Unity Final

Introduction

Welcome to "Cyber Rhythm"! This is an exhilarating music game that combines cyberpunk style with music. Put on your VR headset and get ready for an immersive music adventure!

"Cyber Rhythm" will transport you to a futuristic world infused with cyberpunk aesthetics. In this virtual realm filled with music, you'll be a music warrior, tasked with swiftly smashing obstacles using punch-like actions according to the rhythm of the music. Additionally, you'll need to perform specific moves to earn higher scores! It's not just about maintaining your own rhythm and movements, but also about deftly dodging incoming projectiles to avoid deductions! Only those who possess agility and impeccable rhythm can emerge victorious in this cyber music world!

Team Members (Exhibition Participants)

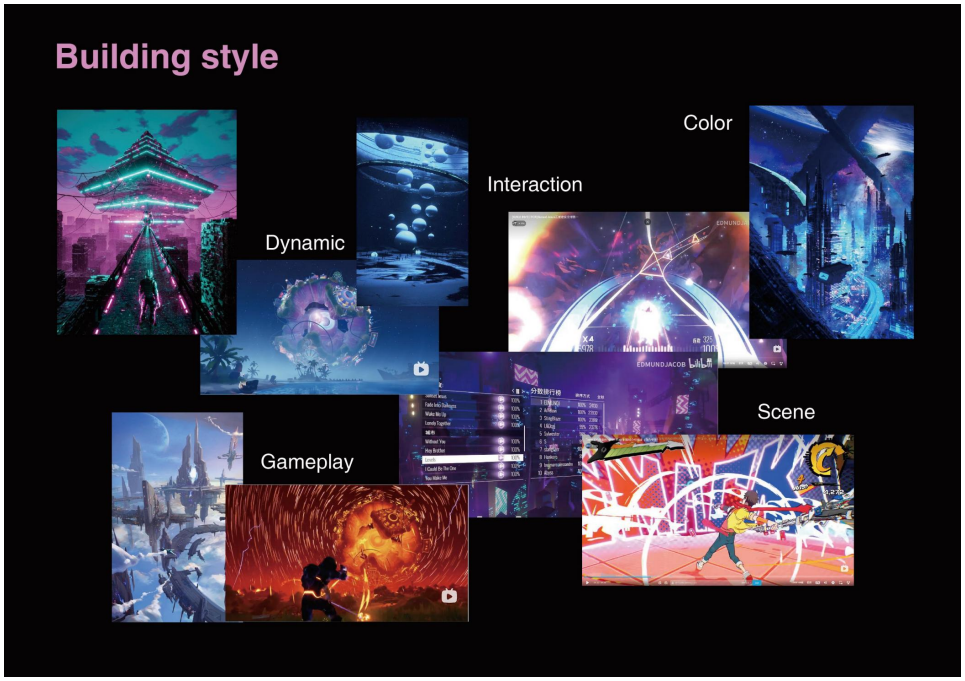
Jinglin Zhang, Kangcheng Deng, Yang Chen, Zhenkai Wang, Yifan Du

Initial Game Division of Labor

Jinglin Zhang

✧ Game concept planning

With five members in our team, it was important to coordinate the time and content of the members' work. At the beginning of the project, we spent a week thinking about the theme and direction of the game. Each team member came up with about five themes and found references to relevant games. After a group discussion, we agreed that a music game was the best choice. Therefore, we started brainstorming about music game themes. I asked everyone to research existing music games, understand their game mechanics and styles, and think about how our games should be innovative. We ended up having several group meetings and finally settled on a direction for the game that combined music with a cyberpunk style.



After deciding on the direction of the game production, I created a project schedule, assigned tasks based on each member's expertise and workload, and set up a timeline for the game production to keep the progress reasonably under control and avoid potential delays. However, during the planning stage, I failed to take into account the serious mismatch between some of the gameplay we wanted to implement and the actual timeline. Each member of the team encountered various difficulties during the production of the game and almost all of the originally planned timelines suffered delays. As an example, we initially planned to create two completely different scenarios with a portal that would allow players to enter another dimension to enhance their visual stimulation and experience. However, in practice, we were so pressed for time as we had three other assignments with deadlines looming that we ended up only being able to use one scene.

