SPRINGBREAK

Lasse Lingens – Producer, Programmer, Game Designer, Music, Graphic Designer
Chris Mnuk – Programmer, Game Designer, Music, Quality Assurance
Christian Fania – Programmer, Game Designer
Tobias Grob – Programmer, Designer, Graphic Designer, Quality Assurance

Contents

Chapter 1. Formal Project Proposal

1.1. Game Description

1.1.1. Overview

“Springbreak” is a 2D platformer Multiplayer (PvP) with extraordinary capabilities, in addition to some already known elements which are present in most platformers (e.g. platforms, traps, collectibles). We give the player not only the possibility to manipulate his surroundings but furthermore the fundamental laws of physics. While jumping through the stage and trying to collect different things dependent on the current level they try to shift the physic to their advantage and reach the amount of collectibles faster than the other player.

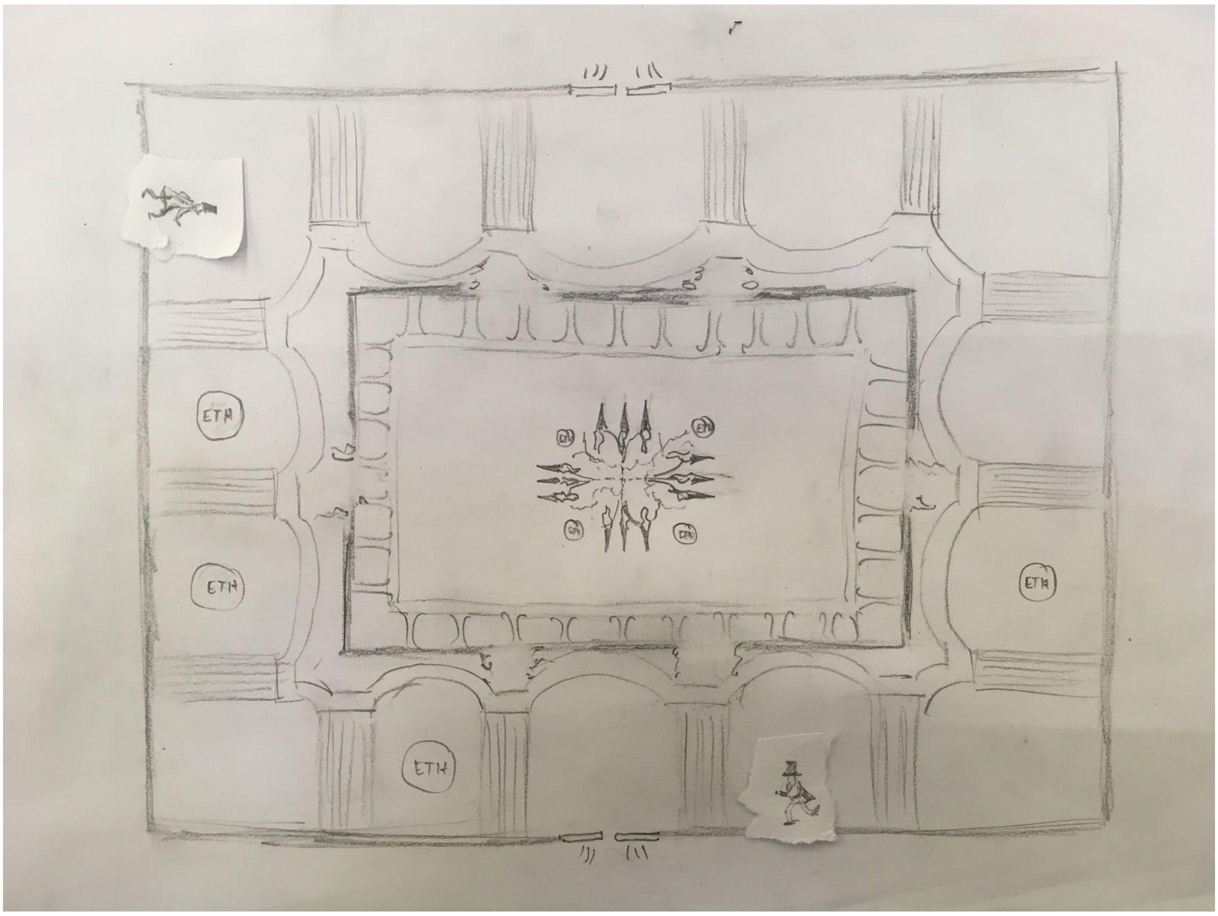
1.1.2. Background Story

See the Story file at git.

1.1.3. Design Decisions

The setting will be 2D-pixel arts, scaled down to be of an enough high resolution to not fall into the old school lowbit pixel graphics.

The players will have a set of certain core abilities already famous for platformers (e.g. jumping, wall jumps, running) and the innovative set of possibilities opened by changing physical laws. One example is the change of the direction of gravity. Using one of the back-buttons (RT) the player indicates he wants to change the gravity direction, while holding that button he presses one of the four arrow buttons to indicate the direction in which the gravity shall be directed until the next change.



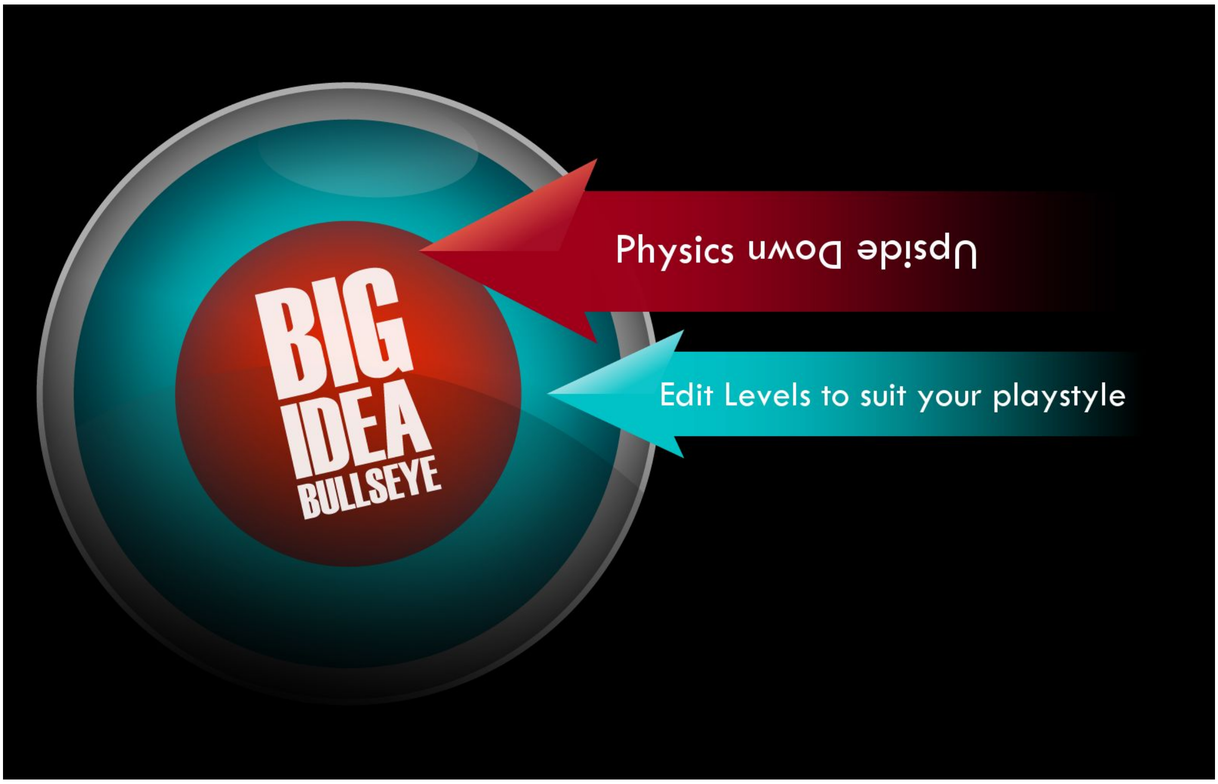
Adapting to a drastic change in the settings all while being under pressure at performing better than the other player and always the need to decide when to use which manipulation will demand a focused mind of the players. At the same time there will be a huge rewarding feel if one of your plans worked out just as you imagined it (e.g. reaching a collectible much faster or letting the other player die).

