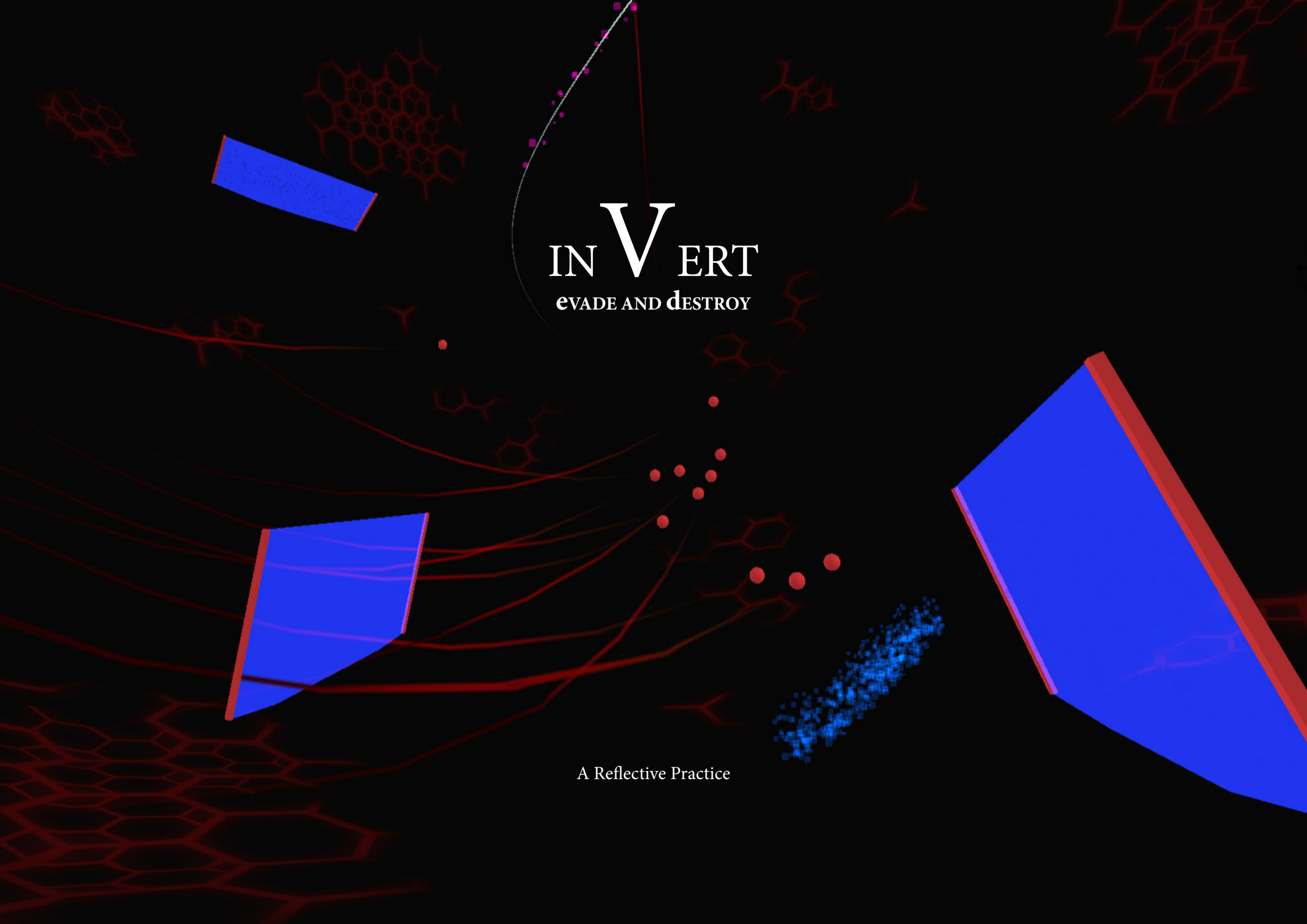

IN V ERT
EVADE AND DESTROY
A Reflective Practice

IN V ERT

eVADE AND dESTROY

A Reflective Practice
08/08/2015

Author: **Hien Quy Tran**
Student Number: 543800

Semester: 4th Semester

Lecturers:
Prof. Thomas Bremer
Prof. Susanne Brandhorst
Oliver Langkowski
Walther Sänger

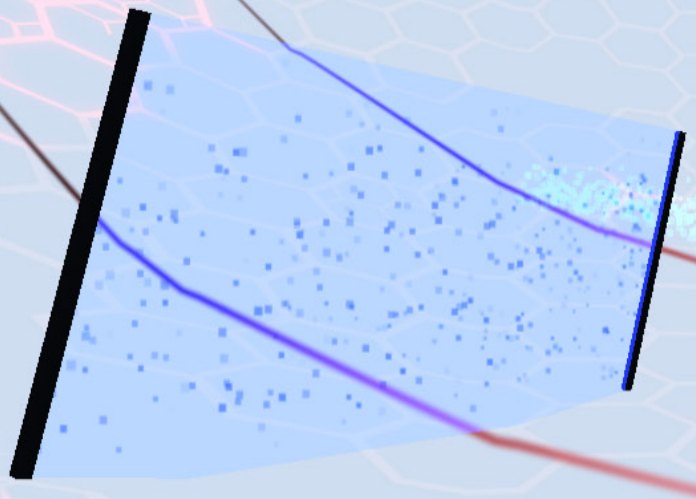
Lecture: Project B. 3 Weeks Experimental Game Jam II
Project: Development of a game with focus on the interface devices

C o n t e n t

1. Introduction to our Project

This project was part of our 4th semester experimental game jam. Our task was to design a game within a three week period. Having this in mind; our time schedule was very tight and the scope of this project strongly limited.

The topic for this group project was: “**Interface Device**”.



