Alexander Goslin

Final Project Report

Introduction:

This report covers the timeline of the project development during the COMP1004 project. The project name is Adventure Simulator. Adventure Simulator is a single page application which boasts a text-based adventure game with branching paths leading to different possible stories to tell. This report will cover how I managed to organise my time and how the project became what it is, and how it relates to the Software Development Lifecycle. It will also document the struggles and issues that I had come across as the developer and how a lot of the issues were fixed along the way.

The SDLC:

1. Analysis
2. Design
3. Implementation
4. Testing
5. Deployment
6. Maintenance

The Agile Software development lifecycle (SDLC) has been successfully implemented into my work routine during this project and others during this course. I implemented a 7-sprint plan which each spanned about 2 or 3 weeks depending on the length of the work I had assigned myself and my availability during the time. Being able to achieve this time management process over a long period of time is a first for me, which showcases how easy, yet efficient structured time planning can be. There were multiple instances of long stretches of time that lacked progress due to bugs in the code and unexpected results that took long periods of time to iron out. I do however believe that the sprint plans could have been more ambitious at moments, several times I caught myself assigning small amounts of work. Later in the project I would increase the workload per sprint however it would have been beneficial to have a more even work routine during the entire project. During the project I had many deviations in my plans and several times changed entirely what the application would function and look like, the SDLC structure to my work gave me good time to work around these changes without falling behind in terms of work progression.

Requirements:

This requirement segment will be divided into three sections: must, should and could. This includes parts several of the project I had envisioned that may have been discarded due to them being too ambitious or being unnecessary in the final product. Those marked as scrapped have been tested and tried but were cut off due to not being functional and difficult to implement in a meaningful way.

Must:

1. Track time played. – functional
2. Offer username function. – functional
3. Have multiple choices and questions. – functional
4. Create save file and save onto the machine. – functional

Should:

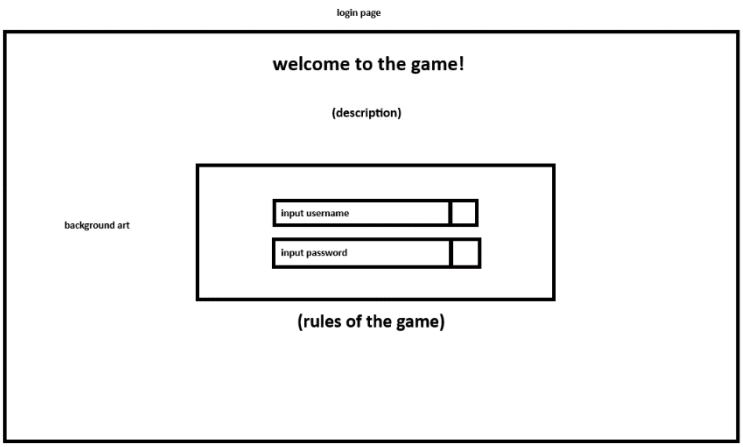
1. Have XP function with a level feature. – functional
2. have a loading feature with old save files able to resume a game. – scrapped
3. Interactive display with custom art. – functional

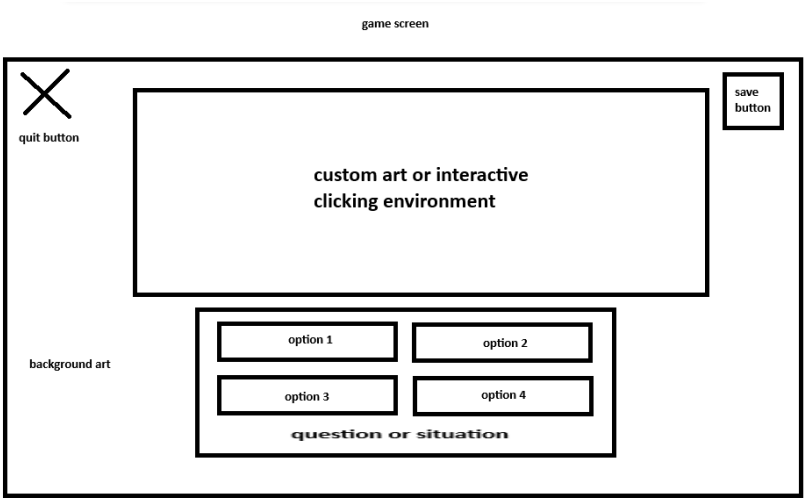
Could:

