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.

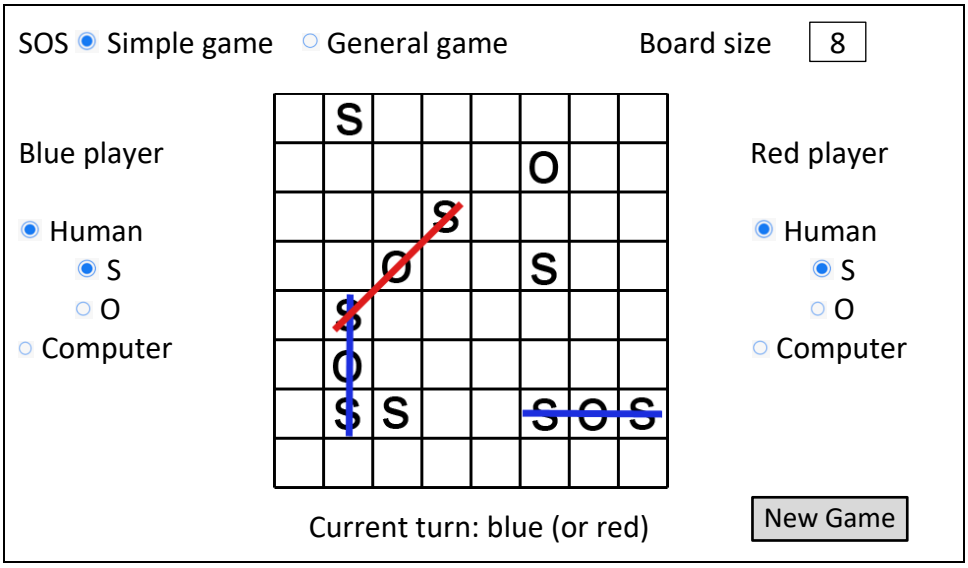

Figure 1. Sample GUI layout of the working program for Sprint 3

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.

2. 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				
..				

3. 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, inPprogress)
8 story one	8.1	AC 8.1 <scenario description> Given When Then	
	8.2	AC 8.2 <scenario description> Given When Then	
	…		
9 story nine if needed	9.1	AC 9.1 <scenario description> Given When Then	
	…		

4. Summary of All Source Code (1 points)

Source code file name	Production code or test code?	# lines of code
	Total	

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

5. 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				
	8.2				
	…				

6. 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)
1	1.1			
	1.2			
	…			
2	2.1			
	…			

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
1	1.1			
	1.2			
	…			
2	2.1			
	…			

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

7. 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)?