

Sprint 3 - ChatGPT



User Story ID

AC ID

Class Name(s) of the Test Code

Method Name(s) of the Test Code

Description of the Test Case (input & expected output)

make a chart listed above.

For user story, 5) A simple game is over
create two test,
one for blue player winning the game,
and the other for getting a draw



Here's a chart based on your request for user story 5, which involves testing the "game over" conditions in your SOS game. The chart includes two test cases: one for the blue player winning and another for a draw.

User Story ID	AC ID	Class Name(s) of the Test Code	Method Name(s) of the Test Code	Description of the Test Case (Input & Expected Output)
5	5.1	`GameLogicTest`	`testBluePlayerWins()`	Input: Game state where the blue player has formed a winning SOS sequence. Expected Output: The method returns `true`, indicating the game is over with the blue player as the winner.

User Story ID	AC ID	Class Name(s) of the Test Code	Method Name(s) of the Test Code	Description of the Test Case (Input & Expected Output)
5	5.2	`GameLogicTest`	`testDrawGame()`	Input: Game state where all spaces are filled without a winner. Expected Output: The method returns `true`, indicating the game is over with a draw.

Feel free to modify any details or let me know if you need further assistance!



I like what you put,
but rename 5.1 handleGameModeLogic_testBlueWin
and rename 5.2 handleGameModeLogic_testPlayersDraw
because I now made handleGameModeLogic in my product code



Here's the updated chart with your specified method names for the test cases:

User Story ID	AC ID	Class Name(s) of the Test Code	Method Name(s) of the Test Code	Description of the Test Case (Input & Expected Output)
5	5.1	`GameLogicTest`	`handleGameModeLogic_testBlueWin()`	Input: Game state where the blue player has formed a winning SOS sequence. Expected Output: The method returns `true`, indicating the game is over with the blue player as the winner.
5	5.2	`GameLogicTest`	`handleGameModeLogic_testPlayersDraw()`	Input: Game state where all spaces are filled without a winner. Expected Output: The method returns `true`, indicating the game is over with a draw.

Let me know if you need any more adjustments or further assistance!



alright, for both ac id input, call `handleGameModeLogic(boolean didPlayerScore)`.

for 5.1, set `didPlayerScore` to `true`
for 5.2, set `didPlayerScore` to `false`



Here's the updated chart reflecting your requirements for the `AC ID` input in both test cases, where `handleGameModeLogic(boolean didPlayerScore)` is called with the appropriate boolean values:

User Story ID	AC ID	Class Name(s) of the Test Code	Method Name(s) of the Test Code	Description of the Test Case (Input & Expected Output)
5	5.1	<code>`GameLogicTest`</code>	<code>`handleGameModeLogic_testBlueWin()`</code>	Input: Call <code>`handleGameModeLogic(true)`</code> to indicate the blue player scored. Expected Output: The method returns <code>`true`</code> , indicating the game is over with the blue player as the winner.
5	5.2	<code>`GameLogicTest`</code>	<code>`handleGameModeLogic_testPlayersDraw()`</code>	Input: Call <code>`handleGameModeLogic(false)`</code> to indicate there is no winner. Expected Output: The method returns <code>`true`</code> , indicating the game is over with a draw.

Let me know if you need any further modifications or assistance!



This is really good, but stay put for now, I might change later



Sure, I'll be here whenever you're ready to make changes or if you have any more questions! Just let me know.



for 5.1 output, make it "Game Over! \nBlue Player wins!"
for 5.2 output, make it "Game Over! \nIt's a draw"



Here's the updated chart with the specified output messages for both test cases:

User Story ID	AC ID	Class Name(s) of the Test Code	Method Name(s) of the Test Code	Description of the Test Case (Input & Expected Output)
5	5.1	`GameLogicTest`	`handleGameModeLogic_testBlueWin()`	Input: Call `handleGameModeLogic(true)` to indicate the blue player scored. Expected Output: ` "Game Over! \nBlue Player wins!"`
5	5.2	`GameLogicTest`	`handleGameModeLogic_testPlayersDraw()`	Input: Call `handleGameModeLogic(false)` to indicate there is no winner. Expected Output: ` "Game Over! \nIt's a draw"`

Let me know if there's anything else you'd like to adjust or add!

ChatGPT can make mistakes. Check important info.