1.1 The Development Team

Hien Quy Tran
Game Concept
Programming
Sound Design

Markus Woltersdorf
Interface Technician
Programming

Maximilian Kiese
Game Concept
Programming

Wanda Sonnemann
Visual Development

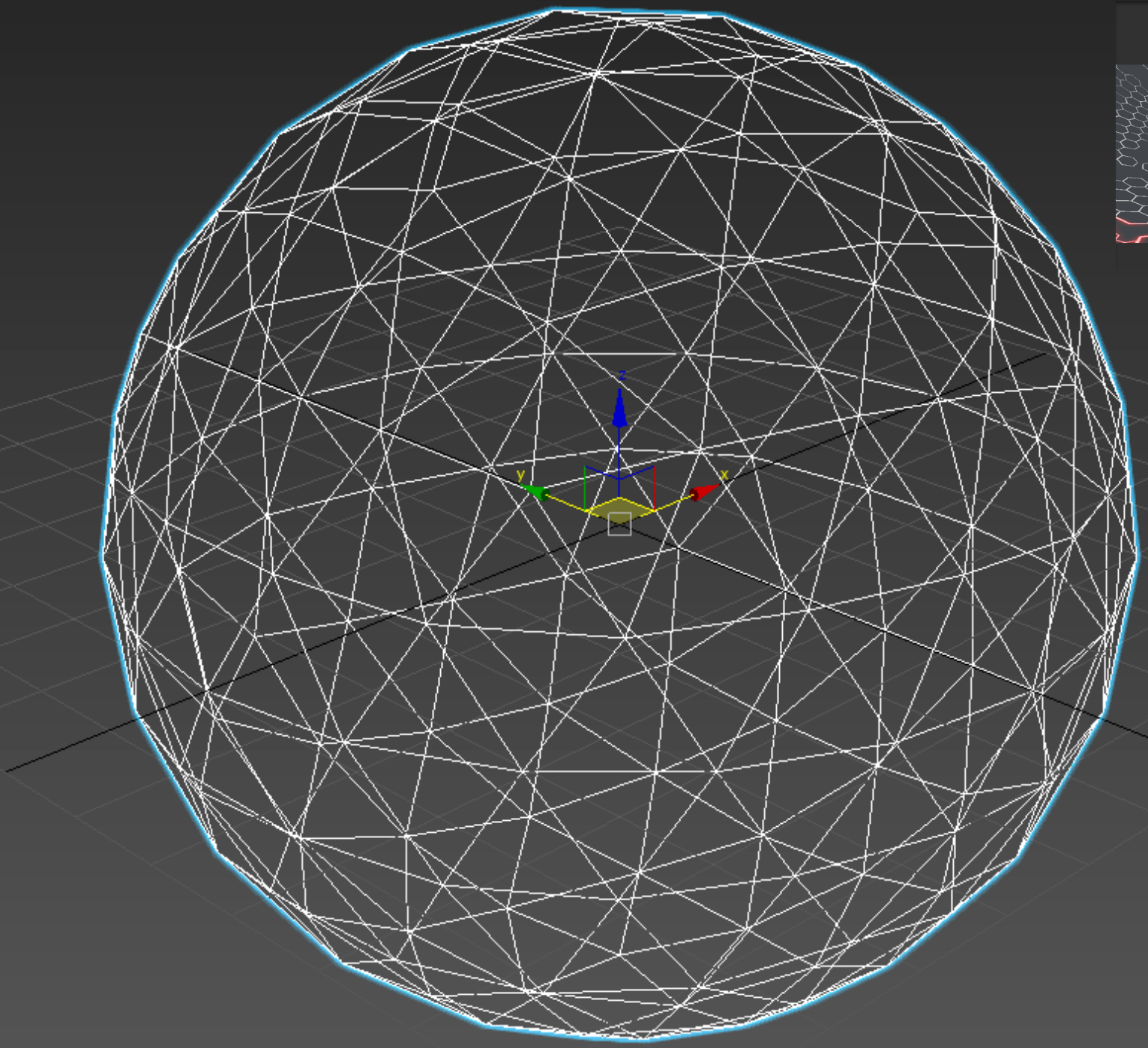
1.2 Concept

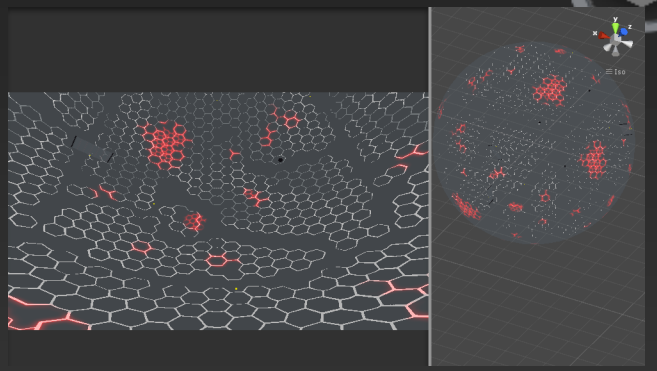
The focus of this project was to design a game specifically addressed towards a chosen interface device and not vice versa.

The interface device we chose is the Oculus Rift. But having the head tracker of the Oculus Rift as the only input device, felt somehow limited to us, for which reason, we decided to include another input device.

We then chose Nintendo's Wii Balance Board, because it is making use of the most natural human body movements, while remaining very simple to handle. It was important to us, to forgo the typical, but mostly unintuitive, on hands controller input, which could destroy immersion.

"We think of the Wii Balance Board as a great expansion to enhance the illusion of Virtual Reality."





Thinking about the Oculus Rift; one of its main advantage is the player's ability to look into any direction and to have a 360° view. In order to fully utilize this feature, we decided, that the player should be surrounded by the playing field, in which the point of interest could be in any direction.

By that we do not meant only all around the player, but also above and underneath. This thought led us to the conclusion, that the playing field should be a sphere, on which the player is moving along its inner surface. The player would for example see the other side of the sphere by looking upwards.

1.3 Game Mechanics

Our game, “inVert - evade and destroy”, is a high score based, single player, segway racing, action game, in which the player is moving along the inner surface of a sphere, trying to reach certain checkpoints while avoiding any contact with enemies.

Enemies are moving straight along their lanes and the amount of appearing enemies is increased by time. At certain intervals, the enemies are starting to chase the player, forming a big mob.

While the player is moving and continuously trying to avoid the enemies, there are also gates appearing on the surface. The player can drive through those gates in order to destroy them. When being destroyed, the gate will burst into pieces and destroy all nearby enemies as well.

The player can use this as an advantage in order to destroy as many enemies as possible. The score is raised for every destroyed enemy and the multiplicator is raised for every destroyed gate. If the player gets in contact with an enemy, the game is over.

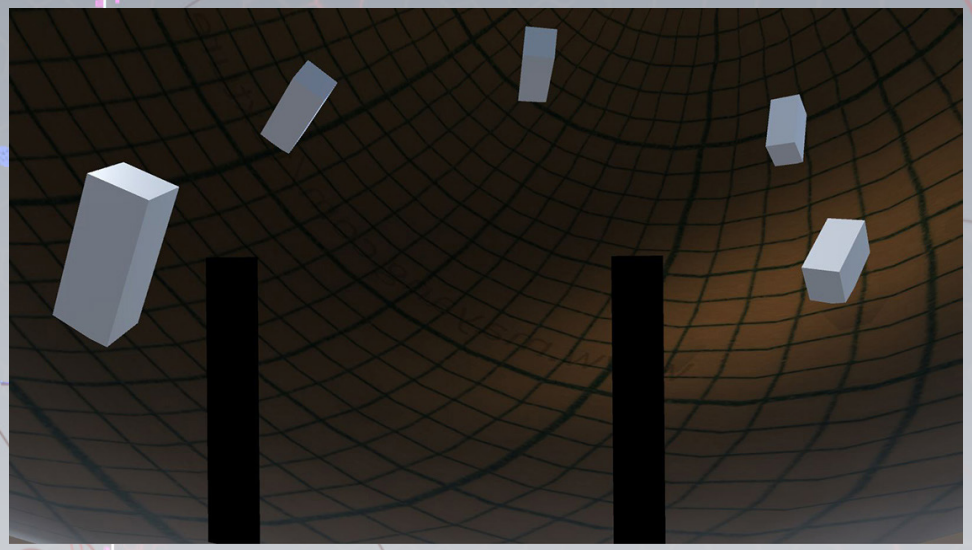
To orientate in space, the player can use the Oculus Rift to look around. Also, the player can move along the surface by leaning forwards, backwards, left or right on the Wii Balance Board.

