



Campus Querétaro

Materia:

Modelación de sistemas multiagentes con gráficas computacionales

(Gpo 301)

Actividad: Laboratorio 3

Profesores:

Alejandro Fernández

Denisse Lizbeth Maldonado Flores

Alumno:

Carlos Isaac Dávalos Lomelí A01706041

Link al video del juego : https://youtu.be/xxv_Ajz-9kE

Link al repo de GitHub: <https://github.com/Charlie-Davalos/Multiagentes>

Juego de un Bullet Hell Shooter

Objetivo

Crear un juego tipo bullet hell shooter en Unity, donde el jugador se enfrente a un jefe con tres modos de disparo diferentes, cada uno con una duración de 10 segundos. Además, implementar un contador de balas en pantalla para optimizar el rendimiento del juego.

Introducción

El bullet hell shooter es un género desafiante de videojuegos donde el jugador debe esquivar una gran cantidad de balas en pantalla mientras intenta derrotar a enemigos igualmente intensos. Este proyecto se centra en la implementación de un jefe con patrones de disparo complejos, proporcionando una experiencia emocionante y desafiante para el jugador.

Implementación de Patrones de Disparo

El jefe del juego tiene tres modos de disparo, cada uno con su propio patrón de balas. Estos modos de disparo cambian cada 10 segundos durante un total de 30 segundos de duración del juego. Aquí están los tres patrones de disparo implementados para el jefe:

Modo 1: Disparo en Cuatro Direcciones

En este modo, el jefe dispara balas en cuatro direcciones: hacia abajo, hacia la izquierda, hacia arriba y hacia la derecha. Para lograr este patrón, se crean cuatro balas y se les asigna una velocidad en las respectivas direcciones.

```
void DisparoCuatroDirecciones()
{
    Vector2[] direcciones = new Vector2[] {
        new Vector2(0, -1),
        new Vector2(-1, 0),
        new Vector2(0, 1),
        new Vector2(1, 0)
    };

    foreach (Vector2 dir in direcciones)
    {
        GameObject bala = Instantiate(balaPrefab, transform.position, Quaternion.identity);
        bala.GetComponent<Rigidbody2D>().velocity = dir * velocidadBala;
    }
}
```

Las direcciones se pueden representar como vectores unitarios

Modo 2: Disparo en Espiral y hacia Abajo

El jefe dispara balas hacia abajo y en espiral hacia arriba.

```
private void ShootSpiralAndDown()
{
    float angleStep = 360f / 4;
    for (int i = 0; i < 4; i++)
    {
        if (i < 2)
        {
            FireBullet(180 + i * 180); // Hacia abajo
        }
        else
        {
            FireBulletSpiral(0 + i * angleStep, -1); // Hacia arriba en espiral
        }
    }
}
```

Modo 3: Disparo Mixto

El jefe dispara una combinación de balas rectas en direcciones aleatorias y balas en espiral en direcciones aleatorias.

```
private void ShootMixed()
{
    // Dispara 3 ráfagas rectas en diferentes direcciones
    for (int i = 0; i < 3; i++)
    {
        FireBullet(Random.Range(0, 360));
    }

    // Dispara 3 ráfagas en espiral
    float angleStep = 360f / 3;
    for (int i = 0; i < 3; i++)
    {
        FireBulletSpiral(i * angleStep, Random.Range(-1, 2)); // Hacia cualquier dirección en espiral
    }
}
```

La implementación de los patrones de disparo en un juego tipo bullet hell shooter en Unity se basa en el uso de vectores y ángulos para determinar las direcciones de las balas. Los modos de disparo descritos utilizan principios de trigonometría y geometría para crear patrones variados. Esto permite crear una experiencia de juego dinámica y entretenida para los jugadores.