Sprint 2

Jonathan Benson

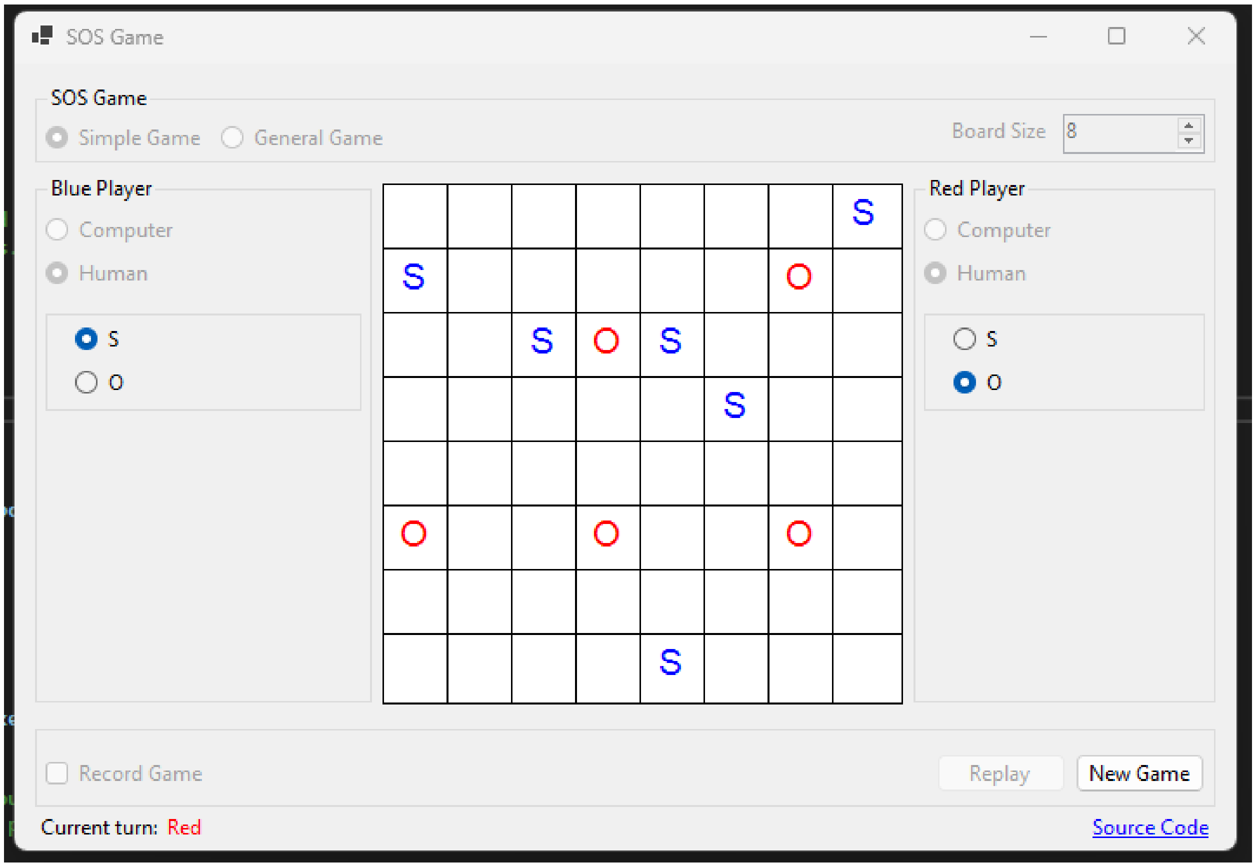
CS449

23 October 2022

GitHub Link for Source Code (sprint_2 folder): https://github.com/jonathanbenson/sosgame

