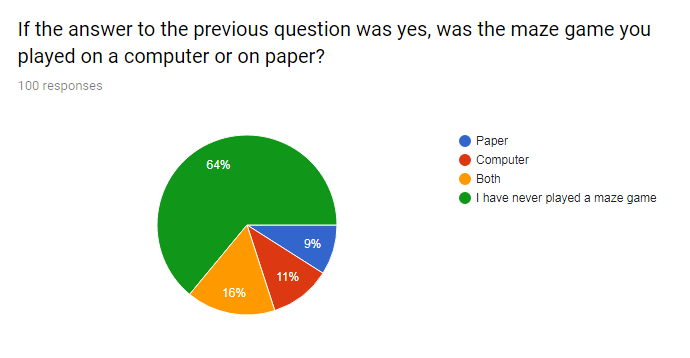
This shows that the majority of end users use mid-range computers. As a result, the end result of this project should be able to run on a mid-range computer.

*Question 3:*

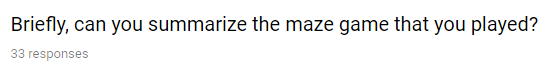
**

The majority of people have never played a maze game. This means that expectations are low and therefore most approaches to this project will be fresh and meet end user requirements.

*Question 4:*

**

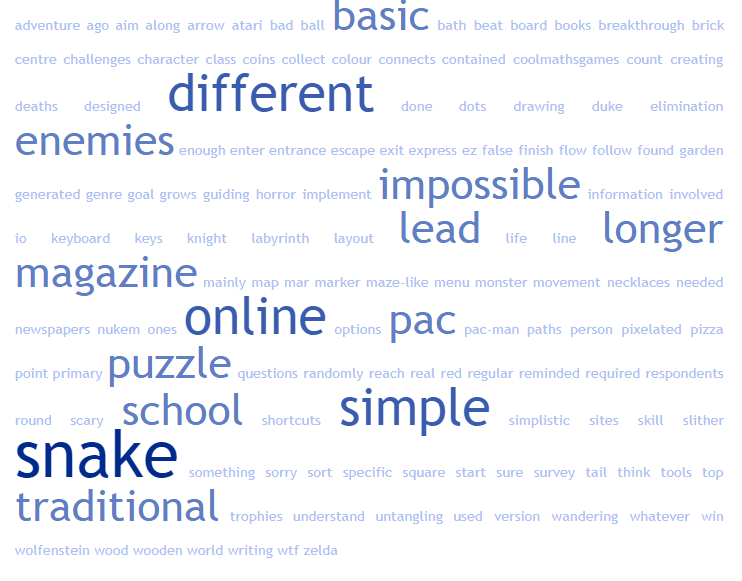
*Question 5:*

**

*Responses:*

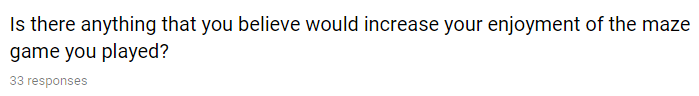
* *Moving through a maze with elemental enemies, needed specific moves to beat the different elements*
* *Ez*
* *There was a maze*
* *Scary maze game, (does the first Zelda game count?), Bad maze games I made, other basic maze games from coolmathsgames and other online gaming sites*
* *It was a horror game where a monster would follow you around and you have to reach to the centre to win*
* *The impossible maze, it was impossible*
* *It was a really simplistic game that came on my primary schools old computer, it was a simple red brick layout in which a maze would be randomly generated and you had to escape while avoiding simple enemies.*
* *I haven’t played one, sorry 😐*
* *It had different paths some lead to deaths some lead to shortcuts but there was one end goal*
* *I don't know what you mean by 'maze game'. When you do a survey, make sure your respondents have enough information to understand the questions.*
* *Traditional maze finding bath from start to finish*
* *A game where your aim is to find your way through a 2 or 3 d map like puzzle*
* *It was a very long time ago - but involved using the arrow keys on my keyboard to move a character through a maze.*
* *Well, I can think of two maze-like games. One genre is found in magazines or newspapers, where you have to enter the maze at one end and find the exit (or the middle or whatever). Tools required: a pen or other writing implement. But this has also reminded me of the computer game Snake where you are a snake in a pixelated world, wandering around trying to collect coins (or something) and you have to avoid your own tail. The longer you play, the longer the snake grows. So it's a sort of a maze like game but you are creating your own maze as you play...*
* *Played paper and wooden ones - made by myself in school wood work class :)*
* *Both really simple games designed for children.*
* *So many, but Labyrinth as a board game and Adventure on the Atari 2600 and Duke Nukem/Wolfenstein 3d were the breakthrough games*
* *Wtf is a maze game?!*
* *Pac-Man - is that a maze game?*
* *Drawing a line to find the way from the entrance to the middle with lots of false options.*
* *I used to have maze puzzle books as a child*
* *It was a pen and paper maze.*
* *Finding your way through to the end, with challenges along the way.*
* *I've mainly played maze games with my children, on paper and also in garden mazes*
* *Guiding a ball round a contained maze from one point to another.*
* *Both top\_down marker movement and also a first person 3d maze run*
* *Snake*
* *It was on a child’s menu at Pizza Express!*
* *Pac Man*
* *Child's maze game in magazine*
* *Basic Square Maze, all about elimination and time.*
* *Played traditional games on paper, pac man online, and a game where I was a knight working through a mar to find trophies*
* *slither.io online. I've done lots of mazes on paper, especially as a child. Untangling necklaces is a way I use this skill in real life.*
* *Flow. Connects different colour dots in a maze. Also regular paper version.*

*Word cloud generated with responses:*

**

Of the people that responded to question 5 (optional question), most agreed that the game that they had played was simple, online and traditional. Games such as PacMan and Snake were mentioned, as was the impossible maze. Many people who had only played paper maze games talked about magazines and similar. This could be interpreted as the most enjoyable games being the simple/traditional ones. Although, opinions of the maze games were fairly neutral.

