

Moon Base Alpha - Case Study

The purpose of this paper is to provide a comprehensive account of the development process that went into making the virtual reality (VR) game known as "Moon Base Alpha," which was developed using the Unity engine. In order to defend the planet Earth from the oncoming asteroids, the player is required to maintain their current position throughout the course of the game and continue to fire their weapon. At the beginning of the game, a timer is set, and the purpose of the game is to eliminate as many asteroids as possible in the allotted amount of time. The player is given access to two different power-ups that are generated at random throughout the game. The red asteroid will wipe out any other asteroids that are inside its radius, while the blue asteroid will make the game last longer.

Brainstorming

In the beginning, my concept involved the recreation of a cinematic sequence in which the participant assumes a stationary stance and engages in combat with enemies. The concept was stimulating, although lacking in fun. Subsequently, I changed to my next idea, which was motivated by my previous completion centred on just one mechanic. Specifically, I have developed a game with a bird's-eye view perspective, wherein the player must save a spacecraft from incoming asteroids. Based on that fundamental mechanism, I initiated the development of a virtual reality game centred on the aforementioned concept. However, where does the fun lie in such an approach? Subsequently, I recollect that during my past involvement with arcade games, what formed the most enjoyable aspect of said games? The spatial aspect of certain games can be described as a race against time. The incorporation of the time element has resulted in a notable growth of the fun value of this concept for a game.

I'm making use of XR Interaction Toolkit, which is a straightforward system for developing VR applications with user interaction. It allows them to implement a

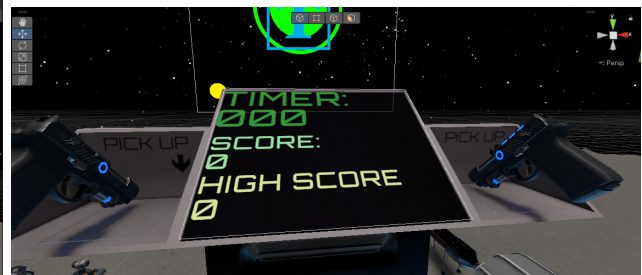
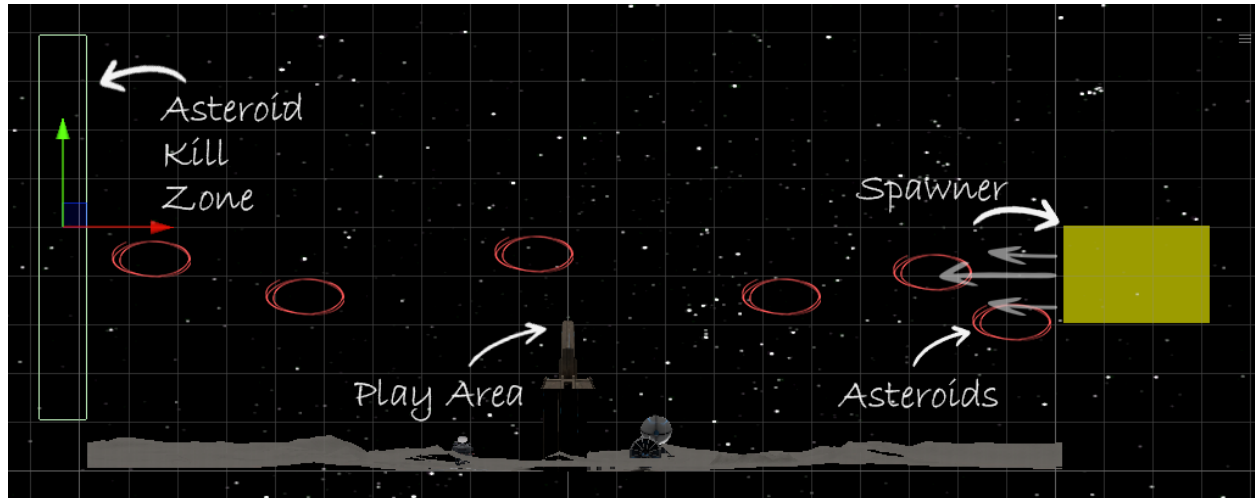
plethora of user-controllable features into their shows, such as picking up and moving things, activating switches, and even teleporting.

Create your own interactive content with the help of the XR Interaction Toolkit's editor interface and library of pre-made components and scripts. These scripts and components can be used to quickly and easily construct interactive features. Virtual reality (VR) input devices, such as positionally tracking headgear and hand controllers, are designed to work with it.[*] The XR toolkit controls how picking up and dropping weapons works in the game.

The Project

I want to improve the player experience by adding psychological and design features to our game. We provide visual and audio feedback to players who destroy asteroids. I also wanted a scoring system that encourages players to compete and score higher. A timer will add pressure to the game. Depending on their approach, this might be irritating or motivating.