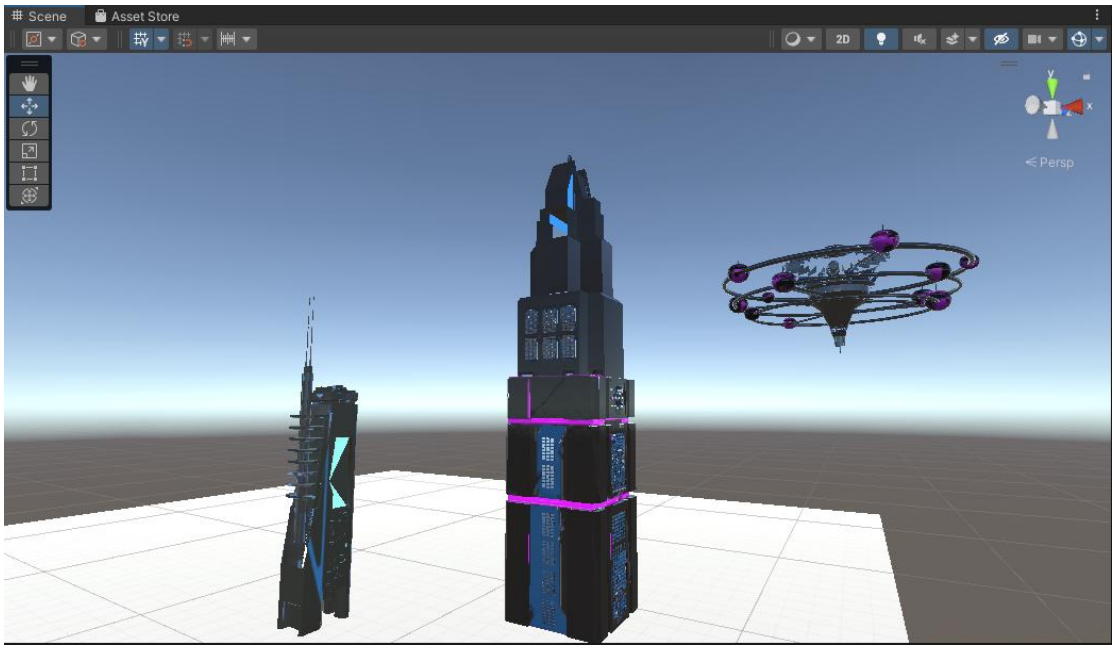
✧ Design of boxing gloves

We did a lot of research on music games and found that most of them use swords or sticks as the striking weapons. However, I wanted to bring a different kind of striking weapon to our game. Fortunately, I was inspired by a pair of boxing gloves hanging in the home of Kangcheng Deng. I searched the internet for images of boxing gloves, but found them all too ordinary. So I started thinking about how to combine them with a cyberpunk style. In the end, I decided to design something with the structure and used some mechanical elements to enrich the appearance of the boxing gloves.

After completing the model, I had planned to add materials in C4D, but there was a problem importing it into Unity. The C4D material balls I used came from a material pack downloaded from the internet, so the exported texture names were very confusing and contained texture maps for the material balls. In Unity I had no idea which map corresponded to which part. I realised that if I continued to import scene models in this way, not only would it be a huge amount of work, it would also lead to material mapping errors. On the advice of Yifan Du, I learned about a software called Adobe Substance 3D Painter which is a good solution to these problems. This software exports the full texture of the model, reducing the amount of work involved in texturing in Unity and providing more accurate results. In the end, I designed boxing gloves in six different colours and after a group discussion we chose a purple colour scheme.

✧ Construction of scene models

Originally I was going to use the scene models available online, but after checking a number of websites, I couldn't find any that were particularly suitable. Some of the scene models that I thought were good were either unavailable for download or overpriced. It then occurred to me that I could build my own scene models and then reuse them in Unity to build the whole city. I initially expected to only need to model 15 buildings, but the actual amount of modelling I did was far more than I expected. I originally thought it would only take a month to complete, but it actually took about two months to model and map. I encountered many problems during the process, mainly because I was not familiar with Adobe Substance 3D Painter software, so I spent a lot of time learning the tutorials. I remember encountering three technical difficulties. Firstly, I was colouring different areas in C4D, but after importing into Adobe Substance 3D Painter, there was only the model and no colour. In the end I found that exporting an FBX file was more suitable than an OBJ file. Secondly, as I had not made the UVs correctly in C4D, there were many misalignments when applying the materials in Adobe Substance 3D Painter. The third problem is the amount of jaggedness that occurs around the edges of the material in Adobe Substance 3D Painter when using colour masks to draw in the appropriate colours. When I encounter these technical problems, I first search for solutions online or consult CHAT GPT, and if I still cannot solve them, I will seek help from my classmates or professional game development friends.