Dying itself will result in a small respawn time punishment. The manipulations themselves all have a cooldown for each player individually to prevent total chaos.

1.2. ‚Big Idea' Bullseye

Changing the laws of nature is a desire of mankind since we discovered the laws of nature and even before that humans desired this impossibility, defying gravity, being faster than the speed of light and so much more. With our game we provide at least a small window to what it could be like, giving the player godlike powers to shape the world according to his imagination.

For us it is really important that the handling is intuitive but to deal with the changes is not trivial and demand permanent adaptations.



1.3. Technical Achievement

Our Story will provide some of well developed levels with a certain theme, however ones the players learned the way the game works it would be a shame to let their own ideas go to waste. Therefore we will provide a level editor with full functionality so the players can create their own stages and compete in their own thought constructs.
Secondary, since we are studying at the ETH we will try to provide exact physics. This means we move the player whenever possible through forces, add friction to different materials, differentiate between the friction when you need to start running and when you already at full speed. Depending on our time this part will always grow during the different development stages.

1.4. Development Schedule

1.4.1. Layered Task Breakdown

1.4.1.1. Functional Minimum

Physical Attributes:

Gravity, Player relative friction, Acceleration, Maximal Velocity, Collision handling

Ingame Objects:

Platforms, Pits, Collectibles, Spikes

Leveldesign:

There will only be one level with a fixed size and rotational symmetry, which makes it easier to design a level for changing gravity directions

Character Skills:

Jumping, Running, Changing Gravity, Dying

Music And Sound:

One repeating song, Sound for running, jumping, colliding, dying and touching a collectible

Graphics:

One Character Model, self designed platform and collectible graphics

1.4.1.2. Low Target

Story:

3 levels which represent big milestones of Escher's life (gotthard tunnel, ETH, political engagement) and are supported motivational explanations by Escher for the goal of the certain level

Physical Attributes:

Moment of inertia, Mass, Material behaviour

Ingame Objects:

Jump Pads, Different Surface behaviour (friction, stickiness etc), Moving objects

Leveldesign:

Add levels that are possibly greater than the screen, resulting in split screen and camera movement.

Level Editor:

Basic Point and Click level editor with undo function, being able to scroll (for levels greater than the screen) and a toolbar with the addable objects

Character Skills:

Finding a logic and pleasing spawn place close to the point of death, losing collectibles upon dying, sprinting, wall jumping

Music And Sound:

Different Themes for the levels, adaptive sounds (indicating the current score by changing the pitch), Giving indicators about certain Events

Graphics:

Model of Escher, the both players and giving each level its own environmental style (eth inner life, grass and mountains etc.)

1.4.1.3. Desired Target

Story:

Provide a Story with guiding the players through Escher's life, add small cuts and video clips for events in between the levels

Ingame Objects:

Local Fields with their own physical laws, Powerups modifying the players abilities (jump higher, run faster et.), more different surfaces, breakable objects and barriers (as well as the possibility to reset them to their first state), Buttons triggering some events in the level (opening doors etc.).

Level Editor:

Have a List of actions for multiple undos and redos, add object selection and grouping as well as scaling and adding the new objects, set physical constants for certain objects (friction, stickiness, force until it breaks etc.)

Music And Sound:

Guiding theme throughout the game, change the music to current state of the level (e.g. speed up if one player collected 90% and therefore the game is coming to an end)

Graphics:

Cutscene in between

Character Skills:

More player-player interaction (being able to lift, jump on the other player to make him lose some collectibles etc.)

1.4.1.4. High Target

Physical Attributes:

Conductivity and conductive material as well as magnetic and electrical fields, temperature changes and their following modification on material (phase transitions etc), time warping (going back in time for a certain amount)

Level Editor:

Level Editor for Computer with export function to the xbox

Graphics:

Add lights and the according shades (2D ray tracer style)

Music And Sound:

Change the Music according to physical changes (low gravity slows it down, high gravity encourages a more bass heavy theme etc.)

1.4.1.5. Notes

A lot of our decisions need some playtesting do decide if they really improve the gameplay, to much correct physics might make it to complicated for the players to grasp all at once and therefore one might become frustrating not being able to see the reason for the behaviour of our game environment and therefore not being able to predict it.

We would like to add more playmodes, 2v2, 1v1v1, 1v1v1v1, some single player puzzle stages.

We have certain ideas about what might be fun to add but it strongly depends not only on time but on testplaying and experience.

One Example would be adding non photorealistic rendering to simulate maybe a state where both players are drunk or on drugs (inevitable for most of the top swiss managers and their right hands). However even if we have the time to implement this we need to test if it fits the game and is still fun to play.

We also want to add some “easter eggs” giving certain sarcastic voice lines if one dies or rewarding the players with the huge amount of amusing math jokes we learned at the ETH.