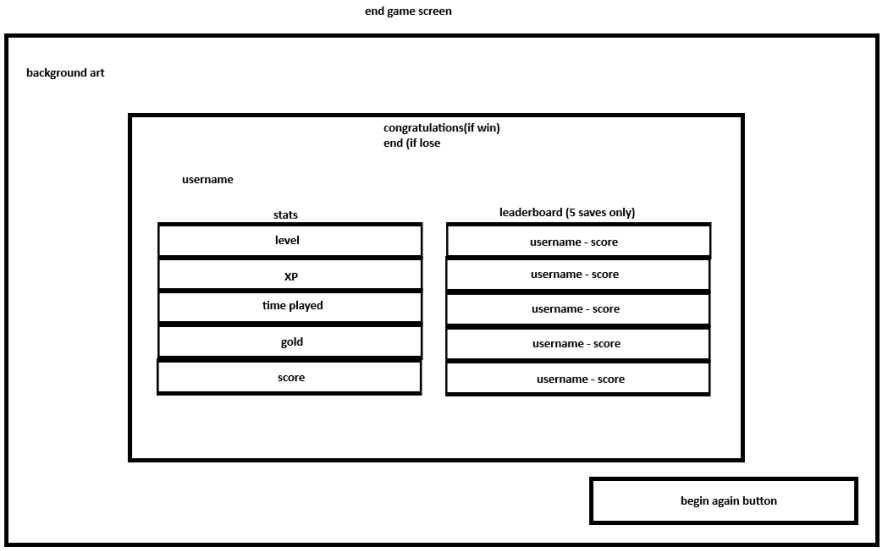
1. Interactive environment with character movement. – scrapped
2. Ability to go back adjust an answer if user makes a mistake. – scrapped

Initial vision:

In this section I will show what my initial vision of the project is and how I imagine the application will turn out to be. This is subject to change over time and some plans may be too difficult to implement as time goes on. Below are some images of how the pages should look and a rough idea of how they will function.







Initial vision review:

Upon looking back at the initial vision now the project has been completed I have realised at the beginning I made many alterations to the product and cut many things out of the product due to difficulty of implementation and time management. However, I do think that in general the structure of the project has remained quite faithful to the initial vision, I am happy with the changes I made although perhaps I could have pushed myself further. The final product also has some improvements that the initial vision did not consider.

Project description:

Video games have been a source of entertainment that has skyrocketed like no other in the last 40 years. From the point of non-existence in the general public to being so widespread as it is today have inspired many people, me included, so I thought that creating a project about a hobby shared by so many would be a great project idea. This project takes inspiration from point and click adventure games which were majorly popular in the 90s, but as technology grew greater, they were quickly outdated due to computers being able to run games with higher interactivity. Even though the game genre is long gone, it still has a unique character which I wanted to explore. This will have a nice interactive user interface with custom art and background.

HTML, java script and CSS have been used to code this application, along with JSON file functionalities to be able to save data from a player once they have completed the game. JSON files are one of the best options for save files due to their security and ease of use as a designer of the application. It also will display many different values that the player has accumulated and gotten during the game and saves them all onto the JSON file which is saved onto the players machine once the game has been completed.

Sprints:

Sprint 1: planning and conception

During the first sprint I am still debating what to make the project, and am playing around with java and HTML programming, poking and prodding at the limitations of the code and navigating my way around how to make a single page application. However, after a week or two I have made my mind up that I want to make a text-based adventure game. I decided upon this because during my childhood I played point and click adventure games on the family computer like Monkey Island amongst others, and I wanted to try and create something similar.

Sprint 2: the beginning of the implementation

In the second sprint I plan on beginning the first step of developing the adventure game itself. I have been putting in research into other text-based adventure games seeing how they tackle the functions and themes of a project like this. I believe I have a good idea of how to structure the text game and what to strive towards. In light of this I have created an application that has a functioning button that changes the website once pressed which will become the decisions made in the game in the future.

