Level Builder

Game Design Document

INDEX

ABOUT THE GAME

Elevator pitch

This is a VR Level Builder for an existing puzzle game, so it’s faster and more comfortable to create and test your own levels, taking advantage of the VR headset capacities

Goal of the game

We want to create a game that allows us the creation and testing of levels for an existing puzzle game, using Oculus Quest as our target platform, in order to play them in the final game.

We desire to create a fast and comfortable editor, as we believe being inside of the created level gives a brand new perspective of it.

Main features

The level builder will let you:

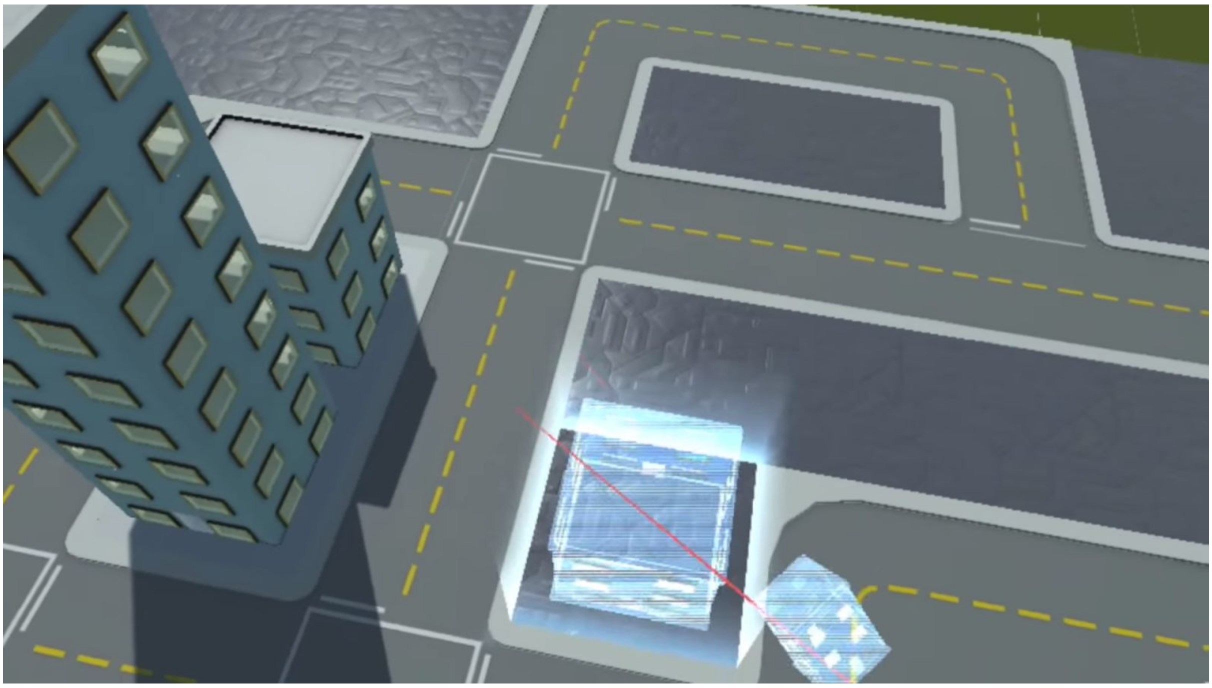
- Place and delete the main elements of the game
  - The cubes that shape the level
  - The goal where the player must reach, the position where it starts and the shadows it will find along its path
  - Axis rotators, the player must walk through to change the axis he is walking for
  - The light source and its pointing direction
- Incrementally save the current status of the level, and load the workspace it in the next session
- Undo and redo the last actions
- Test the level

MAIN REFERENCES

The main references we have used while designing this level builder are mainly games that have VR level builders, so we can see which decisions they have taken and how have they approached every problem.

