Project Notebook - Project: Equilibrium

Julian Geheeb Lucas Leder Yannik Melzer

Master Practical Course Games Engineering: Computer Graphics and Visualization - SS 2021

Contents

1 Formal game proposal

The game is being developed for a Master Practical Course which has an underlying theme that the game is supposed to implement. The given theme is *Chaos and Order*. In the following sections, the basic game idea is explored. Furthermore, it is explained how the design principles of the game fit the theme, which technical achievements can be implemented and the current development schedule is outlined. The chapter concludes with an assessment of the game idea.

1.1 Game Description

1.1.1 Interpreting the Theme

Before coming up with ideas of what games can be made regarding the theme, it is important to understand the theme itself. Usually there are many ways to interpret such a restriction. In this case, it is easy to associate the *Chaos* part with something negative and the *Order* part with something positive. This might result in a game where the gameplay is not equally balanced towards both parts of the theme, e.g. the task of the player is to clean up a chaotic room or in other words *create order in chaos.* While this approach is fine, the team's interpretation is to *have both chaos and order in a balanced state.* It also leads directly to the connections to Yin & Yang, which is a symbolic representation of a balanced state of two opposites, in our case chaos and order. Furthermore, it describes the idea of having *chaos in order*, which is the black dot in the white area, and *order in chaos*, which is the white dot in the black area, see Figure 1. This interpretation will be the underlying idea of the following decisions.

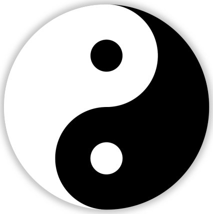


Figure 1: Yin & Yang symbol. Taken from `https://en.wikipedia.org/wiki/Yin_and_yang`

1.1.2 Basic Idea

Project: Equilibrium is a 2.5D bullet-hell game [1] with a top-down view. The basic idea of such a game is that the player has to defeat enemy waves by shooting at them while dodging bullets to be able to reach and defeat the final boss of a level. This genre already incorporates the theme of *Chaos and Order*

[1] `https://en.wikipedia.org/wiki/Shoot_'em_up`

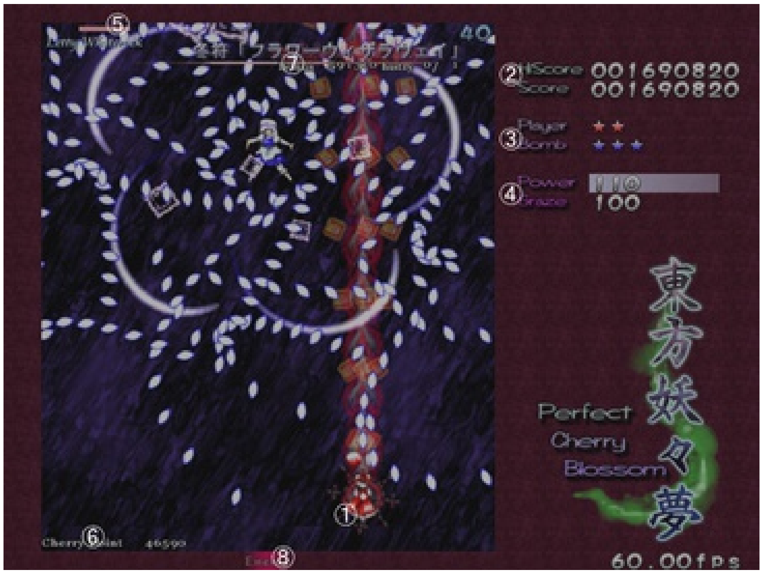
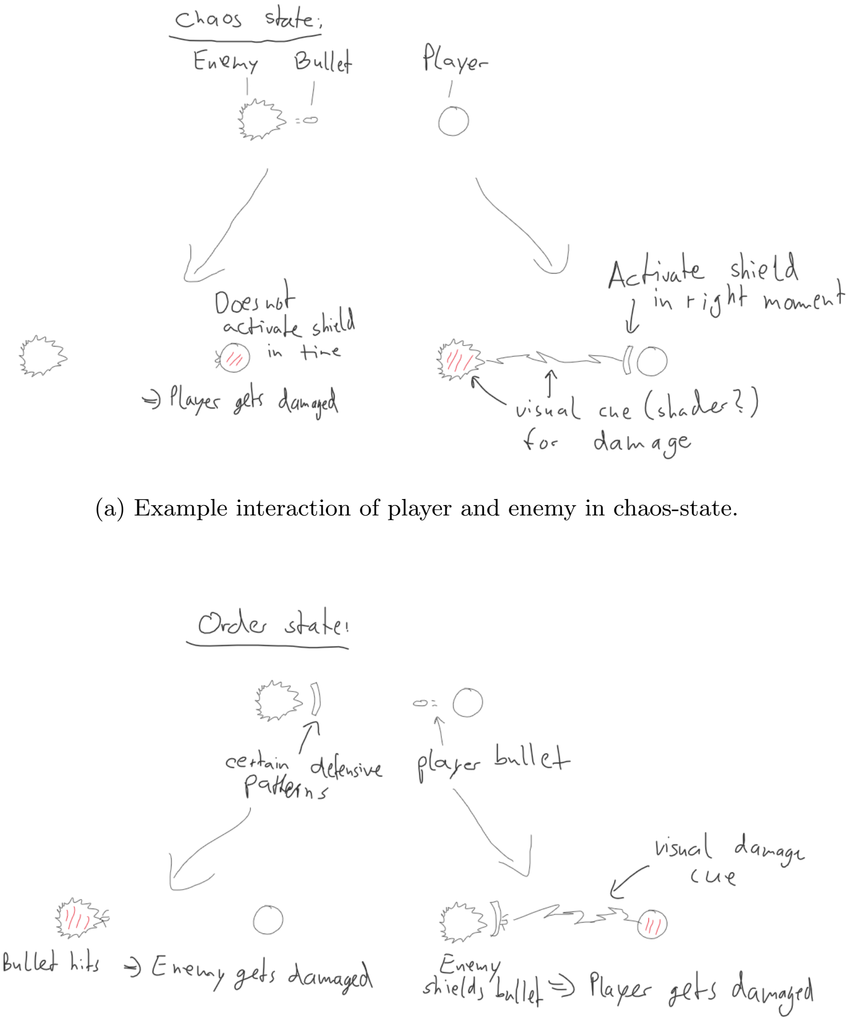


Figure 2: Screenshot of *Perfect Cherry Blossom*. Taken from `https://en.wikipedia.org/wiki/Touhou_Project`

reasonably well. At first glance, the screen in Figure 2 looks clustered and thus rather chaotic. However, having a closer look reveals a pattern, which stands in close relation to order. In a typical bullet-hell game, the actions of the player mostly consist of shooting, which is an *offensive option*, and dodging, which is a *defensive option*. In a broader sense, this can also be seen as a representation of chaos and order. The many bullets on the screen cause chaos, while dodging the patterns brings order into the chaos.

Project: Equilibrium tries to further emphasize and balance those properties by dividing the game world into two states, representing chaos and order respectively. Within a level, the world state constantly switches back and forth. In each state, the behavior of the enemies, as well as the interactions and abilities of the enemies and the player character change. When the world is in the chaos-state, enemy abilities also represent chaos by utilizing offensive options like shooting bullets. The player character represents order within chaos, therefore the player character's abilities are of defensive nature. However, the player should still be able to defeat enemies, e.g. by reflecting their attacks or baiting them into shooting each other. When the world is in the order-state, the roles are reversed. The player character is able to attack by shooting bullets and other offensive abilities. Enemies have defensive abilities with a strong emphasis on patterns. Likewise in this case, the enemies should be able to damage the player, by applying the same principles as in the chaos-state with the player, e.g. they can reflect the player characters attack. An example of this interaction can be seen in Figure 3.


Chaos state:
Enemy
Bullet
Player
Does not activate shield in time
⇒ Player gets damaged
Activate shield in right moment
visual cue (shader?) for damage
Order state:
certain defensive patterns
player bullet
visual damage cue
Bullet hits ⇒ Enemy gets damaged
Enemy shields bullet ⇒ Player gets damaged

(a) Example interaction of player and enemy in chaos-state.

(b) Example interaction of player and enemy in order-state.

Figure 3: Sketch of a storyboard of one scenario in two different states.

1.1.3 Name Design

Finding a fitting name for a product is a very an important task. The name is often the first contact consumers have with the product, so conveying the right image can help to increase the customer count. The name *Project: Equilibrium* consists of two parts, each of them giving a different but important hint to the overall theme and feeling of the game. *Equilibrium* describes the balance between chaos and order. The *Project* part comes from one of the more popular series of the genre, *Touhou Project*, so there is a connection to the potential target group. Upon hearing the name, an assumed reaction could be: "Project? Reminds me of Touhou. Also what is this part about equilibrium/balance? Sounds interesting, I should look into it."

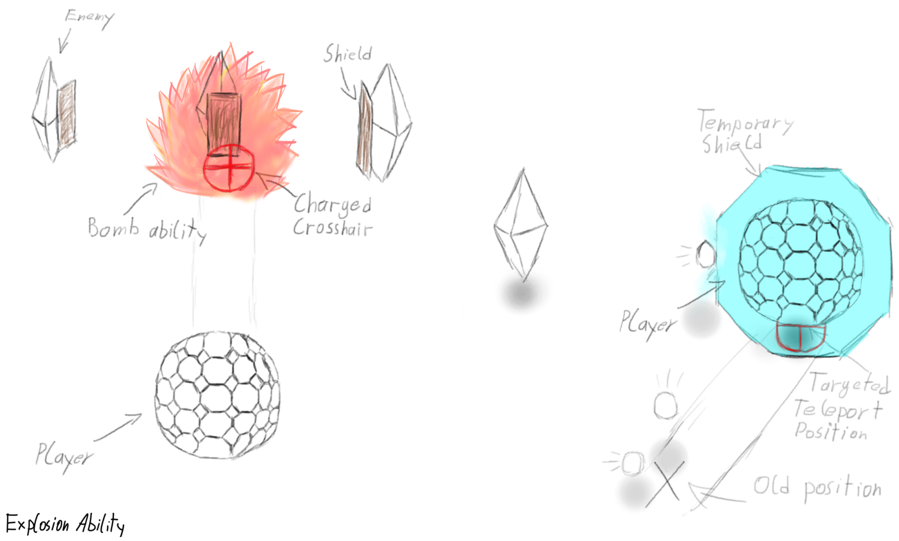
1.1.4 Player Character Abilities

At the time of writing this, the player character is planned to have three abilities per world state available, making it six different abilities overall.

The first ability is shooting bullets in the order state and shielding in the chaos state. The bullets travel a straight line in the direction the player character was facing at the moment of shooting. The shield is also bound to the direction of the player character and covers a set area in front of the player. Hitting an enemy with a bullet results in damage, so does shielding an enemy attack in the right moment.

The second ability is chargeable. The higher the charge, the further away from the player character the ability effect activates. In the order state, it is a small AOE damage ability that does not have any travel time, meaning it can ignore enemies and possible shields between the area of effect and the player character to deal damage. In the chaos state, the player can teleport to the designated area while having a small shield. This can be used to bait enemies into shooting each other by dodging as well as shielding many bullets at once when teleporting to the right area. A sketch of both variations of this ability can be seen in Figure 4.

The third action is only available to the player as an item pick-up. By making it a pick-up, it is possible to add many different abilities without giving the player too many options at a time. Since those abilities exhaust after using them once, they are usually more powerful than the other two abilities of the player character. One example is an AOE damage skill in a large circle around the player in the chaos state and a large shield all around the player in the order state, both with a reasonably long duration.



(a) Ability in the order state.

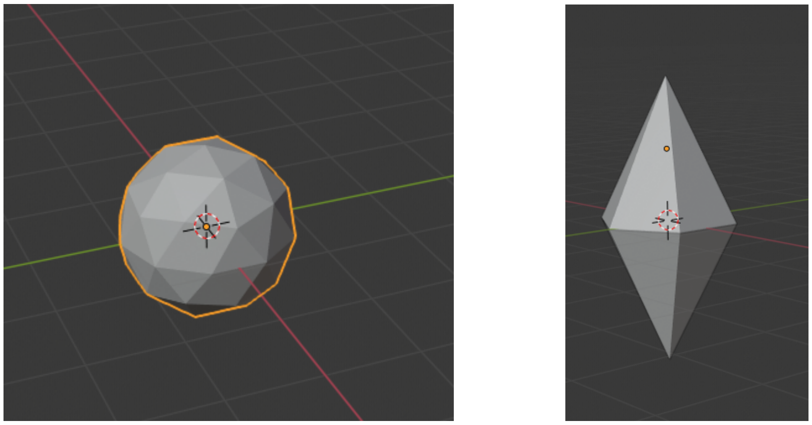
(b) Ability in the chaos state.

Figure 4: Sketches of the second player character ability.

1.1.5 Graphics, Transition and Sound Design

The following decisions are tightly connected and therefore influence each other, which is why they are part of the same section. In order to make the whole picture of many enemies, bullets and the transition not too visually overwhelming, the game is planned to have simplistic 3D graphics that represent geometrical shapes and few to no other colors besides black and white. Examples can be seen in Figure 5. Basic shapes allow for easy recognition even when the screen is clustered. Additionally, they work with a black and white color scheme which helps to integrate the notion of Yin and Yang. Depending on the world state, the colors get inverted, representing their current behavior and state as well. To differentiate the objects from the background, the wireframe is outline by the help of edge shaders.

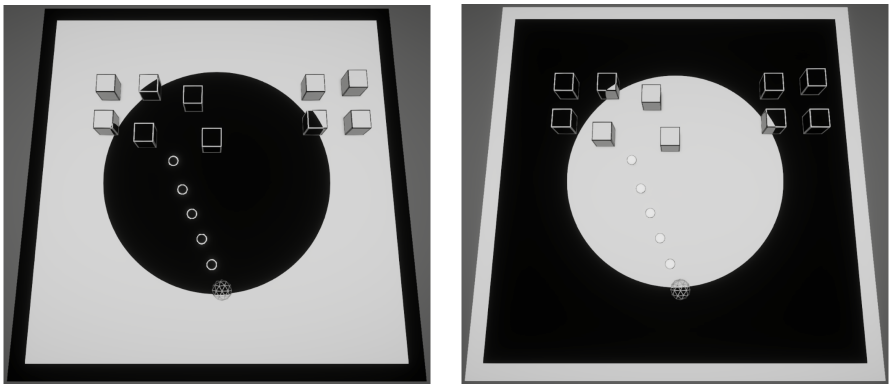
The transition between the two world states is a core feature of the game and therefore needs to be discussed and defined in detail. It is triggered when a gauge is full, but it does not happen everywhere in the game world at the same time. Rather, it starts from a single point or line and spreads throughout the level, swallowing objects, enemies and the player bit by bit. The changes happen per entity, giving the player another strategic element to play with, e.g. dodging the transition as long as possible for certain advantages. An example of desired graphics together with the state transition can be seen in Figure 6. For boss fights or special enemies, the plan is to have the model change while transitioning between states, so they have one model per state.



(a) Example 1

(b) Example 2

Figure 5: Example geometric shapes for potential characters.



(a) Example spread of the chaos state from the middle of the play area.

(b) Example spread of the order state from the middle of the play area.

Figure 6: Mock-ups of the word-state transition.

To put an emphasis on the difference between the two world states, the team is trying to create a noticeable change in music/sound effects when the player transitions between the states. As transition for the objects have a progression depending on how much of the body is covered in the transition, the musical change can progress by the same amount, making it seem like the player is swallowed by the new state. This might be achieved by fading between music or using reverb zones to alter the base music.

1.1.6 Further Sketches

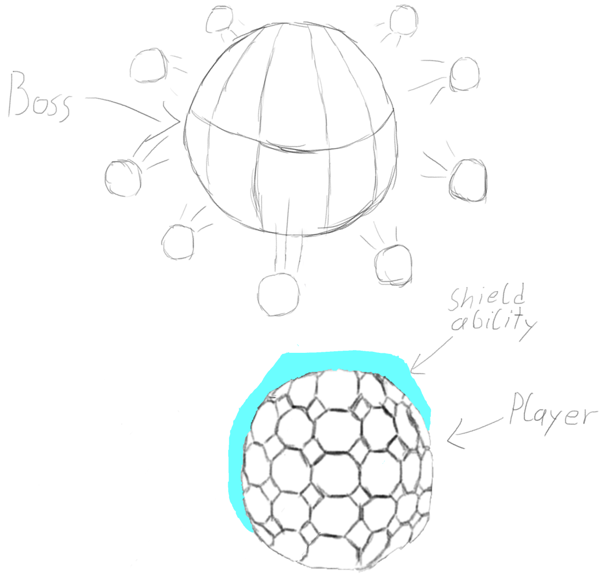
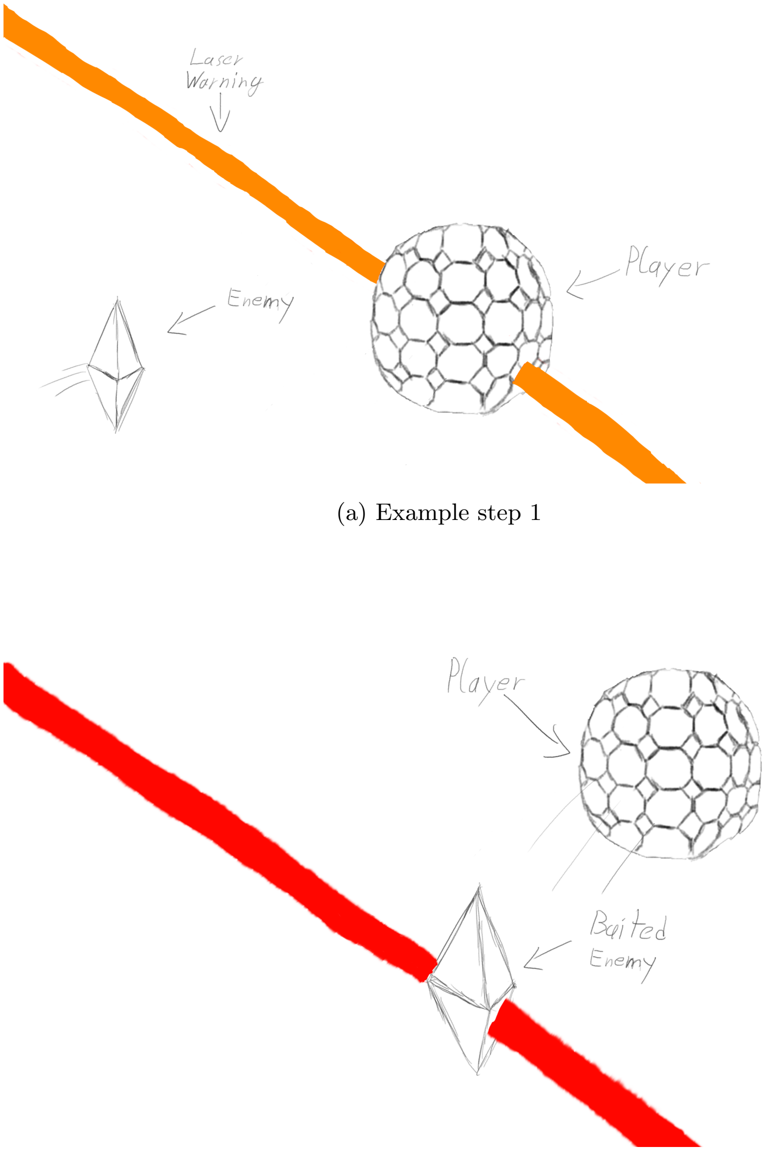


Figure 7: Example boss fight.



(a) Example step 1

(b) Example step 2

Figure 8: Example of how the player can bait enemies in the chaos state.

1.2 Technical Achievement

As described in 1.1.5, the world state transition has many layers. Implementing a smooth transition for gameplay, visuals and sound while maintaining a playable experience for the player is crucial. The team's technical achievement is therefore the transition itself with all its sub-components. The most important ones are listed here:

- Visuals: Transition shader and object shaders in two different versions, one per world state
- Audio: Smooth noticeable change in music based on the player characters transition progress
- Game logic: Change in player character behavior, enemy AI behavior, interactions between player and enemies

1.3 "Big Idea" Bullseye

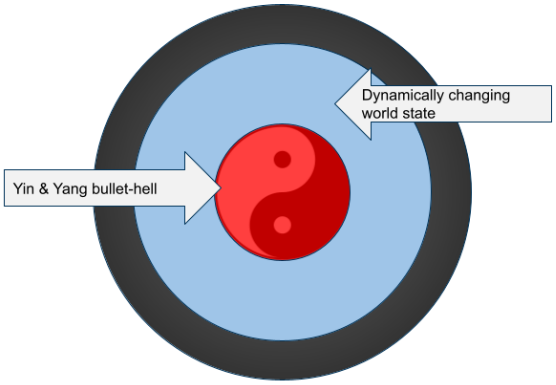


Figure 9: The game's "Big Idea" Bullseye.

1.4 Development Schedule

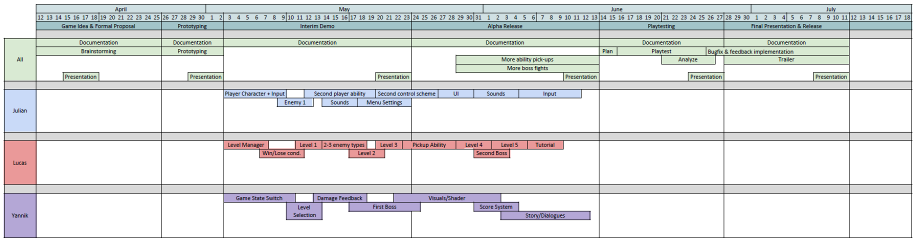
1.4.1 Layered Schedule

1. **Functional minimum**
    - One enemy type
        - Basic model
        - Behavior
        - One ability per world state: shooting/shielding
    - Player character
        - Basic model
        - Basic input
        - One ability per world state: shooting/shielding
    - Switch between game states
        - Basic shader
        - Change of game logic
    - One level
        - Set amount of enemy waves
        - Game over: win/lose condition
2. **Low target**
    - 2-3 enemy types
        - Models
        - Design
        - Behavior
        - One ability per world state
    - Player character: Second ability per world state
    - Basic sounds
        - Bullet sounds
        - Hit sounds
        - BGM
    - Menu: Level selection
    - Second level: Making use of new enemy type
    - Damage feedback, e.g. bullets, parries, ...
3. **Desirable target**
    - Input: Second control scheme
    - Menu: Settings
    - One pick-up ability: One effect per world state
    - One boss
        - Design

    - Two models, one per world state
  - Third level: Making use of boss and pick-up
  - UI
    - Gauge/Timer for world transition
    - Boss UI
  - Visuals
    - Bullet shader
    - Better shader for transition
    - Substitution of potential placeholder models

4. **High target**
   - Input
     - Controller support
     - Custom control scheme
   - Sound design: Sound transition with world state
   - Score system
   - Story/Dialogue
   - More ability pick-ups
   - More boss fights
   - More levels
     - Making use of new bosses and pick-ups
     - Tutorial
5. **Extras**
   - Endless mode
   - Different player characters
   - Local multiplayer
   - Level editor

1.4.2 Timeline



Important: The length of the tasks was sometimes adjusted to fit the text, so it is not a good indicator of how long exactly the tasks will take.

Figure 10: The estimated timeline and task distribution for this project. A clearer/bigger version can be seen at `https://wiki.tum.de/display/gameslab2021summer/Team+Equilibrium`.

1.4.3 Task Overview

In the context of the following table, *p* is short for *person*, e.g. 1h/p stands for *one hour per person.*

Task	**Description**	**Assigned people**	**Estimated time**
Documentation	Writing, sketching, mock-ups, brainstorming ideas, ...	All	50h/p
Presentations	Preparation, discussion, ...	All	4h/p per presentation
Trailer	Editing, storyboarding, ...	All	10h
Prototype	Design, creating, ...	All	9h/p
Character	Design, art, animation, ...	Lucas, Yannik	4h/p
Level Design	Waves, transitions, ...	Lucas	16h
Gameplay Design	Weapon design, enemy types, ...	All	12h/p
Audio	SFX + BGM, implementation	Julian	12h
Visuals	Bullet shaders, particle effects, ...	Yannik	20h
Transition	Game Logic	Lucas	20h
Transition	Shader	Yannik	24h
Transition	Sound	Julian	16h
Enemies	First enemy	Julian	8h
Enemies	Second to forth enemy	Lucas	16h
Enemies	First boss	Yannik	12h
Enemies	Additional enemies and bosses	All	6h per enemy
Player	Input, abilities, ...	Julian	28h
Player	First pick-ups	Lucas	8h
Menu	Level selection	Yannik	4h
Menu	Settings	Julian	8h
UI	Design, art	All	4h/p
UI	Implementation	Julian	4h
Tutorial	Dedicated level, pausing for explanation	Lucas	12h
Story	Writing dialogues, implementing	Yannik	12h
Playtesting	Implementation of feedback from playtesting sessions	All	∞h/p

1.5 Assessment

The main strength of *Project: Equilibrium* is the fast and engaging but simplistic gameplay. Furthermore, it stands out from traditional bullet-hell games because of the state transition, which makes it rather unique. The state transitions are also the most interesting part of the game, specifically the interactions between enemies and the player character during them. If done correctly, the players can have different tactics like delaying or forcing a transition because of certain advantages, which gives the simple gameplay another level of depth for those who are a more serious gamer type. However, getting this transition right is crucial to the success of the idea.

The target audience are mostly fans of shoot'em ups or bullet-hell games. Nevertheless, *Project: Equilibrium* still offers incentives for those who are not typically fans of that genre due to its uniqueness. The players kill enemies while avoiding their own death by constantly managing the different abilities that are at their disposal at the current time. Players who want to take the game more seriously can try to aim for optimization of state transitions and increasing their high-score. The virtual world is rather abstract due to its simple art style, which leads to the story and lore being a secondary contributor to the world. The most important criteria for success is a fluent state transition that maintains immersion and game feeling. Furthermore, the game should be fun to play for bullet-hell fans and beginners alike.

2 Prototype

2.1 Prototype Goals

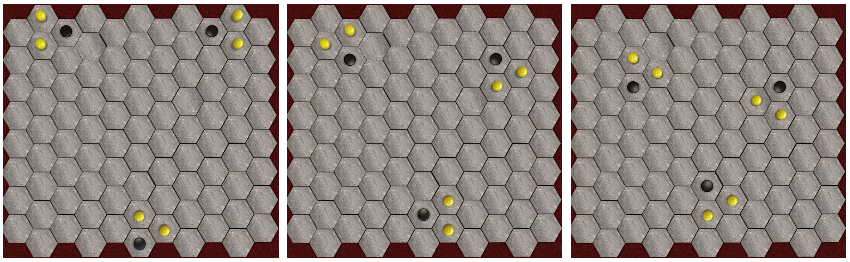
The prototype's goals are to demonstrate and test the core game mechanics, like shooting, shielding and the transition between world states. It should become clear if the game principle has the potential to be fun or if major changes to the game idea have to be made. It should also help to make rough estimates with regard to balancing things like player, enemy and projectile speed, how fast the transition spreads and so on. Furthermore, the prototype should serve to figure out reasonable behavior for the enemy AI and composition of waves.

2.2 Prototype Description

2.2.1 Overview

The prototype is made in Tabletop Simulator[2]. The units move on a grid consisting of hexagonal tiles. These can be flipped to represent the transition between world states. The prototype is played as a turn-based board game. Turns are executed in the following order:
Transition spreading → Spawn waves → Bullets → Player → Enemies.

In the order state enemies have a shield, which occupies two adjacent tiles and moves clockwise around the enemy unit each turn. As can be seen in Figure 11, they move one tile each turn in a predefined pattern. Meanwhile, the player can move one tile every turn, but does not have to move. They can shoot at enemies with bullets, which move two tiles per turn in a straight line.



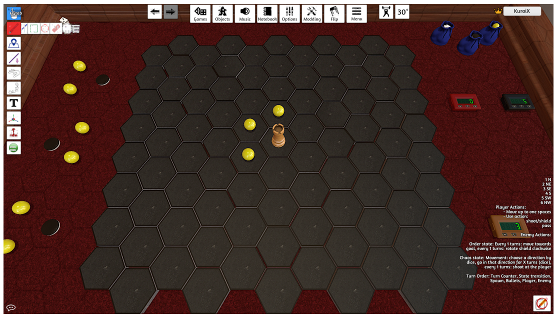
(a) Turn 1 (b) Turn 2 (c) Turn 3

Figure 11: Movement pattern of enemies in the order state.

In the chaos state the roles of player and enemies are reversed. An example of this state can be seen in Figure 12. The latter can now shoot at the player and has no shield anymore. They also move randomly every turn now. How they move is decided by two rolls of a six-sided dice. The first roll determines the direction in which the enemy unit moves, the second determines how far

[2] https://www.tabletopsimulator.com/

it will walk in this direction. This process can also be seen in Figure 13. Enemies can still move only one tile per turn, hence they will move in the same direction in consequent turns until they reach their target tile. The player cannot shoot anymore in this state, but can activate a shield instead. This shield lasts for one turn and has a cooldown of 2 turns. In contrast to the enemies' shields, the player's does not move around him, but is instead three tiles wide to accommodate to this difference. Furthermore, it only stays for one turn.



1 N
2 NE
3 SE
4 S
5 SW
6 NW

Player Actions:
- Move up to one spaces
- Use action:
shoot/shield
pass

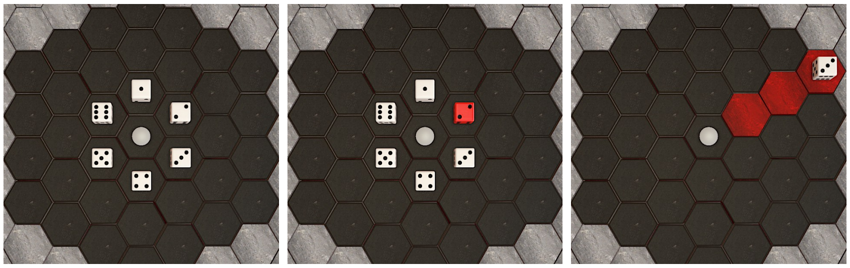
Enemy Actions:

Order state: Every 1 turns: move towards goal, every 1 turns: rotate shield clockwise

Chaos state: Movement: choose a direction by dice, go in that direction for X turns (dice), every 1 turns: shoot at the player

Turn Order: Turn Counter, State transition, Spawn, Bullets, Player, Enemy

Figure 12: Picture of the prototype in the chaos state.



(a) Step 0 (b) Step 1: Direction roll (c) Step 2: Distance roll

Figure 13: Random movement decision process.

As can be seen in Figure 14, both player and enemies can shoot in twelve directions. This means that they can not only shoot in direction of adjacent tiles but also along the edges between these tiles. Since bullets move two tiles per turn, this is possible without creating unusual edge cases.

In both states hitting a shield with a bullet triggers a parry action, reflecting the damage by the bullet to the unit which shot it.

2.2.2 Gameplay

The goal of the player is to eliminate all enemy units without dying. To make this easier and not too frustrating for them, the player has three lives. The transition starts after five turns are completed and the first wave of enemies

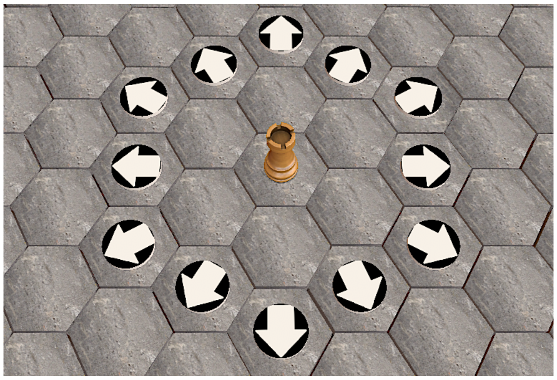
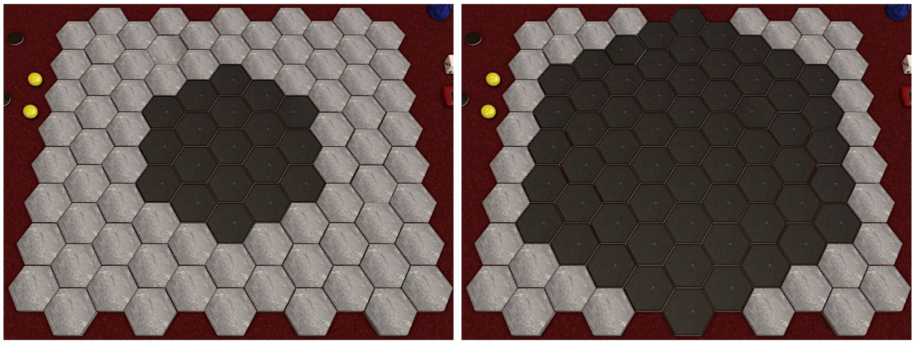


Figure 14: Picture of possible shooting directions.

spawns in the first turn. Since no transition has started and no bullets are present yet, the player begins turn one. If they do not destroy an enemy before it reaches the end of the board, the enemy loops back to the opposite end of the board and continues its pattern. At the beginning of the sixth turn the transition into the chaos state starts. First, only the middle tile is flipped. In the following turns the chaos state spreads two tiles per turn in all directions until the playing field is completely transitioned. The game stays in the chaos state for five turns again, afterwards the order state transition starts, again starting from the middle tile.



(a) Transition turn 2

(b) Transition turn 3

Figure 15: Transition into the chaos state.

After a certain amount of turns the next wave spawns, regardless of whether the player has cleared the current wave.

2.3 Experiences

During creation and playing of the prototype several experiences were made. The realization of how difficult it is to design a fair behavior of enemies and balanced waves is the most notable one. Several edge cases and game mechanics were explored and refined while playing the prototype. For example a decision had to be made of what happens when enemies reach the end of the playing area. It was tested whether enemies should just despawn, start moving in the opposite direction or loop back to the opposite end and continue moving as before. In the end the decision was made to do the latter. It was also decided to let shields parry incoming damage to give the player the option to defend himself more aggressively during the chaos state and make the order state slightly more difficult. Additionally, the speed of enemies, bullets and transition had to be balanced. While the speed of these things in this prototype cannot be directly compared to the speed in the digital game, it will at least give an estimate for initial values to test in the game.

2.4 Revisions to Game Idea

At the time of writing, no major revisions to the game idea where made due to the prototype. Nevertheless, several open questions which we could not find a satisfying solution for before playing the prototype were answered, and many game mechanics were refined. In summary, designing and playing this prototype already helped a lot with the development of the final game, and it probably will continue to do so.