1.4.1.6. Extras

- 2vAI
- Auto-generated levels
- Physical simulations (full rigid body interaction, fluid simulation like adding rain to a map making it possible to swim over some obstacles after a certain amount of time

1.4.2. Task List

Lasse: Background, Music,

Chris m: Sounds, Music

Chris F: Level Editor

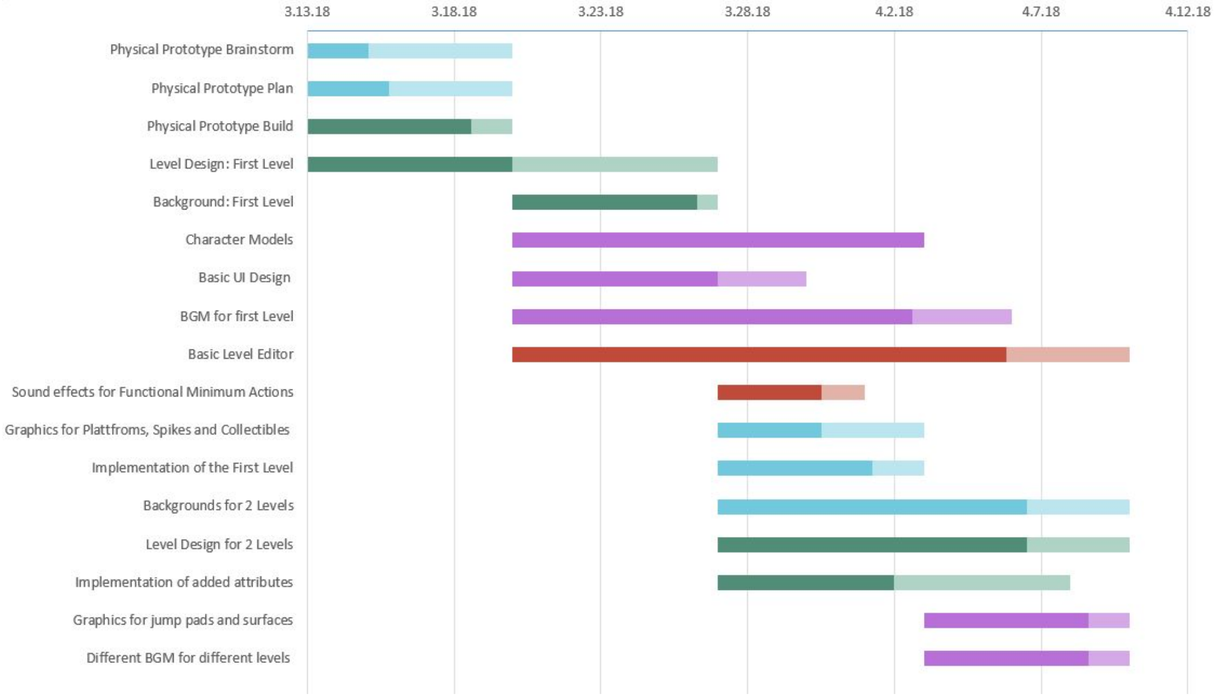
Tobi: Graphics

Story: all together

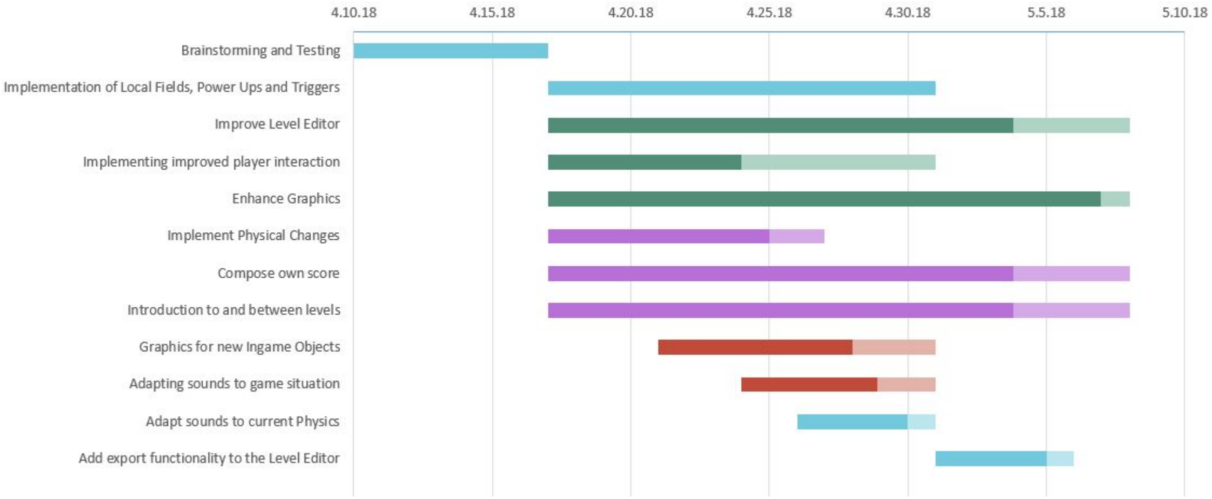
Coding: whoever got time -> Good commentary, interface and explanations

Brainstorming, meeting to discuss and update everyone on progress (once in 2 weeks)