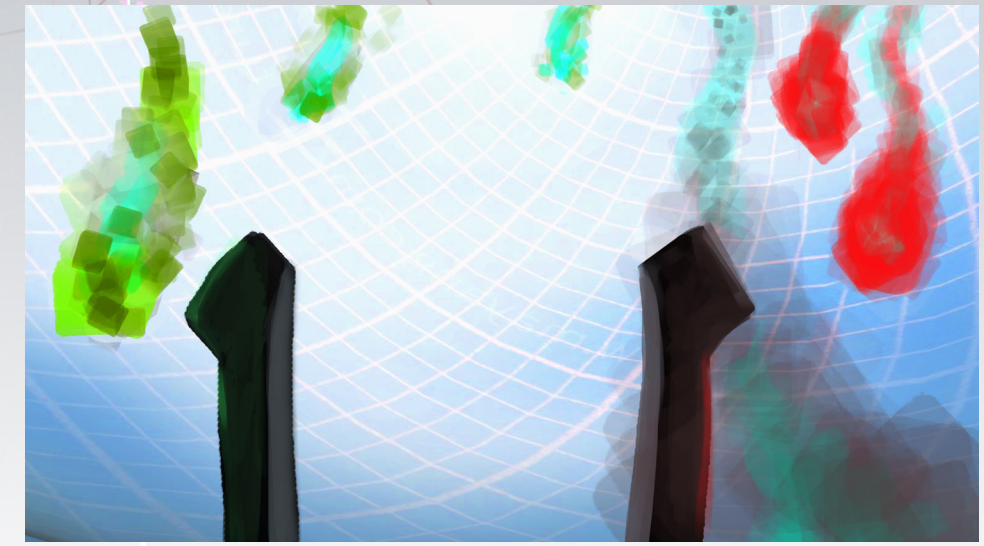
“Using the senses to balance the body, is just as intuitive like turning the head to look around.”

1.4 Art Style

Having the low resolution of the Oculus Rift's development kit and general motion sickness in mind; we were aiming for a rather simple and clean look. We wanted least distraction from gameplay relevant elements as possible and have chosen to use very simple geometries with easily distinguishable silhouettes.

(Following concept arts are made by Wanda.)





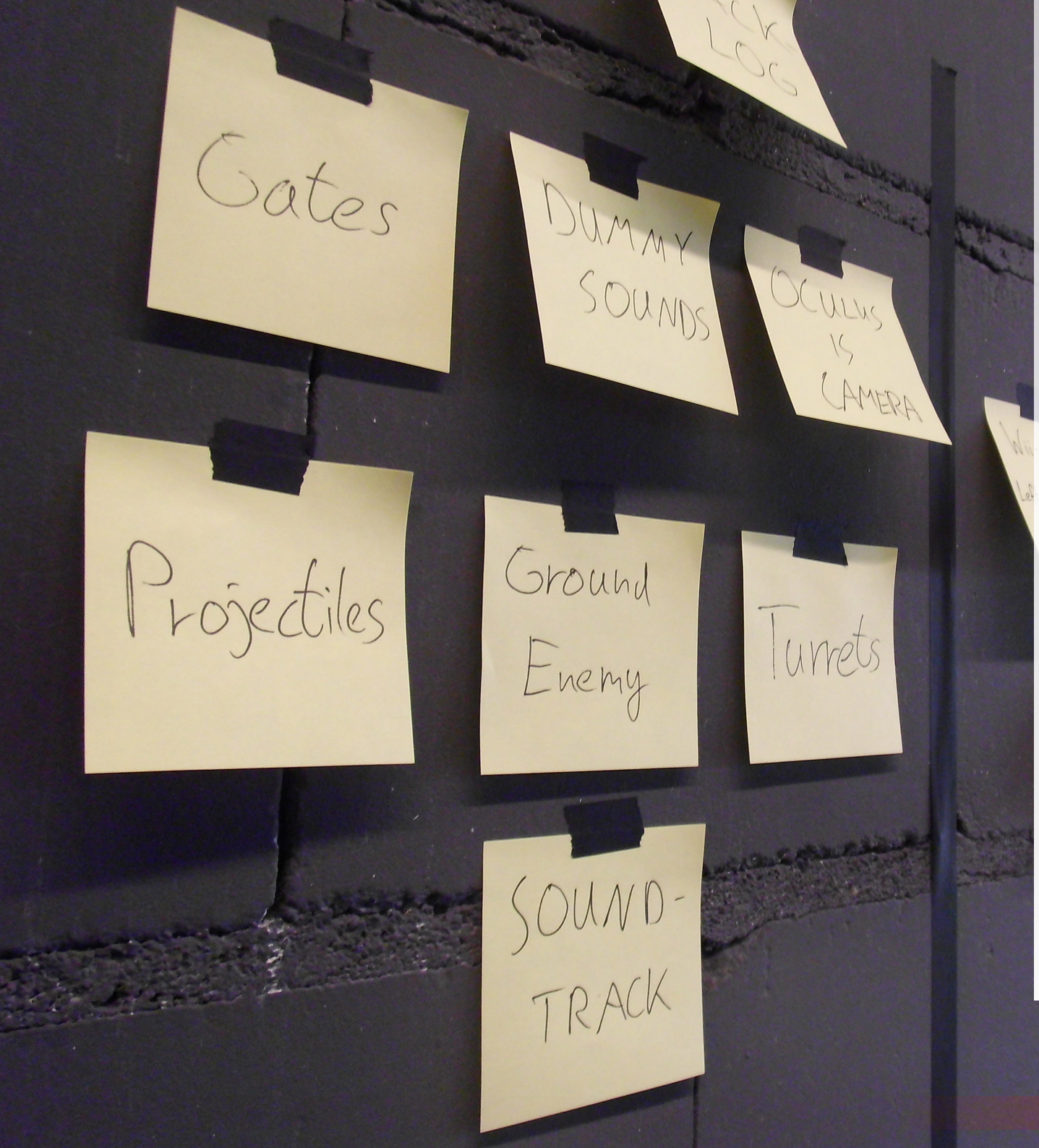
2. Personal Goals and Milestones

Because I have been working with Unreal Engine's visual scripting language in the last two projects, one of my primary goals for this project was to catch up with the text-based scripting language. I have been already using Csharp in the first project: "Armville - Archie's Book Chase".

The Blueprints Visual Scripting Language helped me to understand more complex programming operations, without having to worry about the syntax of text-based languages. Switching back to Unity appeared very consequential to me. It helped me to transfer, as well as utilize the knowledge, which I have been gaining while using the visual scripting language.

I was very curious about the results.

