Validation Testing Strategy

Identifying Functionality of Main Use Cases

Test Number: 1

Purpose of Test: Start game by clicking the “Start” button from the main GUI.

Test Inputs: run gizmoball application by running the game from a desktop. When the GUI has loaded, click the “Start” button to begin gameplay. Board should then change from build game mode into play mode. Check if the game board is now displaying in the interface. Scan through GUI and ensure that gizmos and ball are displayed in the correct position (specified through pre-set board OR build mode map).

Expected Inputs: User clicks the start button from main interface to begin game.

Expected Outputs: Check Start button – Start button + Gizmoball game view

a. Normal Flow – Game begins in run mode letting the user start the Gizmoball game
b. Alternative Path – Button does nothing / Game does not start due to a fault within the system
c. Alternative Path – Interface displays gizmos, however not as the user has built the game, i.e. some of the gizmos are in a different place from when it was originally placed, proving a glitch in the system

Test Number: 2

Purpose of Test: Stop gameplay by clicking the “Stop” button displayed in the game interface. Board should be in run mode AND running, for the STOP command to continue so check that the board has met these conditions. After these conditions have been met, the gameplay should halt (check that the board is in run mode BUT NOT running). Gameplay should have stopped and key presses should not affect the state of the game.

Expected Inputs: User clicks the stop button from main interface to stop game.

Expected Outputs: Check Stop button – Stop button + Board

a. Normal Flow – Games stops running in gameplay mode
b. Alternative Path – Button does nothing / Game fails to stop and continues in gameplay, signalling a possible error in the system. User may have to force game shutdown if game fails to stop.

RUN MODE

Test Number: 3

Purpose of Test: Load Model (from save / game in progress)
Check if player's click on button "Load Model" triggers the file saved previously to load game. Board should be in run mode in order to continue with loading a previous save. Check if file selected is in the correct format (correct file extension). Saved game state should appear in the GUI. User should be able to continue from saved state (so maybe change Start > Continue after a map has been loaded).

Expected Input: User clicks the "Load Game" button from the GUI, looking to load a previously built game

Expected Output: Previously saved file (game data) is loaded into game, and the GUI displays the gizmoball build as expected.

a. Normal Flow – Model is loaded from a saved file and everything is in correct position, i.e. gizmos display in game as previously saved without corruption
b. Alternative Path – Button does not work, preventing the saved game from loading caused by an error in the system
c. Alternative Path – user clicks load game, however the file has been corrupted or deleted, preventing the game from being loaded
d. Alternative Path – game is loaded, however the gizmos are not displayed in the correct place as previously saved

Test Number: 4

Purpose of Test: Reload Model. Check if user's click on button "Reload Model" works as expected. Current map should then be "refreshed" (map should reload from pre-built map or user made map). Check if the correct map has been loaded (check gizmo locations and compare to saved gizmo locations).

Expected Input: click load game from start state

Expected Output: Gizmoball is loaded in run mode gameplay state

1. Check Reload – Reload Button + Board
    a. Normal Flow – Board is reloaded into start state
    b. Alternative Path - Button doesn't work caused by an error in the system

Test Number: 5

Purpose of Test: Save Model
User should be in build mode and at least 4 gizmo's (some progress) should have been placed in build mode (just some basic counter to keep track?) in order for this code to continue. After file is saved check if file exits and if it does continue, if not ERROR.

Expected Input: when user interface is displayed, user clicks the "Save" button, prompting the system to save the game data to a file for re-use at a later time.

Expected Output: Alert message is displayed to the user signalling success of saved game.

1. Check Save – Save button + Board
    a. Normal Flow – Gizmoball game setup is saved to a file correctly in user directory, containing all of the required data
    b. Alternative Path – Button does not work as expected, stopping the game from being saved due to an internal system error
    c. Alternative Path – game is attempted to be saved to a file, however the process has been corrupted, and an alert message is displayed to the user advising them of the error

Test Number: 6

Purpose of Test: Build Game-play
Check if player's click on button "Build Mode" triggers system implementation to let the user set up the build of the Gizmoball game. Board should then change from Run mode into Build mode.

Expected Input: user clicks "Build Mode" button from the run mode gameplay.

Expected Output: Game changes from run mode to build mode, displaying a variation in the interface, letting the user now add Gizmos etc.

1. Check Build Mode – Build Mode Button
    a. Normal Flow – Game enters Build Mode due to button click
    b. Alternative Path - Button does not work as expected and the user remains in run mode, signalling a glitch in the system

Test Number: 7

Purpose of Test: User is able to quit the game from GUI

Check if player’s click on button “Quit” triggers code tied to it. Check game has stopped and game interface shuts down (could possibly print something in console for debugging purposes if required).

Expected Input: User clicks “Quit Game” button from user interface

Expected Output: program stops executing and GUI is closed

1. Check Quit – Quit Button
   a. Normal Flow – Game quits / Halts execution and the program stops running
   b. Alternative Path – Button doesn’t work as expected / Game does not quit and user remains in gameplay

Test Number: 8

Purpose of Test: User is able to add Gizmo in build mode.

Check if player’s click on button “Add Gizmo” triggers code tied to it. (Button only available in build mode so no need to check if in build mode?). Check if a gizmo is added to the board in the correct position (where the user clicked).

Expected Input: user clicks button to add the specific gizmo they would like to add to the game. User can then select coordinates of the game for the Gizmo to be placed in.

Expected Output: Gizmo is displayed in user interface

1. Check Add Gizmo – Add Gizmo Button + Board
   a. Normal Flow – Gizmo is added to gizmo game grid where specified
   b. Alternative Path - Button doesn’t work
   c. Alternative Path – user selects gizmo to add, however the wrong one is displayed in the interface due to a system error

Use Case: Delete Object

Test Number: 9

Purpose of Test: to delete a gizmo the user has previously added to the game build. Check if player’s click on button “Delete Object” triggers event as required. Used to enter a “delete mode”. User clicks button then clicks on an object they want to delete. Exits the “delete mode” after one deletion to prevent accidental deletes. Check if there is something on the board to delete (check if not empty). Targeted object should be deleted, check that there is no object in the position of where the user clicked.

1. Check Delete Object – Delete Object Button + Board
   a. Normal Flow – Specified Gizmo is deleted from game build
   b. Alternative Path - Gizmo is not deleted, preventing the selected gizmo from being removed from the game, making us aware of a possible bug in the game.

Test Number: 10

Purpose of Test: Add Absorber to the game build

Check if player's click on button "Add Absorber" triggers code tied to it. Absorber should be added where the user specified (check where user clicked and that the object added is an absorber). Also check if there is already an object in the space clicked, if there is, do not add an absorber.

Expected Input: User clicks "Add Absorber" in build mode

Expected output: Absorber is displayed in the selected area on the game build

1. Check Add Absorber – Add Absorber Button + Board
   a. Normal Flow – Absorber is added where the user has specified
   b. Alternative Path – Absorber added but in the incorrect position
   c. Alternative Path – user clicks to add the absorber, however it does not display as expected

Test Number: 11

Purpose of Test: User Adds Flipper to the game for the purpose of colliding with the game ball

Check if player's click on button "Add Flipper" works as expected letting the user add it to the build mode of the game. Flipper should be added where the user specified (check where user clicked and that the object added is a flipper). Also check if there is already an object in the space clicked, if there is, do not add a flipper.

Expected Input: User clicks "Add Flipper"

Expected Output: the flipper is displayed in the game grid in the correct position for gameplay

1. Check Add Flipper – Add Flipper Button + Button
   a. Normal Flow – Flipper is added where user specified
   b. Alternative Path – Flipper is added but in the wrong position of the interface
   c. Alternative Path – flipper is not added at all, with a possible bug in the program which may need corrected

Test Number: 12

Purpose of Test: Key Connect / Input keys are connected to the correct events

Check if player’s click on button “Bind Key” triggers code tied to it. User then clicks on a flipper or absorber and a window will pop up. Check if the object clicked is valid. User then enters a VALID key to associate with the object. Then make sure that the key is bound to the correct flipper or absorber. Key bind that has just been made by the user should then be added to the current game.

Expected Input: user clicks the “Bind key” trigger

Expected Output: game reacts to the user interaction

1. Check Key Connect – Key Connect Button + Board
    a. Normal Flow – Gizmo is connected to a key press, and the flipper or gizmo reacts to the user input in the correct way as it should
    b. Alternative Path – Button does nothing
    c. Alternative Path: Gizmo is not connected to a key press

Test Number: 13

Purpose of Test: Key Disconnect / Input keys are disconnected to the events

Check if player’s click on button “Unbind Key” triggers the desired output. User clicks on an object and the program should unbind any key associated with it. Check that the object now has 0 keys bound to it.

Expected Input: user clicks the “Unbind key” trigger

Expected Output: user can establish if the key has been disconnected from the event previously triggered by testing it in gameplay

1. Check Key Disconnect – Key Disconnect Button + Board
    a. Normal Flow – Gizmo no longer has a key connected to it
    b. Alternative Path - Button trigger doesn’t work as expected
    c. Alternative Path - Gizmo still has a key connection

Test Number: 14

Purpose of Test: Add Gizmo-Ball to the game

when the user clicks on the “Add Ball” button it signals the ball to be added correctly as part of the game interface. Ball should be added where the user specified (check where user clicked and that the object added is a ball). Also check if there is already an object in the space clicked, if there is, do not add the ball to these coordinates. Ensure that there are no other balls on the board, if there are, delete the balls and the check again to make sure board has no balls.

Expected Input: User clicks to “Add Ball”

Expected Output: the Ball is displayed as it should, in the correct place

1. Check Add Ball – Add Ball Button + Board
    a. Normal Flow – Ball is added to the board
    b. Alternative Path - Ball is not added to the interface
    c. Alternative Path – Ball is displayed in the wrong game co-ordinates

Test Number: 15

Purpose of Test: Rotate a gizmo such as triangle or square

Black box test used to click on the “Rotate” button and test correctness. User will then click on an object. Check if object is there, if yes, continue. Code should then rotate the object 90 degrees. Check that the object has been rotated (basic variable that is updated after rotate?).

Expected Input: User clicks the selected gizmo for rotation

Expected Output: Gizmo rotates 90 degrees

1. Check Rotate – Rotate Button + Board
    a. Normal Flow – Selected gizmo is rotated
    b. Alternative Path - Gizmo is not rotated (check co-ordinates), and remains in the same position
    c. Alternative Path – gizmo rotates but not to the desired 90 degrees as expected, in this event the system will have to be altered to prevent this from happening and let it rotate correctly

Test Number: 16

Purpose of Test: Clear gizmos from the game build

when the user selects the “Clear Board” button, we would like to check that the gizmos clear from the game as expected. When the code is initiated, check that there is nothing on the board after clear has been run.

Expected Input: User clicks “Clear Board” button to clear everything from the game including the gizmos, ball and the flippers

Expected Output: Everything is removed from the game and the grid is clear

1. Check Clear Board – Clear Board Button + Board
    a. Normal Flow – Clears the board of gizmo and ball
    b. Alternative Path(s) - Button does nothing / Board still has gizmos on it

Test Number: 17

Purpose of Test: Move a gizmo to the desired location within the boundaries of the game

User clicks the “Move” button to trigger the code and alter gameplay gizmos. Check that board is not empty. User clicks on an object, then on the preferred location of where they would like to move it. Check that the user has clicked on a valid object (not empty). After object has been moved check that it is in the correct position by comparing mouse click location to current object position.

Expected Input: user can select the gizmo to be moved by using their mouse to drag the gizmo about the screen.

Expected Output: Gizmo is in a new position of the game

1. Check Move – Move Button + Board
    a. Normal Flow – Gizmo is moved to position specified
    b. Alternative Path(s) - Button does nothing / Gizmo doesn’t move / Doesn’t remove old gizmo position

Test Number: 18

Purpose of Test: Connect gizmos together

Player clicks the “Connect Gizmo” button. User checks that the game is not empty and the gizmos are displaying properly. User clicks on a gizmo and then another, these are now connected. Check that the 2 gizmos are connected / linked together. Exit Connect mode after 1 link is created to minimize accidental links.

Expected Input: User clicks the “Connect Gizmo” button using the gizmoball interface in build mode

Expected Output: Gizmos are connected as the user expects

1. Check Connect – Connect Button + Board
    a. Normal Flow – 2 Gizmos are now connected/linked
    b. Alternative Path(s) - Button does nothing / Gizmos are not connected/linked

Test Number: 19

Purpose of Test: Disconnect the already connected gizmos

Check if player’s click on button “Disconnect gizmo” works as expected. Check that board is not empty and that a current link / connection between 2 gizmos exists. User clicks on a linked gizmo and then the associated gizmo, these are no longer connected. Check that the 2 gizmos are no longer connected / linked together. Exit disconnect mode after 1 link is been deleted to minimize accidental disconnects.

Expected Input: User clicks the “Disconnect Gizmo” button using the gizmoball interface in build mode

Expected Output: Gizmos are disconnected from one another

1. Check Disconnect – Disconnect Button + Board
    a. Normal Flow – Gizmo link is destroyed
    b. Alternative Path(s) - Button does nothing / Gizmos are still connected/linked

Test Number: 20

Purpose of Test: Change Friction in gameplay, which will alter how the ball reacts to collisions and speed from the absorber etc.
Check if slider “Change Friction” trigger event as expected. User chooses value on a slider. Check that the number displayed on the GUI is the same as the value for friction.

Expected Input: User alters the friction level by changing the setting using the slider available in build mode

Expected Output: Friction level will change in gameplay, resulting in slightly different ball movement from collisions

1. Check Change Friction – Change Friction Slider + Board
    a. Normal Flow – Friction slider is changed, making ball movement differ
    b. Alternative Path(s) - Slider does nothing / Friction is changed but not to correct value

Test Number: 21

Purpose of Test: Change Gravity, making the ball fall faster or slower depending on user settings when falling vertically

User changes the “Change Gravity” slider. Check that the number displayed on the GUI is the same as the value for gravity.

Expected Input: User alters the gravity level by changing the setting using the slider available in build mode

Expected Output: Gravity level will change in gameplay, resulting in slightly different ball movement from collisions

1. Check Change Gravity – Change Gravity Slider + Board
    a. Normal Flow – Gravity slider is changed, altering ball speed in gravity
    b. Alternative Path(s) - Slider does nothing / Gravity is changed but not to correct value

Test Number: 22

Purpose of Test: Change Ball Velocity

Expected Input: User alters the Ball Velocity level by changing the setting using the slider available in build mode

Expected Output: Ball Velocity level will change in gameplay, making the speed of the ball change, and in run mode the game ball will move faster or slower depending on the user settings

1. Check Ball Velocity
    a. Normal Flow – Ball Velocity is changed, altering ball speed
    b. Alternative Path(s) - Slider doesn’t work / Ball Velocity moves faster or slower than