✧ Obstacle design

When I started to conceive the obstacle, I didn't want to make a traditional sphere, so I came up with the idea of a polyhedral ball, I designed the polyhedral structure and gave him the corresponding material.

Kangcheng Deng

- ✧ Music selection
- ✧ Generation of obstacles based on music rhythm
- ✧ UI design for game start and end
- ✧ Code implementation for UI

Yang Chen

- ✧ Shattering effects for obstacles upon impact
- ✧ Collision detection for obstacles
- ✧ Scoring calculation for hitting obstacles

Zhenkai Wang

- ✧ Construction of scene models
- ✧ Design of swords
- ✧ Generation of obstacles along a path

Yifan Du

- ✧ Input and control implementation for Unity and VR
- ✧ Code implementation for object generation, collision detection events, and music node events

Feedback collected from the exhibition is as follows

Advantages:

- ◆ The game has a high level of completeness and provides users with a satisfying gaming

experience.

- The immersive music gameplay enhances users' visual enjoyment.
- The three different scene perspectives add to the fun of the game and make users more engaged.

Disadvantages:

- The direction of the obstacles is too uniform, resembling the gameplay of rhythm-based sword games, lacking innovation.
- The rhythm of the music is not strong enough, resulting in a low level of difficulty in the game.
- Although there are shattering effects, adding punching sound effects or controller vibration can further enhance the immersion of the game.
- Players can anticipate the position of obstacles in advance and place the controller at the designated striking point, which reduces the difficulty of the game.

Suggestions for improvement

- Design more unique obstacle shapes and generation directions to increase the creativity and challenge of the game.
- Select music with a stronger rhythm and intense beats to increase the difficulty and tension of the game.
- Add more sound effects and vibration feedback to allow players to directly feel the impact force and rhythm of their strikes.
- Adjust the speed and position of the obstacles to prevent players from anticipating them in advance, thus increasing the game's level of challenge.

Music Game Iteration

Team Members (Involved in the iteration)

Jinglin Zhang, Kangcheng Deng, Yang Chen, Zhenkai Wang

Music Part:

To address the issue of moderate rhythm and low difficulty in the previous music, we have replaced it with a more rhythmically intense track to increase the game's challenge and intensity.

Gameplay Part:

- Addition of Obstacle Generators: We expanded the number of obstacle generators from 7 to 9 and increased the angle of obstacle generation, requiring players to improve their reaction and adaptability.
- Introduction of Multiple Gameplay Mechanics:

a. Linear Obstacles: Players need to press and hold the trigger on the controller at the right time to score points.
b. Action Scoring Mechanism: Five different action cues are set, and players need to perform the same action when the cues appear to score points.
c. Cannonball Mechanism: Players need to dodge cannonballs to avoid losing points. These new

gameplay mechanics increase the difficulty and fun of the game.

d. Enhanced Shattering Effects: Added sound effects and controller vibration feedback when obstacles are shattered, allowing players to directly feel the impact and feedback of their strikes.

e. Improved Scoring System: Previously, players' scores were categorized into four levels (A, B, C, D), resulting in almost all players being classified as A level, making it difficult to accurately assess their performance. The optimized scoring system now displays the players' scores, providing an accurate reflection of their gameplay performance.

◆ Punch Timing and Speed Detection: Implemented detection for punch timing and speed, requiring players to punch the obstacles quickly to score points, enhancing the immersive boxing experience.

Visual Part:

◆ Introduction of a completely different game scene, allowing players to enter another scene through a portal, providing a more immersive gaming experience.

◆ Instead of using a giant ball to trigger scene transitions as before, we have replaced it with a portal, enhancing the visual effect of scene switching.

Division of Responsibilities for Game Iteration

Jinglin Zhang:

✧ Game iteration planning and documentation organization.

