

GitHub Link: <https://github.com/BradyMaes1/SOSGame.git>

User Stories:

ID	User Story Name	User Story Description	Priority	Estimated Effort (hours)
1	Choose a board size	As a player, I want to choose the board size so that I can customize the difficulty and length of the game.	Medium	2
2	Choose the game mode of a chosen board	As a player, I want to choose the game mode of a chosen board so that I can either play until the first SOS is formed or play until the board is full and the player with the most SOSs wins.	Medium	2
3	Start a new game of the chosen board size and game mode	As a player, I want to start a new game and choose the board size as well as the game mode so that I can play again.	High	1
4	Make a move in a simple game	As a player, I want to mark an S or an O on the board, so that my move can be documented and update the current state of the game.	High	2
5	A simple game is over	As a player, I want the game to end if either myself or the opponent forms SOS so that the game can end and report the winner.	High	3
6	Make a move in a general game	As a player, I want to mark an S or an O on the board, so that my move can be documented and update the current state of the game.	High	4
7	A general game is over	As a player, I want the game to end when every space on the board has been	High	5

		filled so that the game can end and report the winner (whoever has the most SOSs). Note: If I have enough time, I wish to incorporate logic that ends the game when no new SOSs are possible, not just the board being filled completely.		
--	--	--	--	--

I used ChatGPT to generate my first two user stories. Here are the prompts and answers from ChatGPT:

First User Story:

I'm making the SOS board game in Java. I need to set up my user stories with the following template:
 As a <role>, I want <goal> [so that <benefit>]

Make a user story for: Choose a board size



Memory updated

Here's a user story for choosing a board size:

As a player, **I want** to choose the board size **so that** I can customize the difficulty and length of the game.

Second User Story:



Great, now make a user story for: Choose the game mode of a chosen board



Here's a user story for choosing the game mode:

As a player, I want to choose the game mode of a chosen board **so that** I can play either a single-player or multiplayer version of the game.

Sorry, I should have specified that by game mode, I am referring to two different game modes:

- 1) The two players play until one player creates SOS, and the game ends with that player as the winner.
- 2) The two plays play until no more board space remains, and the player with the most SOSs formed wins.



Memory updated

Thank you for the clarification! Here's the updated user story for choosing the game mode:

As a player, I want to choose the game mode of a chosen board **so that** I can either play until the first SOS is formed or play until the board is full and the player with the most SOSs wins.

🔊 📄 👍 🗨️ ↺

It is worth noting that in the second user story, I had to correct ChatGPT because it made a wrong assumption about what I meant by "game mode." It assumed that I meant single-player versus multiplayer when I meant a simple versus general game.

Acceptance Criteria:

User Story ID and Name	AC ID	Description of Acceptance Criterion	Status (completed, toDo, inProgress)
1. Choose a board size	1.1	AC 1.1 Successful creation of a valid board size Given a valid integer (N) When I enter this integer as prompted AND the integer is above the minimum AND the integer is below the maximum Then the program should create a new game with NxN dimensions for the board size	toDO
	1.2	AC 1.2 Unsuccessful creation of an invalid board size Given a valid integer (N) less than the minimum size When I enter this integer as prompted AND the integer is below the minimum Then the program should not create a new game with NxN dimensions for the board size AND it should display an error indicating the chosen integer was too small AND it should prompt them again	
	1.3	AC 1.3 Unsuccessful creation of an invalid board size Given a valid integer (N) greater than the maximum size When I enter this integer as prompted AND the integer is above the maximum Then the program should not create a new game with NxN dimensions for the board size AND it should display an error indicating the chosen integer was too large AND it should prompt them again	
	1.4	AC 1.4 Unsuccessful creation of an invalid board size Given a non-integer value	

		<p>When I enter a non-integer when prompted Then the program should not create a new game with this invalid input AND it should display an error indicating that the user must choose a valid integer for their board size AND it should prompt them again</p>	
2. Choose the the game mode of a chosen board	2.1	<p>AC 2.1 Successful creation of a simple game Given the selection of "Simple" When I am prompted to click a button indicating the board game type Then the program should create a "Simple" type game</p>	toDO
	2.2	<p>AC 2.2 Successful creation of a general game Given the selection of "General" When I am prompted to click a button indicating the board game type Then the program should create a "General" type game</p>	
	2.3	<p>AC 2.1 Unsuccessful creation of a game Given no input is received or the user forces unexpected input When I am prompted to click a button indicating the board game type Then the program should not start a game of any type AND it should display an error indicating that no valid board game type was received AND it should prompt them again</p> <p>Note: The GUI will be set up so that the user has two buttons to click between (rather than a text input), so the area of vulnerability is low in this case. However, a user may be able to force an invalid input with a buffer overflow (although highly unlikely in Java).</p>	
3. Start a new game of the chosen board size and game mode	3.1	<p>AC 3.1 Successful start of a new game of the chosen board size and game mode Given a valid board size and game mode was selected When I try to start a new game AND I already selected a valid board size</p>	toDO

