Preliminary Design

Gizmoball is an arcade game very similar to pinball. The aim is to keep a ball moving around the playing area and not let it touch the bottom using different types of gizmos. Gizmos can be static objects, such as circles, squares and triangles or flippers which can hit the ball if it is in their reach. The final system would include a graphical user interface with 2 modes - editor mode and play mode. Below is the list of revised specifications in each mode:

Table of Contents

Revised Specification

Editor mode

- Initialise an empty area where the layout can be built
- Re-configure an already constructed layout

[1] triangle, square and circle bumpers
[2] connects a gizmo trigger to a gizmo action
[3] placing a new ball in the playing area

- Add any chosen type of gizmo to the playing area
- Add a ball to the playing area, specifying its position
- Move an existing ball to a different position
- Edit the velocity and the diameter of an existing ball
- Remove a ball from the playing area
- Set custom values for gravity and friction
- Move a gizmo from one cell to another
- Rotate a gizmo 90° clockwise
- Remove a selected gizmo from the playing area
- Connect gizmos together (link a certain gizmo's trigger to the action of another gizmo)
- Make a key press trigger a particular gizmo's action
- Save the configurations to a file
- Load particular configurations from a file and display it on the screen
- Go to play mode
- Clear playing area
- Exit the application

Play mode

- Start a game (release the ball in the playing area)
- Stop a game (stop the ball in its current position)
- Press keys that trigger gizmos' actions
- Save the current state of a stopped game (maybe not???)
- Load and display a saved game state (maybe not???)
- Go to editor mode
- Exit the application

Use cases

Add Gizmos

Precondition: Editor mode enabled

Trigger: Gizmo[4] type selected from the gizmos panel

Path:

1. The `20L x 20L` layout grid becomes highlighted.
2. The status label informs the user that he needs to select one grid location.
3. The user selects a grid square from the layout by clicking it.
4. If the grid square is occupied by another gizmo, go to `3.` . The status label informs the user that he can now add more gizmos of the same type animated for 2 seconds, go to `2`.

Postcondition: One or more gizmos of the same type have been added to the layout.

Add Absorber

Precondition: Editor mode enabled

Trigger: Absorber shape selected from the gizmos panel

Path:

1. The `20L x 20L` layout grid becomes highlighted.
2. The status label informs the user that he needs to select one grid location representing the right top corner of the absorber.
3. The user selects a grid square from the layout by clicking it.
4. If the grid square is occupied by another gizmo, go to `3`.
5. The status label informs the user that he needs to select one grid location representing the left bottom corner of the absorber, do `3 - 4`and go to `6`.
6. The status label informs the user that he can now add more gizmos of the same type, go to `2`.

Postcondition: One or more absorbers of the same type have been added to the layout.

Remove Gizmo

Precondition: Editor mode enabled

Trigger: Gizmo selected

Path:

1. User clicks delete button.

[4] triangle, square and circle bumpers

2. Gizmo is removed from the grid layout.

Postcondition: The grid layout does not contain the removed gizmo.

Connect Gizmos[5]

Precondition: Editor mode enabled

Trigger: Connect switch has been toggled

Path:

1. The user selects a gizmo from the grid, the gizmo then becomes highlighted. If the gizmo cannot trigger then notify user through the status label, go to `1`.
2. The user selects a second gizmo from the grid which then becomes highlighted in a different way. If the selected gizmo doesn't have an action then notify user through the status label, go to `2`.

Postcondition: The first gizmo's trigger is now connected to the second gizmo's action.

Clear playing area

Precondition: Editor mode enabled and at least one edit action performed

Trigger: "Clear Board" button selected

Path:

1. User clicks the "Clear Board" button
2. All gizmos are removed from the playing area

Postconditions: All gizmos are cleared from the grid layout. Physics properties like gravity and friction are preserved.

Adding a new ball[6]

Precondition: Editor mode enabled, a ball does not exist in the current grid layout

Trigger: "New ball" button clicked

Path:

[5] connects a gizmo trigger to a gizmo action
[6] placing a new ball in the playing area

1. The user enters values in the input fields for the velocity (0L/sec to 200L/sec), diameter (default is 0.5L) or chooses to go with the default values.
2. The user selects a grid location to place the ball at. If the grid location is occupied, go to `2`, else `end`. If The user clicks on an absorber go to `3`.
3. The ball is placed in the right bottom corner of the absorber.

Postcondition: A new ball is now added to the playing area.

Remove Ball

Description: Remove a ball that is already placed in the playing area

Preconditions: Editor mode selected and in remove mode

Triggers: Ball in the playing area selected

Paths:

- Main:
    1. User selects "Remove" button from the edit toolbar and is in remove mode
    2. If the user clicks on the ball, it is removed from the board, otherwise go back to M1.

Postconditions: an existing ball is removed from the playing area

Edit Ball

Description: Edit the velocity and the diameter of a ball on the board

Preconditions: Editor mode selected

Triggers: Ball in the playing area clicked and Properties tab selected

Paths:

- **Main Path:**
    1. User cicks on ball
    2. User selects Properties tab
    3. The user enters new values in the input fields for the velocity and the diameter.
    4. Ball is updated with new values
- **Alternative Path 1:**

1. Values are in the wrong format, notify the user.
2. Go back to M3.

Postconditions: An existing ball is updated with new values

Move Gizmo

Description: Moving a gizmo from one location to another

Preconditions: Editor mode selected

Triggers: "Move gizmo" button clicked

Paths:

1. User clicks on the gizmo to be moved
2. User drags gizmo to the new location
3. If location is already occupied go back to step 1.
4. If location is free the gizmo moves to the new location

Postconditions: Gizmo has changed its location

Rotate Gizmo

Precondition: Editor mode selected and in rotate mode

Triggers: Gizmo selected and rotate button has been clicked

Paths:

1. User clicks on gizmo
2. User Clicks delete button
3. Gizmo is removed from the map

Postconditions: Gizmo has been removed from the map

Edit gravity and friction

Preconditions: Editor mode selected

Triggers: Focus any of the g, mu or mu2 fields.

Paths:

- **Main**

1. User enters a new value in the field or leaves the field unchanged
2. If the value is in the wrong format, notify user and go back to step 1.
3. User clicks the apply button.

- **A1:**

1. If the value is in the wrong format, notify user.
2. Go back to 2.

Save configurations

Preconditions: Editor mode selected

Triggers: "Save" button selected

Paths:

1. User clicks "Save" button
2. A dialog box appears to specify the save location
3. User selects the desired location and clicks "Save"
4. A status label shows that the game configurations have been successfully saved

Postconditions: Playing area is shown ready for new actions; Game configurations have been saved to a file

---

Load configurations

Preconditions: Editor mode selected

Triggers: "Load" button selected

Paths:

1. User clicks "Load" button

2. A dialog box appears to specify where to load the configurations from
3. User selects the desired location and clicks “Load”
4. Game configurations are loaded and displayed on the screen; A status label shows that the game configurations have been successfully loaded

Postconditions: The saved playing area is shown ready for further editing

Game configurations have been loaded from a file and displayed on the screen

---

Stop game

Preconditions: Play mode selected, the game is running.

Triggers: “Stop” button pressed.

Paths:

1. User presses “Stop” button.
2. Game has been stopped

Postconditions: Ball stops (game has been paused)

Run game

Preconditions: Play mode selected, the game has been stopped or new game has been loaded.

Triggers: “Run” button pressed

Paths:

1. User presses “Run” button
2. The ball starts to move with the previous velocity and direction.

Postconditions: Game is running.

Physics loop (high level)

```
for every tick

    Calculate collision time for all gizmos (and balls if added) inside the map. The method

    If estimated time until nearest collision is greater than 0.05sec (time until next frame
```

```
    Set balls new coordinates where the ball will be after this time passes.
    Apply gravity and friction for that time period.

If time until next collision is smaller than 0.05, then
    update the coordinates of the ball taking into account time and velocity,
    calculate and set the velocity of the ball after the collision taking into account f

    If gizmo that ball collides with have trigger then
        trigger the action(calls the trigered() method on the object returned by calcula

Redraw the screen
```

Questions

1. How detailed the requirements should be: (Example: 1. Create your own map or 1. Add gizmos to map 2. Delete gizmos to the map 3. Flip gizmos)
2. Save game and/or save map?
3. Bonus stuff we could add:
4. Config game