I reflected on the problems I encountered in the last game planning and paid more attention in this planning to whether everyone's tasks were distributed properly and whether the workload and time were matched to ensure we could finish the project on time and with the best possible results. I summarised the user feedback received from the last exhibition and shared the document in the group with the hope that each person would think about how to solve or optimise the issues raised in the user feedback. In the first group discussion, members actively expressed their views and suggested solutions. While some suggestions worked, others did not quite fit. As the person responsible for the overall game planning, I needed to choose the most suitable solution at this point and explain why I didn't choose another option. Sometimes I need to come up with some good ideas when the group is running out of inspiration and actively open up the group's minds. During the discussion, sometimes people will stray from the game project to discuss other unrelated things,then I need to bring the conversation back to the production of the game in time.

I think it is fortunate that the members of our group were all very friendly and we worked very well together, we didn't argue when we have different ideas. We helped and encouraged each other in the process of making the game, and even though I was sometimes very anxious and kept pushing their progress due to time constraints, they didn't get angry about it. During these months, we ended the music game successfully and everyone put enough time and effort into it, so I think we did a great job!

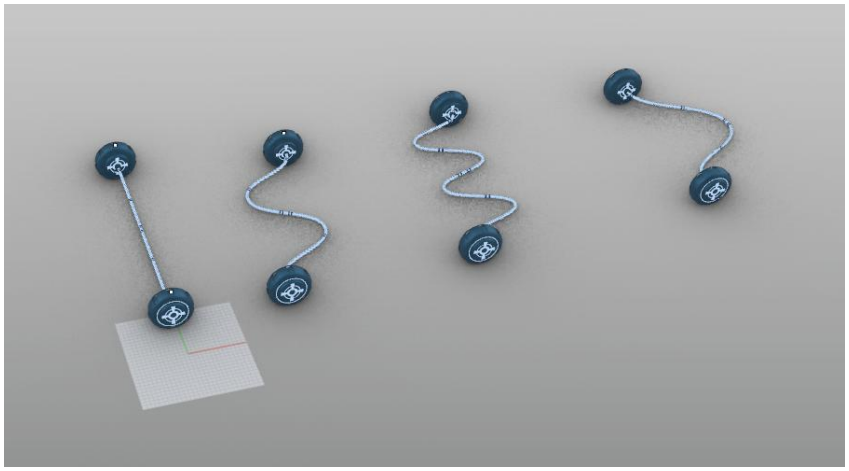
✧ Design of linear obstacles and humanoid actions.

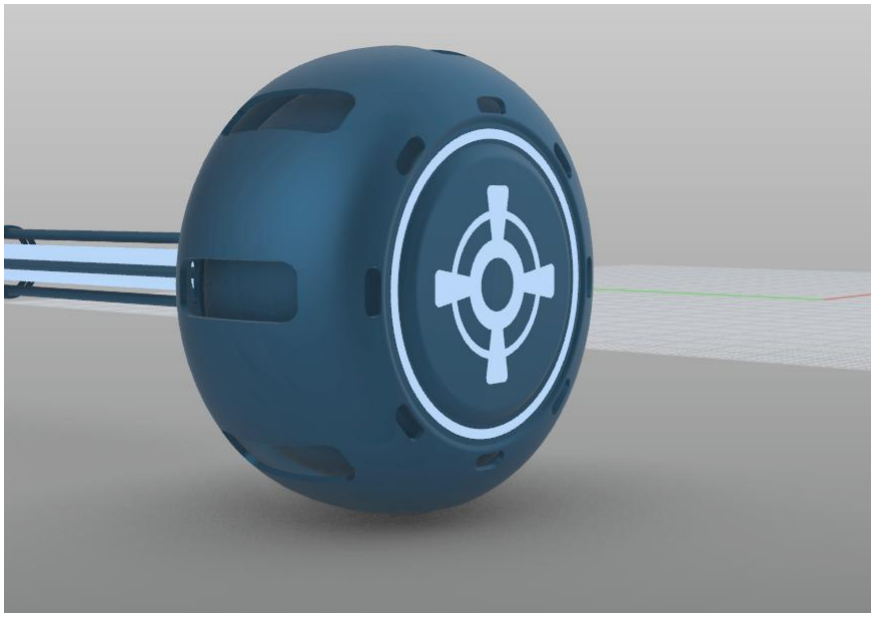
In previous exhibitions, I noticed that some players were bored during the second half of the music. I think there are two main reasons for this: firstly, our music is relatively slow paced and lacks a sense of urgency. Secondly, despite the fact that we set up three different perspectives of the game to provide different visual experiences, there was only one type of obstacle in the game and the gameplay was

relatively one-dimensional. Players found themselves just hitting the ball over and over again for two or three minutes, which could easily lead to a feeling of monotony and boredom. Therefore, one of the key tasks for this iteration of the game was to add gameplay to enrich the player's experience.