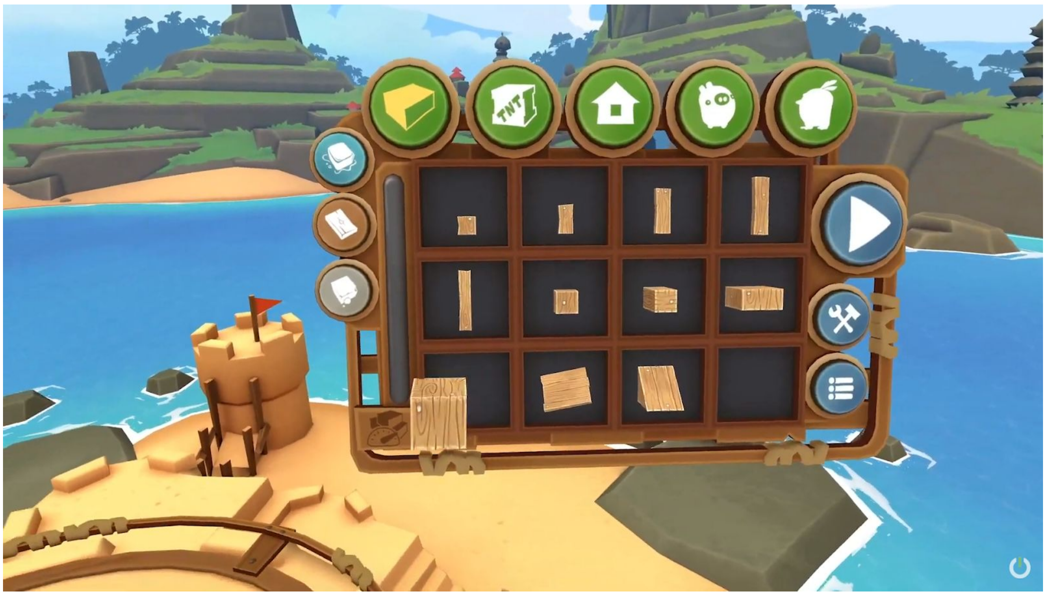
People Gun

Developed by Zach Tsiakalis, People Gun has a grid-based level editor, which places the level elements in a similar way we want our game to work. It also shows the preview of the placed object with a different shader before actually placing it.