Sprint 3: small bug fixes and further implementation

I have begun to try to create the fully fledged game and started on how to get from one decision to the next seamlessly without having any effects on screen that might be jarring. A number of issues I have faced is text moving to places not intended making the bugs and strange movements distracting for the player. The objective of this current sprint is to make the application run smoothly before creating a story or a better user interface and get a better understand of how to create a streamline application.

Sprint 4: storytelling and creative direction

I have successfully created the smoothly running application I have set out to create, and now I shift my focus to creating a story. This is another problem I have run into, a lack of experience when it comes to storytelling. I have settled on a simple story simulating an adventure simulation, but during development I created a very ambitious story with many branching paths, but then when I realized the scope of what I had created I had to cancel it for something of less scale for it would have taken an extreme number of resources to create such a story with as many branching paths. In the end I had created a tree of branching paths on pen and paper of how I would like the story to go.

Sprint 5: functional file work

I have implemented the story into the code of the website, created a script that tracks time played and a function that saves the data into a JSON file onto the player's machine. The website works very smoothly, and its functionality is just as I have envisioned from the beginning. This has probably been the most time-consuming step of the project. Compartmentalising my time into small pieces of coding allowing me to have stress free leisure time has made this project not only be realised sooner, but also increases the quality of work.

Sprint 6: creating a user interface.

Now with the code fully functional, it is time to create a user interface that I believe has a nice aesthetic without being invasive to the user and simple. I have decided on a multicoloured background that I have created, this creation process was very fun, and I experimented with all sorts of different shapes, colours and aesthetics until I landed on one I was happy with. It required some knowledge in SVG which is a vector file format which allows you to store images as a mathematical formula based on points and lines on a grid, this was excellent for me as I have always enjoyed mathematics and gave me an opportunity to express that in a creative way. Along with coloured buttons for the options with custom artwork images for the pages. Overall, I am happy with how the website has turned out and is nearing completion.

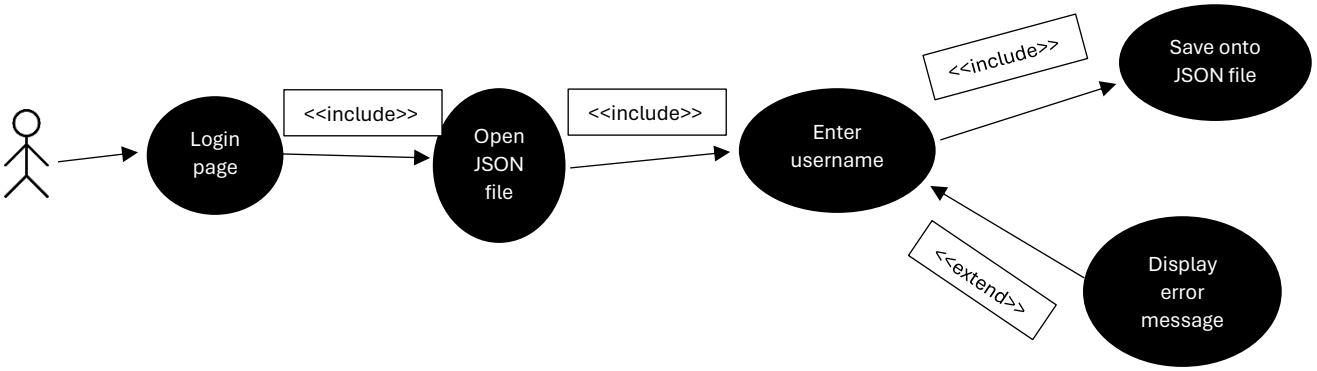
Sprint 7: testing the functionality and removing bugs

The final sprint includes testing and maintenance of the application. I have done numerous tests myself and ironed out many bugs that came with the time function, and the username function in the game. I have also gotten friends to playtest the game and they have come back to me with a few bugs including the background not loading on specific pages, text appearing in unexpected locations and the username function not working with specific non ascii characters. Another more technical bug is saving the JSON file onto the machine. Multiple times when I tried to save it, I could not find it, so I had to find a way to make it save in a place that was easily accessible which took a week to figure out. These issues have been fixed and the application runs smoothly with minimal errors and bugs.