		<p>AND I already selected a valid game mode</p> <p>Then the program should start a new game of the selected game mode and board size</p>	
	3.2	<p>AC 3.2 Unsuccessful start of a new game of the chosen board size</p> <p>Given an invalid board size was selected</p> <p>When I try to start a new game</p> <p>AND I already selected a valid game mode</p> <p>Then the program should refer to the error handling for User Story 1 (AC IDs 1.2 & 1.3)</p>	
	3.3	<p>AC 3.3 Unsuccessful start of a new game of the chosen game mode</p> <p>Given an invalid game mode was selected</p> <p>When I try to start a new game</p> <p>AND I already selected a valid board size</p> <p>Then the program should refer to the error handling for User Story 2 (AC ID 2.3)</p>	
4. Make a move in a simple game	4.1	<p>AC 4.1 Successful selection of a move in an empty space in a simple game</p> <p>Given a user has selected a valid move</p> <p>When it is my turn to play</p> <p>AND I have selected a simple game</p> <p>Then the board is updated to reflect my input as well as store it in memory</p>	toDO
	4.2	<p>AC 4.2 Unsuccessful selection of a move in an occupied space in a simple game</p> <p>Given a user has selected a move in a space that has already been occupied</p> <p>When it is my turn to play</p> <p>AND I have selected a simple game</p> <p>Then the board is not updated to reflect my input</p> <p>AND the program displays an error indicating that I must choose an empty space</p> <p>AND the program allows me to repeat my turn</p>	
5. A simple game is over	5.1	<p>AC 5.1 Successful ending of the game in valid end conditions</p> <p>Given a user has selected a move resulting in the complete formation of "SOS"</p> <p>When it is my turn to play</p> <p>AND I have selected a simple game</p>	toDO

		<p>Then the board is updated to draw a dash through the SOS in the winner's color AND the winner is displayed on screen AND the game comes to an end AND the board and scores are cleared AND the user is prompted to play a new game</p>	
	5.2	<p>AC 5.2 Unsuccessful ending of the game in invalid end conditions Given a user has selected a move that does not result in the complete formation of "SOS" When it is my turn to play Then the board is updated but the game runs another check to make sure it can't continue running AND if the game cannot continue to run, a draw is displayed to the users AND a note of the erroneous ending is made and reported</p>	
6. Make a move in a general game	6.1	<p>AC 6.1 Successful selection of a move in an empty space in a general game forming one SOS Given a user has selected a valid space that forms one new SOS When it is my turn to play AND I have selected a general game Then the board is updated to reflect my input as well as store it in memory AND my score is increased by one AND a dash in my color is striked through the SOS that I just formed AND the program allows me to play another turn</p>	toDO
	6.2	<p>AC 6.1 Successful selection of a move in an empty space in a general game forming multiple SOSs Given a user has selected a valid space that forms more than one new SOS When it is my turn to play AND I have selected a general game Then the board is updated to reflect my input as well as store it in memory AND my score is increased by however many</p>	

		<p>SOSs were created AND a dash in my color is striked through the SOSs that I just formed AND the program allows me to play another turn</p>	
	6.3	<p>AC 6.3 Successful selection of a move in an empty space in a general game not forming an SOS Given a user has selected a valid space that forms one new SOS When it is my turn to play AND I have selected a general game Then the board is updated to reflect my input as well as store it in memory</p>	
	6.4	<p>AC 6.4 Unsuccessful selection of a move in an occupied space in a general game Given a user has selected an invalid space that is already occupied by a player When it is my turn to play AND I have selected a general game Then the board is not updated to reflect my input AND the program displays an error indicating that I must choose an empty space AND the program allows me to repeat my turn</p>	
7. A general game is over	7.1	<p>AC 7.1 Successful ending of the game in valid end conditions resulting in a draw Given a user has selected a move When my move results in the board becoming filled OR my move results in no more possible SOSs being formed Then the board is updated to reflect the last move AND "Draw" is displayed on screen AND the game comes to an end AND the board and scores are cleared AND the user is prompted to play a new game</p>	toDO

	7.2	<p>AC 7.2 Successful ending of the game in valid end conditions resulting in a winner</p> <p>Given a user has selected a move</p> <p>When my move results in the board becoming filled</p> <p>OR my move results in no more possible SOSs being formed</p> <p>Then the board is updated to reflect the last move</p> <p>AND the winner is displayed on screen</p> <p>AND the game comes to an end</p> <p>AND the board and scores are cleared</p> <p>AND the user is prompted to play a new game</p>	
	7.3	<p>AC 7.3 Unsuccessful ending of the game in invalid end conditions</p> <p>Given a user has selected a move</p> <p>When it is my turn to play</p> <p>Then the board is updated but the game runs another check to make sure it can't continue running</p> <p>AND if the game cannot continue to run, a draw is displayed to the users</p> <p>AND a note of the erroneous ending is made and reported</p>	

Data Flow Diagram (DFD)- Global SOS Game:

My data flow diagram can be seen here, but it is also available on my GitHub:

