

Test Plan and Results

Overall Test Plan

Our test plan will be a run through of the game we created. This run through will be a simulated experience of a user interacting with the game. We will create test cases that should lead to expected results of the game such as termination or advancing to the next level. This entails a list of steps a user performs while playing the game. Also, we will make test cases to make sure a user's score is being accurately depicted and displayed accordingly on a leaderboard.

Test Case Descriptions

MT1.1 **Mechanics Test 1**

MT1.2 This test will ensure that once a leaf is selected, the corresponding point value of the tangerine revealed is added to the score

MT1.3 This will be done by clicking a leaf and the points shown will be added to the score

MT1.4 Inputs: User selecting a leaf

MT1.5 Outputs: The calculated number of points on the score

MT1.6 Normal

MT1.7 Whitebox

MT1.8 Functional

MT1.9 Integration

MT2.1 **Mechanics Test 2**

MT2.2 This test will ensure that once an exploding tangerine is selected, the game ends and the board resets

MT2.3 This will be done by intentionally losing the game by clicking an exploding tangerine

MT2.4 Inputs: User selects exploding tangerine

MT2.5 Output: Game ends and resets

MT2.6 Normal

MT2.7 Whitebox

MT2.8 Functional

MT2.9 Integration

MT3.1 **Mechanics Test 3**

MT3.2 This test will ensure that once a tangerine is selected that the correct number of exploding tangerines are placed in each grid

MT3.3 This will be done by intentionally losing the game by clicking an exploding tangerine or completing the level. Under both conditions, the board reveals all spaces and allows observation of the board.

MT3.4 Inputs: User selects enough tangerines to end the level

MT3.5 Outputs: The board reveals the quantity and placement of all tangerines.

MT3.6 Normal

MT3.7 Whitebox

MT3.8 Functional

MT3.9 Integration

MT4.1 Mechanics Test 4

- MT4.2 This test will ensure that once a level is completed, the board resets and the difficulty is increased while keeping the score
- MT4.3 This will be done by having the user go through a level and checking to see if the board resets to its original display. The overall score will also be checked.
- MT4.4 Inputs: User finishing a level
- MT4.5 Outputs: Board is reset, and overall score is kept
- MT4.6 Normal
- MT4.7 Whitebox
- MT4.8 Functional
- MT4.9 Integration

MT5.1 Mechanics Test 5

- MT5.2 This test will ensure that upon surpassing the previous high score, it updated to the users high score upon game completion
- MT5.3 This will be done by achieving a high score and making sure that the high score is updated.
- MT5.4 Inputs: Current score
- MT5.5 Outputs: Making the current score the high score
- MT5.6 Normal
- MT5.7 Whitebox
- MT5.8 Functional
- MT5.9 Integration

UI1.1 User Interface Test 1

- UI1.2 This test will ensure that the appearance of the site is correctly rendered
- UI1.3 This will be done by loading the client of the site and manually check that the visuals are correct.
- UI1.4 Inputs: The site's URL in the browser
- UI1.5 Outputs: The visual interface of the site.
- UI1.6 Normal
- UI1.7 Whitebox
- UI1.8 Functional
- UI1.9 Integration

UI2.1 User Interface Test 2

- UI2.2 This test will verify that the appearance of a flipped leaf and the leaf before it is flipped is correctly rendered
- UI2.3 This will be done by ensuring that leaves remain not flipped if left unclicked. If clicked, the leaf flips.
- UI2.4 Inputs: User's left clicks on a leaf
- UI2.5 Outputs: Leaf is flipped or is remaining not flipped
- UI2.6 Normal
- UI2.7 Whitebox
- UI2.8 Functional
- UI2.9 Integration

UI3.1 User Interface Test 3

UI3.2 This test will verify that upon flipping the leaf, the correct tangerine is displayed

UI3.3 This will be done manually by flipping a leaf and ensuring that upon being flipped, it displays the corresponding tangerine to the points that were added to the score

UI3.4 Inputs: User's left clicks on a leaf

UI3.5 Outputs: The visual representation of both the tangerine and the score

UI3.6 Normal

UI3.7 Whitebox

UI3.8 Functional

UI3.9 Integration

UI4.1 User Interface Test 4

UI4.2 This test will ensure the number indicators for both the vertical and horizontal axis are displayed accurately

