Evaluation

*Requirements Specifications Evaluation*

The requirements specifications were almost fully matched. This excludes the optional requirements specifications based on the end user needs. These specifications weren’t met as they were unable to be implemented with the given time frame and will most likely be implemented upon further development of the program. As said earlier in project documentation, the product delivered to the SQA (And the product that is currently being evaluated) is to be considered in the beginning stages of beta testing. Further features will be added based on community feedback and/or personal ideas for the project.

To elaborate on the requirements specifications being ‘almost fully matched’, the following requirements specifications were met:

* There is a main menu with a start button, continue button and quit button
* There is an in-game menu in which you can quit and save
* There is *one* rendered maze level
* There is a player class represented by a sprite which can be controlled by the user
* The sprite has collision detection
* The sprite can win when it exits the maze
* The sprite can move up, down, left and right on user command
* There is a save and load feature
* The game can run on a Windows 7 computer
* The game can run on a Windows 10 computer
* The code contains a 2D array within the maze class/subclasses
* The code contains recursion in the way that the player moves (see player.moveRight/player.moveLeft/player.moveUp/player.moveDown)

(See data dump file for further info)

The code has been refined as far as possible for its current state of development, further developments can be seen in the last page of this document. Some of the optional user requirements that weren’t met at this stage of development include:

* Better graphics than just ‘Rect()’ objects
* Higher difficulty than a simple maze
* More levels

These were not carried out due to the time frame that was given to make this program. The end product of this project is considered to be the first useable version of the product and is only in the first stages of beta testing. Therefore the above requirements will most likely be implemented when further development is carried out.

*Final testing Evaluation*

The testing phase was carried well, with a detailed test plan which was carried out completely. The way in which the program was tested was systematic and ensured that all program components were tested both individually and together. However, the testing could definitely be more efficient/faster. The way that the program was tested *did* test all components but due to the systematic nature of the testing there was a lot of repetition when testing certain components. This was especially evident in the integration testing, where many components who had already passed integration testing were retested every time another component failed. This meant that more time was spent on testing than was strictly necessary. To improve upon this in the future, it would be much more efficient to omit components that have been proven to work in integration previously from the next iteration of integration testing. Then once the tester thinks that all parts of integration testing are complete, do a full run through of the program to check that components definitely work fully in integration testing. If components do not work together fully, integration testing will start again. This would be a more efficient way to test integrative aspects of the program.

*End User Testing Evaluation*

The end user testing could be much improved in two ways:

* More participants
* Further developments based on feedback

The reason that these two improvements were not made whilst end user testing was carried out was due to time limitations. Very few end users could be found to test the program in the two or three days that were given to carry out end-user testing. As well as this, once the feedback had been received, there was not enough time to implement the features that users asked for. These features will most likely be found in further developments past this stage of development. Generally, the end users were not taken into consideration enough due to time limitations of the project. The main aim of this project up to this point was to achieve the vital components of this program, then test and assess them. Other components that were requested by end-users were seen as optional. They will be taken into account in further development.

*Development process*

*What went well:*

* The development process was originally meant to follow the waterfall methodology, but later changed to follow a methodology more similar to the SCRUM methodology, except with only one developer. A product backlog was created on paper and reviewed at the beginning of each day, selecting things to do for the product. Due to the short time-frame that was given, it made more sense to use an agile development methodology. Even if all requirements specifications of the project weren’t completed in the time given, there would still be working components to show. The SCRUM methodology served well for the purposes of this project and ensured that the project was completed in the correct time-frame. Even if it wasn’t created in the right time-frame, the SCRUM methodology ensured that there would be a somewhat useable product at most stages of development.
* The simple nature of the concept meant that design of the product was straightforward and good for the time-frame provided.

