



Game
Programming
Laboratory

Hansel & Gretel

and the magic of color

Josua Cantieni - Producer, Programmer, Game Designer, Sound Designer
Martina Kessler - Programmer, Game Designer, Visual Artist
Viturin Züst - Programmer, Game Designer, Quality Assurance

ETH

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich



computer graphics laboratory

Contents

1	Formal Project Proposal	1
1.1	Game Description	1
1.1.1	Overview	1
1.1.2	Background Story	1
1.1.3	Design Decisions	1
1.2	"Big Idea" Bullseye	4
1.3	Technical Achievement	4
1.4	Development Schedule	5
1.4.1	Layered Task Breakdown	5
1.4.2	Task List	7
1.4.3	Timeline	7
1.5	Assessment	8
2	Prototype	11
2.1	Prototype Setup	11
2.2	Playing Experience	13
2.3	Findings and Conclusion	14
3	Interim Report	15
3.1	Progress	15
3.1.1	What happened	15
3.1.2	Current state	16
3.2	Challenges	16
3.3	Future Work	18
4	Alpha Release	19
4.1	Progress	19
4.2	Challenges	22

Contents

4.3	Future Work	23
5	Playtest	25
5.1	Playtesting Session	25
5.2	Questions and Comments	25
5.3	Design Revisions	27
6	Conclusion	29
6.1	Final Results	29
6.2	Experience	30
6.3	Personal Impressions	31
6.3.1	Josua	31
6.3.2	Martina	33
6.3.3	Viturin	33

Formal Project Proposal

1.1 Game Description

1.1.1 Overview

We want to create a 2D puzzle platform game for two players. The special thing about our game is that there are hidden platforms of a certain color that can only be made visible with a light of the same color as the platform. For some platforms the light color needs to be mixed with two lamps of different colors. Each player can only hold one lamp, therefore the players need to communicate and cooperate in order to solve the puzzles.

Our game plays in a dark forest and the main goal is to solve the puzzles in order to find the way out. The players only have limited time to achieve this goal.

1.1.2 Background Story

Hansel and Gretel were abandoned by their parents in the dark forest and need to find their way out. They have colored lamps that help them see in the dark. Those magical lamps unveil objects of the same color if the lamp is within reach of those objects. Together the players need to mix the colors of the lamps to unveil objects of mixed colors. They need to find their way out of the forest before time is up because the lamps only have limited fuel.

1.1.3 Design Decisions

Our goal is to create a cooperative game because we think that it is much more fun to win a game together than to play against each other. The idea with the lamps serves this purpose

1 Formal Project Proposal

because they are inherently cooperative if we allow each player to carry only one lamp at a time. To create mixed colors the players need to work together. The color mixing aspect is also instructive because if the players do not know yet how to mix light colors, they need to learn it in order to succeed in the game.

We decided to create a game for two players because mixing colors with more than 2 lamps might become difficult for the players and might make them feel frustrated.

Both players will play on one computer using a shared screen. We prefer a shared screen over a split screen because we want that both players have the same information available.

Most platform games are 2D, therefore we decided to follow this path and make our game 2D as well. One advantage is that with 2D we hope not to have big performance issues. If we created a 3D game it might also be difficult to find a good viewing position for the camera such that both players can see what is going on in the virtual world. We also think that a 2D game focuses more on our main aspect: the color mixing to unveil hidden platforms.

The visual design of the game will be rather simple in order to focus on the lights and the color mixing. Our game relates to the course theme "Grimms' fairy tales" because we motivate our game with the story of Hansel and Gretel. The children need to find the way out of the dark forest.

We introduce a time component in the game in order to make the game more interesting because the challenges need to be solved within limited time. This tests the players ability to communicate in a stressful situation.

Our goal is to support both keyboard and Xbox controllers. We want to support keyboard because we do not want to exclude players that do not have Xbox controllers. However, we personally think that the game will be more fun to play with controllers, therefore we want to support controllers as well. At the moment we think that we only use move, jump and interact inputs.