Immersion was another psychological factor in growth. Immersion is a player's sense of being in the game. This game's realistic sound, visual, and gameplay mechanisms immersed players in its virtual environment. Immersion improves the game's overall quality, which may increase player involvement and enjoyment. [**]



Game Loop



[MoonBaseAlpha Game Loop]

The gameplay loop of the game, which is straightforward but compulsively playable, compels players to keep coming back for more as they try to increase their score or surpass their previous high score. The release of dopamine, which is triggered by the rewards system in the game, further reinforces this behaviour, making it more likely that players will want to continue playing and improve their scores.

As players continue to engage in the game and improve their skills, they may come across new tactics, strategies, and strategies that enable them to play for longer periods of time, score more points, and get more power-ups. The fact that players can try out a variety of gameplay styles and strategies to see which ones are most effective for them contributes to an increase in the game's replay value.

Additionally, the visual and audio features of the game, such as the graphics, sound effects, and music, contribute to the entire experience and enjoyment that the user has while playing the game. For example, the game's soundtrack adapts itself to the various game states, which helps to create an environment that is both immersive and entertaining for the player.

The Huddles in the Project

After I had programmed an asteroid's static movement, I realized I wanted the random speed to gradually grow over time. My game needed more asteroids, so I added the variable "speedChangeTimer" to my AsteroidMovement script and set it so that if it was greater than 1f, the minimum and maximum speeds would increase by 5. This worked, but the results were unexpected. I implemented a rateDecreaseAmount variable in AsteroidSpwaner my script, which will be activated through an if statement if the given circumstance is met. In this scenario, the "rateDecreaseInterval" cutoff is higher than the "timeSinceLastRate Decrease" number. A fast "spwanRate" is one that is greater than or equal to

0.1f. If these two conditions are true, then the code following the "if" expression will be run. Current "spawnRate" will be decreased by "rateDecreaseAmount" and "timeSinceLastRateDecrease" will be set to 0 by the block of code after the "if" statement. After a few seconds, the spawner value decreased, and the game began spawning so many asteroids that the entire playing area was filled with them. To that end, I've programmed my asteroid kill zone so that any asteroids that collide it are destroyed and a time penalty is applied. Because of this, the spawner value will be maintained.

The second huddle was in the timer. Initially, I wanted to use a slider for the timer, and it was working fine, but it was not updating the value when the player hits the time increase power-up. So after a lot of research and trial and error I changed that slider to text mesh, which was easy to implement and was working fine.

Implemented The Midterm Feedback

My midterm evaluation revealed that I needed to make some adjustments to the gun because one of them was malfunctioning. Therefore, in the final version, I created a second prefab of the same gun type, which proved to be effective. As a result, I now have two weapons that function properly; however, they did not have the feel of firearms, so I added some muzzle particle effects as well as some beam burst particle effects. In addition to that, I have given the firearms haptic feedback. This makes playing with my guns more enjoyable.

Comparisons between your experience over the entire development process for a PC title vs. a VR title

As a game developer who has worked on projects for both PC and virtual reality platforms, I can attest to the significance of the contrasts being made here. The building of a convincing and genuine environment for the player to experience in a virtual reality game is of the utmost importance to the overall success of the game. In order to design an immersive experience that gives the impression that

it is taking place in the real world, it is necessary to pay close attention to every aspect and have an in-depth knowledge of how the player engages with the setting of the game.

In a comparable manner, the design of the user interface in virtual reality games is essential to ensure that the player can explore and interact with the game in a way that is both easy and intuitive. Virtual reality (VR) controllers and input devices offer a level of engagement that cannot be equalled by standard PC games. However, this demands careful thought in the design process to prevent overloading the user with complex controls.

How was your experience in the development process? How did you iterate over certain mechanics, designs, or behaviours?

It was a fun and learning experience, the development process is totally different from the PC game development. I have iterated over the speed of the asteroid as the outcome was not as expected. Apart from this designing UI was very fun, it was something different and there were a lot of creative parts attached to the process.

Reviews and Feedback:

The peer feedback praised the game's use of 360 degrees, the inclusion of grabbable objects, and the high quality of its visuals, physics, and technology. One suggestion made in the optional feedback review was that "The special red asteroids should have their unique sound to that the player can differentiate between the normal asteroid and red powerup (special) asteroids." I will be sure to remember this and implement it in future versions of the game. Another piece of criticism suggested adding a bullet trace and reload feature. However, a wave system is required for that to happen, with asteroid spawning occurring in waves 1, 2, 3, etc., and I do not want to develop a wave system since I fear it will disrupt the game loop.

Reference:

* XR Interaction Toolkit: XR interaction toolkit: 2.3.0. XR Interaction Toolkit | 2.3.0. (n.d.).

Retrieved March 16, 2023, from

<https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.3/manual/index.html>

** Schell, J. (2008). The art of game design: A book of lenses. CRC Press.

Madigan, J. (2019). Getting gamers: The psychology of video games and their impact on the people who play them. Rowman & Littlefield.

Learnings References:

CodeMonkey. (2016, October 13). CodeMonkey - Game Trailer. [Video]. YouTube.

https://www.youtube.com/watch?v=-PT-LADWymI&ab_channel=CodeMonk