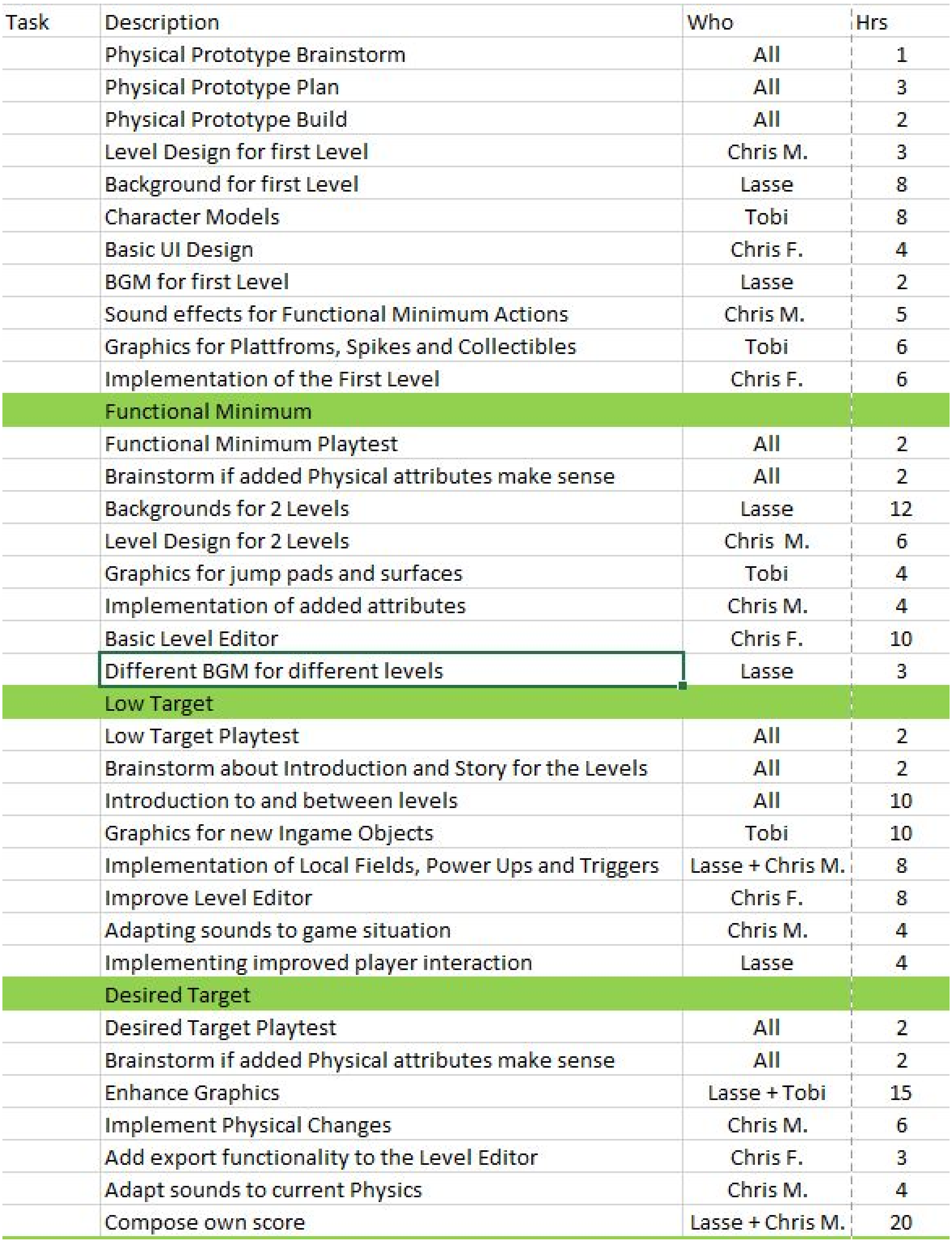
1.4.3. Timeline



Functional Minimum: Dark is the expected end date, light is the Deadline.



Desired Target



Task	Description	Who	Hrs
	Physical Prototype Brainstorm	All	1
	Physical Prototype Plan	All	3
	Physical Prototype Build	All	2
	Level Design for first Level	Chris M.	3
	Background for first Level	Lasse	8
	Character Models	Tobi	8
	Basic UI Design	Chris F.	4
	BGM for first Level	Lasse	2
	Sound effects for Functional Minimum Actions	Chris M.	5
	Graphics for Plattfroms, Spikes and Collectibles	Tobi	6
	Implementation of the First Level	Chris F.	6
	Functional Minimum		
	Functional Minimum Playtest	All	2
	Brainstorm if added Physical attributes make sense	All	2
	Backgrounds for 2 Levels	Lasse	12
	Level Design for 2 Levels	Chris M.	6
	Graphics for jump pads and surfaces	Tobi	4
	Implementation of added attributes	Chris M.	4
	Basic Level Editor	Chris F.	10
	Different BGM for different levels	Lasse	3
	Low Target		
	Low Target Playtest	All	2
	Brainstorm about Introduction and Story for the Levels	All	2
	Introduction to and between levels	All	10
	Graphics for new Ingame Objects	Tobi	10
	Implementation of Local Fields, Power Ups and Triggers	Lasse + Chris M.	8
	Improve Level Editor	Chris F.	8
	Adapting sounds to game situation	Chris M.	4
	Implementing improved player interaction	Lasse	4
	Desired Target		
	Desired Target Playtest	All	2
	Brainstorm if added Physical attributes make sense	All	2
	Enhance Graphics	Lasse + Tobi	15
	Implement Physical Changes	Chris M.	6
	Add export functionality to the Level Editor	Chris F.	3
	Adapt sounds to current Physics	Chris M.	4
	Compose own score	Lasse + Chris M.	20

1.5. Assessment

Our main strength of the game is as the big ideas bullseye suggests, the ability to change the physics. This is the most cool thing in our opinion.
Casual players might enjoy a 1 versus 1 session or get engaged in solving puzzles together. They can have mind-wars what other players might do and can plan ahead or change the physics in order to be able to finish a stage..

1.5.1. Criteria

We use the following criteria to judge if our design is a success:

- The game is fun to play already after a short introduction (ease of entry).
- The game is engaging to play even after a lot of session (fun factor after repetitions).
- The fun can come from being in competition as a player as well as engaging the players mind.
- The game is enough visually pleasing.
- The players are able to come up with working strategies and feel rewarded if they work.

Chapter 2. Prototype

2.1. Prototype Setup

Rules

"Gravity is just a social Construct"

A game for 2 Players

The Assets

The Playing Board:

This is the place where all the fun happens, you will put your *Players* as well as any other objects on it.

The Characters:

Each player has its own figure, if they can't settle who gets which color, they play rock-papers-scissors.

The Barriers:

These are platforms and barriers which the *Characters* can not get through.

The Border:

Don't overstep this one, you will fall in the abyss and die otherwise.

The Spikes:

Careful, these bad boys kill on touch.

The Coins:

You definitely want to get these, at least if you want to win.

The Action Cards:

Players use these to choose actions during the *Strategy Stage*.

The Bag:

Here the player puts his collectibles, as well as his dice.

The Dice:

These two are here to help you counting through the rounds without having to memorize all the numbers.

Game Goal

Collect more than half of the *Collectibles* before your opponent does.

Set Up Stage

The players sit both next to each other, slightly angled so they won't be able to see the cards of the opponent later on and lay the *Playing Board* in front of them.

They first need to agree on a stage. A stage defines the positions of the *Barriers*, *Borders*, *Spikes* and *Coins* as well as the start positions of the *Characters*. We provide an example of a stage however you can get creative and design your own ones. To set the stage up just follow the layout provided by us (or yourself) and put the different objects according to it on the grid.

Each player gets one set of *Action Cards*, consisting of the following cards *Left*, *Right*, *Gravity*, *90*, *180*, *270* and four times the *Nope* card. Further each player is provided with two dice and a *Bag* for the *Coins*. One of the dices will be the *Gravity Dice*, handling the *Gravity* cards cool-down. The other one will be the *Falling Dice* handling the speed with which a *Character* falls, they both start in the *Bag* of the player.

The Round

A *Round* consists of the *Strategy Stage* and the *Action Stage.* After finishing both a new *Round* starts. If a player reaches the *Game Goal* while the other has not after the end of a *Round* this player wins.

The Strategy Stage

Each player chooses always four *Action Cards* without the other player seeing what he chooses. He picks for each of the following categories one card.

1.: *Left* or *Right* or *Nope*

2.: *Jump* or *Nope*

3.: *Gravity* or *Nope*

4.: *90* or *180* or *270* (these three are only playable if *Gravity* was chosen on point three) or *Nope*

When both players have decided which four cards they want to pick, the game continues to the *Action Stage*.

The Action Stage

The *Action Stage* defines how the *Characters* move according to the current state and the *Action Cards* which got played. This Stage is divided into four phases and has to follow the ordering of the phases as given here.

Dice Phase:

All the *Gravity Dices* get their number of eyes on the top reduced by one. If a dice is already at one the player takes it back in his bag and he picks up the *Gravity* card and adds it back to his hand. All the Falling Dices get their number of eyes on the top increased by one. If the dice shows four and the *Jump* cards lies in front of the player, the dice is set to one and the *Jump* card gets turned face down. If the dice is already on six, he stays at six and nothing else happens. If the *Character* is on *Stable Ground* he has to reset the *Falling Dice* by putting it back into his bag.