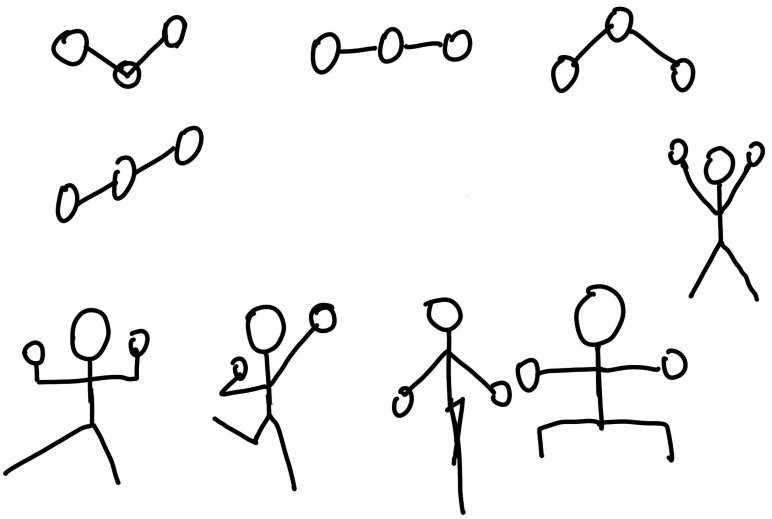
After several group discussions, we eventually decided to employ the "point-line-surface" approach to diversify the obstacles in the game and enhance their visual appeal.

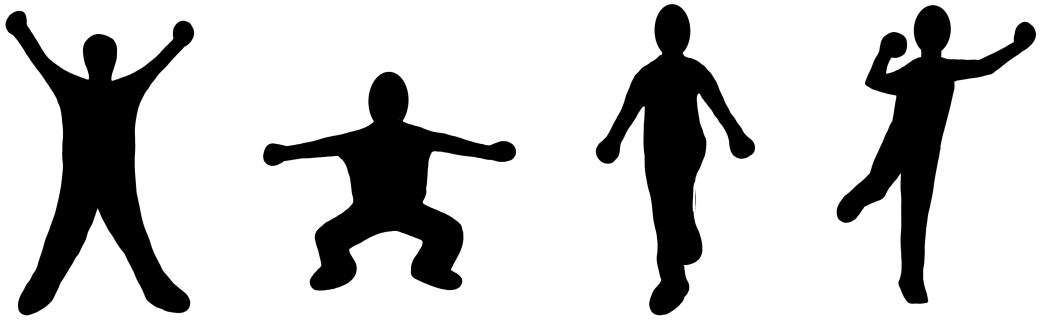
"Point" incorporates the previously designed polyhedron, where players only need to strike it to score points. "Line" requires players to hold down the trigger on the controller from the start of the line until the end.