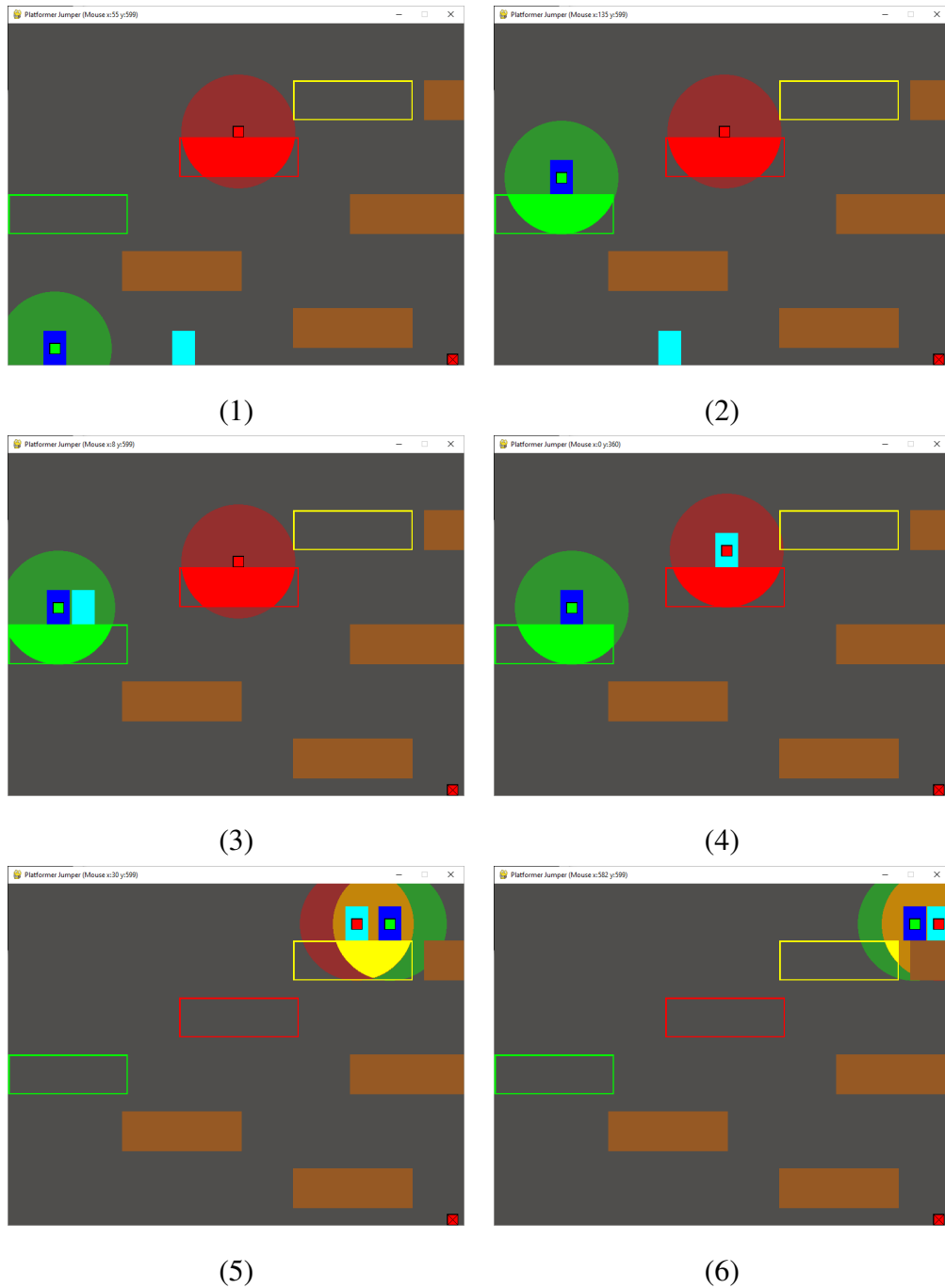
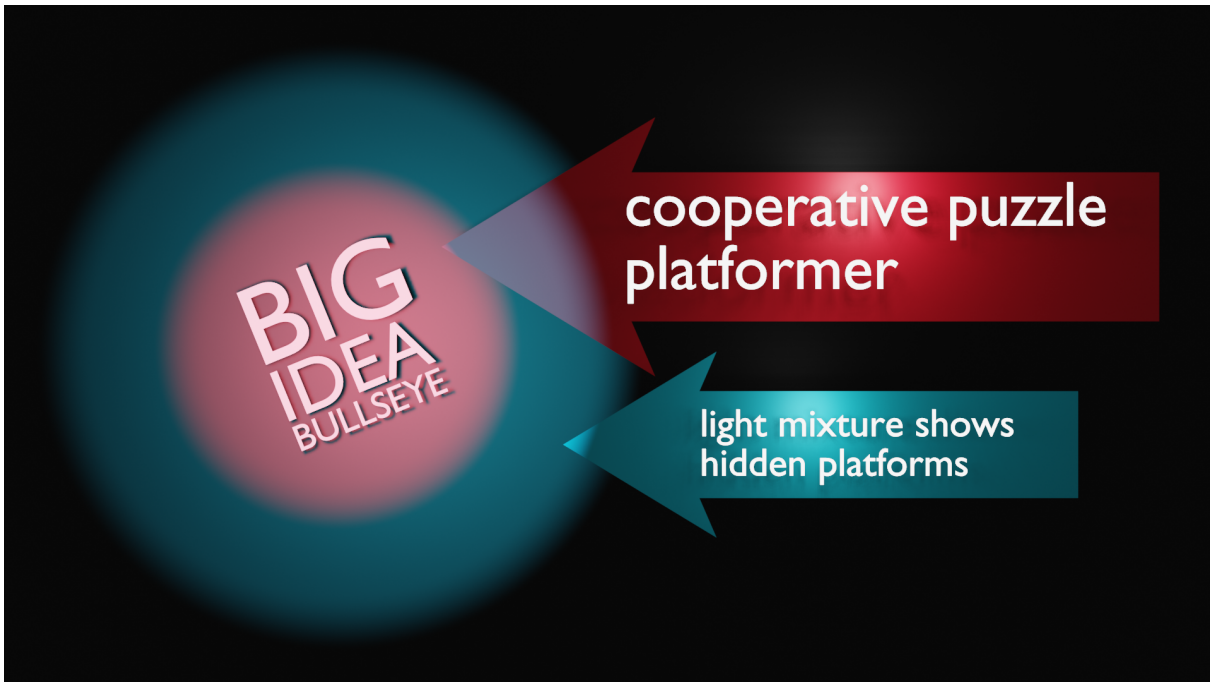


Figure 1.1: Here is a demonstration of how the game mechanic works: In screenshot (1) player 1 picks up the green lamp. Then the player jumps onto the green platform in screenshot (2). He can only stand on this platform because of the green lamp he is holding. Without it the platform would be invisible for the player and he would fall through it. In screenshot (3) player 2 also steps onto the green platform. He then continues and jumps onto the red platform. He can step onto it because there is a red lamp located on the platform that makes it visible. He picks up this red lamp, which can be seen in screenshot (4). In screenshot (5) both players jump onto the yellow platform. The yellow platform is only visible because of light mixing. In screenshot (6) it can be seen that both players reached the goal of this platform game.

1.2 "Big Idea" Bullseye



Our game requires two players to work together. This necessitates clear communication between the players and forces the players to reach a common goal. Without cooperation, it is impossible to win this game.

Secondly, our game uses special lamps with different colors, which unveil platforms of the same color. The players can grab different lamps in their environment to solve a given puzzle. The players can combine lamps of different colors to create a new color; this color mixing enforces our game's cooperative element. It furthermore forces the players to communicate clearly what they want to do with a given lamp.

1.3 Technical Achievement

Our game's primary technical achievement is using different colored lamps, which have to be combined to reach a puzzle's goal. More concretely, the lamps make some platforms of the platformer appear. For some platforms, it is required to mix different colors. This allows us to create levels that force player cooperation and communication because multiple lamps are needed simultaneously. The lamps will be rendered attractively by rendering the shadows cast by the platforms around the player.

To create interesting experiences in such an environment, we need to create and test levels efficiently; thus, we will create a level editor. This level editor will not be available to the player; it only allows us to prototype quickly.

1.4 Development Schedule

1.4.1 Layered Task Breakdown

Here we break down our different milestones for our game.

Functional Minimum

- Level should have
 - Platforms
 - Lamps
 - Two Players
 - Platforms that are only walkable by using a lamp in the right color
 - Changing colors of lamps
- Color mixing of lamps
- Two Players. Use arrow keys and WASD to control them.
- Collision between player and platforms
- Very simple graphics. Everything is represented by simple geometric primitives and colors
- Simple jump physics
- Add a simple, hardcoded test-level to test our mechanics

Low Target

- Level should have
 - Puzzle Elements (levers, doors)
 - Background-image
 - Sound effects
 - No need to have final Graphics but ok looking temporary graphics or sound
 - Death pits
- A Start-Menu UI Screen
- Pause-Menu and/or HUD during Gameplay
- Add a time limit. For example the light radius of lamps is decreasing over time.
- Add controller-support for playing

1 Formal Project Proposal

- Player can select between keyboard or controller
- Replace the players with simple characters with animations