I was also looking forward to design the feedback sounds and improve my capabilities in sound modulation. Furthermore, I was interested in designing a great soundtrack, which reacts and modulates itself on the fly accordingly to the game state.



3. My Task Area

Before anything else, we needed a concept for the gameplay and a working scoring system. In addition to that, I was given the task to completely design and implement the acoustic feedback.

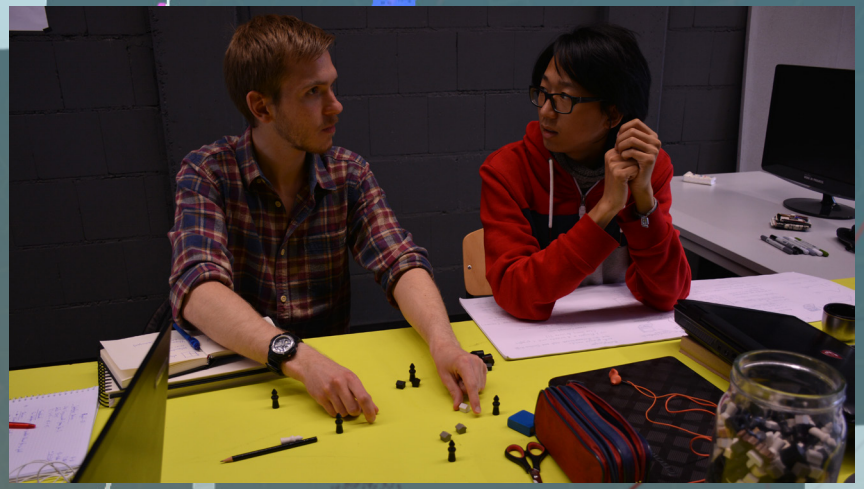
Having three programmers in our team, it was essential to coordinate and share our tasks carefully. One of my other tasks was to create all necessary game objects, like gates and enemies, as prefabs for prototyping purposes. I was also in charge of the spawning system of those.

Other minor tasks I had to tackle over the course of the project involved the creation of a pole-less sphere, a visual feedback by changing the color scheme as well as arranging the title and high score menus.

3.1 Developing Gameplay Concepts

Our mission for this project was not only to work out a game concept, which would support the Oculus Rift and the Wii Balance Board, but furthermore, is noticeable designed around those devices. We wanted to make them truly a part of the experience.

In order to be able to start prototyping as soon as possible it was necessary to develop several gameplay concepts. We agreed on the sphere as the playing field and continued to develop the gameplay concept with having the Oculus Rift and the Wii Balance Board in mind. All the little details, like for example the scoring system and the movement system had to be taken into account.



Max and me developing ideas for gameplay concepts.

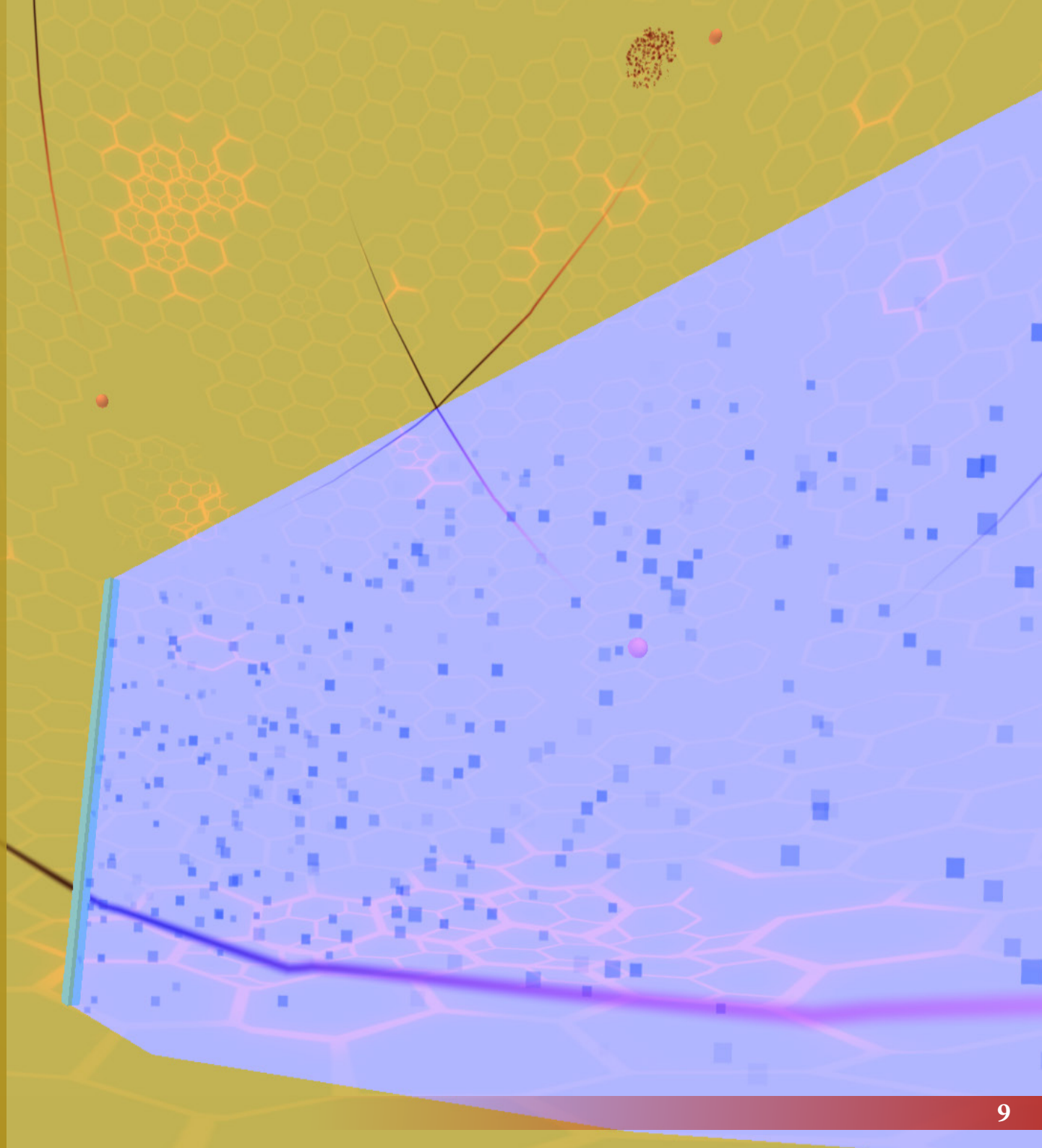
non-digital prototyping

Our original concept was a very different one and included Microsoft's Kinect v2 sensor in combination with a crate full of sand. After iterative processes of non-digital prototyping and redesigning, we came to the conclusion to drop the idea for this project.

This led us to the current idea, with the exception, that we had enemies, which were chasing the player continuously. This did not work out very well. Looking behind oneself, while trying to keep moving straight forward on the Balance Board, turned out to be more difficult than expected

After several changes and tweaks, we finally reached the point, which suited our expectations. It was very interesting to see that concepts, which were working on paper, did not work out while prototyping.