1. Demonstration Video Link: https://youtu.be/KyzlwQHWO70

Screenshot of SOS Game at the End of Sprint 2



Updated User Stories and Acceptance Criteria

Story ID	Story Name	AC ID	AC Name	Description of Acceptance Criterion	Status
1	Choose a board size	1.1	User attempts to select an invalid board size	Given a SOS application, when the user attempts to select an invalid board size, then the board size will stay the same	completed
		1.2	User attempts to select a board size within a game	Given a SOS application that is in a game, when the user attempts to select a board size, then the board size will stay the same	completed
		1.3	User does not select a board size	Given a SOS application, when the does not select a board size, then the board size will be set to 8	completed
		1.4	User selects a valid board size outside of a game	Given a SOS application that is not in a game, when the user selects a valid board size, then the SOS board will be rendered with the correct size and the new board size will be visible in the board size selection	completed
2	Choose the game mode	2.1	User attempts to select an invalid game mode	Given a SOS application that is not in a game, when the user attempts to select an invalid game mode, then the game mode will stay the same	completed
		2.2	User attempts to select a game mode inside of a game	Given a SOS application that is in a game, when the user attempts to select a game mode, then the game mode will stay the same	completed
		2.3	User does not select a game mode	Given a SOS application, when the user does not select a game mode, then the game mode will be set to simple game	completed
		2.4	User selects a valid game mode outside of a game	Given a SOS application that is not in a game, when the user selects a valid game mode, then the new selected game mode will be checked in the game mode selection	completed
3	Start a new game	3.1	User starts a new game during a game	Given a SOS application that is in a game, when the user starts a new game, then a new game mode will start with the correct game mode	completed
		3.2	User starts a new game outside of a game	Given a SOS application that is outside of a game, when the user starts a new game, then a new game will start	completed
4	Make a move	4.1	User makes a move on a nonempty cell	Given an SOS application during a game, when a user makes a move on a nonempty cell, then the cell will remain unchanged and the player's turn will not change	completed

		4.2	User makes a move outside the game board	Given an SOS application during a game, when a user attempts to make a move outside of the game board, then all cells will remain unchanged and the user's turn will not change	completed
		4.3	User makes a move on an empty cell	Given SOS application during a game, when the user makes a move on an empty cell, then fill the cell with the corresponding S or O and mark an SOS on the board with the color of the current player if it completes a SOS	completed
		4.4	User makes a move not during a game	Given SOS application outside of a game, when the user makes a move, then the game state will not change	
		4.5	The user makes a move when it's not their turn	Given SOS application during a game, when the user makes a move and it's not their turn, their move will not count	
5	A simple game is over	5.1	User makes move that wins a simple game	Given a SOS application inside a simple game, when the user makes a move that completes an SOS, then tell the user that they won the game and end the game	toDo
		5.2	User makes move that does not win a simple game	Given a SOS application inside a simple game, when the user makes a move that does not complete an SOS, then switch turns between players and do not end the game	toDo
		5.3	Simple game ends in a draw	Given a SOS application inside a simple, game, when the user makes a move that fills the last nonempty cell and no SOSs have been completed yet, then end the game in a draw	toDo
6	A general game is over	6.1	User makes move that wins a general game	Given a SOS application inside a general game, when the user makes a move on the last empty square and they have more completed SOSs than their opponent, then tell the user which player won the game and end the game	toDo
		6.2	User makes move that does not win a general game, but completes a SOS	Given a SOS application during a general game, when the user makes a move on an empty square that is not the last empty square and their move completes an SOS, then do not switch turns between players and do not end the game	toDo
		6.3	User makes move that does not win a general game and does not complete a SOS	Given a SOS application during a general game, when the user makes a move on an empty square that is not the last empty square and their move does not complete a SOS, then switch turns between players and do not end the game	toDo

		6.4	User makes move that ends general game in a draw	Given a SOS application during a general game, when the user makes a move that fills up all the squares and the number of completed SOSs by each player are the same, then tell the user that there is a draw and end the game	toDo
7	Game replay	7.1	User attempts to replay previous game while playing a game	Given a SOS application during a game, when the user replays the previous game, then the previous game will not be replayed	toDo
		7.2	User replays game outside of a game with no previous game played	Given a SOS application outside of a game with no previous game player, when the user attempts to replay the non-existent previous game, then the non-existent previous game does not replay	toDo
		7.3	User replays game outside of a game with previous game played	Given a SOS application outside of a game with a previous game played, when the user replays the previous game, then the previous game will replay from start to finish on the game board with constant time steps for each turn	toDo
8	Move selection	8.1	User attempts to select another move option besides S or O	Given an SOS application, when the user attempts to select another move option besides an S or O, then the user's move option will stay the same	completed
		8.2	User does not select a move option	Given an SOS application, when the user does not select a move option, then the user's move option will be set to "S"	completed
		8.3	User selects a move option with the values S or O	Given an SOS application, when the user selects S or O, then the user's move option will update to their desired selection	completed
		8.4	User attempts to select a move option outside of a game	Given an SOS application not during a game, when the user tries to select S or O, their move type will remain unchanged	completed

