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 following is a sample GUI layout. It is required to 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.

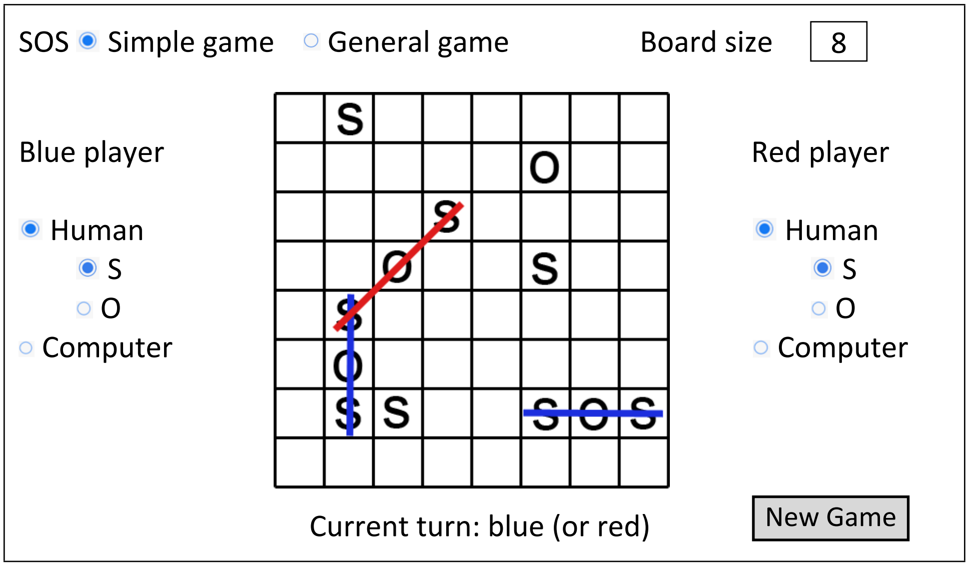

Figure 1. Sample GUI layout of the working program for Sprint 3

1. Demonstration (4 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)

1	Game type	As a player, I need to choose toplay a simple game or general game so that I can start a game.	High	2
2	What to put	As a red/blue player, I need to know what should I put next , so that I can make a move	High	2
3	Current turn	As a player, I need to know the current turn so that I can make a move	High	2
4	Empty board	As a player, I need an empty board of nxn(n>2) grids so that I can start a game.	High	2
5	Place move	As a red/ blue player, I need to place mu move on an empty cell so that I can make a move.	High	2
6	Game over	As a player, I need to know if the game is over ager each move.	High	2
7	New game	As a player, I need a New Game button if I want to start a new game.	high	2
8	Computer player	As a player, I want to player with computer	High	4

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

User Story ID and Name	AC ID	Description of Acceptance Criterion	Status (completed, toDo, inPprogress)
1 game type	1.1	AC 1.1 choose to play a simple game Given a choice of simple game or general game When I choose simple game Then the system should create a new simple game	Completed
	1.2	AC 1.2 choose to play a general game Given a choice of simple game or general game When I choose general game Then the system should create a new simple game	completed
2 what to put	2.1	AC 2.1 red player can see what the next letter should be Given a started game When the game continue And each turn should use different letter Then the system should show the button of the red player's next step letter	completed
	2.2	AC 2.1 red player can see what the next letter should be Given a started game When the game continue And each turn should use different letter Then the system should show the button of the blue player's next step letter	completed
3 current turn	3	AC 3 show current turn Given a started game When the game continue Then the system should show who is current	Completed
4 empty board	4.1	AC 4.1 create an empty board of n*n(n>2) grids Given the grid n of the board n>2 When choose the type of the game that I want to play Then the system should create a empty board of n*n(n>2) grids And it is red's turn to play	Completed
	4.2	AC 4.2 invalid empty board Given the grid n of the board n>2 When the number of n<=2 Then the empty board is invalid	Completed
5 place move	5.1	AC 5.1 a valid red(blue) move Given an ongoing game with red's(blue's) turn When the red(blue)makes a valid move Then red(blue) is placed in the cell And the turn is changed to blue(red)	Completed

	5.2	AC 5.2 an illegal red(blue) move on an occupied cell Given an ongoing game with red's(blue's) turn When the red(blue)makes an illegal move within the board Then the cell is not changed And the turn is not changed	completed
	5.3	AC 5.3 an illegal red(blue) move outside the board Given an ongoing game with red's(blue's) turn When the red(blue)makes an illegal move outside the board Then the turn is not changed	Completed
6 game over	6.1	AC 6.1 end of simple game When the player who create the first SOS Then the game is over And the winner is the player who create the first SOS	Completed
	6.2	AC 6.2 end of simple game When the player who create the first SOS Then the game is over And the winner is the player who create the first SOS	Completed
	6.3	AC 6.3 end of general game(draw) When the board has been filled up Then the game is over And the game is draw if both players made the same number of SOS	Completed
7 new game	7	AC 7 start a new game Given a click on "New Game " button Then the game will restart	completed
8 computer player	8.1	AC 8.1 choose to play with computer Given a choice of human player or computer player When I choose computer player on red(blue) Then the system should create next move after my turn	completed
	8.2	AC 8.2 choose to play with computer Given a choice of human player or computer player When I choose computer player on both Then the system should create move by itself	completed

4. Summary of All Source Code (1 points)

Source code file name	Production code or test code?	# lines of code
SOSGame.java	Production code	217
SOSGUI.java	Production code	297
	Total	514

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 user story/acceptance criterion 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)**
1Empty board	1.1	GameBoardCanvas()	Create a 8*8 board	Complete	
	1.2	drawGridLines()	Draw grid lines on board	Complete	
2 Game type	2	setContentPane()	Set details on board	Complete	
3 Current turn	3	printStatusBar()	Print current turn and winner	Complete	
4 Place move	4	drawboard()	Print S and O on board	Complete	

5 Game over	5	printStatusBar()	Print current turn and winner	Complete	
6 Start again	6	paintComponent()	Restart the game	Complete	
7 computer player	7	setComputer()	Computer player option	complete	

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

Summarize how each user story/acceptance criterion 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	AC ID	Class Name (s) of the Test Code	Method Name(s) of the Test Code	Description of the Test Case (input & expected output)
1 empty board	1	TestEmptyBoard	Test the board empty or not	
2 game over	2	TestCompleteGames	Test if game complete or not	
3 place move	3	TestCrossMove	Test move legal or not	

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)?