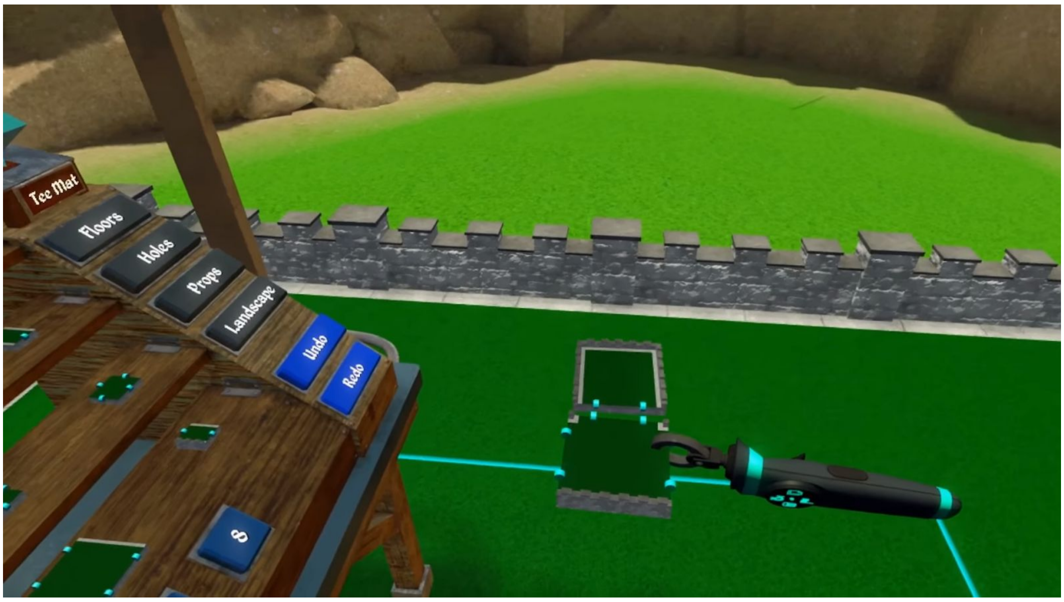
Angry Birds VR: Isle of Pigs

Unlike People Guns, the level editor of this game lets the user place the elements wherever they want, and doesn't constraint them to a grid, but has an in-world item selection overlay, in a similar fashion as our game will work.



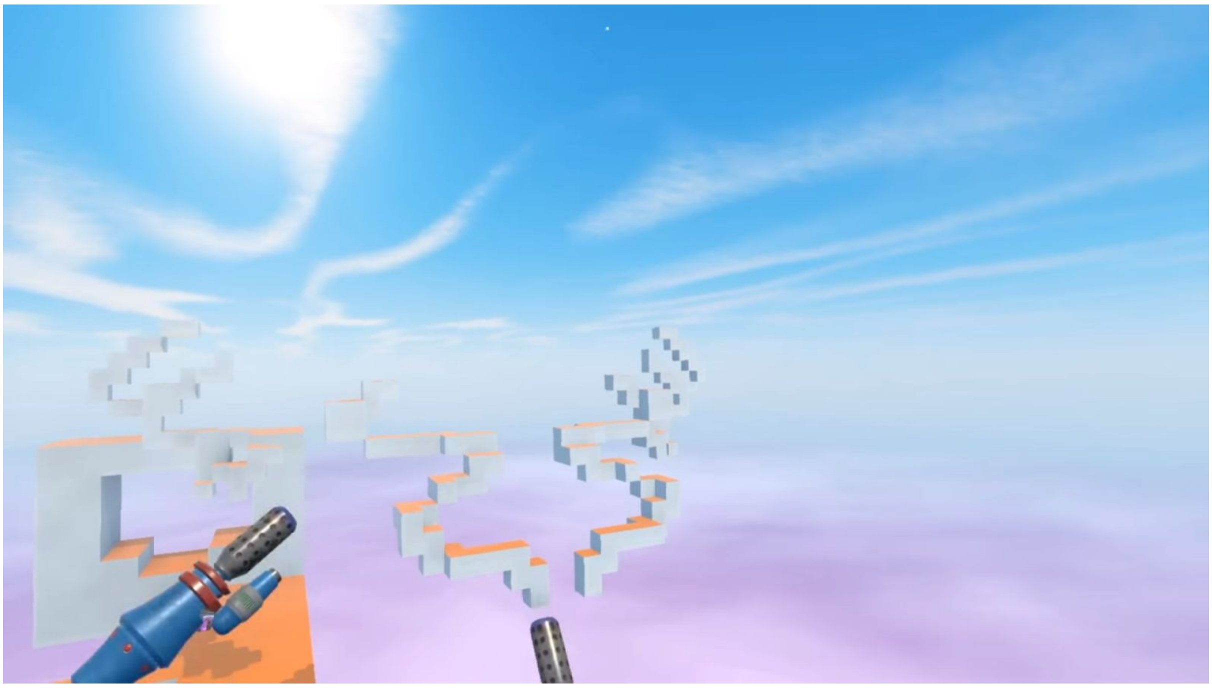
Cloudlands: VR Minigolf

This game (developed by FutureTown) makes the user choose where to place the selected object not with a raycast, but depending on the position of his hand, much like we want our game to work.



Oval

This level creator, done by Reactor Heart, works much like we want our game to work in the block placing aspect, as it lets the user drag and drop to place many blocks pressing the button just once.



