EECS 494 Project 3 Postmortem

Zhenning Zhang*

December 11, 2023

Showcase

Overall, I had a lot of fun by watching players enjoy the level I built and communicating with them.

We luckily grabbed a high table and placed a monitor on it for the Wii mote, which engaged a lot of visitors by just looking at a player playing the Wii mote version. It absolutely contributes to our success in voting, considered that the #votes $\geq$ #players who tried our game.

Me and Grant mainly focused on communicating with players, while Michael and Kevin were sitting behind and solving technical issues. This allocation worked perfectly for us, with the fact that Grant designed level 1, 2, and 4, while I designed the level 3 (purple level).

Speaking of technical issues, we met some problems on Wii mote and the scene transitions. But we were able to fix them really quickly.

From observing the game play

I absolutely learnt a lot of things regarding the game during the playtest. The most notable thing was the difficulty of the game. There are some points where most players were stuck. Some of them caused serious problems while the others were fine.

In short, there were problems about theming and room, flow of the level, the restrictiveness of the room, and the precise inputs. Meanwhile, there are also parts that was not intentional, but worked better than I expected.

(Due to the space limit, I just describe them here. The detailed analysis & discussions of those points are at the end of this file.)

*Game: Tracy and The Magic Brush, done together with Kevin, Michael, and Grant

Creative Process

About the initial idea

Our initial idea was to make a completely new game that uses some part in each of our games. We brainstormed the idea that we would want our new game to have.

But struggling coming up with new idea, Grant promoted his P2 project and we used it. Also, we re-used some of the brainstormed ideas into the game.

Modifications on the P2 game

Things we did come up with is the Celeste like map (continuous rooms for a big level), and introducing each color once at a time (a theme for one level). The continuous rooms is derived from the idea that (mainly because I was playing Celeste at that time and Grant is a big fan of it) we need to have a good flow and tell a good story, instead of just making a collection of small quizzes (Actually, our very first gold-spike / milestone submission was me finishing the continuous levels, although the implementation is different from currently we have).

The idea of introducing each color at one big theme level was mainly due to: a. We want each team member to have a deep understanding of the mechanic and design rooms that let the player explore the mechanics deeply and b. project management to enable working on different levels at the same time. (Worth to note that the "one person per level", while soundly perfect, was not working at the later part. It also caused more problems when the designer of the level obtained different idea from the other teammates)

Development and Iterative Process

The process that transformed our original idea into the final game

In the first week or two, our main focus was to build parts that we would like to have in the game. We brainstormed a lot of new features and objects, tested and implemented them in the individual labs, created working prefabs, and then dragged them into the level where they are desired. (Although some of them were not used eventually, they were still very inspiring.)

After we got those prefabs done, in the week 3-6, we focused mainly on designing level layout (there was also vast efforts on making the trailer, sound effects, Wii control, and art assets). We designed and implemented levels (individually, based on the core mechanics of each color) and modified them based on player response. The black and blue level (layout + aesthetics) were done relatively early on week 4 and 5, and the purple level layout (my level) was done on week 6, and the red level was done on the final week, while none of us really tested it and went to the end.

From week 5 onward, there was a lot of juice going on, including aesthetic changes and sound effects. And there were no big modifications on the level layouts.

Iterative Process

We performed weekly playtesting in class. There were a lot of comments (e.g. about difficulty, usage of colors, etc) and a few feedback. We tried our best to learn and summarize those comments and feedback, as well as noted down and analyzed their response (body language).

As an example of feedback, one of the ideas from the instructor was, since the player will focus on the line when they are drawing, we should move the ink meter to the drawing position. And we indeed did that. Moreover, we (or just me, since I was making the new cursor) learnt from the comment about the "focus" of the player. Under the same logic, I also made the cursor to be centered around the player once they gain or loss any paint, like on regeneration fountain.

Sometimes we did a bad job on adjusting based on the feedback. For example, the crazy difficulty + unreasonable design of some levels (at the end of this file) were well recognized during the playtest session long before the gold build. However, we did not change most parts of them.

