**Platforming Adventure**

**Project Hook**

Designing an entire platformer by myself

* Building levels as a full experience *Read More >*
* Planning for a tight deadline *Read More >*
* Developing skills in new fields *Read More >*

**Project Homepage**

Platforming Adventure is a solo project I created to practice level design. I wanted to lean into rapid iteration, focusing on the *Kishotenketsu* design philosophy popularized by the *Super Mario* games. This involves introducing a mechanic, developing it, and adding an unexpected twist. This is all done in a short amount of time, usually just a level or two, and then the mechanic is dropped to focus on something new.

**Building Levels as a Full Experience**

I believe Level Design is the most important part of a good platformer. It takes thorough planning to create levels that are each exciting and unique. If done properly, any level should be a fun experience on its own. Together, a group of great levels can make any game stand out.

**Objective**

My goal for this project was to build three levels, each focused on a different mechanic. Each level should be:

* Easy to pick up and play
* Short (1-3 minutes)
* Focused on a unique mechanic

Since I was building the entire game myself, I had some limitations. The unique mechanics for each level would also have to be programmed by me, and would cut into some of the level design time.

**Research**

My main goal for this project was to practice a form of level design called *Kishotenketsu*. In this style, a mechanic is introduced and developed at the start of a level. It then has some form of twist that uses the mechanic in new and unexpected ways. At the end, it returns to its simple form for a final challenge.

A video game with a game

Description automatically generated with medium confidence

I researched the *Kishotenketsu* design style through various videos and blogs. I also played levels from several different games to see how it was implemented. I looked into how mechanics were implemented and broken into smaller challenges. One thing I hadn’t considered was that there is a “cooldown” area between each challenge. This would be a safe place to collect some coins or power-ups, a checkpoint, or even just a straight path with no enemies. This gives the player some variety, so the entire level isn’t justthe core mechanic.

**Picking a Core Mechanic**

When I started designing the first level, my goal was to pick a mechanic to focus on. My first attempt was to feature breakable blocks that the player could destroy by attacking. When I tried sketching out some challenges, I discovered that the mechanic didn’t seem to stand on its own. I did more research to figure out why it didn’t work. I found out that players abilities are best used to accent the level’s mechanic, rather than being the mechanic themselves. For example, the dash ability in *Celeste* is present in every level, but no level is just about dashing. Rather, they each have their own unique features that work with the dash to let the player show off their skills. With this in mind, I wanted to find a new mechanic that would highlight the player controller.

A drawing of a game map

Description automatically generated

**Rapid Prototyping**

To find something fun to build my levels around, I created a test level with everything I could think of. I took simple scripts for moving and rotating platforms and made as many combinations as I could. Then, I ran around the level and played with each one to see what they played like. I particularly liked the idea of rotating platforms with spikes on them, and started thinking about other ways I could build off that concept.

**Back to the Drawing Board**

Once I had a core mechanic, I went back to sketching out a level.

**Playtest Feedback**

I had many people play my level over the course of the project. There were many minor pieces of feedback that I implemented over the weeks, but one stands out in particular. I designed my level from a sidescroller perspective. Most playtesters, however, but the

**Planning for a tight deadline**

**Overview**

This project was built from scratch in just ten weeks. While the level design was my core focus, everything had to be built by me. It took a lot of planning to make sure the project stayed on schedule. Throughout the duration of the project, I had to change and cut many features to make sure the game would be finished in time. It’s far from what was originally planned, but I’m happy with how it turned out.

**Creating a Schedule**

I started by making a rough outline of what work I needed to have done by the end of each week. I broke the time into three categories: system, level, and polish. The first two weeks were dedicated to developing the player controller, adding enemies, and updating the camera. The bulk of the time was dedicated to designing the levels, with two weeks planned for each. The last two weeks were for cleaning up external features to make the game more appealing.

A screenshot of a computer screen

Description automatically generated

At the start of each week, I would look at my plan to see what I needed to do. I would then make a list of specific tasks that I need to accomplish that week. As I completed them, I marked each one to keep track of progress. I also tracked how long each task took so I would be able to plan better in future weeks.

A screenshot of a computer

Description automatically generated

**Unexpected Roadblocks**

I had planned to spend two weeks on each level initially. One week would be sketching, whiteboxing, and playtesting, and the other would be bug fixing, setdressing, and polish. When I started working on the first level, the sketching and whiteboxing phase **didn’t go as planned**. I ended up losing most of that week and starting the level over from scratch, which put me behind schedule. I was able to create a much stronger level, but knew I would still need more time to polish it.

**Adjusting the Plan**

At this point, I had a difficult decision to make. If I stuck to my original plan, I wouldn’t have time to polish the levels. Otherwise, I could make just one level, which would prevent me from practicing rapid iteration. It was a difficult decision, but I decided to cut down to just one level. While I had hoped to focus on building several levels with unique mechanics, I knew that I wouldn’t be able to create strong levels if I also had to program the mechanics too. I already had a working level select screen, so I could always add more levels in the future. This way, I could guarantee I reached a level of polish I was happy with.

**Finishing the Game**

Following the new plan, everything went smoothly through the level design process. Thanks to the extra time, I was able to **adapt to unexpected feedback** that came up during playtests.