*What didn’t go well:*

* Due to the small time-frame, many of the features that I wanted to implement could not be implemented. For example: I wanted to make the mazes more randomized, so that the users could get more enjoyment from the game, as there would be a new maze every time. This would have been difficult to implement effectively in the time-frame given, so it was skipped over.
* The simplistic nature of the project resulted in a distinct lack of interesting features. This can be solved in further development.
* The way in which buttons were implemented was not the most efficient way to implement them. The use of ratios instead of just defining the center of the button was more effort than it’s worth. However, by the time a better solution had been figured out, most buttons already had this system, making it difficult to implement a new one. So the old system was kept in place.

*Things that could have been done better as a result:*If my time management was better with other assignments and previous versions of this project, the features which should have been implemented, such as end-user requirements and ideas that I (the developer) had could have been implemented. As well as this, the game would be much more interesting than it is currently.

*Things that have been learned:*

* Better time management
* More realistic outlook on projects, especially when there is more than one thing going on
* The best development methodologies for each project type and time-frame

*Response to feedback:*

Generally my response to feedback when it came to my development methodology was pretty closed off. This was a problem, as it resulted in a large amount of time being wasted on the previous project. If I had been more open to feedback and questions, I may have had a better time-frame to complete the end product.

*Performance*

*What went well:*

* I managed to spend a large portion of my time towards the end of the project on this project. I had the willpower to cut away a lot of my social life in order to complete this project effectively.
* I founded my love for coding and programming again, which hadn’t been stimulated in a while. Towards the beginning of the year I was considering a different career due to my feelings of boredom towards coding.
* I seemed to be almost constantly head of the Gantt chart, which gave me a lot more time to go over the program.

*What didn’t go well:*

* My priorities towards the beginning of the year were not well organised. This resulted in a large portion of the project preceding this one taking up my time.
* My time management skills towards the beginning of the year were not good.
* My personal expectations of what I could and couldn’t do were clouded by confidence.

*Things that could have been done better as a result:*

I time management had been better from the start and I had been more organized and more realistic expectations set, this project would most likely have been finished mid-year, giving me much more time to complete the course and go over this project.

*Things that have been learned:*

* I learned a large amount about object oriented programming and python during this project. This includes; the recursion depth in python and how python as a language is not well suited to recursion,the way in which classes and functions can be integrated with each other to create a program and how to implement and manipulate 2D arrays in python
* I also learned a large amount about development methodologies and their applications throughout this project, which will undoubtedly help me in the future.

*Response to feedback:*

My response to feedback in terms of my own performance was well received. I was extremely self aware when it came to my own organisational and time management skills. As well as this, I listened to feedback when it came to the program itself readily and assessed whether this feedback could be applied to my program. This is evident in the way that I put my head down and worked throughout the last push of my project. This was almost entirely motivated by those who had given me feedback on my previous project’s performance.

*Further development*

Some features that will be implemented in future versions of this game include:

* Randomized maze layouts
* Larger maze layouts (e.g. 20x20 instead of 10x10)
* Easter Eggs (e.g. being able to walk into walls, leading to different mazes)
* Better graphics (Based on user feedback)
* Soundtracks (Based on user feedback)
* A storyline
* Changing buttons to a better system (i.e not making the center based on ratios), making it easier to create buttons in later code

An end goal for this project will be public release.

*Conclusion*

To conclude, this project was successful for the amount of time that was given to complete it. However, many future developments are likely to be completed. The current maze game is relatively simple and not extremely interesting but with a few tweaks, could be made better. This is providing that end user feedback is taken into account. The tweaks that should be made in further development are detailed in the above section.

Furthermore, on a personal level, this project was a great learning opportunity for me (the developer) and taught me many things about development processes, object oriented programming and how to manage/develop a software based project. The experience I earned through this project will no doubt be referenced in many later projects, whether they be personal or work based. The improvements detailed in these documents will be taken into account and the performance improvements will be applied to not only my educational/work life, but also my personal life.

Many of the further developments that are going to be made will make the game much more interesting and generate more user interest. The end goal of this project is for the project to be released publically. Whether or not this game is well received by the public is irrelevant as this is mostly a personal project and I am doing it just because I can and not for any commercial gain.