Instituto Tecnológico y de Estudios Superiores de Monterrey,

Campus Querétaro

TC2008B.301

Modelación de sistemas multiagentes con gráficas computacionales

Actividad Medio término (Bullet hell shooter)

Profesores

Denisse Maldonado
Alejandro Fernandez
Pedro O. Perez

Presenta

Daniel Felipe Hurtado Giraldo A01707774

Querétaro, Querétaro Domingo 19, Noviembre 2023

Índice

Introducción

La actividad que se llevó a cabo en el presente proyecto se centró en la implementación de diversas coreografías para las balas en un entorno de juego tridimensional, utilizando el motor de juegos Unity, reconocido por su versatilidad y capacidad avanzada. El principal objetivo de este emprendimiento fue el desarrollo de patrones de movimiento que no solo destacaran por su complejidad técnica, sino que también resultan visualmente atractivos y cautivadores para los jugadores. Esta iniciativa buscaba enriquecer la experiencia de juego con una estética que fuera tanto visualmente estimulante como desafiante, contribuyendo así a una inmersión más profunda y gratificante en el mundo del juego.

Análisis de los Patrones de Movimiento Implementados en el Código

1. Patrón de Movimiento Recto (Straight)

- Función: StraightMovement(float timer)
- Descripción:
    - Este patrón mueve la bala en línea recta.
    - La posición z de la bala se calcula multiplicando el timer por la speed, lo que significa que la bala se desplaza hacia adelante a una velocidad constante.
    - La fórmula return spawnPoint + new Vector3(0, 0, -z); asegura que el movimiento sea a lo largo del eje z.

2. Patrones de 20 Grados (MinusTwenty y PlusTwenty)

- Función: TwentyDegreeMovement(float timer, float angle)
- Descripción:
    - Estos patrones dirigen las balas en ángulos de 20 grados, ya sea hacia 70° o 110°.
    - El ángulo se convierte a radianes y se utiliza para calcular las componentes x y z del movimiento.
    - La fórmula return spawnPoint + new Vector3(x, 0, -z); combina estas componentes para crear un movimiento diagonal.

3. Patrón Sinusoidal

- **Función:** SinusoidalMovement(float timer)
- **Descripción:**
    - Este patrón produce un movimiento ondulado en el eje x mientras la bala avanza en el eje z.
    - La posición x varía según una función seno, cuya amplitud y frecuencia están definidas por amplitude y frequency.
    - La fórmula return spawnPoint + new Vector3(x, 0, -z); crea un efecto ondulante lateral mientras la bala se mueve hacia adelante.

Reflexiones sobre la Implementación de los Patrones

- **Versatilidad del Código:** El uso de un switch en el método Update para seleccionar el patrón de movimiento demuestra una implementación versátil y modular. Permite cambiar fácilmente entre diferentes patrones de movimiento sin alterar la estructura principal del código.
- **Desafíos y Visualización:** Cada patrón ofrece un desafío único al jugador y contribuye a la estética visual del juego. Desde las líneas rectas simples hasta los movimientos diagonales y ondulantes, estos patrones enriquecen la experiencia del juego.
- **Optimización y Precisión:** La implementación eficiente y precisa de las fórmulas matemáticas asegura que los patrones de movimiento no solo sean visualmente atractivos sino también técnicamente sólidos, contribuyendo a un juego fluido y atractivo.

Reflexión

El desarrollo de coreografías de balas en mi juego Unity fue una experiencia profundamente enriquecedora, que fusionó la programación con la creatividad artística. Este desafío no solo puso a prueba mis habilidades técnicas, sino que también me llevó a una exploración creativa sobre cómo las matemáticas pueden dar vida a un mundo de juego dinámico y visualmente estimulante.

Este proceso me enseñó la importancia de la creatividad en el diseño de videojuegos. Cada patrón de movimiento no era solo una tarea técnica, sino una oportunidad para infundir arte y narrativa en el juego. El proceso de ambientar el escenario con componentes únicos y que aportan a la narrativa, me transporto a esos recuerdos que tenía de niño en donde todo lo que imaginara podría ser real.

En resumen, desarrollar estas coreografías fue más que una tarea de programación; fue un acto de creación artística que enriqueció el mundo de mi juego. Esta experiencia no solo mejoró mis habilidades como programador, sino que también amplió mi comprensión de lo que significa diseñar un juego atractivo y memorable.

Video

https://youtu.be/lmSFumMhRNU

Repo

https://github.com/DHurtado714-itesm/bullet-hell-shooter-unity.git