I learned, that it can be very helpful, to take non-digital prototyping material and go through the rule set of the game in order to identify potential problems early enough.



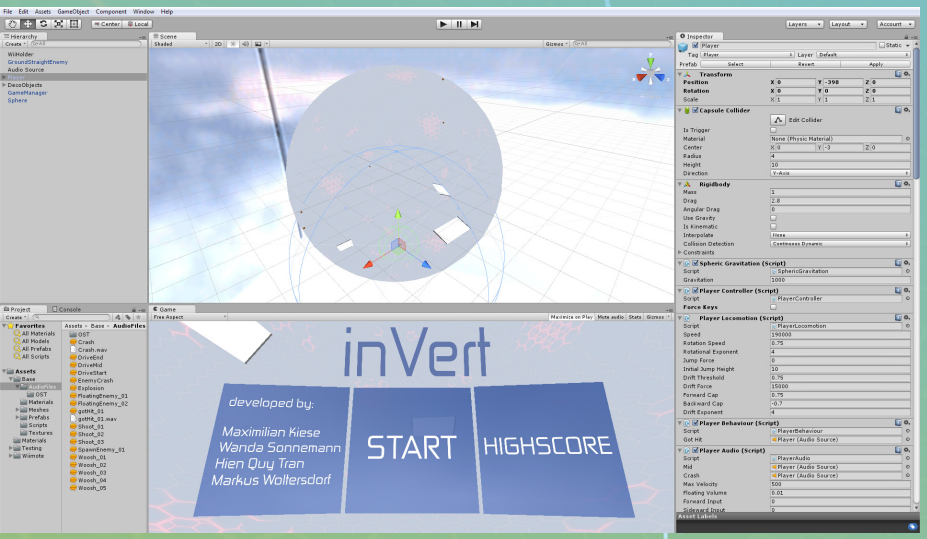
3.2 Creating Prefabs

The developed gameplay concepts helped us to identify, what game objects were in need for prototyping.

My task was to create a stack of scripts and game objects as prefabs, which we could use for prototyping. It was important, that everything was created with the modularity in mind. Meaning, that everything had to be developed in a manner, which would allow us to reuse and recombine it again. This should lead to a more efficient prototyping.

The gates are colliders, which are in between of two poles. It destroys itself on overlap with the player. When being destroyed it checks all game objects within a specific radius on the tag. If the game object is tagged as an "enemy", it will destroy it.

The enemies are sphere meshes, containing a script giving them a constant force in the direction of their local forward vector. This makes the enemies keep moving forward. They also got a script which sets the up vector to look at the center of the sphere. In combination of a slight force opposite to their local up vector, they are getting pushed away from the center and onto the surface of the sphere resulting in something like "reversed gravity". If they collide with a game object, which is tagged as "player", the game ends.



Creating a variety of prefabs was a perfect programming task to begin with. It helped me to get back into text-based scripting. Each of the different game objects came with an unique programming problem. I was confronted with a diverse set of problems without having the difficulty being set to high. Because each game object by itself was representing a rather simple problem, I was able to solve them all one by one. Also it was very helpful to put extra attention on the modularity and consistency of the objects, which, I think, can be helpful for future projects.

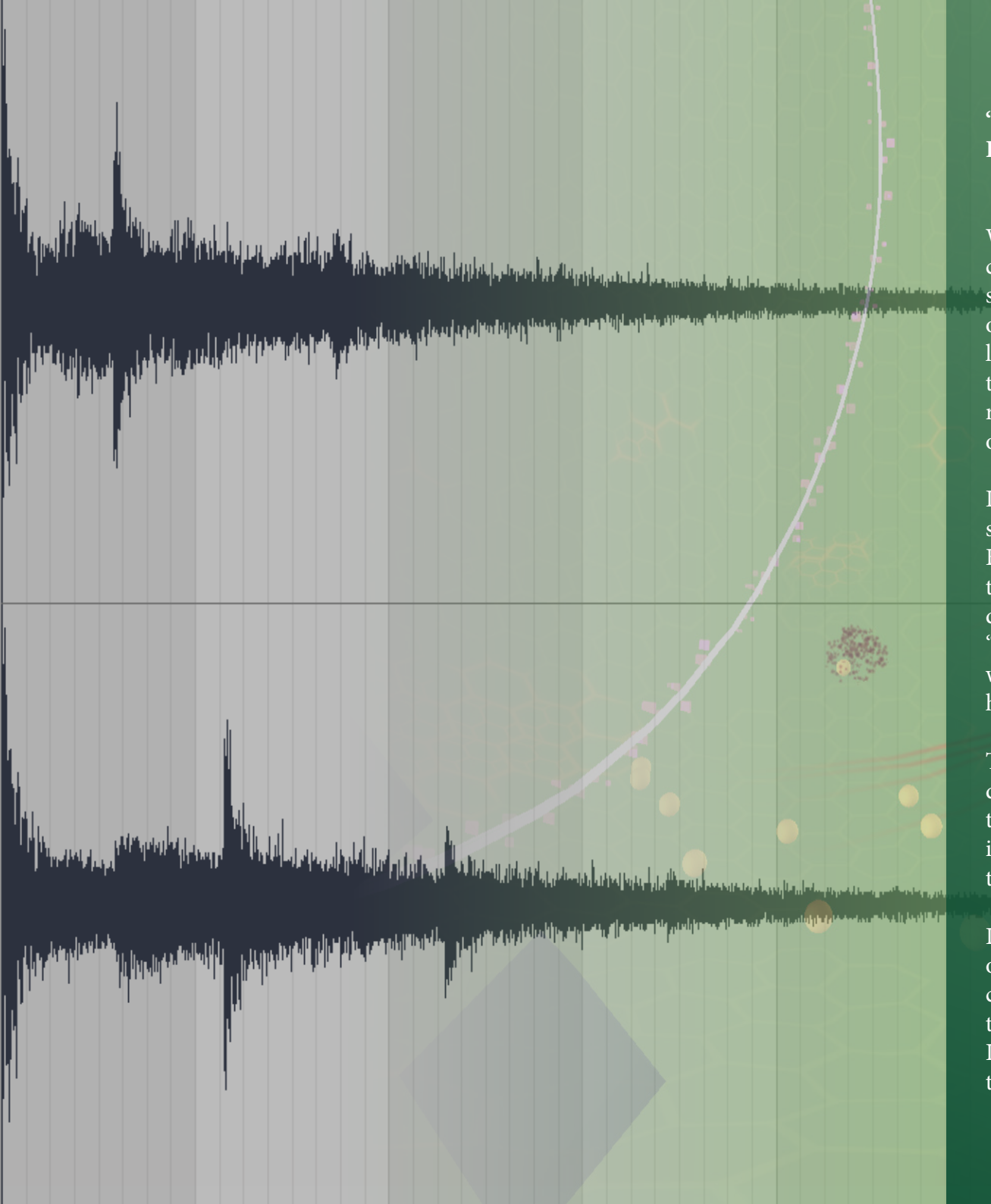
There have been several other game objects like collectibles or projectiles shooting turrets, which didn't make it into the final game. Due to the simplicity of these game objects, I do not see the necessity to further explain them.

