

### **Basic Frame Build**

Identifier: FramePlayerInfo

Description: This test confirms that there is an area to the right of the screen that is reserved for player information

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

Execution Steps: Observe right side of screen

Postconditions: There should be a pink area labeled "Player Information"

Identifier: FrameDeckInfo

Description: This test confirms that there is an area to the bottom of the screen that is reserved for deck information

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

Execution Steps: Observe bottom of screen

Postconditions: There should be an area labeled "Deck Information"

Identifier: FrameGameplayArea

Description: This test confirms that there is an area to the left/middle of the screen that is reserved for the game board

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

Execution Steps: Observe left/middle of screen

Postconditions: There should be an area with the game board

### **Basic Deck Area**

Identifier: DeckPrev

Description: This test confirms that there is a space to show the previous card drawn

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

Execution Steps: Observe bottom of screen

Postconditions: A space for the previously drawn card should be shown in the area labeled "Deck Information"

Identifier: DeckFacedownAndButton

Description: This test confirms that there is a button to draw a card and a face down deck displayed

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

Execution Steps: Observe bottom of screen

Postconditions: An image of a face-down card and a button labeled "click to draw card" should be shown in the area labeled "Deck Information"

### **Basic Gameplay Area**

Identifier: Basic-Gameplay-Area-Test01

Description: This test verifies that when the board is shown after user input is collected, there is a space on the board that displays all of the colored squares (red, yellow, blue, green, and orange) that a user can land on during gameplay.

Preconditions: The program is run via “gradle run” on the command line, number of players is selected, and for each player, their name and preferred token is given to the program.

Execution Steps: After pressing the “OK” button on the last player token selection pop-up, the game window is observed.

Postconditions: In the top left section of the World of Sweets window, the gameplay area is displayed and 52 squares are displayed, 50 colored squares (red, yellow, blue, green, and orange) plus a start and an end (grandmother’s house) square.

Identifier: Basic-Gameplay-Area-Test02

Description: This test verifies that when the board is shown after user input is collected, there are token indicators displayed on the game board showing the relative positions of all the players.

Preconditions: The program is run via “gradle run” on the command line, number of players is selected, and for each player, their name and preferred token is given to the program.

Execution Steps: After pressing the “OK” button on the last player token selection pop-up, the game window is observed.

Postconditions: In the top left section of the World of Sweets window, the gameplay area is displayed and on the start space, the tokens of all of the players are shown in the north, south, east or west sections of the square.

### **Basic Player Information Area**

Identifier: PlayerInfoNames

Description: This test confirms that player names are shown in the player in the player information area

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

Execution Steps: Observe right side of screen

Postconditions: All players names should be shown in the area labeled “Player Information”

Identifier: PlayerInfoTokens

Description: This test confirms that player tokens are shown in the player in the player information area

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

Execution Steps: Observe right side of screen

Postconditions: All players tokens should be shown in the area labeled “Player Information”

**Card Chances**- done via programming tests in SSQWorldOfSweetsTest.java

**Single and Double Cards-** done via programming tests in SSQWorldOfSweetsTest.java

### **Tokens**

**Identifier:** Tokens-Test01

**Description:** This test verifies that the user is allowed to select a token that they will be able to identify with throughout the duration of the game.

**Preconditions:** The program is run via “gradle run” on the command line, number of players is selected, and Player 1 name is entered.

**Execution Steps:** After pressing the “OK” button on the Player 1 name pop-up, the subsequent pop-up windows are observed.

**Postconditions:** The next pop-up that appears lets Player 1 select a token from a list that they will identify with throughout the duration of the game, and after the “OK” button is pressed, subsequent player names and tokens are asked for from the user.

**Identifier:** Tokens-Test01

**Description:** This test verifies that players are not allowed to choose duplicate tokens.

**Preconditions:** The program is run via “gradle run” on the command line, number of players is selected, and Player 1 name is entered.

**Execution Steps:** After pressing the “OK” button on the Player 1 name pop-up, the subsequent pop-up windows are observed.

**Postconditions:** The next pop-up that appears lets Player 1 select a token from a list that they will identify with throughout the duration of the game, and after the specific token is selected and the “OK” button is pressed, player 2 (and player 3 or 4 if selected) are given a list of tokens to choose from without the token(s) that was selected by previous players.

### **Start of Game**

**Identifier:** Start-of-Game-Test01

**Description:** This test verifies that during the start of the game the user can select the number of players that will be playing the game.

**Preconditions:** The program is run via “gradle run” on the command line.

**Execution Steps:** The pop-up windows that appear after executing the program are observed.

**Postconditions:** The very first pop-up window that appears asks the user how many players are present to play the game, 2, 3, or 4.

**Identifier:** Start-of-Game-Test02

**Description:** This test verifies that during the start of the game, the each of the players can input their names and choose a token that will represent their progress during the game.

**Preconditions:** The program is run via “gradle run” on the command line, and the number of players is selected.

Execution Steps: After pressing the “OK” button on the player number pop-up, the subsequent pop-up windows are observed.

