**Project Sprint #1**

The SOS game is described in CS449HomeworkOverview.docx. You should read the description very carefully.

Your submission must include the GitHub link to your project and you must ensure that the instructor has the proper access to your project. You will receive no points otherwise.

GitHub link:

In this assignment, you aim to specify the requirements (i.e., user stories and acceptance criteria) of the target software that allows a human player to play a simple or general SOS game against a human opponent. These requirements will be fully implemented by the end of sprint 3. The minimum features include **choosing the board size,** **choosing the game mode (simple or general)**, **starting a new game**, **making a move (in a simple or general game)**, **determining if a simple or general game is over**. The following is a sample GUI layout.

|  |  |  |
| --- | --- | --- |
| SOS Icon  Description automatically generated Simple game Icon  Description automatically generated General game Board size  8 | | |
| Blue player  Icon  Description automatically generated S  Icon  Description automatically generated O | Chart, line chart  Description automatically generated | Red player  Icon  Description automatically generated S  Icon  Description automatically generated O |
|  | Current turn: blue (or red) | New Game |

Figure 1. Sample GUI layout of the first working program by the end of Sprint 3

Use the following tables to document your user stories and acceptance criteria.

You are required to use the free ChatGPT version to complete 2 user stories and their respective acceptance criteria. You also need to ensure that the generated user stories are correct and refine them if not. At the end of the submission, provide screenshots of your ChatGPT prompts and answers, along with errors the ChatGPT made and that you had to correct. You may also use LLMs hosted locally. Points will be deducted if no screenshots are provided.

1. **User Stories (3 points)**

* **User Story Template**: As a <role>, I want <goal> [so that <benefit>]

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **User Story Name** | **User Story Description** | **Priority** | **Estimated effort (hours)** |
| 1 | Choose a board size |  |  |  |
| 2 | Choose the game mode of a chosen board |  |  |  |
| 3 | Start a new game of the chosen board size and game mode |  |  |  |
| 4 | Make a move in a simple game |  |  |  |
| 5 | A simple game is over |  |  |  |
| 6 | Make a move in a general game |  |  |  |
| 7 | A general game is over |  |  |  |

1. **Acceptance Criteria (AC) (8 points): Add/delete rows as needed.**

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story ID and Name** | **AC**  **ID** | **Description of Acceptance Criterion** | **Status (completed, toDo, inPprogress)** |
| 1. Choose a board size | 1.1 | AC 1.1 <scenario description>  Given  When  Then |  |
| 1.2 | AC 1.2 <scenario description>  Given  When  Then |  |
| … |  |  |
| 2. Choose the game mode of a chosen board | 2.1 | AC 2.1 <scenario description>  Given  When  Then |  |
| … |  |  |
| … |  |  |  |

1. **Data flow diagram (DFD) (4 points):**

Consider a web-based SOS game that allows players from all over the world to play SOS games against each other (similar to chess.com for chess). Using data flow digram examples presented in class, draw the data flow diagram for your global SOS game. You may use the tool of your choice, but it is a good idea to get familiar with draw.io (https://app.diagrams.net/) or other similar tools.