I am very satisfied with the result and I am proud to see, that I have not forgotten the text-based scripting. I felt comfortable and was immediately able to access and transfer all the knowledge from the visual scripting to the text-based scripting.

3.3 Sound Design

Having successfully designed the feedback sounds and soundtracks for previous projects like “Armville - Archie’s Book Chase” and “Odd One Out - Humans Not Allowed”, this was not a new area for me. But this also meant, that the expectations I had for myself were set pretty high.

One goal I had set myself was to modulate all sounds from scratch, without using any sound libraries. The idea was to learn how to pinpoint a specific sound while having total control over the creation of itself and without being influenced by any external factors. This ability does not only gives me unlimited recreational possibilities, but also might come handy for future projects, in which I cannot or do not want to acquire the license for a particular sound.



“Being passionate about making music in general, I also wanted to create a soundtrack for this game”

While movies do not give any control away to the consumer, games do. This makes music in games sometimes difficult to handle. The tension in games is often situational, which makes prearranged composition less efficient and less powerful, if they do not adapt to the game state. Therefore, I wanted the soundtrack to be reacting highly dynamically to the state of the game, in order to enhance the gaming experience.

I have been already experimenting with a slightly dynamic soundtrack in the previous project “Armville - Archies Book Chase ”. However, the soundtrack was mainly linked to the overall progress of the game and in my opinion, did not react well enough to the game state. Also, while “Armville”, related to the overall game progress, has a very well defined starting and ending point, “inVert” does not have any predefined ending point.

The end of the game comes with the event of the player’s death. Since there is a theoretical possibility for the player to survive either a few seconds or not die at all, it is really impossible to pinpoint when exactly a player is halfway through the game.

Instead of binding the soundtrack to the overall progress of the game, I had to find a way to let it react to the current game state through other variables. Because the pacing of our game is constantly raising and falling, I wanted the music to be bound to the pacing and the tension of the current game state.

Currently, the soundtrack consist out of eight cues, which are added in the following order:

Clock:
A closed hi hat giving a constant rhythm and supporting the perception of time. Also being the first cue to be played, in order to prepare the player for the incoming tension.

Atmosphere:
A synthesizer giving a constant fluctuating tone as the general atmosphere of the game. It is introduced with a white noise and prevents the complete silence, when the music drops out.

Kick:
A very strong kick in order to apply pressure and reflect the overall pace of the game. Being the first element in the lower frequency, it sticks out more than others, which increases the effect of pressure.

Hi Hat:
A simple open hi hat is being added after the kick. Makes the arrangement more interesting, but also doesn't take too much attention away from the kick.

Clap:
A typical off beat clap combined with some untypical rhythmical whip sounds to join up all elements and complete the rhythm.

Effects:
Some random sweeper tones and white noises operating as general teasers.

Base and **Lead:**
Providing a rhythmical base in the lower frequency and a melodic synthesizer with overdrive effect in the higher frequency. As the first melodic elements, they loosen up the pressure and bringing in the necessary vibe and character to the soundtrack.

For each passing beat, in which the player does not destroy an enemy, elements of the music are dropping out. The elements drop out in the following order, leaving just the atmosphere behind:

kick → clap → hi hat → clock → lead → base

As soon the player destroys an enemy, all elements are dropping back at once.

The longer it takes the player to destroy an enemy, the more elements are being cut out, which constantly raises the tension and increases the effectiveness of the incoming drop in. It also makes perfect sense, because the longer none of the enemies are destroyed, the more enemies are gathering in the arena.

In the same principle, as a DJ is raising the tension on the crowd by cutting out elements and fading out the lower pitch sounds, my script for the soundtrack is also cutting out elements in order to raise the tension and prepare for the drop in.

Furthermore, the soundtrack is also changing accordingly to the state of the enemies. If the enemies are switching to the mode in which they chase the player, the hi hat, base and lead is replaced by an even more fast paced arrangement.

I have been using Ableton Live to create the feedback sounds and the soundtrack. I tried to be especially careful while equalizing the sounds, so that each sound would have its own frequencies within the sensible acoustic spectrum. This helps the sounds to stick out and remain distinguishable even when several sounds are overlapping.

I am very satisfied with my results in designing sounds and I think, that I have made a huge leap forward. The soundtrack has a very progressive character, which is exactly what I was aiming for. The highlight for me is, the dynamic aspect of the music and how well it adapts to the tension of the game.