Desired Target

- Level should have
 - Background-blocks
 - Background music
- Represent a level as data in a file
- Create a level editor
- Chaining rooms to a level
- Lamp light is occluded by platforms
- Final graphics. Add Texture to platforms, background, and finalize characters
- Design levels for the game. The game should start with easy levels explaining the game mechanic and get gradually harder. If possible, we try to avoid having a explicit tutorial but let the level explain the mechanic itself.
- Incorporate the story. It will be simple to explain the player what the setting is.
- Make the menu and level select screen look nice
- Add settings (for example for audio)
- Add some royalty-free background music
- Add Save and load feature for finished levels

High Target

- Level should have
 - Collectables
 - Background-sprites and more decoration to level
 - Add background animations
- Further expand the level capabilities, for example add level scripting
- Music should support the time pressure, for example by getting faster or more intense when there is less time remaining
- Create different levels with different themes

Extras

- Create an original soundtrack (or at least some tracks)
- Make the game playable for color-blind people
- Polish the level editor to allow players to create levels
- Allow players to unlock and select characters/skins

1.4.2 Task List

When 'All' is in the 'Who'-column the time for each member is the time divided by 3. We also plan to have two meetings of two hours a week.

Task	Description	Who	Time (h)	Actual
1	Look into Monogame	All	12	11
2	Discussing Architecture	All	24	24
3	Set up architecture skeleton	J	8	6
4	Setup window to draw in	J	4	2
5	Draw Level and Players	M	8	6
6	Process Input	V	8	6
7	Jump-Physics and Collision	V, J	24	20
8	Lamps with color mixing	M	16	10
9	Draw Color Platforms	M	6	15
10	Collision Color Platforms	V, J	6	5
11	Color Changing Mechanism	V	6	5

Table 1.1: Task list for Functional Minimum

1.4.3 Timeline

Important Deadlines The Gantt-chart sometimes shows dependencies for a task that overlaps. This means that it is possible to start the task before the dependency is done, but at some point the dependency is needed.

- 23.03.21 Prototype
- 29.03.21 First Playable Demo
- 26.04.21 Interim-Report and Interim-Demo

1 Formal Project Proposal

Task	Description	Who	Time (h)	Actual
12	Discuss Art Style	All	12	9
13	UI Design wireframes	Josua	12	20
14	Add Controller Support	V	6	5
15	Implement Menu UI	J	4	
16	Implement Control selection	J	8	3
17	Implement Pause UI	J	6	
18	Time limit	All, V	24	18
19	Render Background image	M	2	4
20	Render Block Texture	M	12	12
21	Play Sound effect	J	10	
22	Death Pits and Spikes	V	6	4
23	Lever-Door Puzzle Element	V	12	14
24	Add Animation to players	M	20	21

Table 1.2: Task list for Low Target

- 10.05.21 Alpha Release
- 17.05.21 Playtest
- 31.05.21 Final Version
- 07.06.21 Report due and Video

1.5 Assessment

Our game will be strong in pushing players to work together on a common goal. The experience of achieving a goal together with a friend is much more glorious than achieving it on your own. The experience of playing our game will strengthen friendships, and it will improve their communication skill—all of this in an enjoyable, compelling, and intuitive game. The two players will be traveling together in a fairy-tale world with the help of colorful lamps. We want the players to learn about the mixing of light and have an "Aha! moment" when the mixing results in the right color.

Task	Description	Who	Time (h)	Actual
25	Room winning condition	V	6	4
26	Chaining rooms to a level	V	6	3
27	Background-block layer	M	10	
28	De-/serialize Levels	V, J	24	20
29	Implement Level Editor	All	40	32
30	Create some puzzles	All	24	
31	Lamp shadow on platforms	M	12	23
32	Find final graphics	M	20	1
33	Characters with animations	M	24	
34	Tell Story	V	10	
35	Background music	J	12	
36	Settings UI	J	4	
37	Save and load	J	8	
38	Finalize UI	J	8	
39	Finish levels for final Game	All	32	

Table 1.3: Task list for Desired Target

Task	Description	Who	Time (h)	Actual
40	Add Collectables	V	16	
41	Level decorations	M	20	
42	Level polishing	All	20	
43	Scripting	V, J	24	
44	Sound and music polishing	J	24	
45	Level specific special element	V, J	24	

