Simulated Concurrency Programming Game

Contents

[Sprint Backlog: 2](#_Toc126849608)

[Sprint 1: Thursday 02/02/2023 – 09/02/2023 2](#_Toc126849609)

[User Requirements: 2](#_Toc126849610)

[Design Choices: 2](#_Toc126849611)

[Low Fidelity Prototypes: 2](#_Toc126849612)

[Evidence of Progress: 3](#_Toc126849613)

[Sprint Summary: 4](#_Toc126849614)

[Plan for Sprint 2: 4](#_Toc126849615)

# Sprint Backlog:

## Sprint 1: Thursday 02/02/2023 – 09/02/2023

As the first sprint of the project, one of the main focuses is getting the project set up. This involves creating a git repository, setting up a unity project and collecting some of the required packages, such as Antlr4. Another key focus for this sprint is getting the core functionality of creating, editing, and deleting scripts.

### User Requirements:

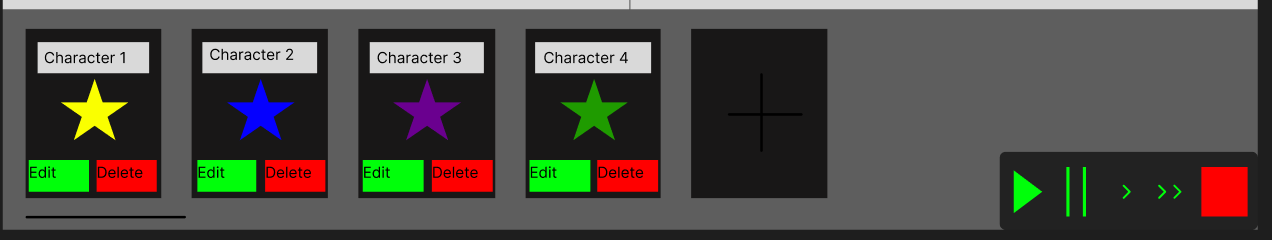
|  |  |
| --- | --- |
| **Story ID:** | **Description:** |
| 15 | As a player, I want to be able to access all options through a main menu and campaign map so that I can navigate to all features of the game |
| 1 | As a player I want to be able to create a new character script so that I can define the behaviour of the players on my team |
| 2 | As a player I want to be able to Edit/ Delete scripts so I can change that behaviour to optimise my team’s chance of victory |
| 7 | As a player, I want to be able to save my scripts so that they carry over between levels, and persist when I exit the game and restart |

Story 15 does include the concept of the campaign map, for this sprint, the menu system will be simplified to only include a sample play scene for development, a settings button (to be fleshed out later), and a ‘quit game’ option.

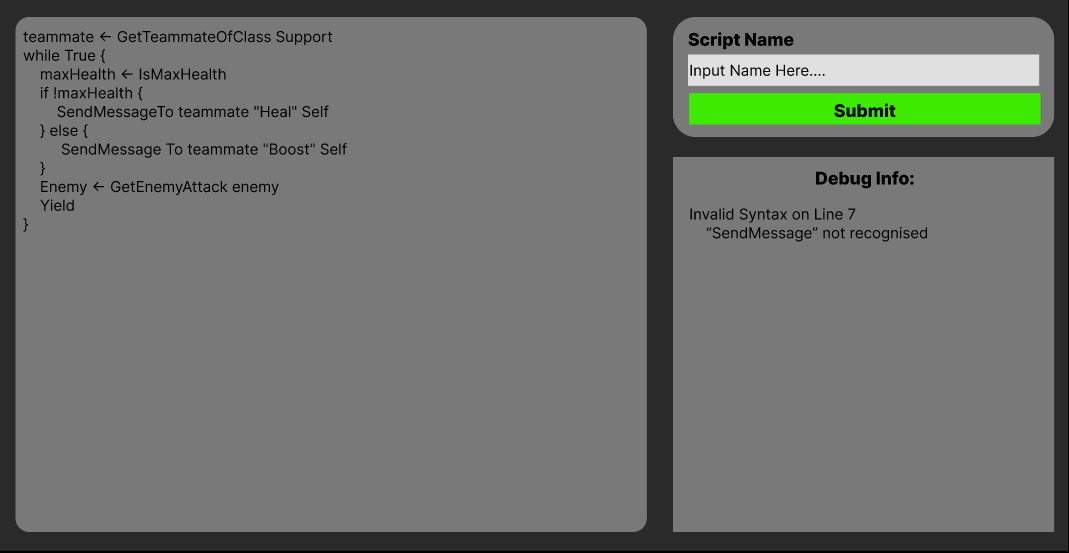
### Design Choices:

Since this sprint I will be implementing the lexer and parser for the language, I have already made some design choices from the original specification. The first of which is to do with iteration. In the early development stage of the language, iteration was designed to work in 2 ways; conditional loops and non-conditional loops (e.g., loop 5 times). I have decided to remove the functionality for non-conditional loops and only have ‘while <condition>’ as a feature. This is for two reasons, firstly, non-conditional loops don’t add any expressive power to the language, and only serve to add more choices to the user. By removing them, I will make the language simpler, and therefore more user-friendly for amateur programmers, which is the main goal of the game. Secondly, for complexity reasons. The language is already surprisingly complex for a small language and removing this feature now will serve to reduce my workload in later sprints.

### Low Fidelity Prototypes:



Referring back to the low fidelity prototype presented in the progress report, the aim of this sprint is to have a functioning version of the character management UI. This includes a box for each of the characters, as well as an Edit and Delete button for each one. By clicking on the ‘+’ button at the end of the list, a new box should appear allowing the user to choose from either an existing script, or to create a new one.



Here is an example prototype for the IDE. It uses a very simple design, with a textbox for entering the code, another for entering the name of the script (which is also the name of the character on the previous screen, and a box for potential debug info. The scope of debugging information is still to be decided. The submit button is used to save and close the script editor. When the button is clicked, the script is passed through a lexer and parser, if it has any errors, then the script will not be saved, and an error will appear in the debug box. Otherwise, the script will save, and the IDE will close.

### Evidence of Progress:

This is the early prototype of the main Menu:



For enhanced user feedback, each of the buttons changes colour when highlighted and clicked. The settings page is not currently implemented as that isn’t one of the core features of the game. Both the Campaign and Skirmish buttons currently lead to the same testing development scene as no other scenes have been implemented yet. Furthermore, the title and background are subject to change, as they are only placeholders for now.



This is a screenshot of the character control bar. It allows for multiple characters, some repetitions of each other. You can also click to add a new character, after which you can select from a list of existing scripts, or create a new one, opening up the IDE. The white boxes will later have images of the characters, so you can see which is which in the battle scene above.

This is the IDE in its current form. There is a section for writing code, a place to enter the name of the script and some debug info. There are various feedback systems such as the button flashing red to show an error, or debug info appearing in the box on the right.

### Sprint Summary:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | Sprint 1 Backlog: | | Setup Repo and Unity | | Install and configure Antlr4 | | Create Main Menu | | Create Script Editor | | Create Script | | Save Script | | Delete Script | | Edit Script | | Create Language Grammar | |  |

Here is the burndown chart for the first sprint. For the first few days, progress was slow, this is mostly due to the difficulty of installing and configuring Antlr4 for use with Unity, this task alone took almost 3 days and was pivotal for moving forward with the project. Once this was out of the way, the other tasks fell into place quite quickly, saving and loading files was the other main challenge to overcome, as I did not have previous experience with file IO in C#. All tasks were completed ahead of schedule, meaning the last few days could be spent cleaning up code and working on extra features not neces sarily mentioned in the spec, for example, setting up Audio functionality for later on.

### Plan for Sprint 2:

Referring back to the Gannt chart in the progress report, my next steps in development are to develop the scheduling algorithm and execution model. This is quite a large task, so I’m going to allocate 2 weeks to the next sprint. Below is the sprint 2 backlog and burndown chart setup.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | | --- | | Sprint 2 Backlog: | | Design and create thread scheduling algorithm | | Connect Lexer and Parser to IDE | | Build AST from scripts on play | | Set character class | | Create execution model | | Integrate interpreter into scripts | |  |