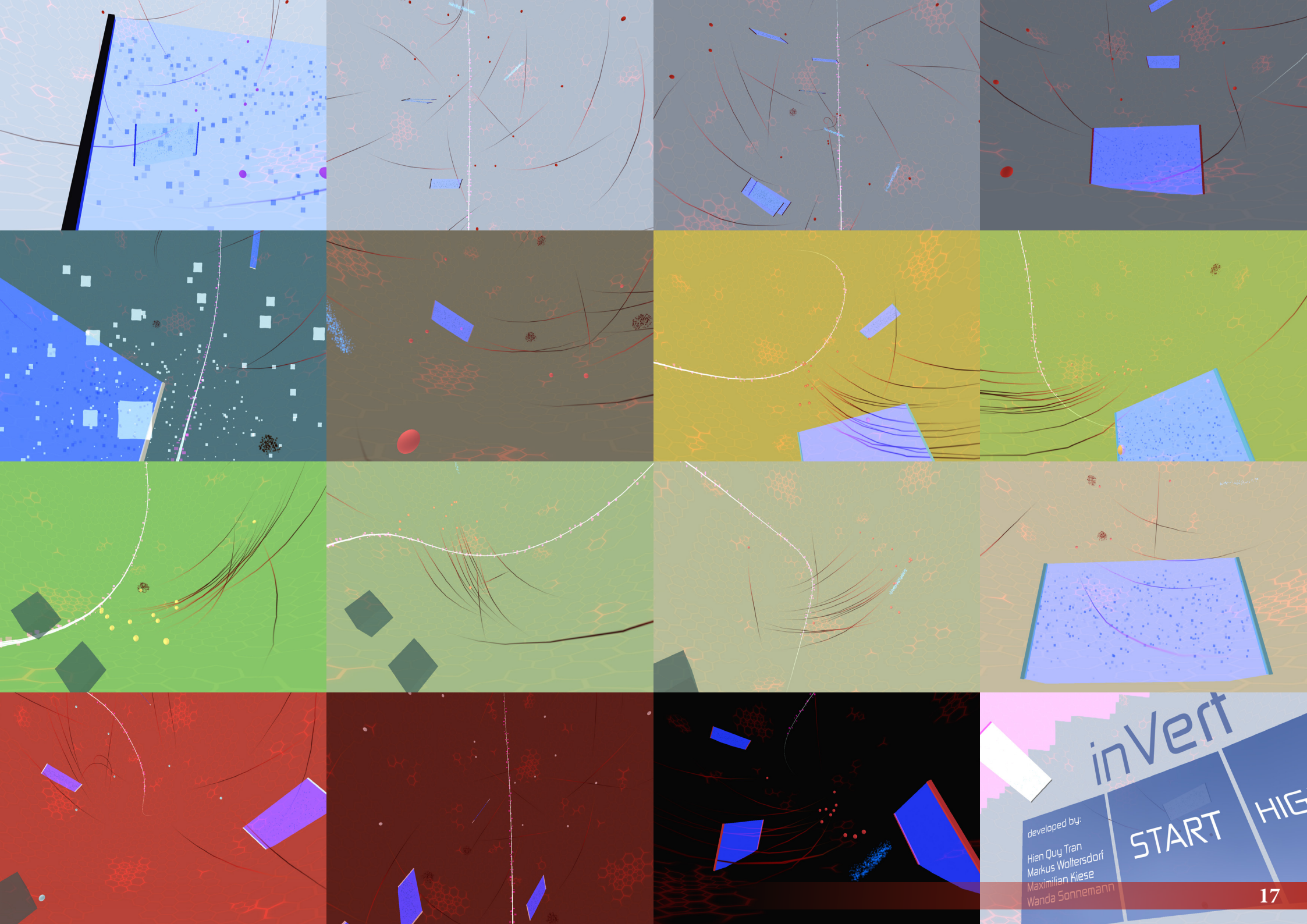
```
    public Color currentColor;
    public Color targetColor;
    public Color currentPoleColor;
    public Color targetPoleColor;
    public Color currentEnemyColor;
    public Color targetEnemyColor;

    float changeTime = 0;
    public int nextColorScheme = 0;

    void Start () {
        currentColor = mainColor.color;
        currentPoleColor = polesColor.color;
        currentEnemyColor = enemyColor.color;
    }

    // Update is called once per frame
    void Update () {
        UpdateSphereColor ();

        if (Input.GetKeyDown (KeyCode.Space)) {
            nextColorScheme = nextColorScheme + 1;
            if (nextColorScheme > 6) {
                nextColorScheme = 0;
            }
            SetRandomTargetColor (nextColorScheme);
            changeTime = 0;
        }
    }

    public void SetRandomTargetColor (int colorScheme) {
        if (colorScheme > alternateMainColor.Length - 1) {
            colorScheme = 0;
        }
        targetColor = alternateMainColor [colorScheme];
        targetPoleColor = alternatePoleColors [colorScheme];
        targetEnemyColor = alternateEnemyColors [colorScheme];
        changeTime = 0;
    }

    void UpdateSphereColor () {
        if (changeTime < 4) {
            currentColor = Color.Lerp (currentColor, targetColor, Time.deltaTime);
            mainColor.color = currentColor;
            currentPoleColor = Color.Lerp (currentPoleColor, targetPoleColor, Time.deltaTime);
            polesColor.color = currentPoleColor;
            currentEnemyColor = Color.Lerp (currentEnemyColor, targetEnemyColor, Time.deltaTime);
            enemyColor.color = currentEnemyColor;
            changeTime = changeTime + Time.deltaTime;
        }
    }
}
```

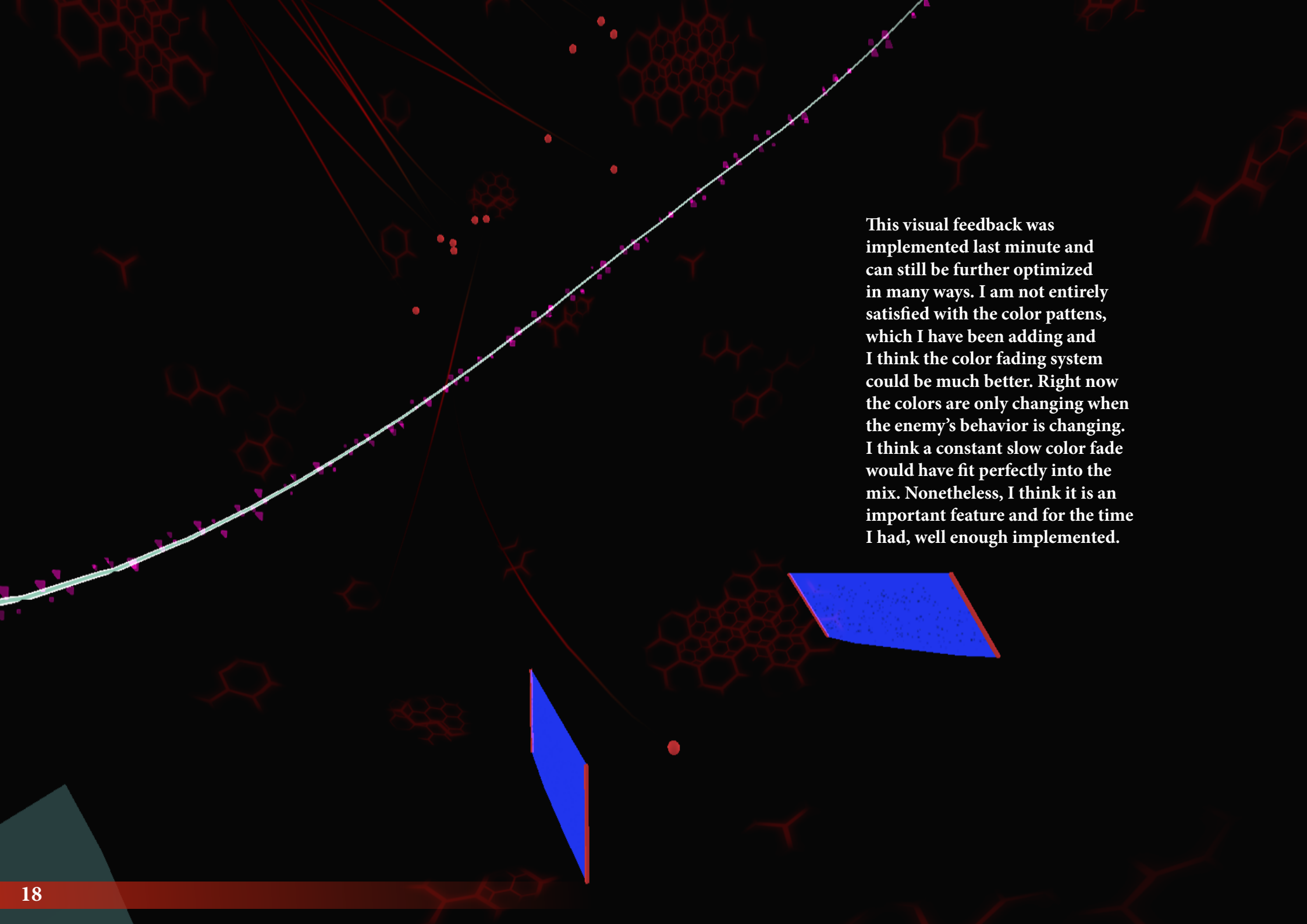

3.4 Visual Feedback

After having successfully implemented the acoustic feedback, I still saw opportunities to enhance the experience by adding more visual feedbacks.

