**Project Sprint #4**

Implement all the features that support a player (**human or computer**) to play a simple or general SOS game against another player (**human or** **computer**). The minimum features include **choosing human or computer for red and/or blue players**, **choosing the game mode (simple or general)**, **choosing the board size**, **setting up a new game**, **making a move (in a simple or general game)**, and **determining if a simple or general game is over**. The computer component must be able to play complete simple and general games. You are encouraged to consider basic strategies for winning simple or general games (e.g., against a poor human player). Optimal play is not required.

The following is a sample GUI layout. You should use a class hierarchy to deal with the computer opponent requirements. If your current code has not yet considered class hierarchy, it is time to refactor your code.

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

Figure 1. Sample GUI layout of the working program for Sprint 3

**Total points: 24**

1. **Demonstration (8 points)**

Submit a video of no more than five minutes, clearly demonstrating that you have implemented the computer opponent features and written some automated unit tests.

1. A complete simple game where the blue player is a human, the red player is the computer, and there is a winner
2. A complete general game where the blue player is the computer, the red player is a human, and there is a winner
3. A complete simple game where both sides are played by the computer
4. A complete general game where both sides are played by the computer
5. Some automated unit tests for the computer opponent.

In the video, you must explain what is being demonstrated.

1. **User Stories for the Computer Opponent Requirements (1 points)**

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **User Story Name** | **User Story Description** | **Priority** | **Estimated effort (hours)** |
| 8 | Using CPU Player(s) in Simple Mode | As a player, I should have the option to play against a Computer Opponent so that I may play a game in Simple Mode without a second human player. | 1 | 3 |
| 9 | Using CPU Player(s) in General Mode | As a player, I should have the option to play against a Computer Opponent so that I may play a game in General Mode without a second human player. | 2 | 3 |

1. **Acceptance Criteria (AC) for the Computer Opponent Requirements (4 points)**

Add or delete rows as needed.

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story ID and Name** | **AC**  **ID** | **Description of Acceptance Criterion** | **Status (completed, toDo, inProgress)** |
| 8. Using CPU Player(s) in Simple Mode | 8.1 | AC 8.1 Red Player CPU (Simple Mode)  Given a selected CPU button in the Red Panel  And an unselected CPU button in the Blue Panel  And Simple Mode is selected.  When the New Game Button is hit  Then a Simple Game will start with a Red CPU Opponent. | Completed |
| 8.2 | AC 8.2 Blue Player CPU (Simple Mode)  Given a selected CPU button in the Blue Panel  And an unselected CPU button in the Red Panel  And Simple Mode is selected.  When the New Game Button is hit  Then a Simple Game will start with a Blue CPU Opponent. | Completed |
| 8.3 | AC 8.1 Blue and Red Player CPUs (Simple Mode)  Given a selected CPU button in the Blue Panel  And an selected CPU button in the Red Panel  And Simple Mode is selected.  When the New Game Button is hit  Then a Simple Game will start with Blue and Red CPU players. | Completed |
| 9. Using CPU Player(s) in Simple Mode | 9.1 | AC 9.1 Red Player CPU (General Mode)  Given a selected CPU button in the Red Panel  And an unselected CPU button in the Blue Panel  And General Mode is selected.  When the New Game Button is hit  Then a General Game will start with a Red CPU Opponent. | Completed |
| 9.2 | AC 9.2 Blue Player CPU (General Mode)  Given a selected CPU button in the Blue Panel  And an unselected CPU button in the Red Panel  And General Mode is selected.  When the New Game Button is hit  Then a General Game will start with a Blue CPU Opponent. | Completed |
| 9.3 | AC 9.1 Blue and Red Player CPUs (General Mode)  Given a selected CPU button in the Blue Panel  And an selected CPU button in the Red Panel  And General Mode is selected.  When the New Game Button is hit  Then a General Game will start with Blue and Red CPU players. | Completed |