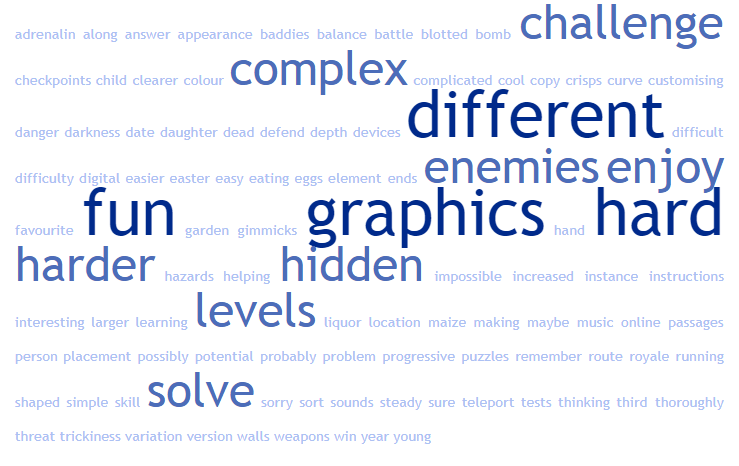
*Question 6:*

**

*Responses:*

* *Better graphics, clearer instructions*
* *Battle royale*
* *More maize*
* *More variation in graphics, different weapons, different types of enemies*
* *Adrenalin*
* *Make it even more impossible*
* *Hard liquor.*
* *And maybe some way of customising the levels, for instance changing the appearance of enemies that sort of thing*
* *Really sorry, haven’t played one!*
* *More depth, like a way to defend myself or checkpoints*
* *See answer above.*
* *Good balance of trickiness - not too easy, not too hard*
* *Not sure*
* *No - I do remember it was my favourite game*
* *Well now I'm thinking it would be cool to have a maze that changed as you played it.*
* *More challenge*
* *A steady learning curve and hidden Easter eggs*
* *Make it easier for me to win!*
* *If it was more complex*
* *If online, progressive levels*
* *Better graphics.*
* *Music, and gimmicks on the maze*
* *I probably prefer running around garden mazes, some challenges/puzzles to solve along the route makes it more fun*
* *More hazards such as bomb placement or teleport location devices*
* *Thoroughly enjoyed both types I played. First person was a lot harder but really fun*
* *An up to date version.*
* *It was for a young child so I was helping my 2 year old daughter - it was very simple! Other digital maze games I enjoy have an element of danger and darkness - always good to have potential threat and baddies to get in the way. And problem solving. Possibly hidden passages.*
* *A third hand (for eating crisps)!*
* *I would like to play a maze game. It sounds fun.*
* *More difficulty*
* *A larger Maze and with different shaped walls blotted around making the turns more interesting.*
* *I prefer hard copy games in colour which are complex- with lots of tests turns and dead ends*
* *As my skill has increased, I like harder more complicated mazes.*
* *More difficult games*

*Word cloud generated from responses:*

**

As this is another optional question, only 33 people responded. Most expressed their need for better graphics, hidden secrets and more difficulty. Although this is what users expressed as their main needs, many of these needs cannot be met. This is due to a lack of time to complete this project and the main end user group is that of the SQA. They have extremely specific needs which must be prioritized before all others. However, the public end user group may be taken into consideration soon after reaching SQA requirements.

***Requirements specifications***

Purpose

The purpose of this solution is to create a functional maze game which will demonstrate advanced higher techniques. Hopefully, more than one level will be able to be created but no guarantees can be made as to that. The end product of this project should be considered in the ‘alpha’ testing stage of development.

Scope and boundaries

Scope:

The game must have:

* Have a main menu, which allows users to Start, load and quit
* Have an in-game menu, which allows users to quit to menu and save. It must also open and close when the user presses the ‘esc’ button
* At least one level with a rendered maze
* A sprite for the player which can move left, up, right and down, have collision detection and win when it exits the current rendered maze
* The ability to save current player/maze data onto a file
* The ability to load data from a file and generate a player/maze from it
* The ability to run on a windows 7 computer
* The ability to run on a windows 10 computer

Boundaries:

* Game may not work on Windows 8
* Game may not work on any OS before windows 7
* Game may not work on computers which do not have a windows operating system
* Game will not have more than one save file (Must have one user/computer)
* Game will not have an in-game tutorial and will have the controls in documentation

From the above scope and boundaries, we can deem the end product to be a maze game with a main menu, in-game menu, a sprite with movement options, collision detection +win function, the ability to save and load data as well as running a windows 7/10 OS.

End users

The end user group is fairly vague in this case, as the end user survey revealed that most age groups have the ability to play this game. The only end user fact that was gleaned from the end user survey was the fact that they:

1. On majority, have midrange computers
2. On majority, have never played maze games
3. Would like higher difficulty/more features in the maze game delivered

There is also an alternative end user group which is that of the SQA/marker. This is the most important end user group as it determines the value of the project at the end of it.

Some features that have been asked for by this user group include:

* A User Interface
* 2D arrays
* Recursion
* A Save/ Load feature

Features asked for by the public end user group include:

* Good Graphics
* High Difficulty
* Hidden features

Unfortunately, these features are unable to be implemented initially due to the time constraints currently given. Implementation must take as little time as possible so only the very basics of the program will be implemented. If there is extra time to complete implementation, there may be extra features implemented

User Requirements

From the above, it can be seen that the user requirements for this project to be successful include:

* A user interface, to be carried out using the in-game menu and main game
* 2D arrays, to be carried out through the drawing of the maze
* Recursion, to be carried out through the movement of the player
* A Save/Load feature

Some optional user requirements include:

* Higher difficulty
* Better Graphics

Due to the time frame, only one end user group can be catered for (that being the SQA), but the end user requirements may be implemented if more time is given or the project is completed early.