In some cases, the player would notice the change in the soundtrack, but would not notice the change in the enemies behavior. This situation could provoke confusion, because the sound changes remarkably, but nothing seems to happen. That is why I have been implementing a script, which changes the color pattern of the game, whenever the music changes drastically. Furthermore, it gives the player a major feedback about the progression of the game.

In order to achieve this, I wrote a script, which accesses all used materials directly. By changing the attributes of an material, which is in use, all game objects that are linked to it, can be altered in their appearance by runtime. This way, for example, the script is able to change the appearance of all enemies at the same time, by just changing the referred material of the enemies. I added several color patterns, so that the script can blend between them.


inVert
START
developed by:
Hien Quy Tran
Markus Waltersdorf
Maximilian Kiese
Wanda Sonnemann



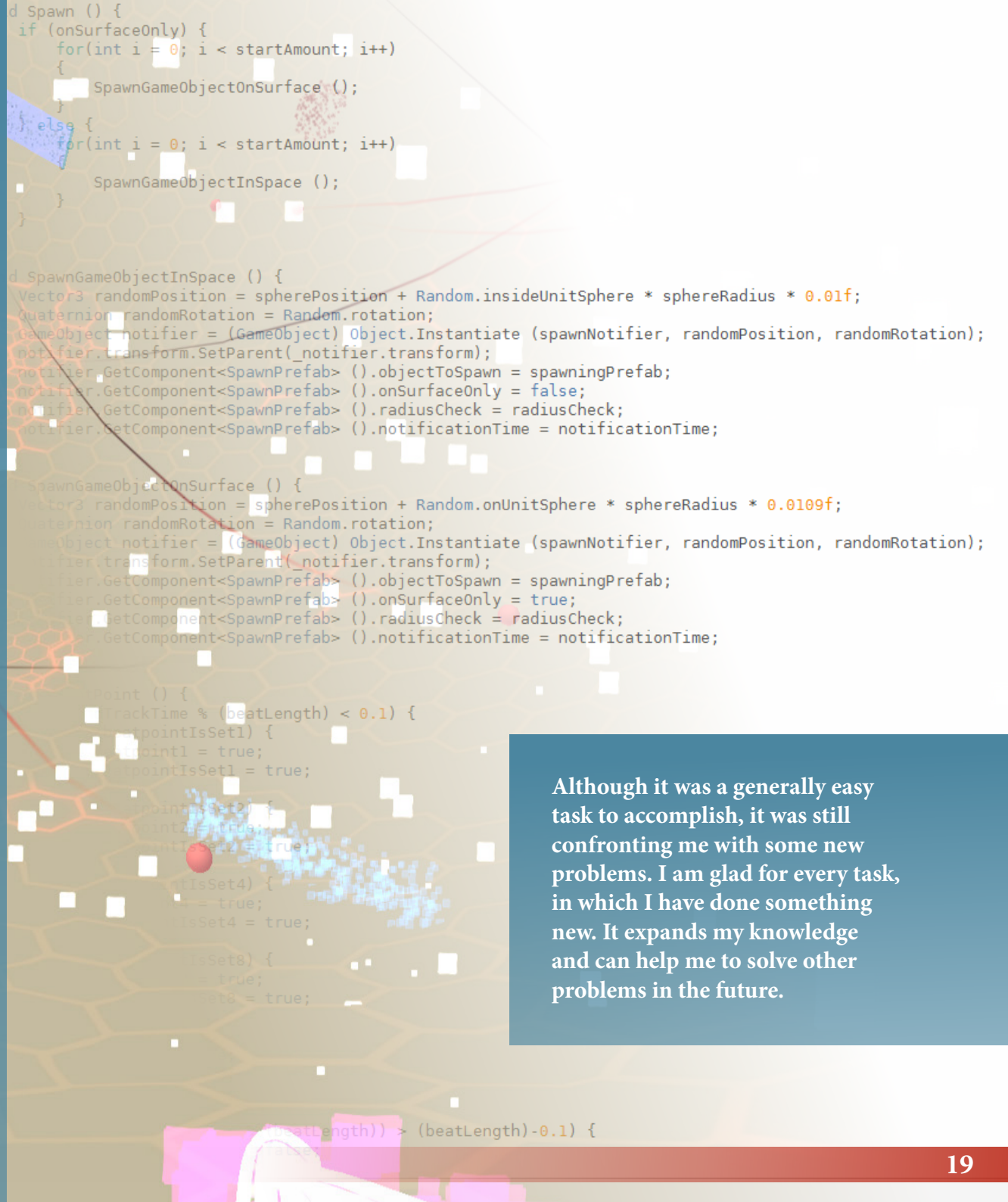
This visual feedback was implemented last minute and can still be further optimized in many ways. I am not entirely satisfied with the color pattens, which I have been adding and I think the color fading system could be much better. Right now the colors are only changing when the enemy's behavior is changing. I think a constant slow color fade would have fit perfectly into the mix. Nonetheless, I think it is an important feature and for the time I had, well enough implemented.

3.5 Spawn Point Manager

Having the stack of prefabs done, it was important to find a way how to utilize the prepared items comfortably for prototyping purpose without having them to be dragged into and out of the scene every time. For that matter, I have written a script, which spawns the desired game objects in a desired pattern under desired restrictions.

Generally, the spawn point manager has a public variable for game objects, in which the user can define the prefab to be spawned. In addition to that, the script knows the length of a beat in seconds and can spawn the objects in definable intervals accordingly to the music. The user can also define whether the game object should spawn on a random location within the sphere or on a random location on the surface of the sphere. This is accomplished by a random normalized vector, going from the center of the sphere. In order to spawn within the whole volume of the sphere, the vector has to be multiplied by a random float between 0 and the radius of the sphere. In order to spawn on the surface of the sphere, the vector has to be multiplied exactly with the radius of the sphere.

There can be multiple Spawn Point Managers in the scene, each handling their own prefab, spawn amount, starting time, interval length and ending time.



Although it was a generally easy task to accomplish, it was still confronting me with some new problems. I am glad for every task, in which I have done something new. It expands my knowledge and can help me to solve other problems in the future.

4. Recap

I am very satisfied with the outcome of this project and would mark it as a great success. I have completely fulfilled the goal to transfer my knowledge from the visual scripting language in order to apply it to the text-based scripting language.

Working as a team of three highly skilled programmers was an interesting experience. Because each one of us had a different skill set and approach, it was always very interesting to exchange ideas and opinions. Also, working together forced us to work tidy and structured.

Being in the position of having full freedom in designing the sounds was great and I am particularly proud about this the results of the sound design.

I am still very interested in the programming of AI and would love to improve in this area at next possible chance. This time there was no complex AI needed.

In overall, I am very happy with the result of this project and I am very satisfied with my performance.

I am looking forward to the next project!

Impressions of working in our studio: