Validation testing strategy

Tests in Play mode

Test no 1. This test will let the user check if the "*Start game*" option functions correctly

i. To achieve the required test result, the user needs to firstly load up or create a build in "*Build Mode*".
ii. After loading up a build and now seeing it in a "*Play mode"* window. The user can see a couple of options that are available to select from. The one that we are looking for is "*Start Game*".
iii. After locating the option, the user clicks on it which initiates the start of the game.

If the test was successful user should see the following actions performed in "*Play mode"* window:

- The game clock begins ticking
- The ball begins its movement
- ***(optional output)*** The error messages which states:
  - "*Load a correct build*".
  - *"Game is already running*".

If user receives the following error messages they should restart the game and complete steps above.

Test no 2. This test will let user check if "*Pause*" option functions correctly

i. To achieve the required test result user needs be running a game in "Play mode" screen. With the clock ticking and ball moving.
ii. Once the Game is running user can see a couple of options that are in "*Play mode*" window. The one that we are looking for is "Stop"
iii. After locating the option, the user clicks on it which stops the game that is running.

If the test was successful user should see the following actions performed in "*Play mode"* window:

- The game clock stops ticking
- The ball stops its movement
- ***(optional output)*** Error message which states:
  - "*The Game is currently stopped".*

If user receives any of the error messages they should close the error message and complete steps above.

Test no 3. This test will let the user check if option "*Tick*" functions correctly

i. To achieve the required test result user needs be running a game in "Play mode" screen. With the game clock stopped and ball not moving.
ii. Once the game is stopped and the ball is not in motion the user can see couple of options in "*Play mode*" window. The one that we are looking for is option "Tick"
iii. After the locating the option user clicks on it which increments the game clock by **1** and moves the ball by one tick of the game.

If the test was successful user should see the following actions performed in "*Play mode"* window:

- The game clock is incremented by **1**
- The ball moves by one tick of the game (by the **one that clock has incremented**)
- The GUI also will update itself, if there are any interaction by other objects with the ball
- ***(optional output)*** The error message which states:
  - *"The game has not been stopped"* – this will pop out right after the user clicks on "*Tick*". This will not stop the game it will lead the game to progress in the background while the error message will be in focus for the user.

If user receives an error message, user should close it and complete steps above.

Test no 4. This test will let the user to see if everything is working while switching between the "*Play mode*" and the "*Build mode*"

i. To achieve the required test result user needs to be currently in "*Play mode*" window.
ii. The user can see couple of options in "*Play mode*" window. The one that we are looking for is option "*Build mode*".
iii. After locating the option user clicks on the button which closes the current "*Play mode*" window and brings up "*Build mode*" window.

If the test was successful user should see the following actions performed:

- The current "*Play mode*" window has been **closed**
- User can see a new *"Build mode"* window
- If the user was running a game they can see the current build of the game in "*Build mode*" window

There are no error messages that user should receive while doing this test.

Test no 5. This test lets the user to see if the option – "*Load game*" functions correctly.

i. To achieve the required test result user needs to be currently in "*Play mode*"
ii. The user has to have a **valid saved build** (*test no 7*)**.**
iii. The user can see couple of options in "*Play mode*" window. The one that we are looking for is option "*Load game*"
iv. After locating the option user clicks on it which opens a file explorer window with the location of all saved game files.
v. The window prompts the user to select a saved file
vi. The user selects a file that he has saved before or transferred onto device
vii. File explorer window closes.
viii. User is prompted with a new *"Play mode"* window that has the **saved build loaded** on the grid

If the test was successful user should see the following actions performed:

- A loaded up build of game form the selected file.
- ***(optional output)*** The error message which states:
  - "*The file is not valid*" – This pops right after the user selects the file and the file explorer window closes.

If user receives an error message this mean that the saved file is not the file that was meant to be used and/or is corrupted by editing it outside of the "*Build mode*". To complete this task, **locate or create a saved build** and complete the steps above.

Test no 6. This test lets the user to see if an option "*Reload game*" functions correctly.

i. To achieve the required test result user needs to be currently in "*Play mode*"
ii. The game should be running or it can be paused as well
iii. The user can see couple of options in "*Play mode*" window. The one that we are looking for is option "*Reload game*".
iv. After locating the option user has to press on it.
v. A new window appears with a message – "*Are You sure you want to reload game*"
vi. User can see two options: "*Yes*" and "*No*". User selects "*Yes*"
vii. User is prompted with a new "*Play mode*" window that has the current game refreshed on grid.

If the test was successful user should see the following actions performed:

- The "Play mode" screen has been refreshed with the same build of the game that was running before.
- ***(optional output)*** Error message which states:
  - "*The game has not been loaded*" – This pops right after the user clicks on "*Reload*" option button.
- ***(optional output)*** After user select "*No*" in the window where it requires a second user confirmation that they want to reload the window. They are taken back to the current game with game **paused or running** *(depends on the state of the game that the user has left it in while pressing the "**Reload**" button)***.**

If user receives an error message this mean that the game has yet been loaded in "*Play mode*". User needs to run the play mode with a build loaded.

Tests in Build mode

Test no 1. This test lets user to check if they can save a game correctly in "*Build mode*"

i. To achieve required test result the user needs to be in "*Build mode*" window
ii. The user has to create a playable build.
iii. After the creation of the build user should locate "*Save game*" option button.
iv. User clicks on it which prompts them with a new window called – "*Save game*"
v. The user can see an empty text field where he ***needs*** to enter a string of any symbols
vi. User is prompted with a new file explorer window where he needs to select the destination of the saved build.
vii. After selecting the desired location user now can see a window prompted to them that declares – "*The game has been successfully saved*"
viii. User needs to press "OK" to close the window.

If the test was successful user should see the following actions performed:

- A new file in the system of the user. That has been created while saving the build
- ***(optional output)*** Error messages which states:
  - "*The name of the file is not valid*" – This prompts up when a user uses symbols in "*Save game*" window that are not recognized by the system while creating new files.

- "*The system is to full*" – This prompts after locating the desired save file location. It means that the system has not have enough space on it for a new file to be created.

If user receives error messages, they should **give a correct name to** the file they want to save. Or if the system has no more space on it - just **clean up some disk space**. After fixing up error complete steps above to save.

Test no 2. This test lets user to check if they can load a game correctly in "*Build mode*"

i. To achieve the required test result the user needs to be in "*Build mode*" window
ii. The user has to have a **valid saved build** (*test no 7*)**.**
iii. The user needs to locate option "*Load game*"
iv. User clicks on it which prompts them with a file explorer window with the location of all saved game files.
v. The window prompts the user to select a saved file
vi. The user selects a file that he has saved before or transferred onto device
vii. File explorer window closes.
viii. User is prompted with a new *"Build mode"* window that has the **saved build loaded** on the grid

If the test was successful user should see the following actions performed:

- A loaded up build of game form the selected file.
- ***(optional output)*** The error message which states:
    - "*The file is not valid*" – This pops right after the user selects the file and the file explorer window closes.

If user receives an error message this mean that the saved file is not the file that was meant to be used and/or is corrupted by editing it outside of the "Build mode". To complete this task, locate or create a saved build and complete the steps above.

Test no 3. This test lets user to check if they can switch between the "*Build mode*" and "*Play mode*" windows

i. To achieve the required test result, the user needs to be in "*Build mode*"
ii. The user can see couple of options in "*Build mode*" window. The one that we are looking for is option "*Play mode*".
iii. After locating the option user clicks on the button which closes the current "*Build mode*" window and brings up "*Play mode*" window.

If the test was successful user should see the following actions performed:

- The current "*Build mode*" window has been **closed.**
- User can see a new *"Play mode"* window.
- If the user had a layout of a game in "*Build mode*" they can see "*Play mode*" window.

There are no error messages that user should receive while doing this test.

Test no 4. This test lets user to check if adding gizmos on the board works correctly.

i. To achieve the required test result user needs to be in "*Build mode*"
ii. The user can see couple of options in "*Build mode*" window. Any of the following options will let the user process with the test – *"Circle", "Triangle", "Square", "Left flipper", "Right flipper"*
iii. User presses the button on one of the selected options.
iv. User then can see a floating see trough gizmo **(depends of the option selected before)** on top of his mouse.
v. User clicks on the grid **(empty space)** to place the gizmo
vi. Gizmo can be seen place on the new updated build

If the test was successful user should see the following actions performed:

- The current build has been updated by the **addition of a new gizmo**
- ***(optional output)*** The error message which states:
  - *"The board is full"* – This error appears right after the user presses on the selected option in "*Build mode*" window
  - *"The selected tile is occupied"* – This error appears right after the user places a gizmo. Unfortunately, the gizmo is not placed after receiving this message.

If user receives the first error message this means that the user is required to clean up space by **deleting, moving gizmos** on the current build. If the user receives the second message it means that the space is occupied by other gizmo, it needs to be **deleted, moved** before placing a new one.

Test no 5. This test lets the user to check if placing the absorber on a build works correctly.

i. To achieve the required test result user needs to be in "*Build mode*"
ii. The user can see couple of options in "*Build mode*" window. We are looking for option "*Absorber*".
iii. After locating the option user clicks on it.
iv. User selects an **empty spot** on the grid to start placing the absorber.

v. The creation of absorber is "*click and drag*" based placing system. After clicking on the grid the user should not release the left mouse button but instead drag and release when it meets user requirements (*It is visible for the user how the absorber gets bigger or smaller while moving the mouse*).
vi. The user can see a placed absorber on the board

If the test was successful user should see the following actions performed:

- There is a new absorber currently placed on the grid.
- ***(optional output)*** The error message which states:
  - "*The board is full*" – This error appears right after the user presses "*Absorber*" in "*Build mode*" window
  - *"The selected tile is occupied"* – This error appears right after user select a **starting point for the absorber** or anywhere in between the **start of the absorber and the end of it** (the place where user release the mouse button)

If user receives the first error message this means that the user is required to clean up space by **deleting, moving gizmos** on the current build. If the user receives the second message it means that the space is occupied by other gizmos, it needs to be **deleted, moved** before placing a new one.

Test no 6. This test allows user to check if moving gizmos functions correctly

i. To achieve the required test result user needs to be in "*Build mode*"
ii. There should be at least one gizmo **created** (*test no10*) on the current layout.
iii. To start moving the gizmos user needs to locate option "*Move*" and press on it
iv. User selects a gizmo by clicking on it
v. The movement of the gizmo is initiated by user holding the left mouse key while pressing on the gizmo ("*Click and drag*").
vi. User releases left mouse key to drop the gizmo to the desired location
vii. The gizmo has been moved and it is visible to the user.

If the test was successful user should see the following actions performed:

- The gizmo's location has been updated
- The layout of the grid has been updated
- ***(optional output)*** The error message which states:
  - "*The gizmo cannot be placed in the desired location*" – This error message appears right after user places the gizmo where another gizmo already exists or if the user is placing gizmo on the walls of the grid.
  - "*There is no gizmo to move*" –This error message appears right after user selects the option "*Move*". It appears if there is not a single possible gizmo that can be moved.

If user receives the first error message this means that the user is required to find a space that is not occupied by other gizmos. The user can achieve it by **deleting, moving gizmos** on the current build. If the user receives the second message they need to **add** a gizmo first before selecting "*Move*" option.

Test no 13. The test would allow user to check if function "*Rotate*" options correctly

i. To achieve the required test result user needs to be in "*Build mode*"
ii. There should be at least one gizmo **created** (*test no10*) on the current layout.
iii. To start Rotating the gizmos user needs to locate option "*Rotate*" and press on it
iv. Now the user presses on the Gizmo that they wish to rotate.
v. After the click, gizmo that was clicked, have been rotated **90 degrees clockwise.**
vi. User exists the "*Rotate*" state by clicking on an empty space on the grid or pressing on option key "*Rotate*" again.

If the test was successful user should see the following actions performed:

- The location of the gizmo was updated (**by rotating 90 degrees clockwise**)
- The layout of the grid has been updated
- ***(optional output)*** The error message which states:
    - *"There is no gizmo to move*" –This error message appears right after user selects the option "*Rotate*". It appears if there is not a single possible gizmo that can be moved.
    - *"The gizmo cannot be rotated to the desired location"* – This error message appear right after the user clicks the gizmo that is meant to be rotated.

If user receives the first error message they need to **add** a gizmo first before selecting "*Rotate*" option. If user receives the second error message this means that the user is rotating a gizmo which will overlap with another gizmo. The user can fix it and achieve the desired results by **deleting, moving gizmos** on the current build.

Test no 14. This test allows user to check if deleting the gizmo works correctly

i. To achieve the required test result user needs to be in "*Build mode*"
ii. There should be at least one gizmo **created** (*test no10*) on the current layout.
iii. User selects option "*Delete*" that is visible in "*Build mode*" window
iv. After selecting the "*Delete*" option user clicks on gizmo that they wish to delete.
v. The gizmo after a click has been removed from the current layout.
vi. User exists the **delete state** by clicking on an empty space on the grid or pressing the option key "*Delete*".

If the test was successful user should see the following actions performed:

- The gizmo that was selected, while in the **delete state** (*a state when user clicks on "Delete" option and it termites after the second click on the option button),* has been removed from current layout
- The layout of the grid has been updated
- ***(optional output)*** The error message which states:
    - *"There is no gizmo to delete*" – This error message appears right after user selects the option "*Delete*". It appears if there is not a single possible gizmo that can be deleted.

If user receives the error message they need to **add** a gizmo first on the layout before selecting the option "*Delete*".

Test no 15. This test allows user to test if the function for clearing the board functions correctly

i. To achieve the required test result user needs to be in "*Build mode*"
ii. There should be at least one gizmo **created** (*test no10*) on the current layout.
iii. User selects "*Clear Board*" which is visible in "*Build mode*" window
iv. All of the gizmos that are placed on the board have been removed after the click

If the test was successful user should see the following actions performed:

- The board has been cleared and there are no more gizmos left on the board
- ***(optional output)*** The error message which states:
    - *"The board is empty, nothing to remove"* – The message appears following the user selects "*Clear board*"

If user receives the error message they need to **add** a gizmo first on the layout before selecting "*Clear Board*".

Test no 16. This test allows to check if the connection of two gizmos functions correctly:

i. To achieve the required test result user needs to be in "*Build mode*"
ii. There should be at least two gizmos **created** (*test no10*) on the current layout.
iii. User locates and presses the option "C*onnect gizmos*"
iv. User has entered the **connect state** by pressing the "*Connect gizmos*" key.
v. The user selects first gizmo (*by pressing on it once with the left mouse click*) that is going to be **the connective gizmo** (*the one to be connected to*) while in the **connect state**.
vi. User selects the second gizmo (*by pressing on it once with the left mouse click*), the one that is going to be **connected to the first one**.
vii. The refreshed view of the layout is now available to view for the user.
viii. Gizmos are now connected.

ix. User exits the **connect state** by clicking on an empty space on the grid or pressing the option key "*Connect*".

If the test was successful, user should see following actions performed:

- The gizmos that were select by the user have been now connected and it is visible on the layout.
- The build- was updated
- ***(optional output)*** The error message which states:
    - "*There are not enough gizmos to connect*" – This pops up following the user pressing on "*Connect*". There are not enough gizmos on the board to enter the **connect state**.

If user receives the error message they need to **add** at least two gizmos first on the layout before selecting "*Connect*".

Test no 17. This test allows user to check if disconnecting the gizmos functions correctly.

i. To achieve the required test result user needs to be in "*Build mode*"
ii. There should be at least **one set** of gizmos that are connected.
iii. User locates option "*Disconnect*" and clicks on it.
iv. **Disconnect state** is initiated.
v. While user is in **disconnect state** they select a gizmo that is connected with another one.
vi. The gizmo that has been **selected** is disconnected.
vii. User exit disconnect state by pressing on grid or pressing with the left mouse key the on button "*Disconnect*".

If the test was successful, user should see these actions performed:

- User should be able to see that two connected gizmos which were selected are now disconnected.
- ***(optional output)*** The error message which states:
    - "These gizmos are not connected" – This will pop out right after user tries entering the disconnect state.

The error message means that user has to **add at least two** gizmos and connected them together before pressing the button "*Disconnect*"

Test no 18. The test would allow user to check if setting key binding with flippers works

i. To achieve the required test result user needs to be in "*Build mode*"

ii. There must be at least one *flipper* **placed** on the current grid with one placed gizmo also.
iii. User finds an **option button** located in "*Build mode*" window called "*Key connect Gizmo*". After the click user enters "*Key connect"* mode.
iv. User selects a *flipper* that is not yet connected to a key.
v. User presses on the *flipper.*
vi. A **prompt appears** requesting: "Please press a key on you keyboard to get this flipper working".
vii. User presses a keyboard key.
viii. The prompt that was displayed previously has disappeared, after entering a key of the keyboard. "*Key connect"* mode is terminated.
ix. User proceeds by entering the "*Play mode*" window with a working build by clicking on a button "Play mode" (located in "*Build mode*" window)
x. User starts the game by clicking button "*Start Game*" located in "*Play mode*" window.
xi. After the game starts user presses the key which previously selected.
xii. After the key press user should the *flipper* **moving**

if the test was successful user should see following actions performed:

- The *flipper* that was selected should have the binding action set to a keyboard presses.
- While in the "*Play mode*" window the user should be able to see *flippers* moving according to the key presses.
- ***(optional output)*** The error message which states:
    - "*There are not enough gizmos on the board*" – This message appears after the button click of "*Key connect Gizmo*".

The error message means that user has to **add at a gizmo and a flipper** to enter "*Key connect"* mode.

Test no 19. This test should let the user check if setting a gravity works correctly

i. To achieve the required test result user needs to be in "*Build mode*"
ii. A text field that has a nametag – "*Gravity*" located in the "*Build mode*" window. User should locate it.
iii. **Change the gravity**. That is achieved by entering an inbound numeric value in the text field located before.
iv. User clicks on a button "Gravity" which is located by the text field in where the user was typing.
v. After the button click, **new value of gravity** has been set.
vi. User proceeds by entering the "*Play mode*" window with a working build by click on a button "Play mode" (located in "*Build mode*" window)
vii. User starts the game by clicking button "*Start Game*" located in "*Play mode*" window.
viii. The user can see the ball moving accordingly to the set gravity.
ix.

If the test was successful user should see following actions performed:

- The ball should be moving and changing its speed to the values entered in the "*Gravity*" text field accordingly, each time running the game.
- ***(optional output)*** The error message which states:
    - "*The value entered in Gravity field is not recognized, enter a value that is*" – This will pop up and focus in front of the view, right after user presses the "*Gravity*" button.

The error message signifies an incorrect value entered, this means:

1. Entered value is not numerical it has different values e.g. letters, symbols, spaces etc.

Test no 20. This test should allow user to check if disconnecting a bound *flipper* works correctly

i. To achieve the required test result user needs to be in "*Build mode*"
ii. There must be at least one *flipper* **placed** on the current grid that has a **key binding action attached** to it. (after a key press of a keyboard the flipper moves)
iii. User locates an **option button** which is in "*Build mode*" window called "*Key Disconnect Gizmo*".
iv. User clicks on it
v. "*Key disconnect"* mode is entered.
vi. User selects a *flipper* by clicking on it *that* is **connected** to a keyboard key.
vii. A **prompt appears** stating – "*The key binding has been removed*"
viii. User **exits** "*Key disconnect"* mode.
ix. User proceeds by entering the "*Play mode*" window with a working build, by clicking the button "Play mode" (located in "*Build mode*" window)
x. User starts the game clicking the button "*Start Game*" located in "*Play mode*" window.
xi. After the game starts user presses the key that was bound with the *flipper* (**Before step vi**).
xii. After the key press, user should the view *flipper* **not moving** as the binding was removed.

If the test was successful, user should see following actions performed:

- The *flipper* that had the binding action **set before step iv** should be not moving anymore with the presses of the keyboard key.
- ***(optional output)*** The error message which states:
    - "*There are no connected gizmos on the board*" – This message appears right after the the button click of "*Key disconnect Gizmo*".

The error message means that user has to **add a gizmo and a flipper** to enter "*Key connect"* mode by pressing the button "*Key connect Gizmo*" and proceed to this test then.

Test no 21. This test will let the user check if setting the friction of a ball functions correctly:

i. To achieve the required test result user needs to be in "*Build mode*"
ii. There should be a text field that has a button – "*Friction*" next to it in the "*Build mode*" window. User should locate it.
iii. **Change the friction**. That is achieved by entering an inbound numeric value in the text field located next to button.
iv. After entering a value user proceeds with pressing the Friction button to update the friction for the current build.
v. User proceeds with entering "*Play mode*" with a **working build** different value entered every time in the text field.
vi. Follow how the balls speed changes after bouncing of gizmos while the **game is running** and the ball is **moving**.

If the test was successful user should see following actions performed:

- The ball should be moving and changing its speed to the values entered accordingly each time, with the game running at the relative speed.
- ***(optional output)*** The error message which states:
    - "*The value is not recognized, enter a value that is*" – This will pop up and focus in front of the view, right after user presses the "*Friction*" option button.

The error message signifies an incorrect value entered, this means:

1. Entered value is not numerical it has different values e.g. letters, symbols, spaces etc.

Test no 22. This test will will let user check if adding a ball on the layout functions correctly:

i. To achieve the required test result user needs to be in "*Build mode*"
ii. There should be **two** text fields that have a nametag – "*Ball size*" and "*Ball velocity*" located in the "*Build mode*" window. User should locate it.
iii. **Change ball size.** This is achieved by entering a numeric value in text field ("*Ball size*") located before.
iv. **Change ball velocity.** Achieved by entering numeric inbound value, in the text field (*"Ball velocity"*) – **located in step ii**.
v. The button next to the text fields which has a nametag – "*Add ball*" is supposed to be pressed by the user.
vi. The ball is placed on the grid by user. The measurements are changed according to the values entered in "*Ball size field*".
vii. To test the velocity user is required to **launch** a build in "*Play mode*".
viii. User has to press "*Start*" option button located in the "*Play mode*" window to start the game.
ix. User can see the ball moving according to the set velocity.

If test was successful user should see following actions performed:

- The change in size of the ball, on the grid once the user has pressed "*Add ball*" button.
- The change in velocity of the ball once the user has entered the "*Play mode*" window and started the game by pressing "*Start*" button.
- ***(optional output)*** The error message which states:
    - *"Entered value in Ball size field is not recognized, enter a value that is"* – This appears right after the user presses "*Add ball*" button in the "*Build mode*" window.
    - *"Entered value in Ball velocity field is not recognized, enter a value that is"* – This appears right after user presses "*Play mode*" button in "*Build mode"* window.

Error messages signify an out of bound value entered in one of the fields, this means:

1. Entered value is not numerical it has different values e.g. letters, symbols, spaces etc.