UI4.3 This test will be done by clicking the leaves on the vertical and horizontal axis and ensure that the number of expected exploding tangerines and value of the other tangerines add up to the indicators for both the vertical and horizontal axis

UI4.4 Inputs: User's left clicks on a leaves

UI4.5 Outputs: The visual representation of both the tangerine and the indicators for both the vertical and horizontal axis

UI4.6 Normal

UI4.7 Whitebox

UI4.8 Functional

UI4.9 Integration

BE1.1 Back-end Test 1

BE1.2 This test will ensure the backend pulls the correct data from the score table to populate UI.

BE1.3 This test will run the same query on the database to pull users highest score from score table. This test will be ran using LINQ query in code base and separately using SQL server management studio as well.

BE1.4 Inputs: The input for this test will be LINQ query from codebase and SQL query and random manually entered values in the database.

BE1.5 outputs: All the user saved records for columns name, score and ranking from score table

BE1.6 Normal

BE1.7 Blackbox

BE1.8 Functional

BE1.9 Unit Test

BE2.1 Back-end Test 2

- BE2.2 This test will ensure the backend saves highest score on the database when player passes the previous high score and this will be done after the game is ended.
- BE2.3 This test will run the same query on the database to insert user score if the player beats the previous highest score. This test will be ran using LINQ query in code base and separately using SQL server management studio as well.
- BE2.4 Inputs: The input for this test will be LINQ query from codebase and SQL query and random manually entered values in the database. For LINQ query, it will get the record from UI component.
- BE2.5 outputs: Saves the new high score on the score table.
- BE2.6 Normal
- BE2.7 Blackbox
- BE2.8 Functional
- BE2.9 Unit Test

Test Case Matrix

	Normal/ Abnormal	Blackbox/ Whitebox	Functional/ Performance	Unit/ Integration
MT1	Normal	Whitebox	Functional	Integration
MT2	Normal	Whitebox	Functional	Integration
MT3	Normal	Whitebox	Functional	Integration
MT4	Normal	Whitebox	Functional	Integration
MT5	Normal	Whitebox	Functional	Integration
UI1	Normal	Whitebox	Functional	Integration
UI2	Normal	Whitebox	Functional	Integration
UI3	Normal	Whitebox	Functional	Integration
UI4	Normal	Whitebox	Functional	Integration
BE1	Normal	Blackbox	Functional	Unit
BE2	Normal	Blackbox	Functional	Unit

Results

MT1 - SUCCESSFUL. When user selected leaf, a tangerine was revealed with a corresponding point value which was then added to the total score

MT2 – SUCCESSFUL. When user selected a leaf which had an exploding tangerine, the game ends and board resets to retry game

MT3 – SUCCESSFUL. When the user selects a leaf which had an exploding tangerine, the game board is revealed to reveal all the normal tangerines and exploding tangerines. Once board was revealed, was able to confirm there were the right amount of exploding tangerines on the grid

MT4 – SUCCESSFUL. When the user completed the game level, the board reset, and the difficulty of the game got harder. There were more exploding tangerines on the grid resulting in the increase of difficulty.

MT5 – SUCCESSFUL. When a user gets a score that is higher than the current high score, the high score is updated to the new high score.

UI1 – SUCCESSFUL. When user inputs site URL in web browser, the site appears correctly. The game was shown along with necessary information. ****NOTE**** URL no longer works as the free trial of the cloud service has expired after the CEAS expo was finished

UI2 – SUCCESSFUL. When user clicked on a leaf, the leaf had been flipped. This renders correctly from UI aspect.

UI3 – SUCCESSFUL. When user clicked on a leaf, a tangerine rendered correctly with the correct score on it.

UI4 – SUCCESSFUL. When the user clicked on a leaf, it revealed either a tangerine or exploding tangerine. From there, we were able to verify the correct amount of exploding tangerines and normal tangerines were correct for each column and row.

BE1 – SUCCESSFUL. After entering in random data, the database updates to random data entered ensuring that the database updates correctly.

BE2- SUCCESSFUL. The backend saves the highest score on the database when the player passes the previous high score after the game has ended.