Table 1.4: Task list for High Target

1 Formal Project Proposal

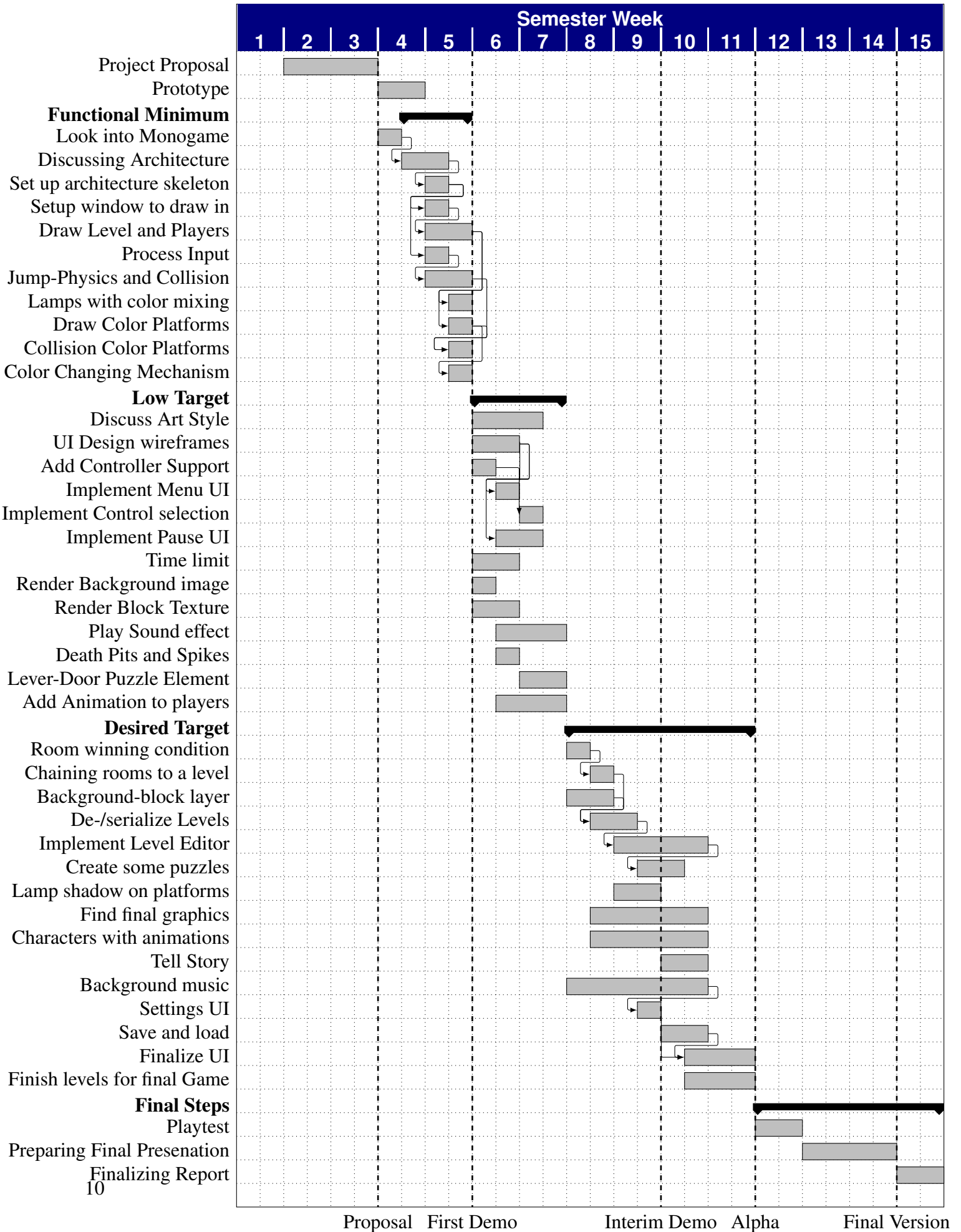


Figure 1.2: Gantt Chart

Prototype

In this chapter we describe our experience with the physical prototype we built. We also built a digital prototype of which you can see some screenshots in figure 1.1, but we focus on the physical prototype in this chapter.

2.1 Prototype Setup

We created the physical prototype with different colored paper, which allowed us to test the different colored platforms. Using six different colors, we were also able to test the color mixing process. We furthermore created the color-changing stations to test different colors during one game and tested different lever mechanics. The color changing stations can be seen in figure 2.1

One of us set up a game level on a table using small squares of different colored paper. The level was shared between the teammates using an overhead camera and Zoom. Two players were able to play Hänsel & Gretel, and one of us updated the game state. The different color of a player was kept track of by a colored Venn-Diagram and two markers. Using this the players could also look up the colors that are required for mixing although we do not plan to include this in the final game. The whole setup as seen by the players is in figure 2.2

With the physical prototype, we tested different lamp radii. For this, multiple different foil circles were created.

Given this level setup, we played two game modes: The first was using action cards in which the two players unveiled their next action at the same time. This setup resulted in a block-by-block playthrough of the game. The second variant used a more open approach by giving different actions different difficulties and dices decided if the player fails or succeeds. In this play mode, the players can describe their next action using normal language, similar to Dungeons and

2 Prototype



Figure 2.1: The small triangles with the black base are stations which can be used by the player to change the lamp color.

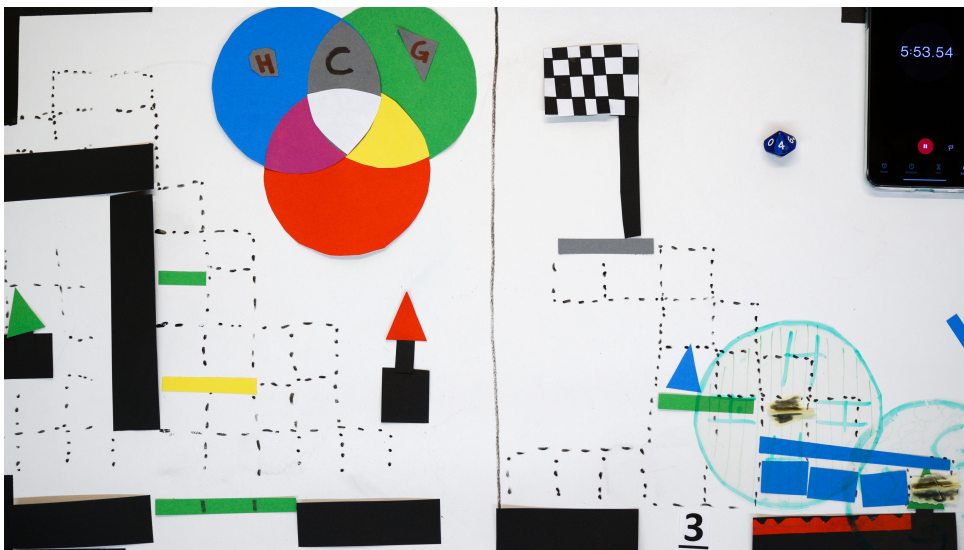


Figure 2.2: We see the two players and their lamp radii in the bottom right. The current lamp color of each player can be seen in the Venn-Diagram at the top. The part of the blue platform on which Hänsel (the left player) is staying is active because Hänsel has the right color, Gretel on the other hand, has the wrong color and is not close enough to Hänsel to stand on the blue platform. Furthermore, we see that the level is nearly solved, and the two players have the right colors available to go to the finish.

Dragons. In this game mode, the game master enforced a time limit. The whole setup in action can be seen in figure 2.3



Figure 2.3: The whole setup as seen by the game master

2.2 Playing Experience

The action cards playing mode was very slow, because the players had to decide which action they wanted to take for every step, then communicate their decision via Zoom. One action card we had was an arrow that indicated into which direction we wanted to go. Due to the mirroring of the camera in Zoom it was a challenge to point the arrow into the direction such that the game master could interpret our action correctly. On the other hand, this playing mode helped us to observe the mechanics of our game and to detect possible problems with it. Another advantage of this playing mode was that the players clearly communicate which action they want to take, and if they make a mistake, they will fail. So there was no need for a probabilistic component modeling the difficulty of an action, because the difficulty lied in making the correct decision for every step.

The other playing mode in which we described which actions to take using normal language was more interesting to play as a board game due to the dices deciding whether an action succeeded or failed. But if we failed, it did not make much sense to start over because we already knew that we would take the same decisions again and the only reason we failed was because of the dice. So we just did not restart the game after a failure, but still it felt like we did not succeed, which modeled the difficulty of the game.

2.3 Findings and Conclusion

As the first thing we noticed that creating a level that is solvable is not that easy. Sometimes after the creator of the level showed the level to the players one player pointed out that it is not solvable, sometimes it took some time in the level. This aside, we got some interesting ideas for levels that we could test really quick. We think that we could also use this to test a level mechanic before we have the level editor in place.

We noticed that having levers that cause an (probably unforeseen) action to happen can make the levels more interesting. For example the levers could change the color of a color changing station which allows the player to pass a pit. We plan to have a save zone in the level where the player can discuss as much as they like to come up with a strategy to solve the level. When they leave the zone the time limit starts. We noticed that for levels where nothing unexpected happens the players can discuss the strategy in advance and afterward not much communication is needed anymore. The levers can add an element that forces the players to discuss the changes under time pressure. We decided to remove the safe-zone in later harder levels which forces the players to think and communicate fast.