Postconditions: Pop-up windows appear asking for player names and player tokens for each of the players that are going to play the game.

### **Front of Card**

Identifier: Front-of-Card-Test01

Description: This test verifies that when the board is shown after user input is collected and throughout gameplay, users can click to draw cards and the cards that are drawn are displayed to the user signifying the space or number of spaces that they should advance in the game.

Preconditions: The program is run via “gradle run” on the command line, number of players is selected, and for each player, their name and preferred token is given to the program, the “OK” button is pressed signifying the first player's turn, and the “Click to draw card” button is pressed.

Execution Steps: After the “Click to draw card” button is pressed, the game window is observed.

Postconditions: In the bottom section of the World of Sweets window, on the right, there is a space where the last card drawn is displayed, and the user is able to press the “Click to draw card” button and is shown a new card which displays a single or double red, yellow, blue, green, or orange card representing the space that they will move to.

Identifier: Front-of-Card-Test02

Description: This test verifies that when the board is shown throughout gameplay, the user can differentiate between the pile of cards that is the “deck” that has the back of card showing, and the pile of cards that shows the last card drawn that has the front of card showing by being presented with differing card designs between the front and back.

Preconditions: The program is run via “gradle run” on the command line, number of players is selected, and for each player, their name and preferred token is given to the program, the “OK” button is pressed signifying the first player's turn, and the “Click to draw card” button is pressed.

Execution Steps: After the “Click to draw card” button is pressed, the game window is observed.

Postconditions: In the bottom section of the World of Sweets window, on the left, there is a space where the “deck” of cards is displayed with the back of card showing, and on the right, there is a space where the last card drawn is displayed, and the user is able to differentiate between these two types of cards because the back of card has a rainbow pattern with the logo “WORLD OF SWEETS” on it while the front of card has either a single rectangle or double rectangle with one color filling it over a white background.

Identifier: Front-of-Card-Test03

Description: This test verifies that both single and double cards appear in the game.

Preconditions: The program is run via “gradle run” on the command line, number of players is selected, and for each player, their name and preferred token is given to the program, the “OK” button is pressed signifying the first player's turn, and the “Click to draw card” button is pressed.

Execution Steps: After the “Click to draw card” button is pressed, the game window is observed.

Postconditions: In the bottom section of the World of Sweets window, on the right, there is a space where the last card drawn is displayed, and if the draw button is pressed many times, it can be seen that both single and double cards appear in the deck by observing the single or double rectangle colored design on the card.

### **Back of Card**

Identifier: Back-of-Card-Test01

Description: This test verifies that when the board is shown after user input is collected and throughout gameplay, there is a space in the window displaying a “deck” of upside down cards with the back of the card showing, simulating a physical deck being drawn from.

Preconditions: The program is run via “gradle run” on the command line, number of players is selected, and for each player, their name and preferred token is given to the program.

Execution Steps: After pressing the “OK” button on the last player token selection pop-up, the game window is observed.

Postconditions: In the bottom section of the World of Sweets window, on the left, there is a space where the “deck” of cards is displayed with the back of card showing, and this deck with the back of card picture persists throughout turns and deck draw clicks.

Identifier: Back-of-Card-Test02

Description: This test verifies that when the board is shown throughout gameplay, the user can differentiate between the pile of cards that is the “deck” that has the back of card showing, and the pile of cards that shows the last card drawn that has the front of card showing by being presented with differing card designs between the front and back.

Preconditions: The program is run via “gradle run” on the command line, number of players is selected, and for each player, their name and preferred token is given to the program, the “OK” button is pressed signifying the first player's turn, and the “Click to draw card” button is pressed.

Execution Steps: After the “Click to draw card” button is pressed, the game window is observed.

Postconditions: In the bottom section of the World of Sweets window, on the left, there is a space where the “deck” of cards is displayed with the back of card showing, and on the right, there is a space where the last card drawn is displayed, and the user is able to differentiate between these two types of cards because the back of card has a rainbow pattern with the logo “WORLD OF SWEETS” on it while the front of card has either a single rectangle or double rectangle with one color filling it over a white background.

### **Deck Click to Draw**

Identifier: DeckOverwrite

Description: This test confirms that when a card is drawn it will replace the previous card that was drawn

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

Execution Steps: Click draw card button

Take note of card shown

Click draw card button

Postconditions: There should be a different card image from what was previously noted

Identifier: DeckLastCard

Description: This test confirms that when a card is drawn it is showed in the correct location

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

Execution Steps: Click draw card button

Postconditions: There should be an image of a face-up card at the bottom of the screen

### **Turns/Game Flow**

Identifier: NextTurn

Description: This test confirms that the game informs the players whose turn it is next

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

Execution Steps: Click draw a card button

Postconditions: An alert should pop up telling whose turn it is now

Identifier: TurnCycle

Description: This test confirms that after the last player takes their turn, it goes back to the first player's turn

Preconditions: World of Sweets has been launched and player info entered for 4 players

Execution Steps: Click draw a card button 3 times

Click draw a card button again

Postconditions: An alert should pop up saying that it is the first player's turn