2. Summary of Source Code

Source Code file Name	Production code or test code?	Notes	# Lines of Code
BoardPainter.cs	Production Code	Drawing board items (grid lines, S's & O's, SOS's)	215
Form1.cs	Production Code	GUI operations and control event handlers	371
AccessibilityManager.cs	Production Code	Manage accessibility and visilibility of GUI controls	162
ComputerPlayer.cs	Production Code	Computer player logic	28
Game.cs	Production Code	SOS game base class	474
GeneralGame.cs	Production Code	SOS game logic associated with a general game	32
HumanPlayer.cs	Production Code	Human player logic	32
Move.cs	Production Code	Logic associated with a move	87
Player.cs	Production Code	SOS player base class	70
SimpleGame.cs	Production Code	SOS game logic associated with a simple game	26
SOSEngine.cs	Production Code	High-level SOS application logic (starting games, etc.)	147
SOSLine.cs	Production Code	Simple class model for a completed SOS line	48
GameTest.cs	Test Code	Testing for the Game class	162
AccessibilityManagerTest.cs	Test Code	Testing for the AccessibilityManagerClass	120
PlayerTest.cs	Test Code	Testing for the Player class	47
SOSEngineTest.cs	Test Code	Testing for the SOSEngine class	88
ComputerPlayerTest.cs	Test Code	Testing for the ComputerPlayer class	30
HumanPlayerTest.cs	Test Code	Testing for the HumanPlayer class	32
MoveTest.cs	Test Code	Testing for the Move class	41
		Total	**2212**

3. Production Code vs User stories/Acceptance Criteria

Story ID	Story Name	AC ID	Classes	Method Name(s)	Status	Notes (optional)
1	Choose a board size	1.1	Game	Game.Game	complete	The Game.Game constructor will throw an ArgumentOutOfRangeException if the board size is not between the values of 8 and 12
		1.2	AccessibilityManager	AccessibilityManager.IsBoardAccessibile, AccessibilityManager.IsAccessible	complete	The AccessibilityManager.IsAccessible determine the accesibility of all controls at any given time. The AccessibilityManager.IsBoardAccessibile method determines if the board is accessibile during a game (which it is not).
		1.3	Game	Game.Game, SimpleGame.SimpleGame, GeneralGame.GeneralGame	complete	The Game.Game constructor will set the board size to 8 if the user does not choose a board size. The SimpleGame and GeneralGame constructors also follow suit.
		1.4	Game	Game.Game	complete	The Game.Game constructor will set the board size to the user's specified number upon game initialization
2	Choose the game mode	2.1	SOSEngine	SOSEngine.StartGame	complete	The SOSEngine.StartGame method will throw an exception if the game mode is not either simple or general
		2.2	AccessibilityManager	AccessibilityManager.IsGameModeAccessible, AccessibilityManager.IsAccessible	complete	The AccessibilityManager.IsGameModeAccessible determines that the user is not able to select a new game mode inside a game
		2.3	SOSEngine	SOSEngine.StartGame	complete	The SOSEngine.StartGame method will set the game mode to simple if no game mode was specified
		2.4	SOSEngine	SOSEngine.StartGame	complete	The SOSEngine accepts the choice of either simple or general game mode when starting a new game
3	Start a new game	3.1	SOSEngine	SOSEngine.StartGame	complete	The SOSEngine will start a new game with the same settings as the current game (game mode, player types, board size)
		3.2	SOSEngine	SOSEngine.StartGame	complete	The SOSEngine will start a new game with the desired settings