The digital prototype we made showed us that the concept with the lamp and color mixing by itself works and is fun to play.

Interim Report

3.1 Progress

3.1.1 What happened

We have worked a lot on our game, and in the meantime, we have finished the functional minimum and most of the tasks of the low target. In the process we renamed our game from "*Super Hänsel & Gretel*" to "**Hansel & Gretel: Together through the darkness**". We have had long discussions about the art style and which graphics we wanted to use. Finally, we found nice graphics that support the atmosphere of a dark forest where Hansel and Gretel were abandoned.

Josua managed to ask his sister to do the characters for our game. So currently we still have the temporary characters which will be replaced for the final game.

We have not yet finished the implementation of the menu and pause UI. We realized that it is not possible to design the UI without having decided on which graphics we wanted to use because we want the UI design to match with the graphics of the game. Therefore we waited with the design of the UI elements until our decision about which graphics to use was made. This is finished now, and we can now continue with the creation of the UI. Most of the UI screens have a rough draft already, including a logo for our title screen (you can see it on the title page).

We have not yet started searching for good sound effects because we prioritized the other tasks in the low target.

Regarding the game engine and the game logic, we have reached all the tasks of the low target. Furthermore, the serialization and the deserialization of levels and rooms are implemented. A very rudimentary level editor is available to ease us with the design of new levels and puzzles. We are happy that we waited until now to start with implementing a level editor because many

things changed in the representation of our levels and rooms during the development stage. If we had started any earlier, we would have created more work for us.

3.1.2 Current state

Here are some screenshots of how our game looks at the moment. *Please note that the characters are not final right now and still placeholder.*



Figure 3.1: The basic mechanic is the color platform. Here Hansel stands on the red platform with the red lantern color. This allows him to be able to stand on the platform.

3.2 Challenges

The discussion and set up of the software architecture went really well. Until now there was no need for a major refactoring of the code. We are really happy about this because it was our goal to create a stable software architecture where new features can easily be incorporated.

A challenge was the implementation of a physics engine from scratch. We decided to implement from scratch to ease the implementation of the colored platform mechanics.

A lot of time went into discussing the graphics we wanted to use. None of us is a designer so we first had to figure out how to discuss this topic and how to find good resources. We ended up buying a bundle from itch.io and another background, which suited the style we imagined for our game. We still had to design stuff ourselves, which, again, took quite some time. Designing stuff is hard.



Figure 3.2: In this image Hansel has the blue color and Gretel has the red color. When mixed together a magenta color is created. This allows the player to stand on the magenta platform.



Figure 3.3: In this image Gretel is imprisoned due to a red platform blocking her way. As Hansel has the blue color, he can pass the red platform and reach the lever above. Using the lever Gretel is set free and they can continue their journey.

3 Interim Report

As it took such a long time to get the final graphics we could not really design the UI properly. We learned that as we still tried to design the UI but noticed, that to make design fitting the gameplay you should know how the game will look like.

Originally we planned to automatically render the borders of blocks. For example the code would detect if a block is in a corner and render the appropriate texture. But due to the huge amount of possibilities we gave up on this idea and now place the tiles manually. The editor provides this functionality.

3.3 Future Work

We are working on the improvement of the game rendering such that the atmosphere in the dark forest is nicely visualized and the lights look attractive.

We still have to improve the level editor majorly and with it create interesting levels for our game. It should also be possible to tell a story with our game.

As we mentioned in chapter 3.1 the UI already has some rough drafts so we start from there implementing the screens.

The sound effects should be implemented right after the UI is done. There should be sound effects for walking, jumping and so on and also sounds in the menu when selecting things. For background music we will search for some royalty free music online.

One big thing we still have to do is find out how we communicate to the player that the time for the specific room is running out. We don't want a timer in the screen to maintain the immersion so we have to think of a good idea.

Alpha Release

4.1 Progress

We improved our game a lot in the last few weeks. One thing that can be noticed right away is that we incorporated the characters created by Josua's sister Flurina Cantieni. We think that those characters fit much better into the game than the placeholder characters used before.

The color platforms are now rendered more accurately. Before, the whole block was rendered solid when at least one pixel of the block was within the lamp radius. We have improved this such that only the part of the block that is within the lamp radius is rendered solid. There were also some advancements in the light rendering. The light of the color flames flickers now smoothly.

We agreed on a way to communicate the time limit to the player without weakening the immersion in the game. The brightness of the lamps of Hansel and Gretel decreases linearly with time. When there is only a certain amount of time left, currently 30 seconds, the lamps start to flicker and the flickering increases as time advances. When time is up, the lamps go out, the characters die and the game is reset.

The editor has been improved a lot. Now there are GUI elements to help and speed up the creation of rooms. There is now a side panel with multiple tabs to work with.

In one tab are the settings for the room that is edited. For example the time that is allowed for this room can be set. Also there are different help utilities here for example to deactivate the lamps and shading and disable the foreground to make work on the background easier.

Then there is the tab about the tilesets. While earlier you had to scroll your way through all the many tile variations we had until you found the one you were looking for, you now can just click on the tile to select it.

4 Alpha Release



Figure 4.1: Solid color platform rendering. Only the parts of the block that are within the lamp radius of Gretel are rendered as solid red blocks, the rest of the platform is rendered not solid.

Finally there is the level tab where you can load different rooms in a chain to test the room transition.

There is also a tab that appears when an entity is selected which shows some information about the entity.

In the level editor you can also Zoom in to see if your tiles are set correctly. Also there is an autosave and tons of warnings if you have unsaved changes to help us to save our rooms and not lose any progress.

The biggest selling point of our editor though is the live test feature. Hitting the Enter key runs the room you just created in engine and you can test if the level you made actually works with ease.

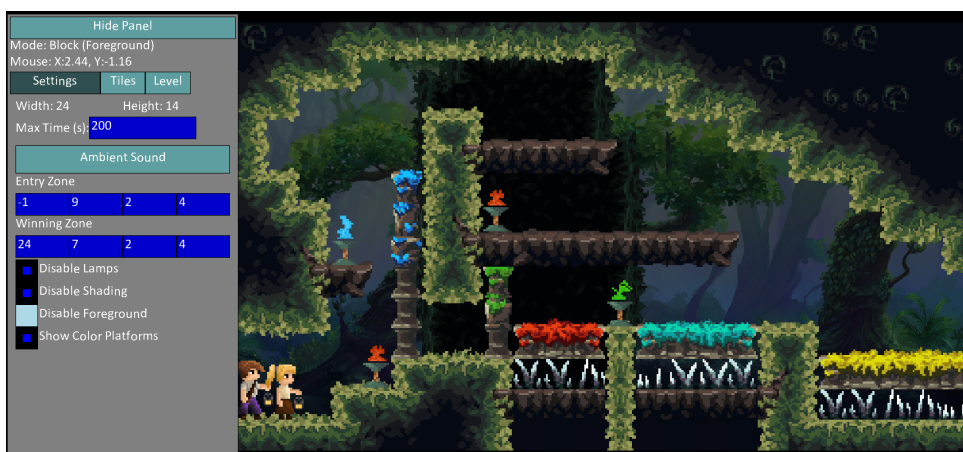


Figure 4.2: In this part of the editor, we can set the different settings of a given room. For example, we can set the winning and spawning zones. Furthermore, the rendering setting of the level can be adjusted; this allows us to see the changes we make.