Gravity Phase:

If one or both players have played the *Gravity* card, the *Playing Board* gets rotated to the total sum of the degree cards picked in clockwise order. (e.g. one player chose 90, the other 180, the board gets rotated 270 degrees in clockwise order).

If a *Gravity* card was played it gets laid open in front of the player who played it and gets the *Gravity Dice* showing 6 eyes to the top added on it.

Top/Down Phase:

If a player played the *Jump* card, he lies that card face up in front of him. The players calculate their top down movement by the following rules: If the *Jump* card of a player lies face up in front of him, he adds +3 to the negative *Falling Dice* eye number. They move according to the result. If it is a positive result they move up the amount of the result, otherwise they move down the amount of the result. While moving up or down they have to check if any *Collisions* would occur on their path. Both players move at the same time.

Left/Right Phase:

The *Characters* move left or right according to the following rule: if a player played his *Left* card he moves one site to the left, if he played his *Right* card he moves one to the right. While moving left or right they have to check if any *Collisions* would occur on their path. Both players move at the same time.

Collisions

This event might only occur during the *Top/Down Phase* or *Left/Right Phase* if a *Character* touches certain objects some events might trigger.

Collectible Touch:

If *Character* would land on a field holding a C*ollectible* or would walk through a field of a C*ollectible*, the C*ollectible* is taken by that player and added into his *Bag*.

Barrier Touch:

If *Character* would land on a field or walk through a field with a *Barrier* or an *Edge* on it, he instead stops directly before that field (in the direction he was coming from).

Player Touch:

If the *Characters* would touch each other during the *Left/Right Phase*, they don't move at all. If they touch during the *Top/Down Phase* the player who had the higher starting position gets his *Falling Dice* reset and added to his *Bag* and if possible his *Jump* card added to his hand. The other player dies.

Spike Touch:

If *Character* would land on a field or walk through a field with a *Spike* on it, he dies.

Other Events

Player Death:

If a *Character* dies, he is set to the bottom left or right corner with a *Barrier* under him (as close to the corner as possible). He is set to left if he started the game in the left corner, if he started in the right he is set to the right corner. His *Jump* card is added back to his hand. His *Falling Dice* is put back in his bag. He can not move during this round.

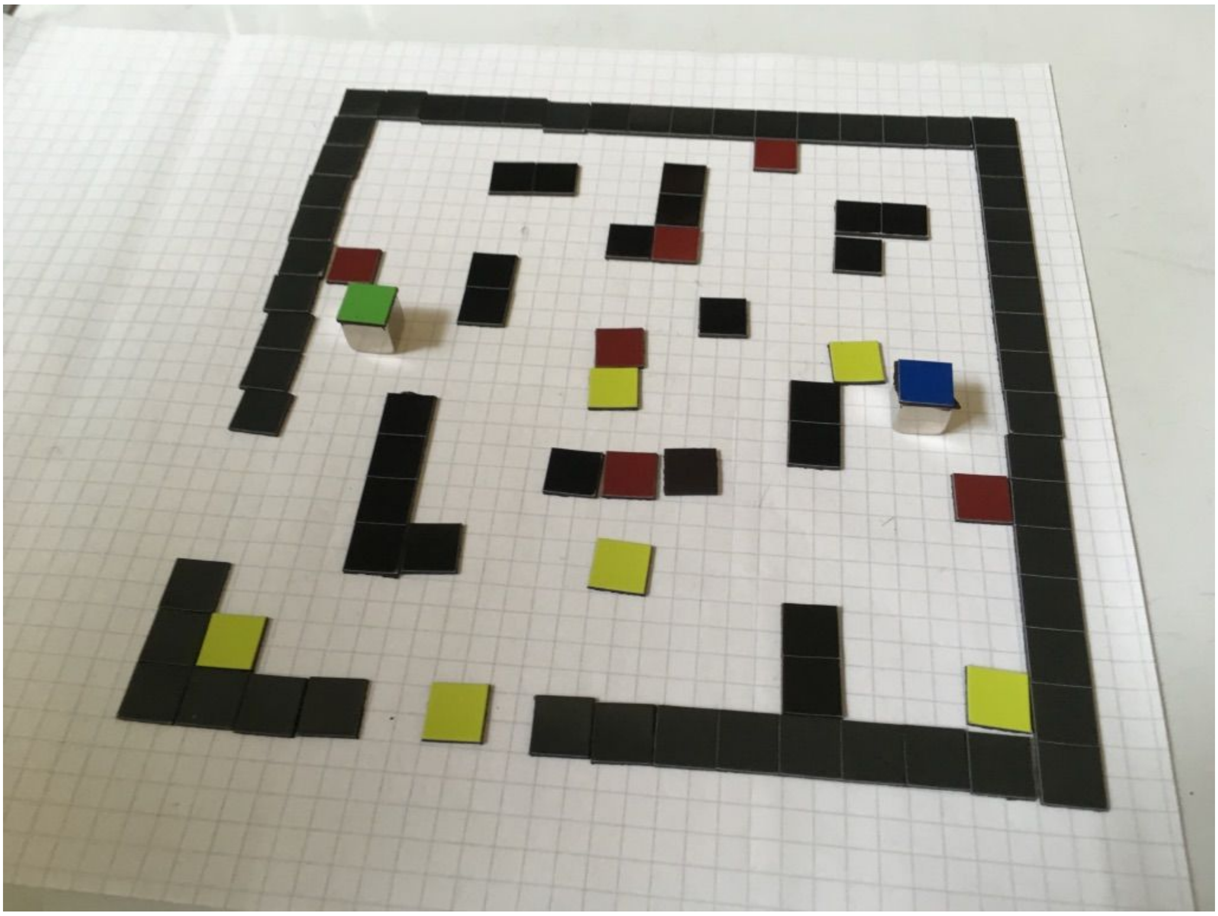
Stable Ground:

If a *Character* has a *Barrier* or an *Edge* under him, he is on *Stable Ground*.

Game End:

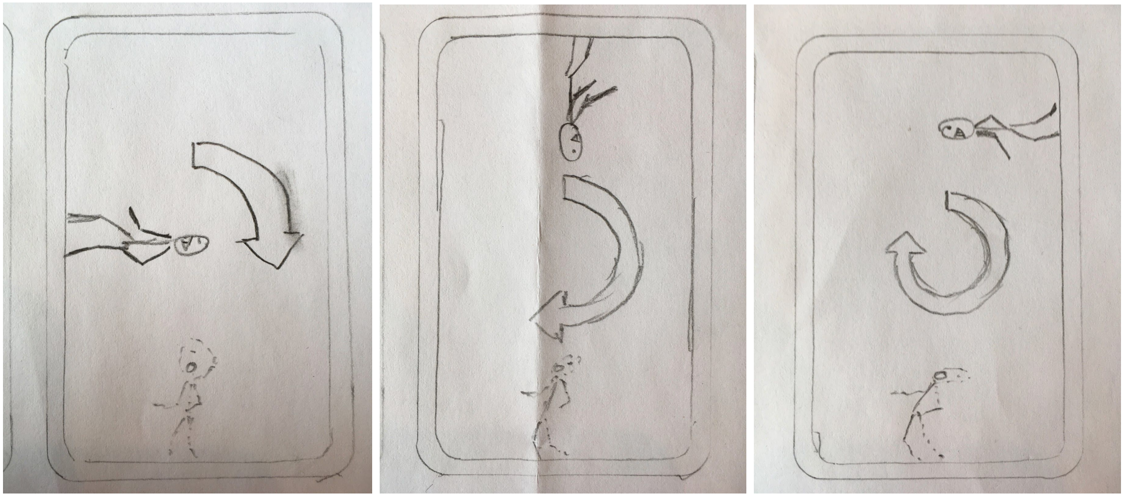
The *Game Ends* if one of the players fulfilled the *Game Goal* or a player chooses to give up under the overwhelming skills of his opponents. Draws are not possible.

The Board

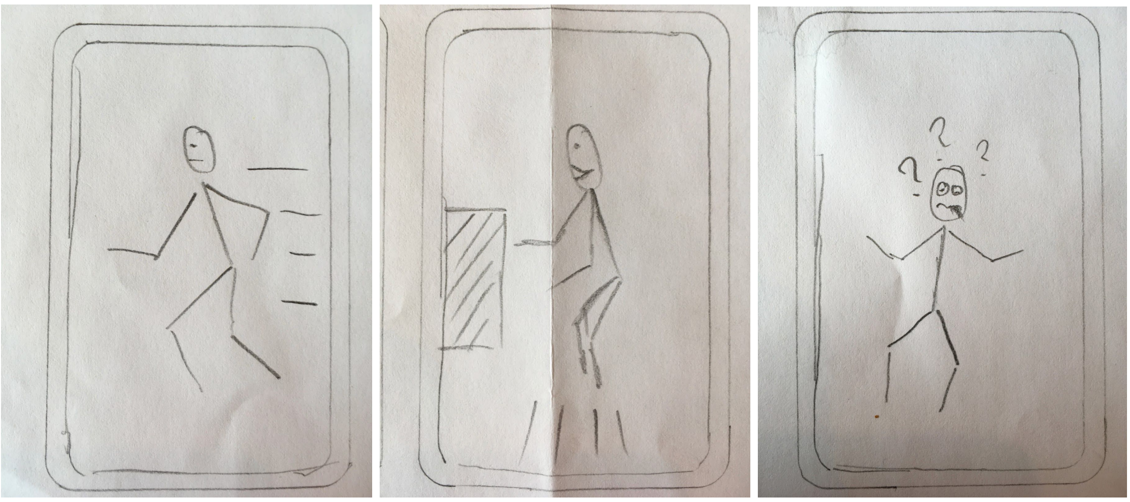


The board with a game level of 16x16 tiles. Black tiles represent walls, red tiles spikes, yellow tiles collectibles. The green and blue cubes are each players characters.

The Cards



The gravity action cards. Turning the gravity by 90, 180, or 270 degrees is possible.



The other action cards. Not all cards are shown. From left to right: Running left, jumping, and the “Nope” card.

2.2. Playing Experience

In playing the game we found that turns were we it was possible to use the gravity rotation, were the most interesting ones. Further, overall it was rewarding to get a collectible and the game was fun and engaging as one quickly tried to plan ahead to deny the enemy and get it for oneself.

2.3. Findings and Conclusion

- Explain what you have learned from creating the prototype.

Dense levels turned out to be more interesting than loosely populated levels.

- What has proved to be harder (or easier) than expected?

In the beginning it seemed rather difficult to capture the mechanics of a real-time game with continuous space in a boardgame. However using a turn-based approach with action cards made the basic design surprisingly simple. But then the details of the physics calculations needed a lot more thinking and tinkering than expected though.

- What design revisions have you made to your game based on your experience creating the prototype?

We learned just how important setting all the physics and movement parameters for the real-time game just right is, like for example how long cool-downs for gravity changing are, and got first approximate values for it.

It also lead us to revise how a cool-down for gravity works. The gravity cools down for each player individually and has to be short enough to keep the phases interesting and long enough to keep a players strategy predictable at least for a few turns (which we expect to translate to a few seconds in the actual game but that is subject to be found out).

We learned that the gravity rotation is sufficient for an interesting game idea and have thus decided to cut everything else in regards to physics manipulation.

Further we decided on having a single global gravity for all players. Early ideas included having player-local gravity.

CHAPTER 3. INTERIM REPORT

(Max 5 pages)

3.1. PROGRESS

Lasse:

Collision Handling:

The collision handling started with the real basic concept of checking if two rectangles, which represent the objects, overlap. To distinguish further from which direction they collide one needed to compare the current state to the state after applying a velocity update. By extracting the differences and changes of these two states of the game, I was able to detect a collision as well as the direction of the two objects colliding. A few errors showed up while further developing the game. At example the diagonal collision of two objects was left undetected as well as some other minor bugs which could almost always be corrected soon after they were detected.

The Project Structure:

We started with a really rough draft just to figure out how monogame works and what awaits us while developing with this framework. Afterwards we all discussed about the structure our code should have. The rework of this was more time intensive than challenging. With the help of the team and a lot of feedback we were able to maintain a structure which helps us to keep track of the changes of other team members. Developing a physical model as well as making it encapsuled was important for us as a team and delayed us a bit at the start. From there on the addition of new features like new objects, new forces etc. was comparably easy.

Level Save and Loader:

I started making my own manual loader and saver but after some great talk with the team and some further research i found the “Json.NET” serializer which makes it possible to simply save objects as text files even as a full deep copy and not only a shallow one. Only minor changes were necessary (at example adding specific constructors). However the loading and saving is much simplified and completely expandable now.

Player Player Interaction:

Adding a second player was mostly adding the controls for this player. The next step was the player-player interaction. The main interaction is killing the other player by jumping on top of him. It worked at first, however it was not reacting correctly to gravity changes. With the global changes of the controls, Chris F. fixed this bug. The player death was as first a simple workaround. Placing him outside the field and not drawing him. Later on we added a boolean which simply excludes him from all interactions.

Music:

I started to compose music whenever i had the feeling for a creative time. However we decided to prioritize our code at first and keep on adding the self made soundtrack later on.

Story:

We first drafted a Story about two angels guiding Escher. However this is not relevant anymore. After feedback during class we decided to minimize our graphical set we set our main stage to the ETH. With that we needed a new story to come up with. The new story is narrated by a man (i won't spoil who it is, you have to find out yourself) who tells his kids a good night story. The story is a letter written by Escher to his daughter and delivered after his Death. It is his last testimony, a secret no one would believe him. It is about two boys getting lost in space time at the ETH through a rift and the adventures they went through while Escher was guiding them by keeping contact through the rift. For more details please read the first draft of the first part of the story at git.

Chris M.

UI:

I designed a start menu where the player can start playing, check the controls, enter the level editor and exit the game as well as a UI to visualize the progress and necessary information for the player. I severely underestimated the time needed to create even simple assets like buttons and arrows because I've never really done this before.

Sound effects:

I inserted sound effects for all actions like jumping, earning collectibles, changing gravity and dying. For this I used assets from copyright-free asset stores.

Chris F.

Level Editor:

The level editor itself has proven itself to be surprisingly simple to implement, in particular the backend, as we had the saving and loading routines already and building on top of them turned out to be rather doable. However as per usual, good UI design is tricky. Visualisation of what one was actually doing or about to do took a bit until a good solution was found, and so far no good solution has been found for the saving and loading menu in particular on XBOX.

Animation:

We've also experimented with a stackable animation effects system, which has so far lead to a lot more visual goodies being implemented than were originally intended as it made them easier, however, it is still possible that in the end we might have to scrap the system as at the time of this writing there are concerns with how well it works together with traditional atlas based sprite animation.

Camera/Splitscreen:

Further, splitscreen has been surprisingly simple and what originally was intended as an optional stretch goal, has thus already become a part of the game, allowing us to build larger levels if desired, there is still some concerns during playtesting that this might be a trap where ease of implementation leads to something that we'd be better off without, so here too there is a chance it's going to be removed.

Tobi

Graphics:

Our barrier sprites currently are unaware of the adjacency of barriers. Changing the graphics of a barrier sprite to align better with neighbours requires a different level data structure for barriers. So it's harder to put in a seemingly simple change than expected.

Animation:

Animating a sprite using key frames can be used quite good with the current animation stack. What was harder to include were the states of a player to base different animations on. The code seems the diverge and loose its structure which made it especially harder.

Concluding we finished layer one and are partly in layer two and three. The most time intensive things missing from layer 2 are assets. While the game itself seems to function great already. Therefore the people focusing on the code started working on layer 3.

3.2. Challenges

There were no outstanding challenges in the project itself except for the ones mentioned above. Most challenging for us is the work overall in a team.
We further decided to use less of the objects that we have planned. Since they are relatively easy to implement and add we will test them out. For now however the game seems to work better with a more minimalistic approach since this puts the focus on the players interactions and abilities.

3.3. Future Work

Our main goal for this week is finishing and fixing all the bugs of a normal level and it's gameplay as well as adding the UI. From there on, also this week, we want to find a good way to insert our story as well as the way we want to engage the player with the story.

Chapter 4. Alpha Release

(Max 5 pages)

4.1. Progress

Lasse:
I started by improving the player controls over the character to make it more intuitive and interactive for the players. They are able to sprint as long as they want to at the cost of lower friction and therefore harder handling of the avatar in dangerous situations. Further the player is able differ the height of his jumps depending on how long he holds the jump button down. Also the player can now activate a slow-motion mode, where he moves slower for a certain amount of time to maneuver through difficult situations more safely. This ability recharges itself to a given maximum amount and is drained if used.
After that i added two more interactive objects to the game, force fields which simply put a certain amount of force on the player passing through as well as a powerup which gave the player an extra charge for their gravity manipulation ability.
With that I reworked this ability completely, allowing the player to use it without any break between the usage but instead having a certain amount of charges which replenish after a given time.
After that a few minor bugs in the UI and the collision handler were fixed.
The story has now an implemented structure. Due to the inability of monogame to play videos, i had to hardcode the story structure. I sadly had to interrupt this work due to the lack of team collaboration.
I had to switch to converting our project to UWP and making it ready for the XBOX.

Chris F.:
Chris implemented a particle system indicating the current gravity direction. The particles are subtle enough to not distract the player but well enough visible to help him orientate. The idea was to randomly spawn them and let them fade away after a certain amount of time.
He further kept the level editor up to date with the changes and added additional functionality. At example being able to set the values for jump pads and force fields.
He also worked further on the UI and fixed a few bugs as well as improving some elements. Example given the FPS and UPS counter.

4.2. Challenges

Converting the project to UWP was unbelievably time consuming. One example for a but was using the graphics.ToggleFullscreen() function, provided by the GraphicsDevice of monogame, while this method worked fine before it cause an unhandled exception in the UWP without indicating the source of this error. Another part was the saving and loading of levels due to the stricter accessibility rules of the UWP.

The other major issue were the missing results and lack of communication from half of the team. This caused extreme motivation loss as well as the reconsidering of the work that had to be done.

4.3. Future Work

The plan is to add a Challenge mode for two players in which they simply compete with each other. Also adding other winconditions except for collecting a certain amount of coins would be desirable. If that is done and everything works as intended the next step would to finally implement the Story.

Chapter 5. Playtest

(Max 5 pages)

5.1. Playtesting Session

Because we had some major issues with the usability (e.g. not being able to return to the main menu from the game or not being able to load a level so it would be time consuming to test later levels) we could only do a quick session. We asked interested friends to play a few levels of the game and also creating their own level to see how intuitive it is to construct a new level from scratch.

We tried to give as little input as possible and only observe how they play. They should navigate through the menu, see how intuitive it is to select different options and then playing the game.

We only had time to test some normal levels so the story mode is still to be tested.

5.2. Questions and Comments

We did not ask any questions specifically. We just asked them to give us a feedback to whatever they found good or bad and only then inquired further if we wanted to get more detail on a critique point.

We got a few comments on the Font that we used for almost everything. Some liked it but others found it hard to read so we will change the font where it doesn’t serve a style purpose.

The menu was easy to navigate in but it’s not always clear how to get back. Also options like changing volume of music and sound effects and selecting different controls layouts were missing which we should include.

Overall the game was enjoyable, but the controls were sometimes confusing so we should provide different options and also improve the UI in game. A slowdown bar would be helpful and the gravity counter is never explained.

The level editor was good to use but with the controller we should change the behaviour of the left stick because it’s a little slow to navigate through large levels.

Some people liked the splitscreen and others didn’t so we’re not really sure what to do yet with it but we should certainly separate the screen better since we don’t have a big border it can get very confusing until where the splitscreen goes.

5.3. Design Revisions

We will overhaul the UI with a different font and clearer elements so the players can better navigate and they should be easier to grasp.
We will add more options. Especially, the controller layout will offer a few different options to be customizable.
We’ll introduce a clearer border for the splitscreen and for levels that are small enough to fit on one screen we will discuss about removing the splitscreen.

The story is yet to be tested so we don’t know how enjoyable it is yet.

Chapter 6. Conclusion

(Max 5 pages)

6.1. Final Results

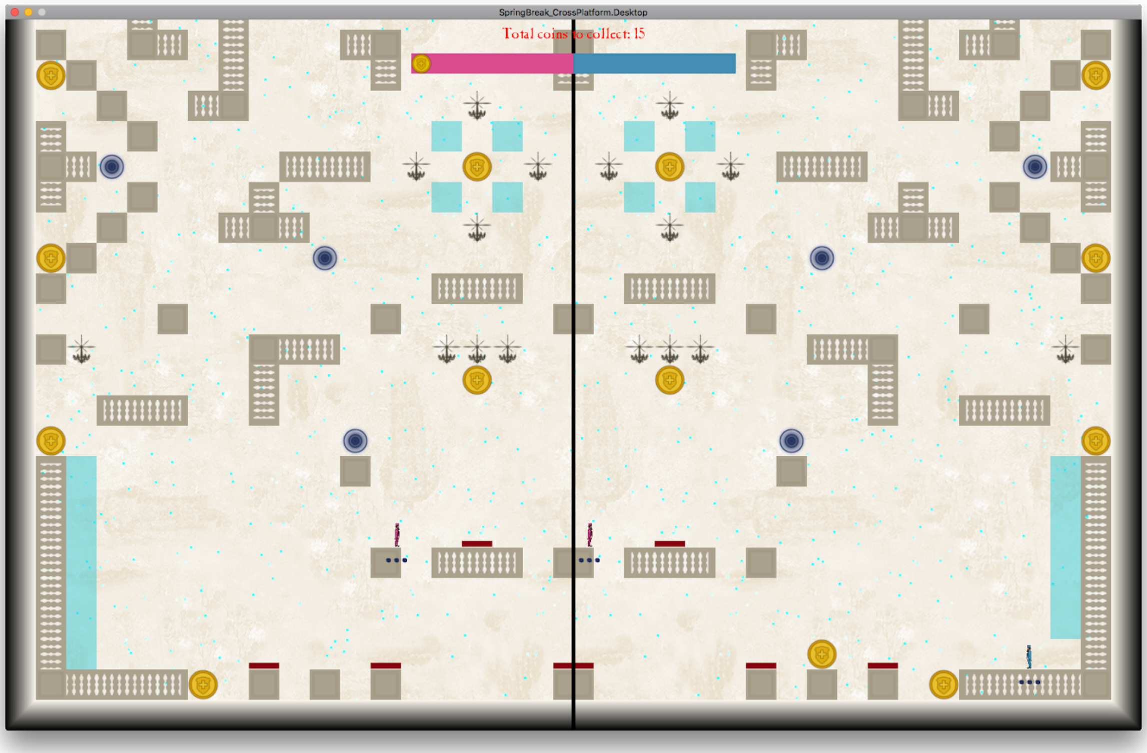
Our final game has a lot of significant changes in graphics, game play introduction, music and as well as how the code is set up.
The main code is now set up as a shared project to use for platform dependent code (for UWP and the Monogame Cross Platform setup). This is due to the fact that not all of us use Windows as the main developing platform and such developing and testing new stuff would be cumbersome otherwise. We also changed our level format to XML, as the JSON format had some platform implementation differences.
These code based changes now allowed to for quickly testing and running the new graphics and animations.
Major changes in the graphics are high detail vector graphics for the Escher character in our game. Actually there would be enough details to cover half the screen height of a 1080p resolution. The character itself in the game doesn’t need this detail level but that allowed a reuse for logos, splash screens and icons which also made it into the final game.
The character has now 17 frame running animation which arguably looks very smooth. We also changed the color of the character to a lighter tint of the colors of the initial color palette, so players can better differentiate between the characters.
When a character dies, it now fades out and fades in during spawning. This should avoid the during the play testing session noted confusion of the play tester where the players couldn’t figure out what was happening when a character dies especially due to character-character interaction.
We changed the initial spike graphics to a spikey candelabra. The candelabra also rotates according to the gravity.
The game now also includes new gravity power-ups graphics (look at the blue circles in the screenshot below).
The world has now background graphics made with a seamless texture.
Instead of a number to indicate how many times a player can use the gravity change function we use dots and hopefully.

We scaled up the world by 50% to make the game better viewable from further distances.

We introduced an adaptive camera splitting or unsplitting the screen initially. The transition between the split and unsplit mode was not smooth enough, and would have needed much more work. So we decided to an easier solution. Instead the camera now has one mode (either all one screen or split screen) depending on the level size.

Copyright and licensing issues (for presenting our game) had a major impact. The first claim of using copyrighted work wouldn't be a problem was limiting us.
We had to change the better fitting classical background music to a free content version of a mix of metal and classical music.
We also couldn't rework and put in the story mode, as there was a lot of non-free content in the story.

Luckily, the final game includes a tutorial mode, with 5 levels of increasing difficulty showcasing all the action and abilities the characters have.



Screenshot of the final game.

6.2. Experience

Comparing to the initial pencil level sketch seen in chapter 1 some visual elements definitely made it into the final game.
We did not include some game elements at all (like trap doors, breakable objects, magnetic fields, etc.) due to more focusing on the key aspect of changing gravity.
We also focused on having one visual level style instead of many.

Initially we were able to follow the schedule in the first weeks but soon we significantly deviated from it, as we had different expectations as well as some communication problems.

Having a clear structure of what happens in the game was definitely contributing to our game. The prototype showed us that only using 4 gravity directions makes the game already quite interesting.
Play testing also showed us major unclear issues for the players as well as some bugs not found otherwise.

6.2. Personal Impressions

Evaluating the criteria

- The game is fun to play already after a short introduction (ease of entry).
  => With the tutorial levels, players can quickly get into the game. And it seemed they had fun.
- The game is engaging to play even after a lot of session (fun factor after repetitions).
  => For us, as well as play testers it was a lot of fun to continue playing several rounds of the same level.
- The fun can come from being in competition as a player as well as engaging the players mind.
  => The game is highly competitive, and definitely engages the player mind in terms of skills of using the right actions.
- The game is enough visually pleasing.
  => The game could have some polishing, but doesn’t look bad.
- The players are able to come up with working strategies and feel rewarded if they work.
  => As the mode is highly interactive, the players tend to play more skillful than strategic.

Questions

- What was the biggest technical difficulty during the project?
  - The physics controller.
- What was your impression of working with the theme?
  - Fitting our game idea to the theme was not as easy as expected.
- Do you think the theme enhanced your game, or would you have been happier with total freedom?
  - It definitely help to generate ideas out of "nowhere".
- What would you do differently in your next game project?
  - We would clearly communicate on how and when we want to work. We would meet more often.
- To what extent did you meet your project plan and milestones (not at all, partly, mostly, and always)?
  - Not at all.
- Are you happy with the final result of your project?
  - It was fun! In general we are happy with it.

Final Remarks

Concluding to the criteria and the questions the project was (after some rough phase) definitely a success!