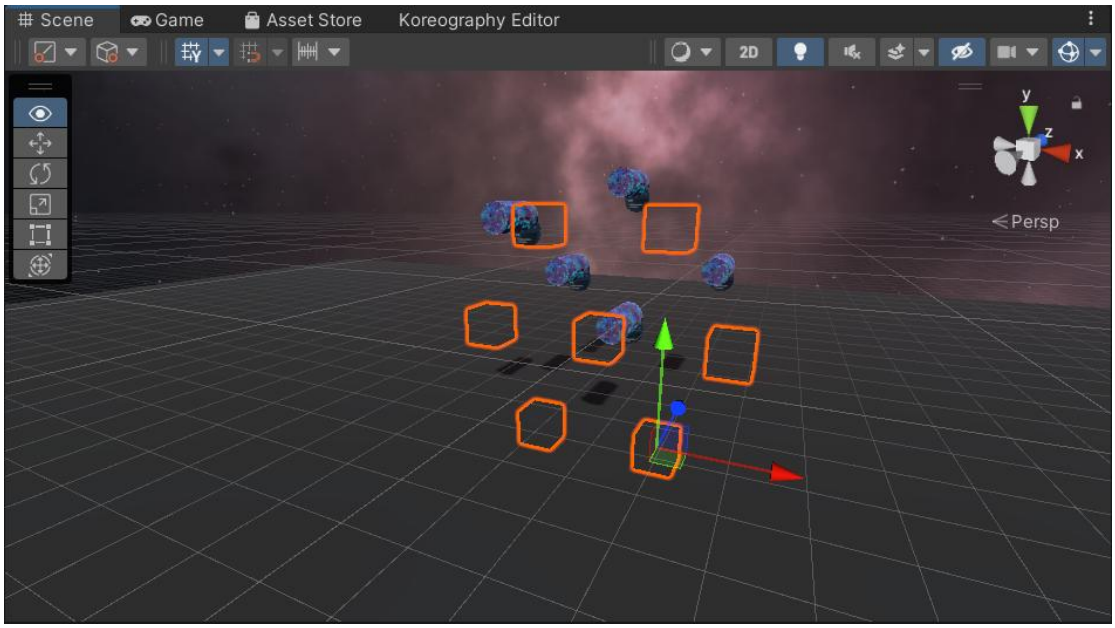
"Surface" represents walls that require specific actions to be performed. However, when I tested walls with action shapes in Unity, I encountered several issues, such as large walls obstructing players' view of obstacles behind them. As a result, we decided to create hollow action markers. I came up with several different actions and had Yang Chen pose as a model in the same positions. After taking photos and importing them into AI, I traced the outlines of the actions.

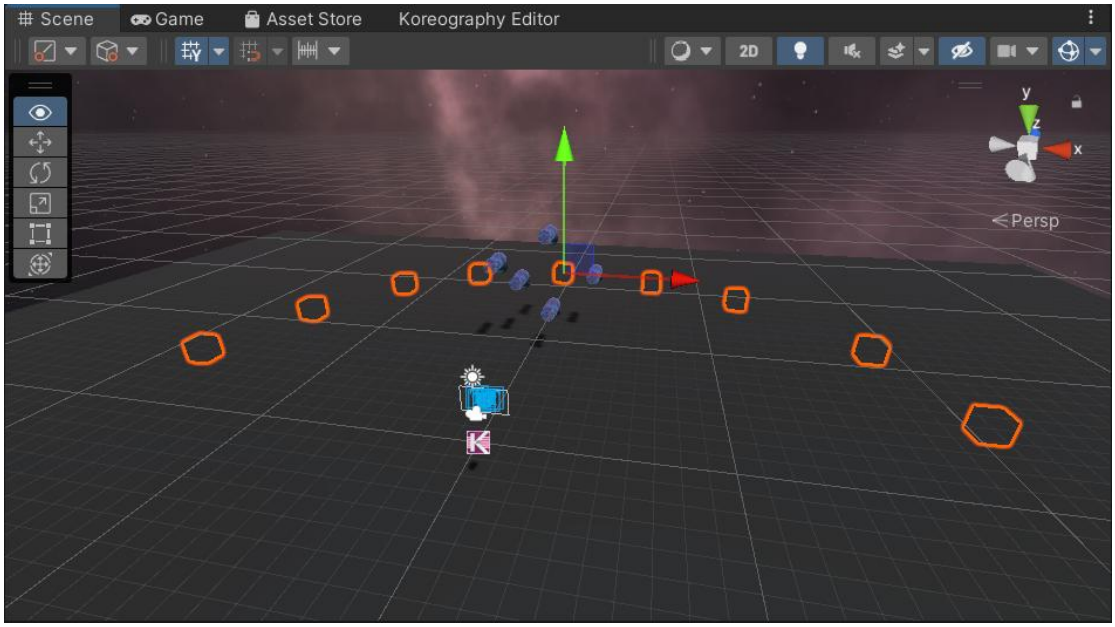




✧ Addition of obstacle generators.

