This is my introduction for Comp1004, we have been asked to create our own SPA (Single page application) using HTML and other resources we have learnt during our first year.

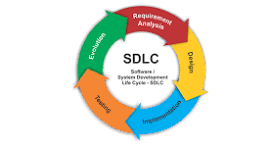
This project is mainly about our computer skills, how good our code is etc, but its also a test of our planning skills, our time management and how we would work in a real-world company. Completing tasks set by either our bosses, or by clients requesting a certain thing. The changes they might make halfway through.

This will be carried out using the SDLC (Software development Cycle), I will explain this later on, but it’s a method used by real people to effectively manage time and communicate with other colleagues how far along the task is.

“If you fail to plan, you are planning to fail” – Benjamin Franklin

As I am going to be doing Games development, I’m going to research into completing a task that is relevant to the course.

DISCUSS THE SDLC



Sprint 1 (Jan 1 – Jan 15)

My first sprint will be the overall design of the project, as this needs to include enough information for me to do a short video presentation I will try and work out the overall design of the project, this will change as we get further into the sprints, but having a goal to aim towards I believe this will help.

I decided to do a text-based dungeon crawl, multiple rooms that you have to make it through by deciding what actions to take against certain creatures and room obstacles. My skills aren’t high enough to create a digital game with graphics so a simple text based games with a few pictures will be great.

To do this I attended lectures about how to use HTML to create a basic webpage, and I started to draw a basic outline of what I wanted my pages to look like.

It would start with a small description of the room you are in, and any monsters that are in the room. They will have a couple of options as to actions they can take, for example

* Fleeing the room.
* Using their weapon to attack.
* Trying to bribe the monster.
* Dodging the trap that they have triggered.

I learnt from the lecture we can have clickable options below so this is how the player will choose to proceed, by clicking on 4 options.



A very rudimentary drawing but this is what I will try and turn into digitally using HTML coding and C++, if the player picks the right choice they will go forward in the dungeon, if they choose incorrectly many things might happen;

* Losing health, they will start with a set amount of health if they lose it all they fail the game.
* They may have to take a longer route through the dungeon, I’m considering setting a time limit on the game, so a penalty of picking the wrong choice might stop them completing the game.
* They might have to go back to the start and try again.
* If they choose correctly they move through to the next dungeon with no penalty.

Sprint 2 (Jan 15-Jan 29)

The next two weeks will be me trying to learn HTML and get the first page up and running, I began using Visual studio, but for some reason after a long time and many only tutorials, my computer at home wouldn’t accept the DOCTYPE, couldn’t understand why. I chose to move over to visual studio code, as the online help for this is more substantial and my computer at home will run it, making it easier to complete my work.

I decided not to apply the SDLC to this sprint, as the requirements analysis is my personal learning journey, but this was my first attempt at using visual studio in HTML.

A screenshot of a computer

Description automatically generated

A very simple first page, but it is what we have learnt in lectures so far, so I will wait for next weeks lecture to begin flushing this out.

Sprint 3 (Jan 29 – Feb 5)

This will be a continuation of the last sprint, I will be spending the next two weeks learning to use HTML.

I also decided to start mapping out what I wanted my first level to be, I wanted a very simple map to start with a few elements.

* A normal path going through the dungeon.
* A shortcut, where you could skip a few rooms.
* A detour, where the player has made a mistake and taken a longer path around the dungeon.

A diagram of a algorithm

Description automatically generated

The level will have a start point and an end point, where you finish the dungeon.

I decided to base my game of a world I had be making for a long time called “Age” and make the various levels places in my world, luckily, I had various scenarios that would give enough variety to keep the player playing, but will discuss this in a later sprint.

I also decided to make the game a kind of quiz game, where you must solve a riddle to pass onto the next level, making a mistake would take you on a detour or make you lose a life.

Sprint 4 (Feb 5th – Feb 19th)

A very successful sprint, really solidifying my knowledge of HTML by playing around with text, colour, images and even music.

I also started to work on the code of the game:

Requirement Analysis

As I have decided to make this game a text-based game, I need first to create some code either in HTML or C++ so that the player has an area they can type into and code that reads their answer and compares it to the answer I have.

Design

I had a few options for my design, either I could use hyper links within the HTML that would give you four possible answers to the riddle, if you clicked on the right answer, it would send you to the next room, but felt this wasn’t utilising all the skills I have learnt in my other modules.

My decision was to build a very short game using HTML, using hyperlinks to jump between the pages depending on what answer they selected, with further sprints and help from professors, I would use, C++ code or Javascript, using a “cin” input that would read the players answer for the last level.

Implementation

I began with trying to use the hyperlinks to jump between them, this was relatively easy, giving the player four possible choices, if they chose wrong they jumped back to the start and had to answers the questions again, if they chose correctly they moved onto the next level.

Testing

This worked well, it looked simple but worked, here is a small image of the rooms

A screenshot of a computer program

Description automatically generated

I simply copied this code five times onto the other pages and filled in the riddles, taking you to an end page congratulating you on completing the game.

Evolution

There are many things I would improve including;

* Making the page looks more exciting, adding borders and images.
* Implement my shortcut and detour rooms, I’m hoping to choose some riddles with multiple answers that can be correct, not sending you back but sending you on a detour so you don’t have to do all the levels again.
* The biggest thing I wanted to try was voicing the Owl that asks you the questions, my friend is blind and would like him to play.

Sprint 5 (Feb 20th - March 3rd)

With a short break I decided to try out voicing the characters for my game, making the riddles voiced and allowing my blind friend to play, all he must do is give me the answer and I will click the option.

I found free software through my IPhone that could record my voice and I could place into my game. This went perfectly and now one of my rooms has me talking through the riddle, I would like to voice the rest of the rooms but will hold of in case I decide to change the audio, riddles or levels.



If I manage to code the riddles, I will try and put in an option to answer the riddles with numbers, allowing my blind friend to play, so if he believes they third riddle is the answer he would put in 3 on the keyboard.

Sprint 6 (March 4th-March 18th)

Sprint 7 (March 19th-April 2nd)

Realised my files were saving in the wrong place and so weren’t being updated on the GitHub.

This sprint started with me learning CSS, which as described to me, HTML is the structure of the page, CSS is the style of the page. This is where I am going to make my SPA look more satisfying and not just plain text and white background. I started by creating a new page, copying and pasting room 5 into it and started playing around

OTHER

I wanted a log in screen and this is the design I thought would look best, it would be simple code I learnt from my first modules in Comp1000

It would store

USER STORIES

As a user I want to play more puzzles

As a user I want to be rewarded for guessing a riddle right and punished for getting one wrong