Figure 4.3: We can select different tiles directly from the menu.

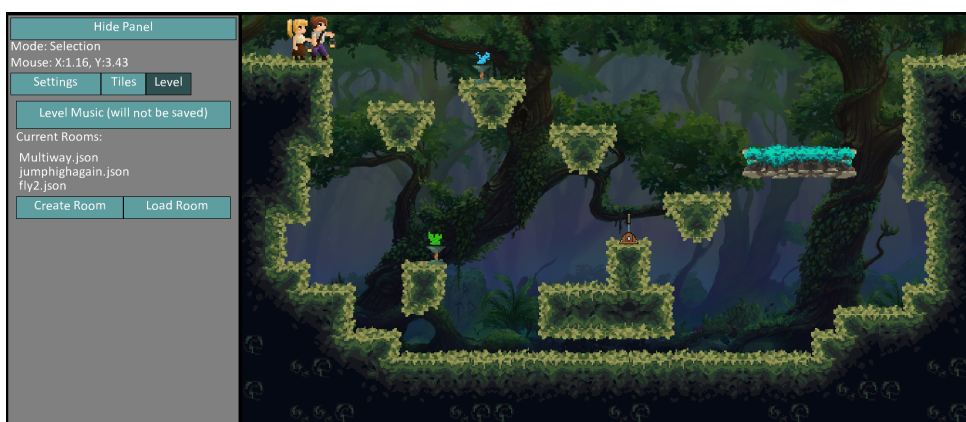


Figure 4.4: Chaining multiple rooms together in a level can also be tested from this menu in the level editor.

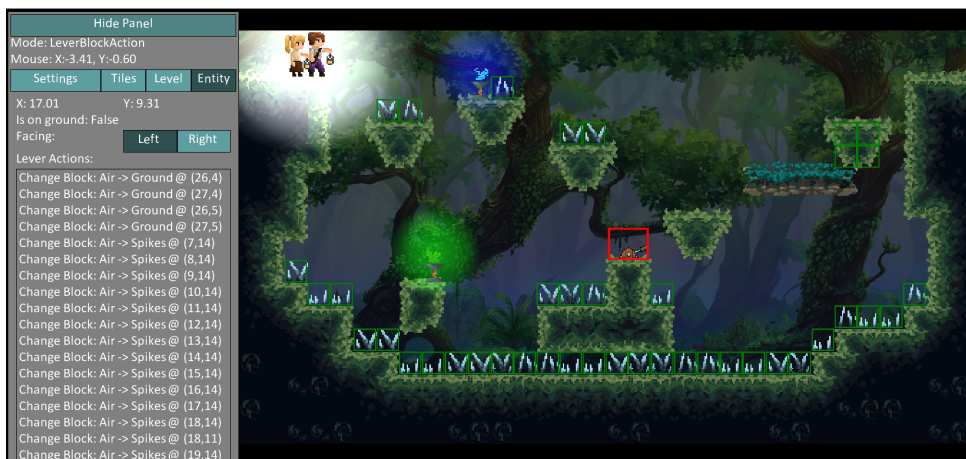


Figure 4.5: We can inspect and set the lever programming directly from the editor.

We finally added sound effects. It was a chore to integrate the sounds into our architecture, but read more about that in the next section. We use the sound engine FMOD to drive our sound system. To integrate FMOD we use the package ChaiFoxes.FMODAudio.Desktop, which aims at using FMOD crossplatform easier. Using FMOD has the advantages that editing the sound

and using runtime effects is relatively easy. Instead of implementing sound cues that randomly play one sound of a list ourselves we can just use the functionality of FMOD. To play a sound we just have to call the appropriate event of FMOD with the right parameters and we can hear something. Also FMOD enables us to test and change the sounds at runtime using Live Update.

The physics engine needed finetuning in a lot of places. Mainly we had to adjust how deadly spikes are handled to represent better what the player is seeing. Furthermore, the entry and winning zones on the side of the rooms are now calculated to allow us to place them where ever we want.

4.2 Challenges

In the beginning where we planned our architecture we did not account for playing sounds at all. This made it difficult to find a way to integrate sound properly. There are different types of sound triggers that needed to be integrated. Some sounds just play when an event happens, some on a specific frame of an animation, some play all the time and some only when an entity is in a specific state. The design we ended up with is probably not the cleanest, but it fits into our architecture, although there are now a lot of hardcoded events that cause sound.

To play sounds we saw two options: using MonoGame internals and code a lot of datastructures ourselves or use a sound engine like FMOD. Using FMOD had the disadvantage that we would need to learn how the tool works and how to integrate it into our game. But coding the datastructure ourselves seemed tedious and in the end we probably would integrate similar features as FMOD would provide us. After some testing and watching a stream VOD of how FMOD was used in the game Celeste, we decided to use FMOD. We found a nuget package that helps integrate FMOD but the library sent the wrong version to the FMOD library which caused some confusion but we managed to work around that. In the end we are glad, that we chose FMOD as sound design is so much easier now.

The only thing that is not easy about the sound design now is finding the appropriate sounds. How do you search for that? How do you describe a sound that has no name? Nevertheless we ended up with something and have almost all sounds we need. We just need to agree on a background music which our team-members have different expectations of.

We noticed that level design takes a lot of time and a good level editor is crucial to help us with this task. Therefore, we added much more functionality to it than initially planned. All the functionality is described in the previous paragraph. Even with our improved editor, it is still cumbersome to find the fitting tile manually. An automatic way to determine which tile to use would be really helpful. By far the most useful tool is that we can directly start the game within the editor and try out the levels. This is what makes our editor so special, because we don't know any open source editor that provides this feature.

4.3 Future Work

One big task that still remains is the UI. We made some improvements but there are still some screens missing (for example the settings and the credits).

There is also a lot of polishing to do that we will base on our play-testing experiences. We will need to polish our levels and the controls. There are probably more things that we hope to discover in the playtesting sessions. Also we plan to decorate our levels.

We may also need to change or add some sounds.