About Time Estimates in Project Management

I was bad at estimating the task time during the entire project. Especially for the level design part, I spent much longer time than I expected designing and implementing the level (For example, I expected

4 hours for designing + implementing $\frac{1}{3}$ of the level, but it turned out to be more than 12 hours, which is fair).

But for technical stuff, it is not that off. I sometimes overestimate fixing technical problems, but underestimate creating technical solutions. For example, I planned to use 2-3 hours on the Zelda-like cursor, but it turned out to be 4 hours, as I was constantly inspired and started to add more logic inside.

Dry Code

The code is 80% dry. As a group project, this is already a good result I would say. Since we took Grant's P2, the very first thing after we made that decision was to inspect the codes. And I found the problem of the code to be very severe. The components were linking together and there was often a huge chunk of code inside one script / component. And there is no use of EventBus.

Therefore I re-write the code to make it somehow extendable for the upcoming development, and it was indeed the most worthwhile coding thing that I did. Even during the first few weeks of the development, we always need to go back to Grant's code and try decoupling things to improve the readability and the consistency. But later on, when it was becoming dryer and dryer, the development process started to become smooth.

One specific example is the development of the cursor. It utilizes events for paint pickup, fountain regeneration, choosing color, drawing, and etc. As we have a really solid code base, developing based on those events was very straightforward and easy.

I don't think there is any downside of having a dry code base, except I could be mad when my teammate is reinventing things we have.

The use of AI assistant

In this project, the AI assistant (ChatGPT) is always the first step when I meet some problem. As the questions were always very abstract, I seldom copy and paste things from it. But sometimes ChatGPT is giving over complicated results. For example, I can just use hotspot in changing the cursor sprite offset, but ChatGPT offered an answer that was way too complicated.

Future Improvement

Right Things

Including the polish, the deepness & completeness of core mechanics and levels, the aesthetics and theming, the overall quality & flow of the game and trailer, and dryness in terms of codes.

Wrong Things

Mostly about decision making and project management.

I sometimes feel this is too "Grant's own game," instead of "our game". We did not get a chance to touch the core mechanics, there was no discussion on theming and art style.

And one of things that's a little annoying to me was the modifications on my level, where some parts that players didn't like was not my intention. I think it is always fine when someone is changing something, but at least there needs to be some discussions, especially one parts that affecting the players' experience directly.

Project management goes wrong when nobody is using Jira and it is messed up (wrong assignment of jobs + missing tasks). Another thing about project management was, we do not have enough time to design and implement the final level. That was somehow due to us saying "anyone can take the final level," then it ends up with nobody doing anything on it.

Measures to ensure a better result in future creative endeavors

In terms of improving the creative endeavors.

Having a more scientific or reasonable way to measure a level.

Being willing to change or even remove the level based on players' response.

Better project management by creating jobs individually.

A more balanced power distribution, i.e. "our game".

AI assistants in future projects

The use of AI assistants is good for now. For the future I might try using AIs to generate some assets to reduce the man hours on arts, so we have more time on other core parts.

EECS 494 Reflection

Largest Takeaways & Things I get

Generalizable concepts / idea.

(Theming / juice & the real thing behind)

(Teaching / tutoring / guiding and exam)

(Intuitive learning)

(Deep mechanics)

(Branching and choices)

(Importance of playtest and iterative process, time to implement a function)

(Slowly improves to become perfect)

(Concept of interesting decisions short run and long run)

(Very generally designing something fun and its measurements / criteria)

(Confirmation bias & playtest)

(Specialization / Professionalism)

(General Creative process, how it is like)

Generalizable techniques.

(The composition v.s. inheritance structure of code)

(Project management + Assign jobs / spec myself)

(Team work basics)

(Better understanding of class responsibilities in coding)

(PubSub / Eventbus Systems)

(DRY principle and referencing)

(GitHub usage)

(Prefabs)

(Parallax & Aesthetics)

Appendix: Level Design Details

On the playtest, I felt most nervous about (since I was afraid that the player cannot understand the game mechanics and just leave) two rooms: the room of the first drawing, and the crazy difficult puzzle room at the blue level.

