

SennottSquareSUPERCoders Sprint 3- Unit Tests

Candy Cards

Both Unit Tests coded into SSQWorldOfSweetsTest.java file.

Swap Players Card

Identifier: SwapTest01

Description: This test confirms that the swap player card works as advertised

Preconditions: World of Sweets has been launched and player info entered

Execution Steps: Click the draw card button until a swap card button is drawn and the symbols of the players begin to change

Post Conditions: the positions of the players should have shuffled in some way

Other Unit Test coded into SSQWorldOfSweetsTest.java file.

Timer

Identifier: TimerTest01

Description: This test confirms that the timer appears in the game window and is incrementing at the proper interval

Preconditions: World of Sweets has been launched and player info entered.

Execution Steps: Observe the game window

Postconditions: At the bottom right corner of the screen, a timer is displayed and the time is incremented by one second every second when checked against an actual clock.

Identifier: TimerTest02

Description: This test confirms that the timer is set to the proper time when loading a previously saved game.

Preconditions: World of Sweets has been launched, player info entered, and several turns have been taken and time has elapsed playing the game.

Execution Steps: Observe the current time displayed by the timer in the bottom right corner of the screen, click the "Save and Exit" button, click "OK" on the following pop-ups to confirm the save, load World of Sweets, choose to load a game, and select the load file to the game that was just saved, and observe the window.

Postconditions: The time that was observed previously when the game was saved and exited is the new time that the loaded game starts with.

Save

Identifier: SaveTest01

Description: This test confirms a file was created and contains data pertinent to the current game being played.

Preconditions: World of Sweets has been launched and player info entered

Execution Steps: Click the "Save Game" button, close the game, and open the folder that the game is run from and see if a file containing the player names and current date exists.

If so, open the file in notepad and check that that playerinfo is accurate.

Postconditions: The game will have saved the data from the last saved session.

Other Unit Test coded into SSQWorldOfSweetsTest.java file.

Load

Identifier: LoadPrompt

Description: This test confirms that the user is asked if they want to load a file at the start of the game.

Preconditions: World of Sweets has been launched.

Execution Steps: Observe popups.

Postconditions: There is a window asking the user if they want to load a save file.

Identifier: LoadState

Description: This test confirms that when a save file is loaded, the game state is correctly loaded.

Preconditions: There is a valid .wos save file in the project root directory.

Execution Steps: Launch WoS

- Choose “yes” when asked to load a file

- Select save file from dropdown list

- Click OK

Postconditions: The game has loaded and the timer, player info, player turn, and tokens are how they were when the game was saved.