Adventure Game Project

Introduction:

I have created a quick, choice-based fictional adventure game, using HTML, CSS and JavaScript. This project uses SDLC carefully to achieve a sufficient and detailed game. I am using Visual Studio Code as the IDE, because I much prefer it for HTML than regular Visual Studio.

SDLC:

The software development lifecycle for this project was important to structure and define functionality in a clear way. Initially, I was brainstorming ideas and using the resources given to me to create a plan for a fictional, choice-based game. This took me a few days, by the end of the week I had settled on what to do. This idea emerged from inspiration of the culture of adventure games, books and movies. The design phase started as I took notes for features I could implement for the user interface and the code structure. I created sprints to analyse what to focus on for each couple of weeks of development.

Sprint Log:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Deliverable | Sprint 1 | Sprint 2 | Sprint 3 | Sprint 4 | Sprint 5 | Sprint 6 | Sprint 7 | Key |
| User interface using CSS |  |  |  |  |  |  |  | Cancelled |
| Username input |  |  |  |  |  |  |  | Partially introduced |
| Game containers using HTML |  |  |  |  |  |  |  | Introduced, not tested |
| GameData properties |  |  |  |  |  |  |  | Testing |
| Random function |  |  |  |  |  |  |  | Fully tested |
| Load data using JSON |  |  |  |  |  |  |  |  |
| Physical Level design |  |  |  |  |  |  |  |  |
| Images |  |  |  |  |  |  |  |  |
| Functional level design |  |  |  |  |  |  |  |  |
| Health point features |  |  |  |  |  |  |  |  |
| Dynamic items |  |  |  |  |  |  |  |  |

This was my sprint log, using a colour key to depict what I could and what I couldn’t implement. To begin with, I started by creating the basic user interface using CSS. Then I created containers for data so that I could dynamically alter the situation of the game.

I would then later move onto creating gameData, to track multiple properties of the game, to make everything dynamically accessible and to minimize code usage. I used properties such as collectedShield, collectedSword, startTime, endTime.

I used the math keyword in JS to implement a random number generator feature, I wrote a function, tested it, then called it for one of the scenarios.

Just after this, I then created the JSON file to track the users progress of the game.

Some things I couldn’t get round to completing successfully were a physical level design, for the user to be able to look at and have some more immersion into the story and any images during playing the game. Another feature would have been to introduce a health point system, to give the player more of a chance to adapt, rather than trial and error. One more was dynamic items for example, a potion, armour, or ranged weaponry.

Design document:

Here I will provide an image of the basic opening page using prompt in JavaScript and asking for the user to enter a username.A screenshot of a computer

Description automatically generated

This is one of the newer functions I have implemented to the game, where one option is defeat, and will reset the game, the other is an item needed to progress through backtracking and completing something that wasn’t doable before. This feature makes the game uniquely entertaining.

A screenshot of a computer

AI-generated content may be incorrect.

This is a random number generator, simulating a dice roll, much like dungeons and dragons. To creep past you need to get an item, and to kill the dragon you need to roll a 4 or above, giving the player two choices in winning. If you creep past without the item, you will lose.

A screenshot of a video game

AI-generated content may be incorrect.

Project vision:

For the general person who is bored on a daily basis, and they need something to pass time, especially young people or teenagers. The vision for this game aims to be an easy to play game with and exciting level of risk.

Background:

Adventure is a single web page application, fictional, choice-based game. This game involves entering a username and choosing multiple different paths to reach the end. This is a small version of games where choices are highly important, and some crucial thinking is needed to successfully traverse through the scenarios. This game aims to paint an image in your head of the scenario that is being described, to immerse you in a simple but effective way. Simple games like this are good for a relaxing environment and should be played to relieve stress and promote a good mood.

As this game is an inspiration of many others, legal considerations should be made, this game was created mainly from the idea of the genre, not from any specific game or brand. The following image is a UML diagram of the potential implementations that were considered. The use of time management more effectively would have been optimal for these ideas to come to life. The include text represents what I should be adding, the extend is what ideas were there but could not be implemented.

User stories:

As a player, I would like a longer, more challenging game.

As a player, I would like a wider variety of tasks.

As a player, I would like a more strategic, and detailed game.

A diagram of steps to a path

Description automatically generated with medium confidence

Issues and challenges:

This project came with various issues, for one having to manage the scoreboard to effectively force the user to follow the rules about collecting the sword and the shield. I changed the gameData properties from a score system to a Boolean system. This was to prevent any score abuse, this was the code used for this:



I also needed for all the data to reset at the end of the game to force players to start over once they had been defeated. This is the function for that:

A screen shot of a computer program

AI-generated content may be incorrect.

<https://github.com/Luke-Edge/Comp1004-Project>