The confusing start

After the player gained the very first paint bucket (black), they were required to

i. draw two consecutive long lines and

ii. move really fast

in order to get passed (the line disappears very quickly). The worse thing is that, if they fall from second place (which happens very very often), they need to

iii. pass the first part of the challenge again.

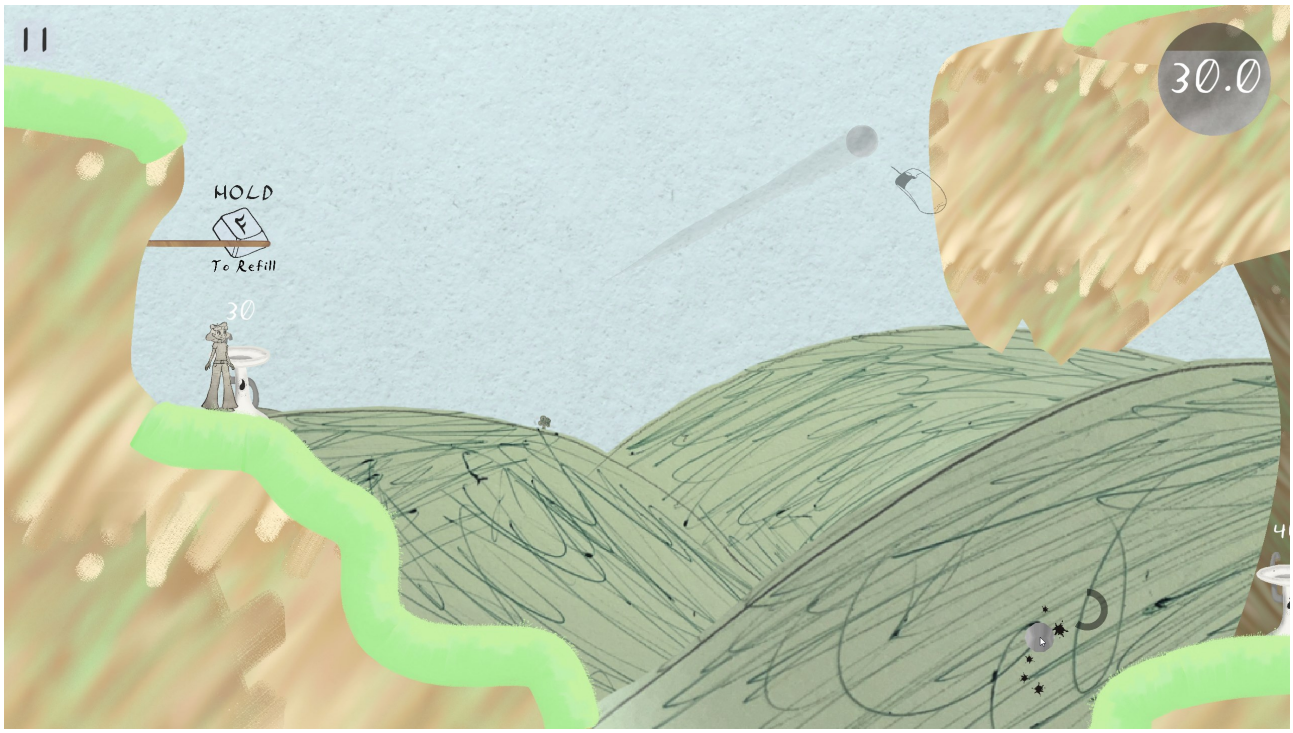

Figure 1: The Confusing Start

When the player is not really confident about the mechanics of the game, (I felt) it is really frustrating.

Another reason that makes that a frustrating level is

iv. the vast open space that we give.

When the player ever draws for the first time, I would expect it to be a very limited space that the only way to get through is drawing the line at a very very obvious position. The room just after that part is ideal, but as the very first level, it is not that appropriate. (I want to use the example of The Legend Of Zelda, Breath of The Wild, which the game offers a very clear path at the beginning, instead of just dropping us off in the wild.)

(One potential reason that stops us from changing is the nice implementation of the bridge that adds story to it, but I think we can always change the layout in a smart way to take both advantages)

The crazy difficult blue room

In the blue level, there is a difficult part taken from Grant's P2. Some players cannot pass this level and need assistance from us.

I think that room does not fit the ice settings (slippery even makes it harder). The room is not very well integrated into other parts as well, you can clearly observe the other parts of the blue level telling the adventure story and have a well defined sense of location, which is not in the level (It's just some floating platforms).

Figure 2: The crazy difficult blue room

What I would expect for the blue level is, the level has a very good flow, and the player does not need to make precise input to get passed. It is indeed the case when we as game designers are playing it, but it is not so for the new players.

Difficulty Settings

To explain more on that, I need to talk about the difficulty settings. We do have a difficulty setting, but that's only changing the amount of paints we have. It's the only thing we can do given the lack of time. Sometimes when there's a hard level, we tried to say giving more paints is going to solve everything (I knew it's not going to work, so I did not design any too difficult level).

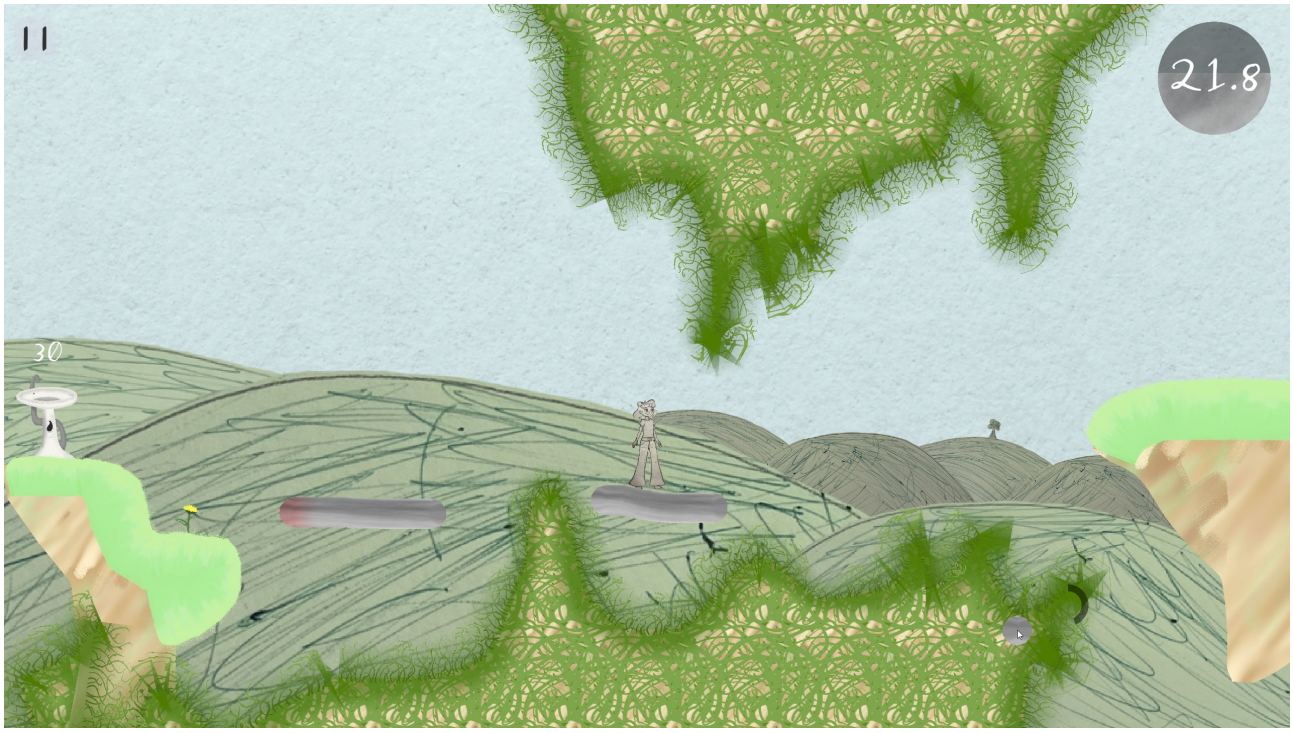

Figure 3: Precise inputs are required regardless of paint amounts

However, it's an undersight. Simply giving more paints does not help with important problems such as precision inputs (movements and drawings) and hazards. Therefore even if we changed the difficulty settings, the hard level is still hard.

(More level designs on the next page)

The purple level

A difficult room

For my level, I think it basically works fine except for the room that needs multiple purple trails.

That room confused basically everyone, so I kind of feel that the players were not so focused on the vanishing trails. Since the room is one of a few rooms that the player needs to focusing on the vanishing mechanics, we shall just lower the importance on it (I think the only reason why trails are vanishing is to prevent softlocks).

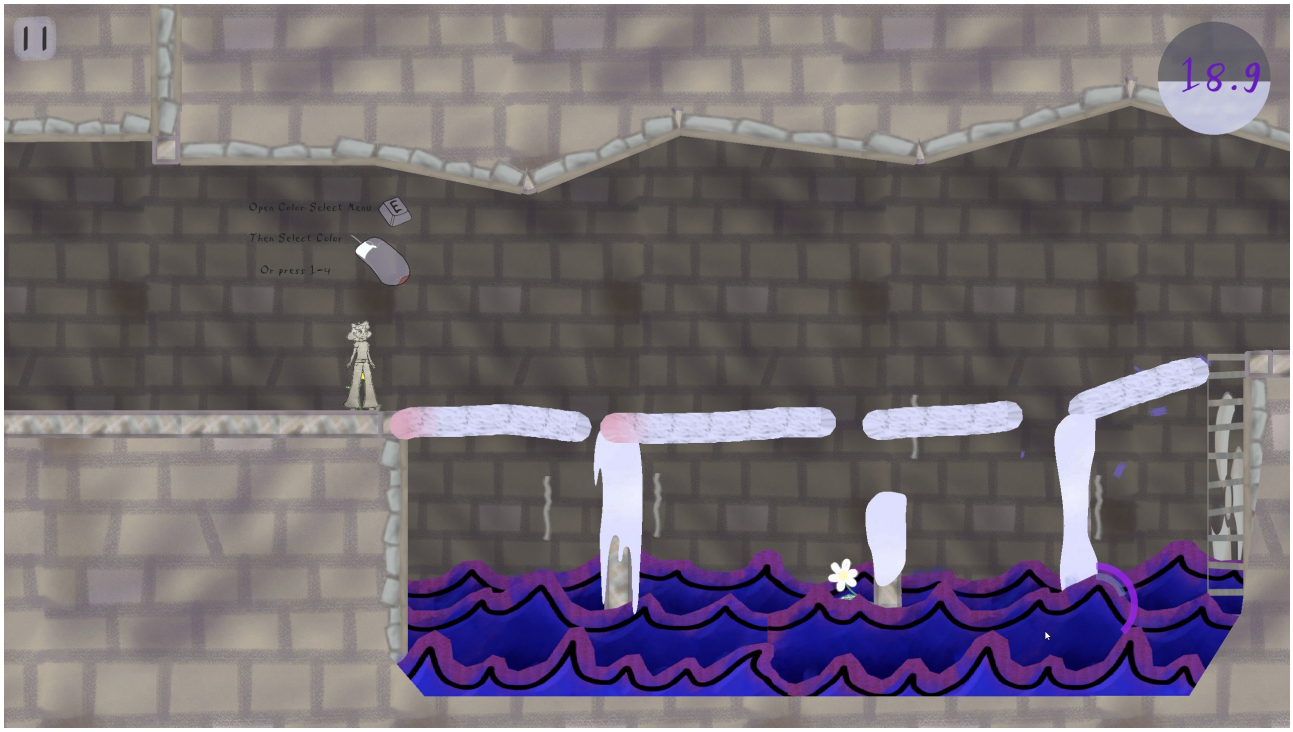

Figure 4: The room required multiple purple trails

Also, the room is kind of boring since we already showed the mechanics once right in the previous room. And in the upcoming rooms we never touch that mechanics again. (tbh my original design of them room is that the wind is blowing at the opposite direction, so it is just a room for storytelling)

The difficulty of rooms

Let's say, for every room, there is a distribution on the probability player can figure out the solution immediately, ranging from 30% to 90%. But it is a question to me, as a level designer, that why that is the case.

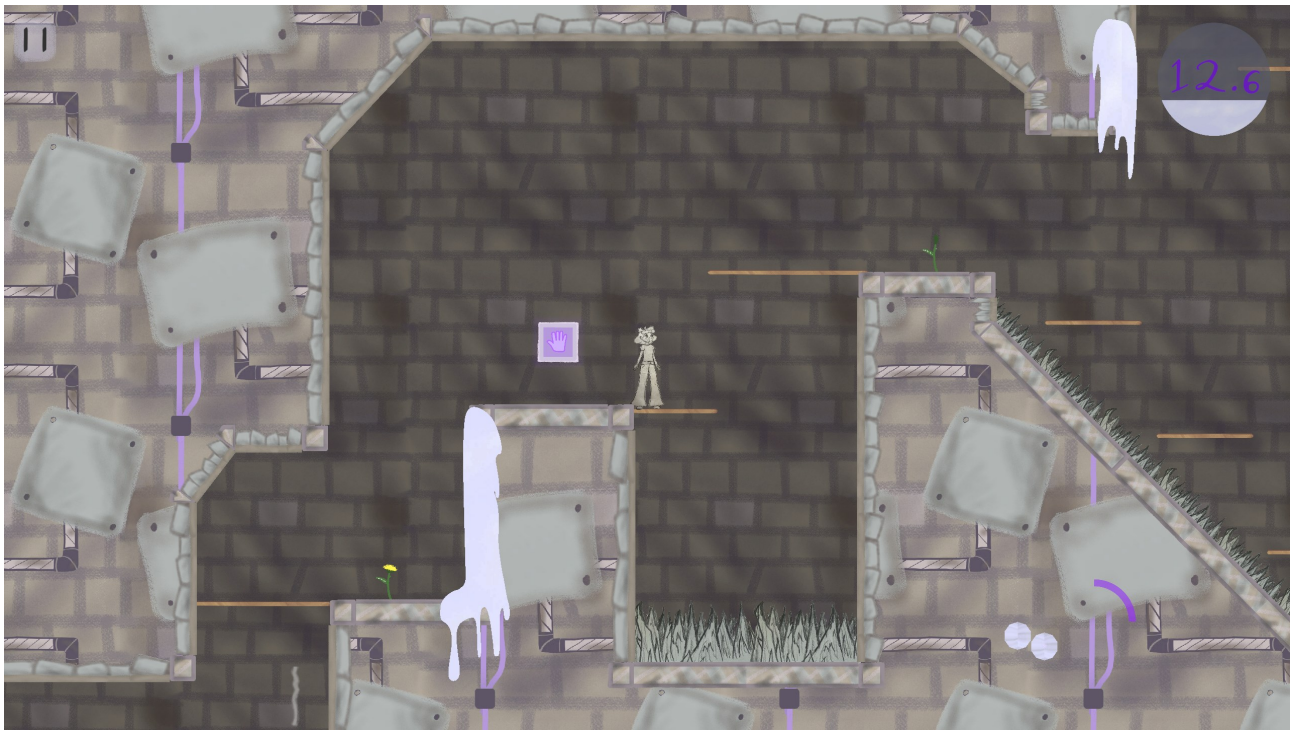

Figure 5: About 50% player immediately know how to get through

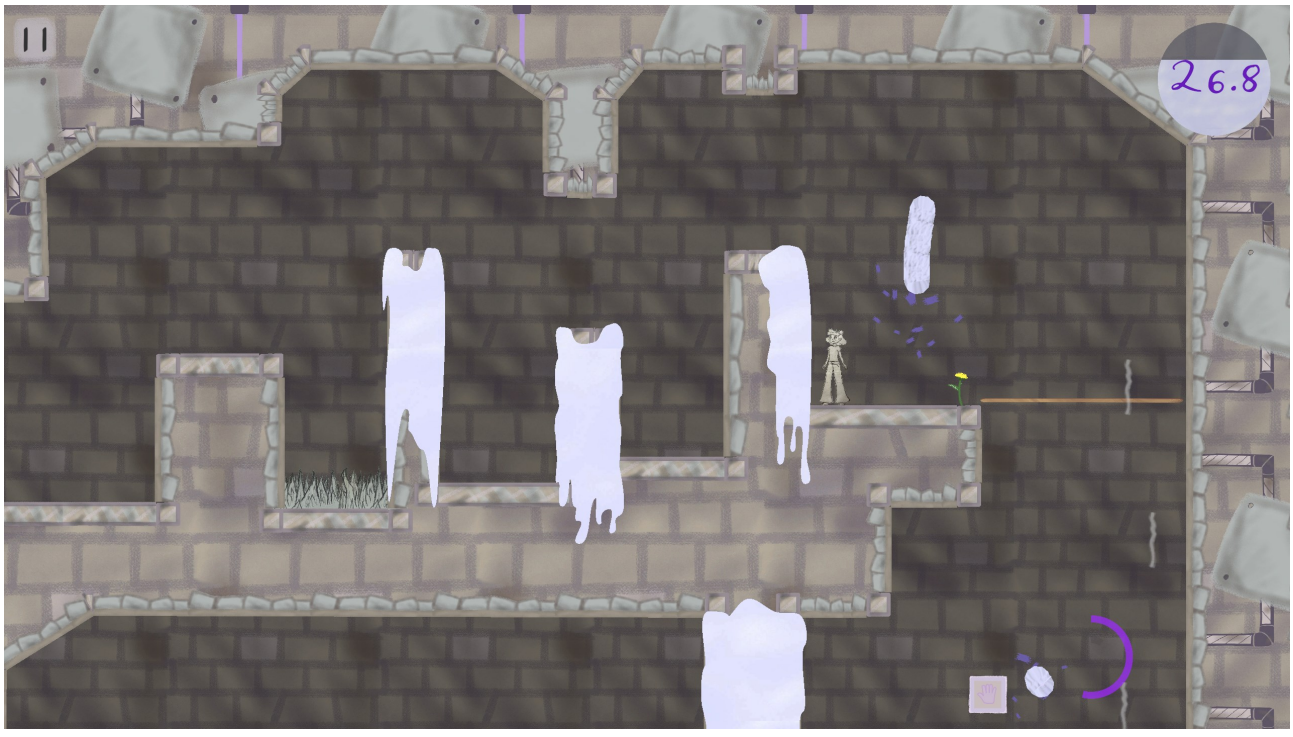

Figure 6: About 90% player immediately know how to get through

Unintentional but good design

One positive thing that I found was how players are playing the end of purple level very intuitively. There was a room where the player needed to open 4 switches in order to get passed.