4	Make a move	4.1	Player, Game, Move	Player.MakeMove, Game.MakeMove, Game.IsMoveValid, Move.DoesConflict	complete	Player.MakeMove is called which calls Game.MakeMove, and then Game.IsMoveValid checks if the move is valid with the help of Move.DoesConflict. An exception is thrown inside Game.MakeMove to let the GUI know of invalid move.
		4.2	Player, Game, Move	Player.MakeMove, Game.MakeMove, Game.IsMoveValid, Move.DoesConflict	complete	Refer to 4.1 notes. Game.IsMoveValid checks if the move is outside of the board
		4.3	Player, Game, Move	Player.MakeMove, Game.MakeMove, Game.IsMoveValid, Move.DoesConflict	complete	Refer to 4.1 notes. Game.IsMoveValid checks if the move conflicts with another move
		4.4	AccessibilityManager	AccessibilityManager.IsBoardAccessible, AccessibilityManager.IsAccessible	complete	The accesibility manager determines that the user is not able to make a move on the game board outside of a game
		4.5	Player, Game, Move	Player.MakeMove, Game.MakeMove, Game.IsMoveValid, Move.DoesConflict	complete	Refer to 4.1 notes. Game.IsMoveValid checks if the move's turn is the same with the current turn of the game
8	Move selection	8.1	Player	Player.SetMoveType	complete	Player.SetMoveType will make sure the only moves a user can use are MoveType.S and MoveType.O
		8.2	Player	Player.Player	complete	Player.Player constructor will set the default move type to S
		8.3	Player	Player.SetMoveType	complete	Player.SetMoveType will set the new move type to either MoveType.S or MoveType.O
		8.4	AccessibilityManager	AccessibilityManager.IsBlueSOAccessible, AccessibilityMana	complete	The accessibility manager will determine that the SO options for both the red and blue players are inacessible outside of a game

	ger.IsRedSOAccessible	

4. Tests vs User stories/Acceptance Criteria

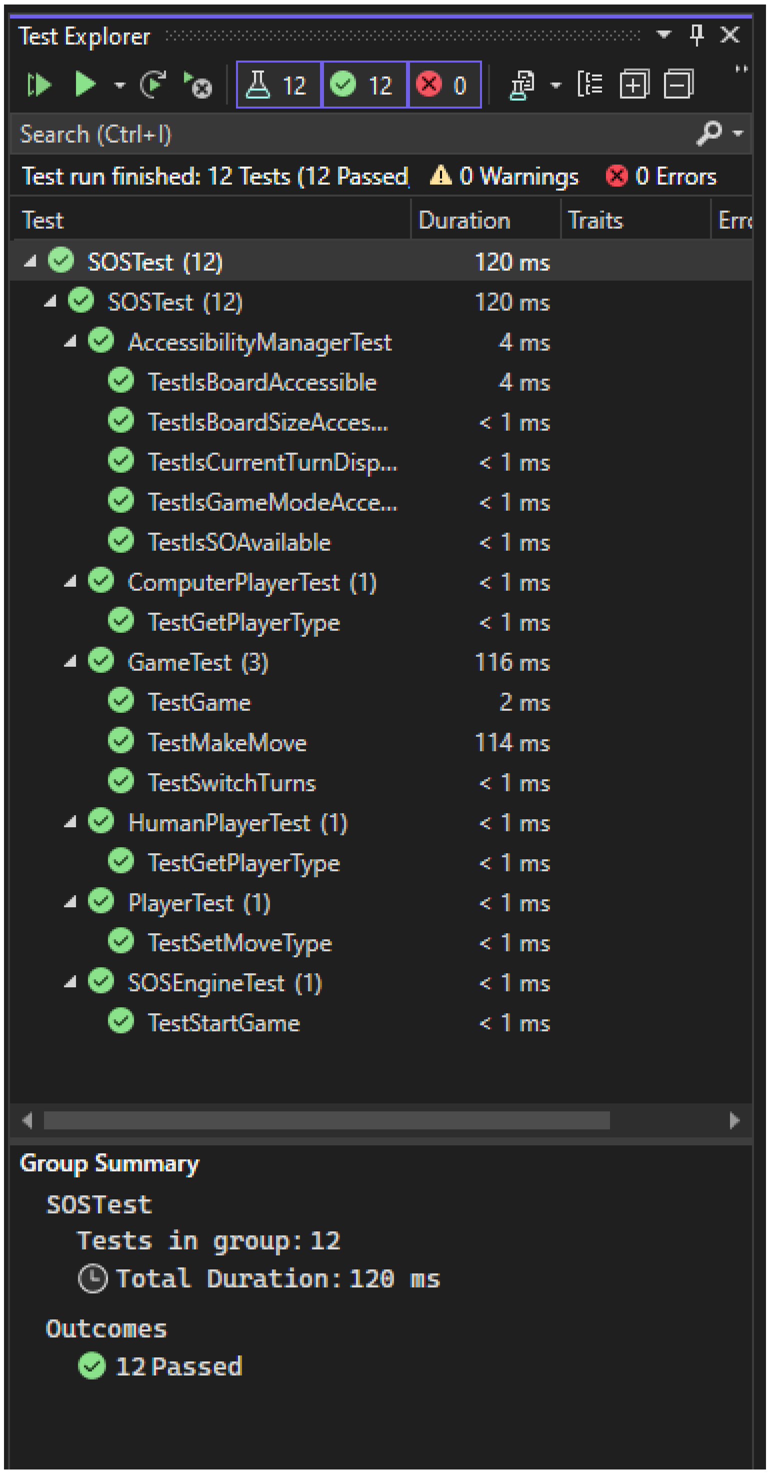
4.1) Automated tests directly corresponding to the acceptance criteria of the above user stories