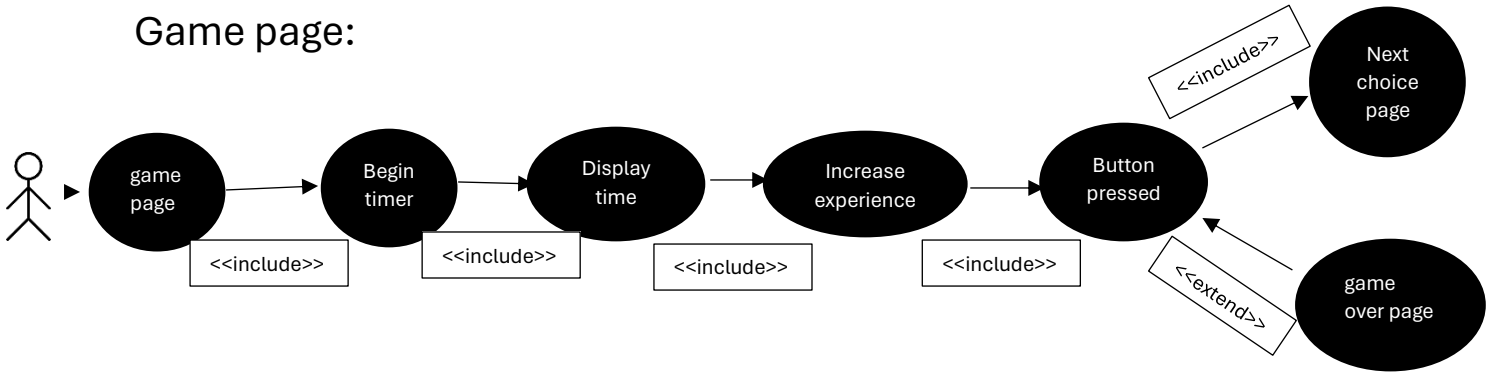
As a whole the records of my sprints have remained manageable, and I haven't felt that I have fallen behind on delivering what I intended to do for that segment of time. This could be due to multiple factors such as simply not delivering enough and not pushing myself or organising my time correctly. Near the start I did have too much planned for the sprints but the further I went the more I felt like I had to do, perhaps that is a symptom of poor work and time management that could be improved upon. The entire sprint process was very beneficial to the process of creating a large-scale project such as this and I will implement this in the future of my university education and even beyond that.

UML diagrams:

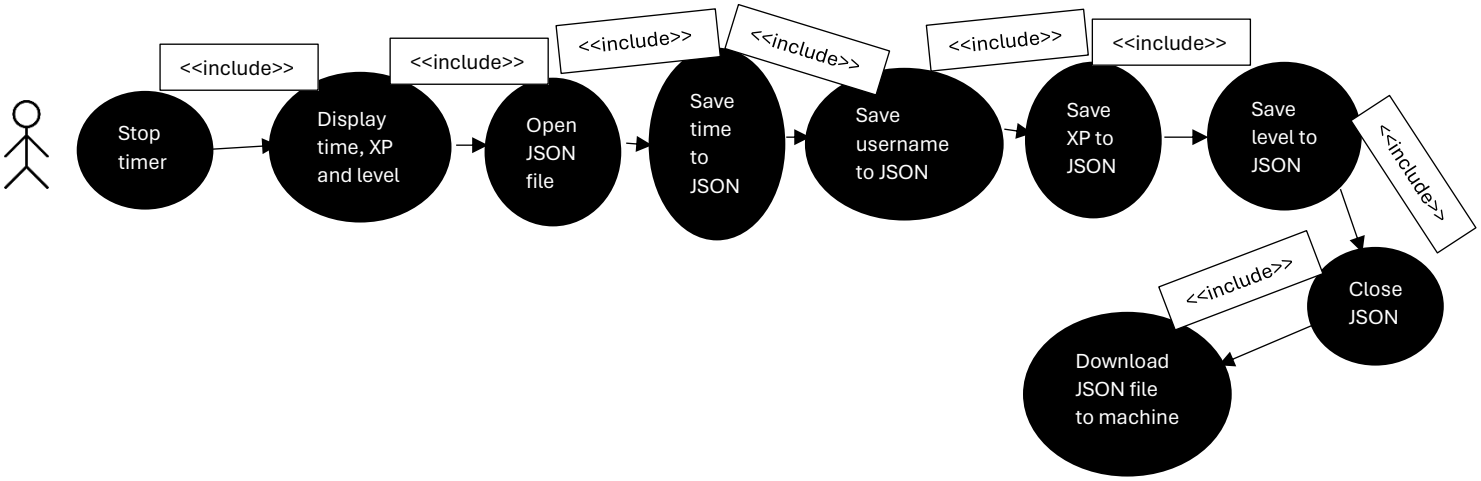
Login page:



Game page:



Game over page:



Testing:

Test 1:

Time function works while not on the browser – successful.

Test 2:

Website can open JSON file – successful.

Test 3:

JSON output to local storage – successful.

Test 4:

Website can save data onto JSON file with correct structure – successful.

Test 5:

Website can scale onto resolution of any size – successful.

Test 6:

Website can accept any symbol for username – successful.

Test 7:

Choice buttons continue to the next page – successful.

Test 8:

Website is easily used and not jarring to look at – successful.

Test 9:

Website functions on different OS – successful.

Test 10:

Website background is centred correctly behind all else – successful.

Test 11:

End game screen displays all value and username – successful.

Test 12:

Save files will not overwrite – successful.

These tests were all verified on the 28th of May 2024 but were all tested during the development process at during their integration into the code at dates unknown.

Reflection:

In terms of the project itself I think I have done quite well for myself, of course there were a few things I wish I had done and implemented now that I look with hindsight. But I think I pushed myself adequately and created a good project and good effort with the time I had. If I could go back and do it all again, I think I would develop a save and load function, as I had tried to do that although I couldn't quite grasp how to read a save file and input the values into a meaningful position in the game, but if I had then it would have been impressive. When it comes to time management, I think the SDLC alongside with a more open-minded approach to completing the work was very beneficial and effective when it comes to long term projects such as this. Compartmentalizing your work into small segments which makes your work seem more achievable than looking at it all as one big continual line of work makes the motivation to do it a lot more common.

Initially the Adventure Simulator had to be cut back due to a very ambitious outlook on how much time it would take to create a story and website of that scale. There were many instances where I had given myself too much and too little work, such as the story of the adventure itself was going to be incredibly large spanning 15 endings each with the same amount of decisions required to get there, which would have taken many hours of creating the same code over and over when that time could instead be used to create more technical functionalities in the project instead of making a glorified novel to go through. I am happy with that decision to take the general approach of the project into a more technical line instead of focusing on the storytelling where the website itself would not be particularly impressive, just long for the sake of it.

Following Agile, my work output has been better than ever before even still when I had encountered bugs and errors that blocked me from progressing further in the development process. This was because the Agile work routine gave me time to fix these bugs with ample time and no stress since the work was all achievable even with minor setbacks along the way. Several of these bugs included the JSON file not opening and storing data correctly and not in the format I wanted it to appear in. Another bug was trying to get the HTML and the java script to function together, it took me many attempts to fix it alone and finally solved the problem, having to redesign the entire website for the bug fix to function.

Overall, I am very happy with the way the project has become and the process of documenting the steps that I have taken, and creating the code itself were all very rewarding. If I had the opportunity to do the project all over again, I would do many things similarly regarding time management and overall direction, but still would have changed a few things such as focus on story which wasted much time along with having a more explorative approach to new untaught functions I could have implemented. In terms of the final product, I am proud it functions well and has a nice look to it without it being jarring to the user.

GITHUB REPO:

Github repo – https://github.com/AGoslin1/COMP1004-2024.git

References:

Agile sprint project frame - https://www.agilebusiness.org/dsdm-project-framework/introduction.html

UML class diagrams and the general way to make one - https://www.lucidchart.com/pages/uml-class-diagram

Text based adventure games I researched for inspiration –

https://en.wikipedia.org/wiki/List_of_text-based_computer_games

help with JSON files used in sprint 7 –

https://blog.hubspot.com/website/json-files#:~:text=Opening%20JSON%20files%20is%20far,and%20even%20command-line%20interfaces.