Functional Requirements

Detailed functional requirements, expanded upon from the scope include:

* A main menu created with Rect() objects, which can:

1. Start the game from the first level (i.e the first Rendered maze)
2. Load saved player and maze data if there is any and use it to render a maze and player. Otherwise, display an error message
3. Quit the game by uninitializing pygame and python

* The rendered maze loaded by user pressing start on the main menu must be rendered using 2D arrays and the Rect() object and must have walls on all sides of the screen.
* Every maze defined will have an ID dependent on the order they appear to the user (e.g. first maze that appears will be ‘1’)
* A sprite for the player which can:

1. Move left, right, up and down using recursion
2. Be defined as a Rect() object
3. Detect any collisions with other Rect() objects, using Rect attributes
4. Use collision detection to detect when it has won the maze (i.e when rect collides with edge of screen, it wins)

* Open an in-game menu when pressing ‘Esc’ and closes when pressing ‘Esc’ and can:

1. Quit the game, getting rid of all player and maze objects and returning to the menu
2. Save the game, storing the ID of the maze and the player position in an external txt file.

* The ability to run on a windows 7 computer
* The ability to run on a windows 10 computer

Inputs and Outputs

*Main Menu I/O:*

* Upon clicking within the Rect() defining the Start button’s borders (input), a maze will be rendered and the sprite for the player will appear at the first available ‘empty’ space. (i.e the first place where a wall rect is not placed.) (Output)
* Upon clicking within the Rect() defining the Continue/Load button’s borders (Input), the program will check if a file called ‘sav.txt’ exists within the same folder as itself. If not, an error message will display, saying ‘No Save File Found’. And if it does exist, the program will use the data to load a maze and player consistent with the last time the user saved. (Output)
* Upon clicking within the Rect() defining the Quit button’s borders (input), pygame and python will uninitialize and close. (Output)

*Maze I/O (Once the Start button has been pressed or a maze/player has been loaded):*

* Upon pressing the Right Arrow key on the keyboard (input), if there is not a ‘wall’ Rect in the next ‘block’ to the right, the sprite the player controls will move one ‘block’ to the right. (Output)
* Upon pressing the Left Arrow key on the keyboard (input), if there is not a ‘wall’ Rect in the next ‘block’ to the left, the sprite the player controls will move one ‘block’ to the left. (Output)
* Upon pressing the Down Arrow key on the keyboard (input), if there is not a ‘wall’ Rect in the next ‘block’ downwards, the sprite the player controls will move one ‘block’ downwards. (Output)
* Upon pressing the Up Arrow key on the keyboard (input), if there is not a ‘wall’ Rect in the next ‘block’ upwards, the sprite the player controls will move one ‘block’ upwards.
* Upon pressing the ‘Esc’ key on the keyboard whilst the in-game menu is not open (input), the in-game menu will open in the middle of the screen, covering the current maze and player (if the player is in the middle of the screen). (Output)
* Once the player’s sprite touches the edge of the screen window (input), the ‘win’ function will be called, which clears the screen and creates text in the middle of the page saying ‘You win!’. If there is another level, it will move onto the next level. (Output)

*In-game I/O* *(Once esc key has been pressed whilst in the maze):*

* Upon clicking within the Rect() defining the Save button, the ID of the maze, as well as the x and y positions of the player will be stored on an external ‘sav.txt’ file in the same folder as the program.
* Upon clicking within the Rect() defining the Quit button, the screen will be cleared and the main menu will be drawn onto the screen.
* Upon pressing the ‘Esc’ key on the keyboard whilst the in-game menu is open (input), the in-game menu will close, covering over where the menu was with black, then drawing the maze and player over it again. (Output)