User Story ID	User Story Name	AC ID	Classes	Methods	Description
1	Choose a board size	1.1	GameTest	GameTest.TestGame	Input: ints from -100 to board size lower limit (6) and board size upper limit (12) + 1 to 100. Expected output: exceptions thrown on all test inputs
		1.2	Accessibility ManagerTest	AccessibilityManagerTest.TestIsBoardSizeAccessible	Input: a SOSEngine object with a game that is underway Output: the accessibility manager should recognize that the game is underway and deem the board size inaccessible
		1.3	GameTest	GameTest.TestGame	Input: none (Game.Game called wihout supplying the board size) Expected output: the board size defaults to 8
		1.4	GameTest	GameTest.TestGame	Input: ints from board size lower (6) limit to board size upper limit (12) Expected output: board size is correctly set to the supplied value
			Accessibility ManagerTest	AccessibilityManagerTest.TestIsBoardSizeAccessible	Input: an SOSEngine with no game underway Output: the acecsibility manager recognizes that not game has started yet and deems the board size selection accessible
2	Choose the game mode	2.1	SOSEngineTest	SOSEngineTest.TestStartGame	Input: An invalid game mode passed to SOSEngine.StartGame method Output: an exception is thrown
		2.2	Accessibility ManagerTest	AccessibilityManagerTest.TestIsGameModeAccessible	Input: a SOSEngine with a game that is underway Output: the accessibility manager should recognize that the game is underway and deem the game mode selection inaccessible

		2.3	SOSEngineTest	SOSEngineTest.TestStartGame	Input: No game mode supplied to SOSEngine.StartGame Output: a new game will start with the simple game mode
		2.4	SOSEngineTest	SOSEngineTest.TestStartGame	Input: simple and general game modes passed to SOSEngine.StartGame method Output: new games will start with their corresponding valid game modes
			AccessibilityManagerTest	AccessibilityManagerTest.TestIsGameModeAccessible	Input: an SOSEngine with no game underway Output: the accessilbility manager recognizes that no game has started yet and deems the game mode selection accessible
3	Start a new game	3.1	SOSEngineTest	SOSEngineTest.TestStartGame	Input: an SOSEngine with a game underway Output: an SOSEngine with a new game underway that has the same settings as the game before (board size, player types, game mode)
		3.2	SOSEngineTest	SOSEngineTest.TestStartGame	Input: a SOSEngine with no game underway Output: a SOSEngine with a game underway that has the desired settings passed to it (board size, player types, game mode)
4	Make a move	4.1	GameTest	GameTest.TestMakeMove	Input: a Game with a Player that makes a move on a nonempty cell Output: an exception is thrown when Player.MakeMove is called
		4.2	GameTest	GameTest.TestMakeMove	Input: a Game with a Player that makes a move outside the game board Output: an exception is thrown when Player.MakeMove is called
		4.3	GameTest	GameTest.TestMakeMove	Input: a Game with a Player that makes a move on an empty cell Output: a new Move is added to the list of moves in the Game

		4.4	AccessibilityManagerTest	AccessibilityManager.TestIsBoardAccessible	Input: a SOSEngine with no game underway Output: the accessibility manager determines that the board is inaccessible
		4.5	GameTest	GameTest.TestMakeMove	Input: a Game with a Player that makes a move when it is not their turn Output: an exception is thrown and no move added to the board
8	Move selection	8.1	PlayerTest	PlayerTest.TestSetMoveType	Input: a Player that sets their move type to a move other than MoveType.S or MoveType.O Output: An exception is thrown and the Player's MoveType does not change
		8.2	PlayerTest	PlayerTest.TestSetMoveType	Input: A Player that has not set its move type yet Output: the Player's move type will be set to MoveType.S as default
		8.3	PlayerTest	PlayerTest.TestSetMoveType	
		8.4	AccessibilityManagerTest	AccessibilityManagerTest.TestIsBlueSOAccessible, AccessibilityManagerTest.TestIsRedSOAccessible	