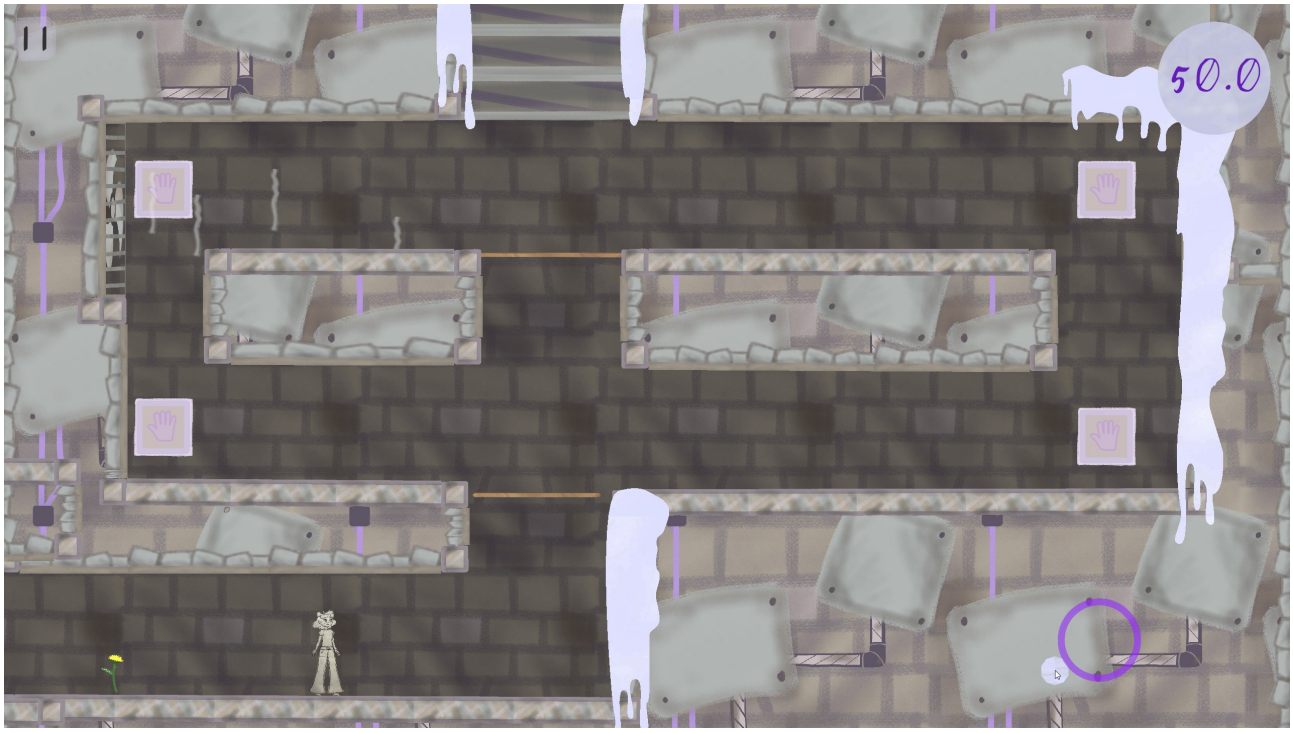

Figure 7: An unintentional good design

In the room, the players might need to go through the right path twice. Some players easily passed it at the first try, but then failed to do it for the second time since they were a bit nervous by the chasing purple shadow. It is an interesting design that players know they can pass (the room provides them confidence), but they need to beat the sense of fear themselves in order to get through.

It's like a perfect balance between players being confident, well perceived the intended path, but still getting some level of difficulty (i.e. the difficulty does not come from the weird geometry.)

Additional Remarks on the purple level

I found most players tried to block the wind and the purple clone by drawing a trail. As it is impossible to do so in the current version, they sort of overestimate our development ability (kind of sad / funny). The purple clone is really just a clone of the player $2.5s$ before, and the wind is just an area effector.

But in our original plan, we do want objects to interact with trails. For example, icicles can break the trails if we toggle an option, and purple trails would make crates sticking to whatever connected by the purple trail. We did not implement them mainly due to the time constraint and afraid of being Warrio-like. (The time and effort of combining mechanics grow exponentially.)

The red level

The red level is done right before the gold submission, non of us (group members) tested it.

But generally, I think the biggest issue is the nondeterministic hazard upon respawn. i.e., every time the player respawn, the order and timing of the hazards appear to be different. So in certain cases, there is a softlock in a way that the hazard sequence makes the level unbeatable (we knew this, but the players were keep trying).

Other notes on playtesting

In the playtest, when I found early parts that are not very reasonable, I was really sad and afraid.

Since we designed the levels in the same order players play it (i.e. we made black first, then blue, then purple, ...), The earlier levels were confusing, while the latter levels are better (while the player is learning deeper about the mechanics, the levels themselves at the beginning are more confusing).

So I was afraid that the player leave the game just because the first few rooms are a bit confusing. That is absolutely something to improve for the upcoming projects (more willing to modify the levels we thought are done).