1. **Summary of All Source Code (1 points)**

|  |  |  |
| --- | --- | --- |
| Source code file name | Production code or test code? | # lines of code |
| Board.java | Production Code | 802 |
| GUI.java | Production Code | 1501 |
| BoardTest.java | Test Code | 317 |
| GUITest.Java | Test Code | 18 |
| Total | | 2,638 |

**You must submit all source code to get any credit for this assignment.**

1. **Production Code vs New User stories/Acceptance Criteria (2 points)**

Summarize how each of the new user story/acceptance criteria is implemented in your production code (class name and method name etc.)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **User Story ID and Name** | **AC ID** | **Class Name(s)** | **Method Name(s)** | **Status (complete or not)** | **Notes (optional)** |
| 8 | 8.1 | Board, GUI, SimpleCellActionListener | redCPU(), cpuRED(), cpuWinCheck() | Complete |  |
|  | 8.2 | Board, GUI, SimpleCellActionListener | blueCPU(), cpuBLUE(), cpuWinCheck() | Complete |  |
|  | 8.3 | Board, GUI, SimpleCellActionListener | redCPU(), cpuRED(),blueCPU(), cpuBLUE(), cpuWinCheck() | Complete |  |
| 9 | 9.1 | Board, GUI, GeneralCellActionListener | redCPU(), cpuRED(), cpuWinCheck() | Complete |  |
|  | 9.2 | Board, GUI, GeneralCellActionListener | blueCPU(), cpuBLUE(), cpuWinCheck() | Complete |  |
|  | 9.3 | Board, GUI, GeneralCellActionListener | redCPU(), cpuRED(),blueCPU(), cpuBLUE(), cpuWinCheck() | Complete |  |

1. **Tests vs New User stories/Acceptance Criteria (2 points)**

Summarize how each of the new user story/acceptance criteria is tested by your test code (class name and method name) or manually performed tests.

6.1 Automated tests directly corresponding to some acceptance criteria

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User Story ID and Name** | **Acceptance Criterion ID** | **Class Name (s) of the Test Code** | **Method Name(s) of the Test Code** | **Description of the Test Case (input & expected output)** |
| 8 | 8.1 | BoardTest | testRedCPUWin() | Input: Two S’s from Blue Human Player  Expected: Red is the winner. |
|  | 8.2 | BoardTest | testBlueCPUWin() | Input: Two S’s from Red Human Player  Expected: Blue is the winner. |

6.2 Manual tests directly corresponding to some acceptance criteria

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User Story ID and Name** | **Acceptance Criterion ID** | **Test Case Input** | **Test Oracle (Expected Output)** | **Notes** |
| 8 | 8.1 | Simple Mode selected. Red Human Player moves, deliberately throwing the game in favor of CPU. | Blue CPU win | This was to test GUI behavior when the CPU wins. |
|  | 8.2 | Simple mode selected. Blue Human Player moves, deliberately throwing the game in favor of CPU. | Red CPU win. | This was to test GUI behavior when the CPU wins. |
|  | 8.3 | Set both players to CPU and set mode to Simple. | Automatic CPU moves and a winner. | This test was done after the previous two and done in hopes of a complete game. |
| 9 | 9.1 | General mode selected. Red Human Player moves, deliberately throwing the game in favor of CPU. | Blue CPU win | This was to test GUI behavior when the CPU wins. |
|  | 9.2 | General mode selected. Blue Human Player moves, deliberately throwing the game in favor of CPU. | Red CPU win. | This was to test GUI behavior when the CPU wins. |
|  | 9.3 | Set both players to CPU and set mode to General. | Automatic CPU moves and a winner. | This test was done after the previous two and done in hopes of a complete game. |

6.3 Other automated or manual tests not corresponding to the acceptance criteria

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number** | **Test Input** | **Expected Result** | **Class Name of the Test Code** | **Method Name of the Test Code** |
| N/A |  |  |  |  |
|  |  |  |  |  |

1. **Present the class diagram of your production code (3 points) and describe how the class hierarchy in your design deals with the computer opponent requirements (3 points)**?

Attached this in a separate PowerPoint File.