In the original design I only created 7 obstacle generators, the obstacle generators were fired in a straight line towards the player, so I placed the generators directly opposite the player. However, during the iterations I wanted the generators to fire at a wraparound angle to increase the angular range of the obstacle generators and therefore the difficulty of the game. So I added three obstacle generators to allow the obstacles to be fired within visible range of the player. This meant that I needed to create two different arrays, one for generating obstacles at no angle and another to change the launch angle depending on the position to ensure that all obstacles were launched towards the player.






```
void AddLONGEvent(KoreographyEvent koreoEvent)
{
    if (b % 2 == 0)
    {
        GameObject[] longob = new[] { LONG,LONG1,LONG2,LONG3 };
        int a3 = Random.Range(0,longob.Length);

        Transform[] id = new[] { originTrans0, originTrans1, originTrans2, originTrans3, originTrans4, originTrans5, originTrans6 };
        int a = Random.Range(0, id.Length);
        Instantiate(longob[a3], id[a].position, longob[a3].transform.rotation);
    }
    else
    {
        Transform[] id180 = new[] { Trans180_0, Trans180_1, Trans180_2, Trans180_3, Trans180_4, Trans180_5, Trans180_6, Trans180_7, Trans180_8 };
        int a2 = Random.Range(0, id180.Length);
        Instantiate(LONG, id180[a2].position, LONG.transform.rotation);
    }
}


1 reference
void AddDOGEEvent(KoreographyEvent koreoEvent)
{
    Transform[] dogeid = new[] { DogeTrans1,DogeTrans2,DogeTrans3,DogeTrans4,DogeTrans5 };
    int a3 = Random.Range(0, dogeid.Length);
    Instantiate(DOGE, dogeid[a3].position, DOGE.transform.rotation);
}

1 reference
void AddWALLEvent(KoreographyEvent koreoEvent)
{
    Instantiate(Wall, originTrans4.position, Wall.transform.rotation);
}

1 reference
void AddGATEEvent(KoreographyEvent koreoEvent)
{
    Instantiate(GATE, originTrans4.position, GATE.transform.rotation);
}
```

```
1 reference
void AddCubeEvent(KoreographyEvent koreoEvent)
{
    if (b % 2 == 0)
    {
        Transform[] id = new[] { originTrans0, originTrans1, originTrans2, originTrans3, originTrans4, originTrans5, originTrans6 };
        int a = Random.Range(0, id.Length);
        if (a < 3)
        {
            Instantiate(cubePb, id[a].position, cubePb.transform.rotation);
        }
        if (a == 3)
        {
            if (c % 2 == 0)
            {
                Instantiate(cubePb, id[a].position, cubePb.transform.rotation);
                c = c + 1;
            }
            else
            {
                Instantiate(cubeTb, id[a].position, cubeTb.transform.rotation);
                c = c + 1;
            }
        }
        if (a > 3)
        {
            Instantiate(cubeTb, id[a].position, cubeTb.transform.rotation);
        }

    }
    else
    {
        Transform[] id180 = new[] { Trans180_0, Trans180_1, Trans180_2, Trans180_3, Trans180_4, Trans180_5, Trans180_6, Trans180_7,Trans180_8 };
        int a2 = Random.Range(0, id180.Length);
        if (a2 < 4)
        {
            Instantiate(cubePb, id180[a2].position, cubePb.transform.rotation);
        }
        if (a2 == 4)
        {
```

✧ Video cover design.

I was also responsible for the overall visual control of the game in this instance. I wanted the cover of the video to match the cyberpunk feel of the game as well. For the colours, I chose the predominant cyberpunk colours of blue and purple. I designed the font some special effects to make it look like a splash screen look and overall look cooler.

✧ Video recording.

Kangcheng Deng:

✧ Choose new music

✧ Obstacles are generated according to the rhythm of the music

✧ Ray detection of linear obstacles

✧ touchdesigner sound effects interaction

- ✧ UI creation

Yang Chen:

- ✧ Collision effects for punching gloves and obstacles.
- ✧ Implementation of collision detection for obstacles.
- ✧ Design of portals.
- ✧ Refinement of scoring system.

Zhenkai Wang:

- ✧ Construction of new scene models.
- ✧ Implementation of punch timing and speed detection.
- ✧ Code implementation for sound effects.
- ✧ Integration of game resources.
- ✧ Video editing.