OUTCOME OF THE DESIGN PHASE

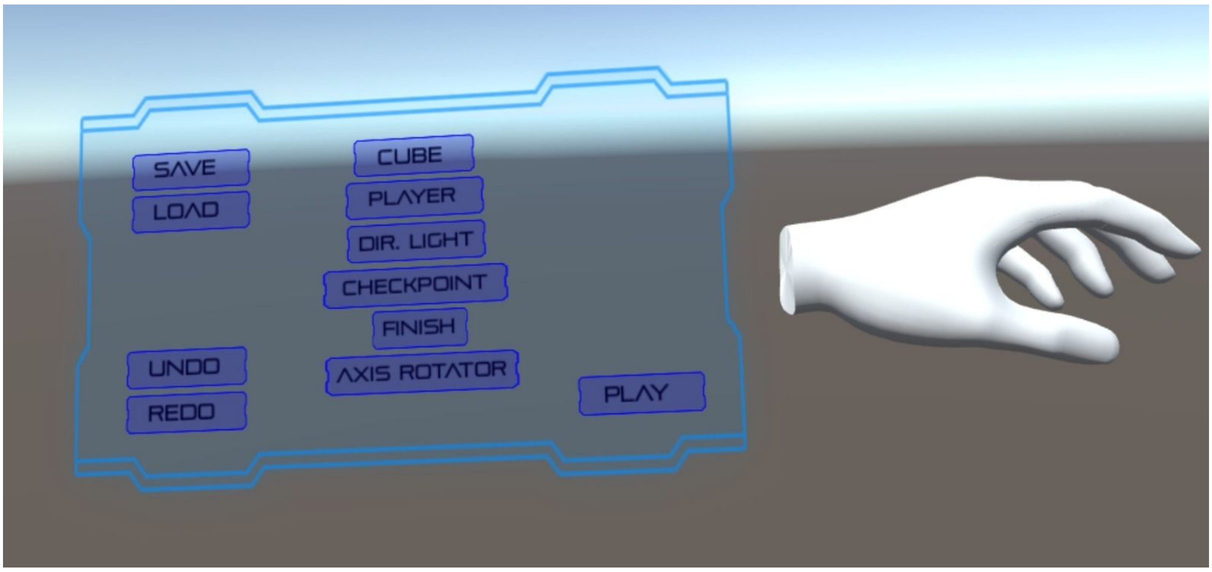
We've tried two different approaches of the block placing mechanic:

- Placing the block on the users hand position
- Choosing where the block is placed with a raycast

After some tests and tweaks, we've decided to go with the first idea, as we thought it was more comfortable for the user and his perspective of the depth of the world. We also added a different shader to emphasize which block we are placing, which helps to make the block stand out before placing it.

We've also made some slight changes on the input system, changing the controls so it feels more natural to the user and making an overall more enjoyable experience.

We also tried attaching the UI to user's left wrist, so it follows him no matter where he is, but due to he small size of the text and the amount of options available, it was confusing and difficult to use. Instead, we implemented the option to move the menu with the left trigger.



THE TEAM

The team is formed by:

- Nil Belmunt - Developer
- Àlex Gomez - Developer
- Òscar Masferrer - Producer
- Àlex Michalec - Developer & Tester
- Roger Sala - Developer
- David Solà - Developer & Tester
- Adam Tubau - Developer
- Albert Vegara - Developer

Each member has their own tasks, which are distributed every week according to their previous week work and the overall project needs and progress, prioritizing the longest tasks and the ones that create dependencies with other members.

Only four of the team members have a VR headset, so they'll have some extra job testing the job done by the other members.

PRODUCTION PLAN

Software used

We've decided to use Unity as or game engine, in the 2018.4.7 F1 version as it's the latest stable version for VR building.

For the team management, we have used trello, in order to distribute the work between the members of the team and keep track of which work is done and which work is pending.

As for communication between ourselves we've chosen Discord as our main chatting tool, for longer messages and conversations. For shorter messages we've used whatsapp, as a more informal tool.

Main Challenges

The undo-redo commands are a huge challenges, as they need a lot of testing in order to be sure of them working flawlessly, so we have a person working solely on it.

Due to not all members having VR headsets, testing becomes harder for the ones not owning them. Because of this, we have a person testing every new feature, and another one making the changes needed to fix the errors we find.

Future Goals

On the short term, our main priorities are:

- Fixing the block placing mechanic so **no branching paths** can be built
- Creating a map **load screen**
- **Implementing the remaining mechanics** in VR, like the world shrinking
- Decide whether to stay on this unity version or upgrade to a more recent one

On the long term we have to:

- Implement a play sistem to test the game in vr
- Merge the level builder with the puzzle game project
- Add the final models and implement art on the project
- Implement the final assets on the UI

MECHANICS AND GAME SYSTEMS

Mechanics

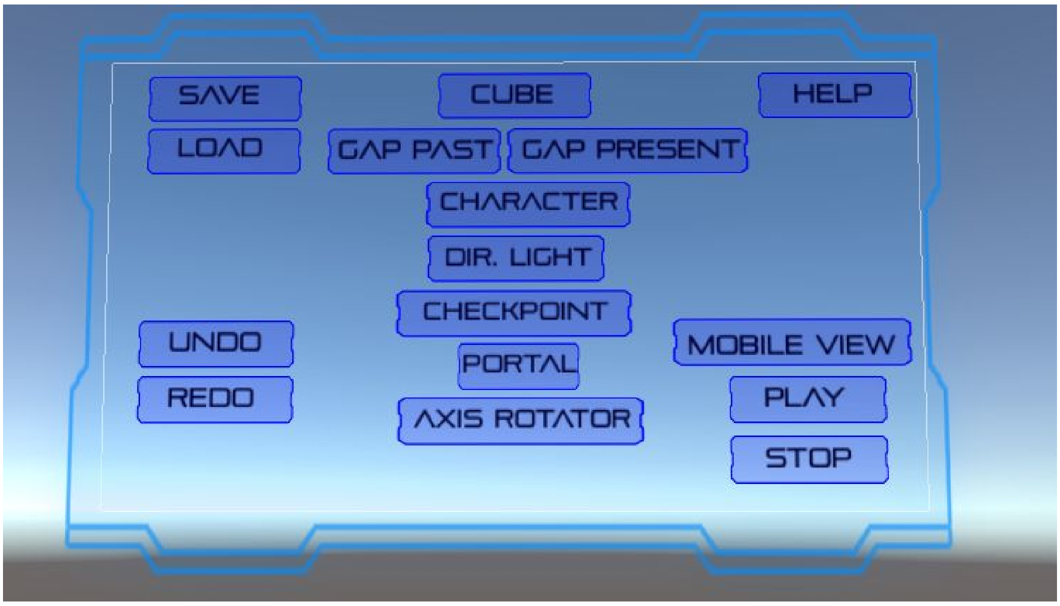
The user will be able to:

- Place new items in the world on the hand position, using the right controller’s top trigger
    - Select which item he wants to place, by selecting it from the list in the main ui of the game with the left controller’s top trigger
- Get into delete mode, holding the right controller’s bottom trigger
    - If the user is in delete mode, delete items with the right controller’s top trigger
- Shrink the whole level, in order to check how the game will be seen on a mobile phone
- Undo and redo actions, pressing the buttons on the world overlay with the the left controller’s top trigger
- Save and load a map, pressing the buttons on the world overlay with the the left controller’s top trigger
- Test the game, by pressing the play button

UI Design

We’ve designed and implemented a rather simple but functional UI. Our main goal was to achieve a compact UI, capable of containing all the editor functionalities while keeping it easy to understand.

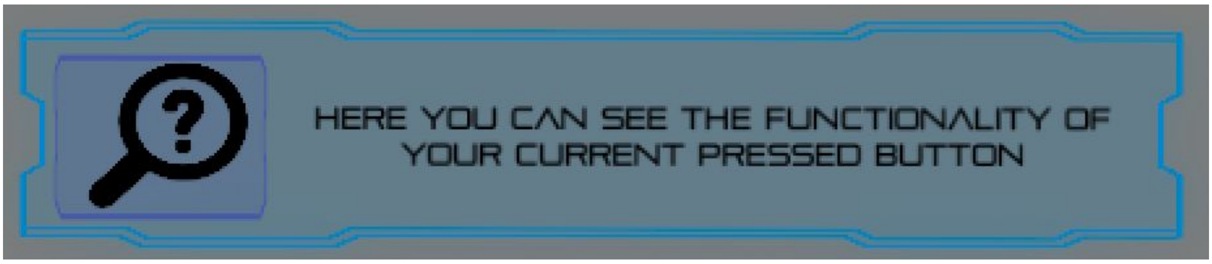
Thanks to testing, user feedback and iteration, we have been able to improve the final user experience & usability to ease the understandability of each of the functionalities.



The main menu contains all the editor functionalities. Each button has a hover functionality that changes the text from black to white when the user points at it with the left controller. Upon being clicked, the button background changes from blue to yellow in order to show the user what functionality is currently using.

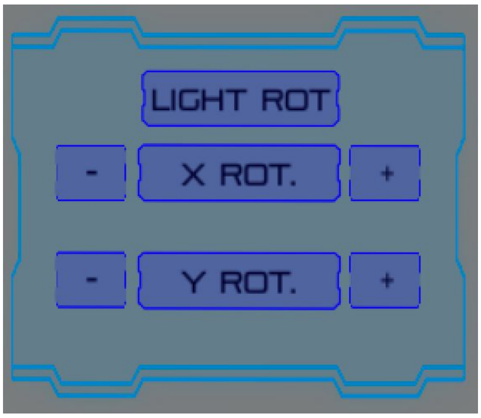
<u>List of the functionalities:</u>

Cube	Use this to build the basic geometry of the level.
Gap Past	Past cubes, only visible in the shadow.
Gap Present	Present cubes, only visible in the light.
Directional Light	Use this to place a directional light in the level.
Checkpoint	Place a checkpoint within the level.
Character	Use this to place the character in the level.
Undo	Reverse the last command executed.
Redo	Repeat your previous command.
Portal	Place a portal inside the level.
Play	Start playing your level.
Stop	Pause the gameplay.
Save	Save your current level progress.
Load	Load a saved level layout.
Mobile View	Swap to mobile view.
Axis Rotator	Changes the character walking axis.
Help	Shows the tutorial again



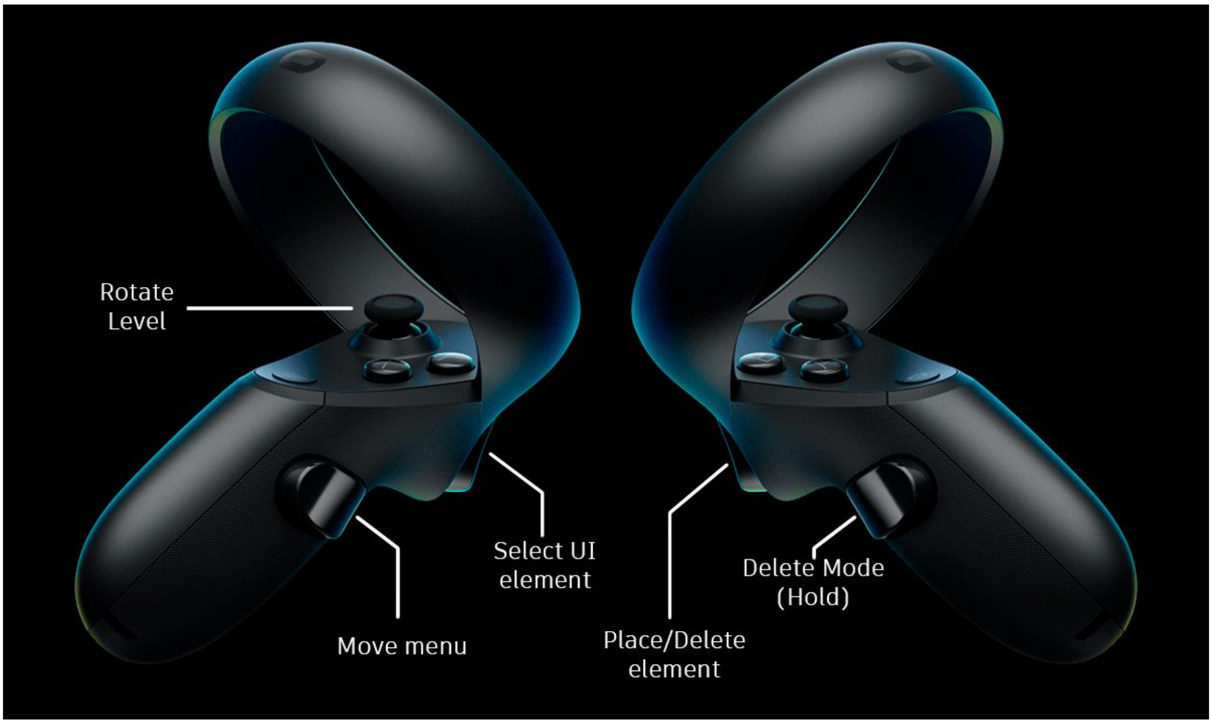
HERE YOU CAN SEE THE FUNCTIONALITY OF YOUR CURRENT PRESSED BUTTON

The help tooltip displays basic information about the current pressed button in order to explain its functionality. It also shows a basic preview image of the object if it’s spawnable.



When a light source is placed inside the level the light rotation settings menu will show up in the UI. The user will be able to rotate the light while playing the game in both X & Y axis.

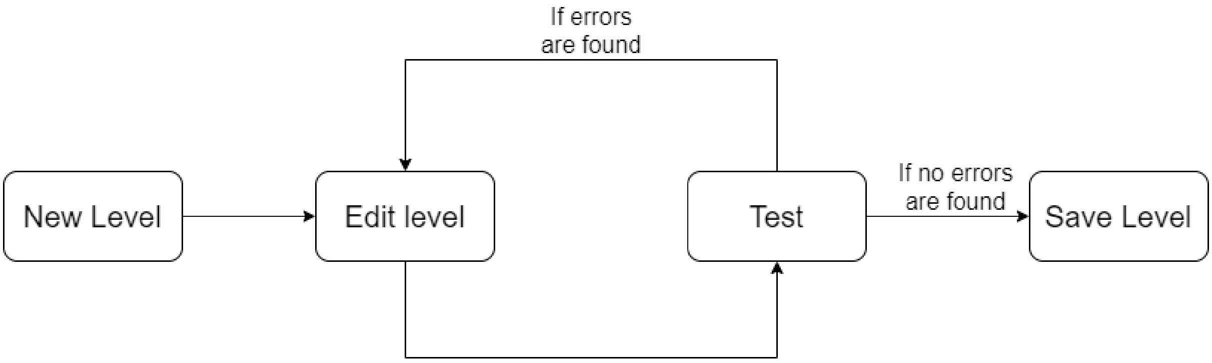
User input



We’ve mapped the user input so it uses both controllers are used, but constraining the right one to build the level, and the left one to the actions not involving the building itself like changing the place block, moving the menu and changing the rotation of the level.

Game cycle

1. **Load level/Create new level**
2. **Edit level**
3. **Test level**
4. **Fix level**
5. **Save level**


If errors are found
New Level
Edit level
Test
If no errors are found
Save Level