Playtest

5.1 Playtesting Session

We had different types of playtesting sessions. One type was to let our friends and family play. We tried to have teams of two such that we could observe them. In those sessions we usually let them play through the entire game and afterwards we asked them our questions. In some cases we had to play as the second player as there was no one to fill the team, but we noticed that this type of playtesting is not optimal. The reason for this is that we tried not to give any hints such that we could observe how well the tutorial levels explained the game mechanics to them. But the interaction with the second player was clearly missing and we think that this interaction is crucial to the game play.

Lastly Josua had the chance to go to the school where his father works to let students play and test the game. The students were around the fourth to sixth grade. In this playtesting session we had half an hour for each team which resulted in letting them play for 20 to 25 Minutes and 5 to 10 Minutes of asking questions. The kids did not know in advance what kind of game it was except that they had to work together.

5.2 Questions and Comments

We asked our participants at least the following questions:

- What is your gaming experience on PC or Console?
- What was your impression of the game?
- How well did the communication with the other player go?

5 Playtest

- How did the controls feel?
- Did you always know where your character was located on the screen?
- Did you always know which lamp color your character had?
- What did you like best?
- Would you change anything?
- Did you miss anything?
- Would you buy and play this game?

The game was received overwhelmingly well and everybody seemed to like it. Some kids even asked when the game would release on Steam and could not stop playing. Multiple participants mentioned the teamwork and the color mixing as one thing they liked best, which is great to hear. At least for the kids, everyone would buy and play the game. It certainly exceeded our expectations of how well the game would be received.

Amongst this we received also other positive feedback:

- Nice rendering of the lights
- Nice sound effects for the color platforms
- Nice background music. One player mentioned that they probably would have raged if the music was not that calming.
- Fun game, it is not frustrating
- Nice that it is a cooperative game
- Nice death animation

On the gaming experience side, it was interesting to see how differently they played and that also the players with no experiences were able to play our game.

It was also interesting to observe how the player interacted with each other. When dying some players were laughing and some were annoyed. It is definitely a game that requires patience from the more skilled player but that is also a good team building exercise.

One negative point was that the jumping does not feel very natural because you need to remain on the button/ key to jump also when falling down, but when you land on a platform and do not release the button/ key, you immediately jump again. This was perceived as not optimal and it was especially annoying when landing simultaneously on a mixed color platform. Quite often one player unintentionally jumped again and both players fell down.

During the session, there were multiple comments and questions from the testers. Viturin's testers tried to break the game in some way or another. On one level, in particular, they pushed for over an hour to find another way to the exit, then the one that was intended. Thanks to their scrutiny, we were able to fix multiple rooms and adjust some puzzles.

5.3 Design Revisions

Initially, we planned to have a time limit because we thought that this was needed to motivate the players to solve the puzzles, but we found out that the opposite was true: It demotivated the players that they had limited time because our levels were challenging enough. So we removed the time limit and we now focus on creating interesting and challenging levels that motivate the players to play our game.

Our central part of the game is the use of light colors, but this central part was not reflected in our title "Hansel & Gretel - together through the darkness". One of our playtesters noted this and suggested to change the title in order to include the word "color". We had a short brainstorming session and decided to rename our game to "Hansel & Gretel and the magic of color". We think that this title reflects the mood of our game very well and it summarizes our central game mechanic.

With observing how the player play our game we could find weak points in our level design. For example, there were certain rooms where a jump was a bit difficult due to the player bonking their head and thus not making the jump. We removed some blocks to make those jumps easier. We also found that some rooms had flaws that the players could just jump over the obstacle and thus skip the room which we had to fix. Then we noticed that some mechanics we took for granted are actually not that easily understood, for example jumping simultaneously onto a platform. So we added some rooms and moved the rooms around a bit to explain the mechanics better. We also noticed that we needed a way to reset a room. Currently, you can press the key "O" on the keyboard to reset the game, but this is just temporary and there is no way to reset the game on the controller. We decided to refrain from introducing another button on the controller and we added spikes to all levels where a reset could be needed, such that the players can kill their characters with the spikes and start a room over if needed.

Then there were also details like that we changed the color of the first encounter with our color platform, as a lot of the people thought it was fire or lava and we wanted to prevent that confusion. Also we noticed that the players could jump out of the top of the rooms which was not intended and we had to fix this. And we adjusted the physics of the spikes as they were a bit funky at times. For example if you fell into spikes with high velocity you could survive the spikes.

A major problem we detected was that players were confused which color they had, when they were standing next to each other and were mixing a color. We fixed this by rendering the color into the lamp of the players character.

When the characters die, their horizontal velocity remained as before, which resulted in the characters slipping in a very strange way. This was noticed by many playtesters. We changed it such that the horizontal velocity is set to zero when one character dies, such that the characters can only fall down, but they can not move left or right after the death of one of the players.

We also need to adjust our controls a bit, as they did not feel completely right yet.

Conclusion

6.1 Final Results

One of the significant changes between the alpha release and the final release was the removal of the time limit. We did not create any levels which used the time limit in any way; hence we removed it.

We changed the UI in many places to allow for intuitive starting, loading, input selection, and character selection.



Figure 6.1: We worked a lot on the improvement of the UI. One new screen is the character select screen. The players can choose their input method and whether they want to play Hansel or Gretel.

We made further improvements to the editor. These improvements allow for the placement of dialog entities and the placement of characters on the ground.



Figure 6.2: We added a pause screen where the controls for each player are displayed. There are also options to resume, reset the room or return back to the main menu.



Figure 6.3: We added a dialog between Hansel and Gretel, which tells a shortened and adapted version of Grimm's fairytale "Hansel and Gretel". Josua's sister, Flurina Cantieni, also created this beautiful candy house for us. It illustrates the dialog visually.

After the feedback from the testers, we rearranged rooms to allow for a better difficulty increase. We also improved the sound effects, which resulted in a better immersion.

6.2 Experience

Overall, our initial design idea materialized pretty well into our final game. The only major change to our initial idea was the removal of the time limit. In hindsight we should have developed this idea more in order to make it work.