Screenshot of Automated Test Run



```
Test Explorer
12  12  0
Search (Ctrl+I)
Test run finished: 12 Tests (12 Passed   0 Warnings   0 Errors

Test                              Duration   Traits   Erro
SOSTest (12)                        120 ms
  SOSTest (12)                      120 ms
    AccessibilityManagerTest          4 ms
      TestIsBoardAccessible           4 ms
      TestIsBoardSizeAcces...       < 1 ms
      TestIsCurrentTurnDisp...      < 1 ms
      TestIsGameModeAcce...         < 1 ms
      TestIsSOAvailable             < 1 ms
    ComputerPlayerTest (1)          < 1 ms
      TestGetPlayerType             < 1 ms
    GameTest (3)                    116 ms
      TestGame                        2 ms
      TestMakeMove                  114 ms
      TestSwitchTurns               < 1 ms
    HumanPlayerTest (1)             < 1 ms
      TestGetPlayerType             < 1 ms
    PlayerTest (1)                  < 1 ms
      TestSetMoveType               < 1 ms
    SOSEngineTest (1)               < 1 ms
      TestStartGame                 < 1 ms

Group Summary
  SOSTest
    Tests in group: 12
    Total Duration: 120 ms

  Outcomes
    12 Passed
```

4.2) Manual tests directly corresponding to the acceptance criteria of the above user stories

User Story ID	User Story Name	AC ID	Test Case Input	Test Oracle
1	Choose a board size	1.1	Attempt to increase the value of the board size numeric updwon control below its lower limit (6) and over its upper limit (12)	The value of the board size does not decrease lower than the lower limit or increase above the upper limit
		1.2	Attempt to change the board size after the game has started	The value of the board size does not change
		1.3	Start a new game, but do not select a board size	A new game will start with the correct board size of 8
		1.4	Start a new game, but select board sizes within the lower limit (6) and upper limit (12)	The new games will start with the chosen board sizes
2	Choose the game mode	2.2	Start a new game, and try to select the game mode different from the current game mode	The game mode should remain unchanged
		2.3	Start a new game without selecting a game mode	The game mode should default to simple game in the new game
		2.4	Start two new games: one with simple game and the other with general game mode	The two games should start with the correct coresponding game modes
3	Start a new game	3.1	Start a game, and then start a new game inside of the game	The new game should start with the same settings as the first game (board size, game mode, and player types)
		3.2	Start a new game by selecting the game mode, board size, and player types	The new game should start with the desired game mode, board size, and player types
4	Make a move	4.1	Make a move on a nonempty cell in a simple or general game	A message box should show saying an invalid move has been made and no changed to the game state should happen
		4.2	Make a move outside the game board	No changes should be made to the game board

		4.3	Make a move on an empty cell	The cell should be filled with an S or O depending on the current move type, and color depending on the player (blue or red)
		4.4	Make a move on the board before a game has started	A message box will show telling the user to start a new game and no changes will be made to the board
8	Move selection	8.2	Start a new game, and then make a move on an empty cell	The cell should be filled with the default move type, S
		8.3	Start a new game, then select a move option, and then make a move on the board	The corresponding cell should be filled with the S or O move option that the user picked
		8.4	Before a game has started, select a new move option	No changes should be made to the move option

4.3) Other automated or manual tests not corresponding to the acceptance criteria of the above user stories

Number	Production Code Class.Method	Test Code Class.Method	Test Input	Expected Result
1	AccessibilityManager.IsNewGameButtonAccessible	AccessibilityManagerTest.TestIsNewGameButtonAccessible	A game that hasn't started yet	FALSE
2	^	^	A game that has started	TRUE
3	AccessibilityManager.IsCurrentTurnDisplayAccessible	AccessibilityManagerTest.TestIsCurrentTurnDisplayAccessible	A game that has not started	FALSE
4	^	^	A game that has started	TRUE
5	ComputerPlayer.GetPlayerType	ComputerPlayerTest.TestGetPlayerType	The player type of a new ComputerPlayer object	MoveType.Computer
6	HumanPlayer.GetPlayerType	HumanPlayerTest.TestGetPlayerType	The player type of a new HumanPlayer object	MoveType.Human
7	Game.SwitchTurns	GameTest.TestSwitchTurns	The player's turn when the game just started, so the blue player	After switch turns, it should be the red player
8	Move.DoesConflict	MoveTest.TestDoesConflict	A move that conflicts with another move	TRUE