**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.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

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 | As a player, I want to select a board size so that I can customize my game experience. | High | 4 |
| 2 | Choose the game mode of a chosen board | **As a** player, **I want** to choose the game mode **so that** I can play either a simple or general game. | High | 3 |
| 3 | Start a new game of the chosen board size and game mode | **As a** player, **I want** to start a new game **so that** I can begin playing immediately. | High | 2 |
| 4 | Make a move in a simple game | **As a** player, **I want** to make a move in a simple game **so that** I can play the game. | High | 3 |
| 5 | A simple game is over | **As a** player, **I want** to know when a simple game is over **so that** I can see the result of the game. | High | 4 |
| 6 | Make a move in a general game | **As a** player, **I want** to make a move in a general game **so that** I can participate in a more complex game. | High | 4 |
| 7 | A general game is over | **As a** player, **I want** to know when a general game is over **so that** I can understand the outcome of the game. | High | 4 |

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 <game setup – board size>   1. **Given** I am on the game setup screen, **when** I select a board size, **then** the selected size must be between 4 and 8 (inclusive). | inProgress |
| 1.2 | AC 1.2 <game board size – user conf and board initialization>  **Given** I have selected a board size, **when** I confirm my selection, **then** the board should be initialized with the chosen size. | inProgress |
| … |  |  |
| 2. Choose the game mode of a chosen board | 2.1 | AC 2.1 <after board size selection – game mode selection>  **Given** I have selected a board size, **when** I choose a game mode, **then** I should see options for "Simple" and "General." | inProgress |
| 2.2 | AC 2.2 <selected board size and game mode is displayed correctly> **Given** I have chosen a game mode, **when** I confirm my selection, **then** the game mode should be set and displayed on the screen. | inProgress |
| 3.Start a new game of chosen board size and game mode | 3.1 | AC 3.1 <new game initialized>  **Given** I have selected a board size and game mode, **when** I click "Start Game," **then** a new game should be initialized. | inProgress |
|  | 3.2 | AC 3.2 <grid size and game mode displayed correctly in GUI>  **Given** a new game is initialized, **when** I view the game board, **then** it should display the correct size and game mode. | inProgress |
| 4. Make a move in a simple game | 4.1 | AC 4.1 <cell click on board shows player’s selection>  **Given** I am in a simple game, **when** I click on a cell on the board, **then** my move should be registered in that cell. | inProgress |
|  | 4.2 | AC 4.2 <player move selection reflected properly on game board GUI>  **Given** I have made a move, **when** I check the game state, **then** it should reflect my move. | InProgress |
| 5. **A simple game is over** | 5.1 | AC 5.1<simple game over message>  **Given** I am in a simple game, **when** a player has won or there are no available moves, **then** the game should display a "Game Over" message. | inProgress |
|  | 5.2 | AC 5.2<game over – game state check – display winner or draw>  **Given** the game is over, **when** I check the game state, **then** it should indicate the winner or if it's a draw. | inProgress |
| 6. Make a move in a general game | 6.1 | AC 6.1<player can execute valid moves on game board>  **Given** I am in a general game, **when** I click on a cell on the board, **then** my move should be registered and check for valid moves. | inProgress |
|  | 6.2 | AC 6.2<when player make a move and checks game state game GUI will reflect player move and game status updates>  **Given** I have made a move, **when** I check the game state, **then** it should reflect my move and the game status should update accordingly | inProgress |
| 7. General Game is over | 7.1 | AC 7.1<display Game Over message upon player win or no available moves>  **Given** I am in a general game, **when** a player has won or there are no available moves, **then** the game should display a "Game Over" message. | inProgress |

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.

A diagram of a computer

Description automatically generated