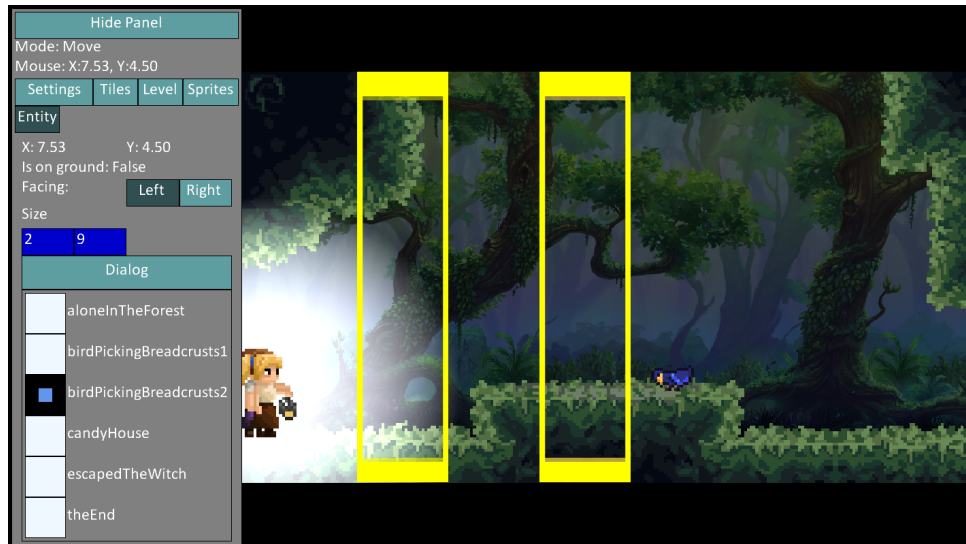


Figure 6.4: Dialog entities can be added in the editor.

Up to the beginning of the Low-Target we were pretty much on schedule. But we then noticed that we incorrectly estimated the priority for certain components. For example UI and Sounds should have been planned to do later in the process, although at the same time we should have accounted better for them at the beginning. So the schedule itself fell apart, but we still had a priority list we followed and thus managed to finish everything we planned.

The project structure was helpful to organize the whole project. It was nice to have some milestones to work to and organize the development schedule accordingly. For example for the alpha release we really tried to push everything such that we have a game that is almost finished. Then we had a bit a calmer time for finishing and polishing the game in the last few weeks.

We also think that we should have asked for more feedback. Everytime we asked for feedback the input we received was very valuable and helped a lot. As we don't get too much feedback by the design of the course, we should have asked for more.

6.3 Personal Impressions

To wrap everything up, here are our individual impressions of the course.

6.3.1 Josua

I was hoping to take this course since the beginning of my bachelors, and had high expectations for the outcome of the course. Thus I made sure to have as much time for the course as possible and I would say it was worth it.

I am happy with the result of our work, although I was hoping to implement a few more things, for example, at least one different level with different textures, an additional game mechanic and a bit more challenging puzzles (basically what Studio Gobo said). But at some point our capac-

6 Conclusion

ities were reached, but nevertheless, for time available between us, it is a very good outcome and I am proud of it.

One of the biggest technical challenges for me was probably implementing the animations from the MLEM library into our architecture and model. We designed our components to be almost state-free, and all state about the game is saved in the game class in `CurrentLevel`. This way it is easy to just replace the state with a different state (for example when resetting the room) and the game would just function correctly, without us having to think about this event happening. Unfortunately the sprite animation from MLEM had their own state, for example how long the animation has been running and, what actually made it complicated, the condition for a specific animation being shown. When we replace the state, the reference to an entity, for which an animation should be shown, is thrown away. We originally needed the entity to read out the state the entity was in to determine which animation to show. It was a big hassle. In the end we introduced Entity-IDs where each entity got a unique ID and this way we could look up the entity object by using the entity-ID, which stayed the same for each reset of a room.

Speaking of libraries, including FMOD was also very tedious. I used the ChaiFoxes wrapper library, but this library is built for a different FMOD version, but it could in theory also be used with a newer version. But even with the FMOD version for which the library was built for it did not work for some reason, so I had to manually initialize FMOD, which then solved the problem. Using FMOD made it difficult to port our game to Linux, as dotnet core did not find the libraries on Linux, but I managed to solve this (even if not nicely) in the end.

But for the code quality overall, I am happy how it ended up. Apart from forgetting to think of how to implement the sound in the design of the architecture, I think that our architecture was very stable and made our life easier (apart from the animations of course). I don't know whether it is the best way to model a game, but it was certainly good enough. But next time we should design the architecture in person on a white board, that would be easier.

For the next game project, if it would be done in a team again, I need to learn a bit to delegate some tasks. Of course I had the most time available but I ended up doing so much, that I could not do the things I really wanted. For example there is still potential in the sound design. I knew that when I had a certain vision how to do something, it would be very difficult to communicate to my team mates how I envisioned this and I felt, that it would be faster if I would do it myself.

I consider the playtesting in the school our greatest success. I was so surprised how much the students loved our game, and they were asking my father even days afterwards where they could get the game. It was a really rewarding experience.

As for the theme, I loved it. I immediately had ideas what to do with the theme. I did not think of the theme as a restriction, as we were still very free how to interpret the theme. Of course we could have done the same game without a theme, but I believe that the theme helped us in finding a gameplay. Also I don't think we would have a story without the theme.

I liked that this year the target machine was Windows-PC and not XBOX. I am a PC-Gamer myself (with the exception for Nintendo-Consoles) and don't really see a point of the XBOX for myself, if you have a gaming PC. But personal opinions aside, I think that not needing to worry about how to publish a game for the XBOX console helped us in further polishing our game and getting a better result in the end.

What I liked about the MonoGame engine, is that we had (almost) total freedom in how to represent our data and that we could design the whole pipeline ourself. Of course the downside is that we had to do a lot by ourselves and that MonoGame from time to time was a bit fiddly, but we also learned a lot in the process.

Overall, taking the course for me was worth it. It was fun developing a game into a state in which I am happy to publish it and make it available for everyone. I think that our game is a success. I don't plan to continue working on this game, but I will definitely use the gained knowledge in the future for a new game.

With that being said, thank you very much for organizing this course. It was a blast.

6.3.2 Martina

My goal for this course was to create a game that I am proud of. I am happy to say that I reached this. I like our game mechanic, our graphics and generally the mood of our game. But it is not just me, also my family and friends liked it. This means a lot to me because together with my team we were able to create something that can entertain others. And it is also a great opportunity to show to my family and friends how much fun it is to study Computer Science!

So far, I have only talked about the positive aspects of our game. Of course there are so many things that could still be improved. Especially, I would have liked to have more time for level creation.

I am happy with my decision to take this course. The mood during the course was always very positive and inspiring. This created a great environment for me to do my best and create an amazing game. However, I would have appreciated to have more feedback from the TAs. The feedback after the presentations in class was usually very short and I would have loved to hear their opinion on our game in greater detail.

6.3.3 Viturin

I am delighted with the final game and especially the path we took to program and design it. I am proud that we have implemented most of the mechanics from scratch. For me, the physics engine was tough to get working; it is up to this point a weakness of our game. I would personally use a library for the physics stuff next time around. The Grimms fairytale theme was excellent, and I am very proud of the final game. In my opinion, we reached the desired target. Monogame allowed us to feel the game mechanics working, which I enjoyed.