



Tecnológico de Monterrey

Instituto Tecnológico y de Estudios Superiores de Monterrey.

TC2008B | Gpo 101

Modelación de Sistemas Multiagentes con Gráficas Computacionales
“Actividad Integradora de Gráficas Computacionales”

Profesores:

Alejandro Fernández Vilchis
Denisse Lizbeth Maldonado Flores
Pedro Oscar Pérez Murueta

Jorge Armando Garzón Varela | A01707102

Fecha de entrega: 28 de agosto de 2025

Actividad Integradora de Gráficas Computacionales

Introducción

En la actividad de medio término se nos presentó el reto de desarrollar un Bullet Hell Shooter, aplicando los conceptos aprendidos en los laboratorios de Unity y gráficas computacionales. Entre los tres niveles de dificultad propuestos, decidí trabajar en el nivel Fácil – El Jefe Solitario, ya que me permitió concentrarme en la implementación de un jefe con tres patrones de disparo mecánicamente diferentes. Este nivel resultó ideal para reforzar los conocimientos adquiridos hasta ahora, al mismo tiempo que me retaba a diseñar y programar patrones de ataque que fueran visualmente claros y jugablemente interesantes.

Los bullet hell shooters son un subgénero de los juegos de disparos en 2D donde la característica principal es la gran cantidad de proyectiles en pantalla. El jugador no solo debe atacar o sobrevivir, sino que su principal reto consiste en esquivar patrones densos y complejos de balas que, muchas veces, forman figuras geométricas o trayectorias poco convencionales. Para lograr esto en el desarrollo, es necesario aplicar transformaciones computacionales como traslaciones, rotaciones y escalas, además de optimizar el rendimiento con métodos como el object pooling, ya que la cantidad de objetos activos puede ser muy alta.

De esta forma, el objetivo principal de esta actividad integradora fue crear un jefe en Unity con tres patrones de disparo diferentes, aplicando transformaciones computacionales y técnicas de optimización. Para lograrlo, se implementaron sistemas de rotación, traslación y control de tiempo, así como un object pooling para el manejo eficiente de balas. La meta fue conseguir una experiencia visual que representara fielmente la esencia de un bullet hell shooter, manteniendo un balance entre un espectáculo visual y un buen rendimiento técnico.

Patrones del Jefe

Primer Patrón – Disparo Lineal con Movimiento Dinámico

El primer patrón utiliza el script Shoot.cs, que dispara balas rectas hacia la dirección frontal del jefe de manera constante. El disparo se controla con un temporizador que define la cadencia de fuego. Mientras dispara, el jefe se desplaza por las cuatro esquinas del escenario siguiendo un recorrido en forma de cuadrado. En cada esquina realiza varias rotaciones de ida y vuelta y después ajusta su orientación a un ángulo múltiplo de 90° antes de continuar al siguiente punto. Este patrón es sencillo pero efectivo, ya que combina disparos con movimiento, obligando al jugador a esquivar proyectiles y anticipar la trayectoria del jefe. Este patrón es único y diferente a los otros dos patrones implementados porque mezcla disparo con movimiento, creando una presión constante pero predecible para el jugador.

Segundo Patrón – Disparo Radial con Curvatura en las Balas

En el segundo patrón, el jefe activa el script RadialShoot.cs, que dispara múltiples proyectiles en todas direcciones de manera uniforme, dividiendo la circunferencia en ángulos iguales. Algunas balas tienen una curvatura positiva y otras negativa, lo que produce trayectorias diferentes y más difíciles de leer. Durante este ataque, el jefe se coloca en el centro del escenario y comienza a realizar rotaciones completas en ambos sentidos, de modo que los

disparos radiales cambian constantemente de orientación. Este patrón genera un efecto visual inicial en forma de flor y luego, al iniciar la rotación, genera una espiral que da la impresión de una “barrera de balas” que cubre toda la pantalla y obliga al jugador a moverse constantemente para encontrar huecos por donde escapar. Este patrón es único porque crea un patrón geométrico que evoluciona dinámicamente, pasando de una especie de flor estática a una espiral cambiante que incrementa la dificultad por la rotación y la curvatura de las balas.

Tercer Patrón – *Disparo Radial con Diferentes Velocidades*

El tercer patrón está a cargo del script StarShoot.cs, que dispara proyectiles en forma de estrella, variando la velocidad de las balas según el ángulo mediante una función coseno. Esto provoca que algunas balas sean más rápidas y otras más lentas, lo que genera un efecto de estrella de cinco puntas en movimiento. Durante este ataque, el jefe realiza rotaciones completas de 360°, avanzando en bloques de 90°, haciendo que las figuras estrelladas se vayan reacomodando y cruzando entre sí. Este patrón es el más complejo, ya que las balas llegan en momentos distintos y desde direcciones impredecibles. Este patrón es único porque combina velocidad variable con rotación, creando una figura dinámica en forma de estrella que desafía la anticipación y los reflejos del jugador, a la vez que genera un patrón visible.

Conceptos Utilizados

Para implementar el jefe y sus patrones se usaron los conceptos aprendidos y/o investigados:

- **Time Manager:** Simula un reloj interno del juego y dispara eventos cada vez que cambia un minuto o una hora. Gracias a este sistema, el jefe activa cada patrón en momentos específicos del juego, lo que le da una estructura y progresión al combate.
- **Object Pooling (BulletPool):** Permite manejar decenas de balas sin perder rendimiento, ya que en lugar de crear y destruir objetos continuamente, se reutilizan instancias desactivadas. Este sistema también incluye un contador de balas visibles en pantalla, mostrado en la interfaz de usuario, que sirve como indicador de desempeño.
- **Clase Bullet:** Define el comportamiento individual de cada proyectil, incluyendo la posición, velocidad y la posibilidad de curvatura del mismo. También controla el tiempo de vida de la bala para que se desactive automáticamente en cierto tiempo.
- **TimeUI:** muestra en pantalla el reloj interno del juego en formato horas:minutos, actualizando dinámicamente cuando se producen los eventos del Time Manager. Esto aporta claridad al jugador sobre el momento en el que aparecerán nuevos patrones.

Reflexión sobre Retos y Aprendizajes

Durante la implementación de los patrones del jefe se presentaron varios retos. El primero fue sincronizar el movimiento y el disparo, ya que era necesario que las trayectorias de las balas fueran entendibles para el jugador sin que el patrón resultara injusto. Otro reto importante fue el rendimiento, pues un bullet hell requiere que haya muchas balas en pantalla al mismo tiempo, lo que podría generar caídas de frames si no se optimiza adecuadamente. La solución fue usar el object pooling, que se convirtió en una de las lecciones más valiosas de la práctica, al demostrar cómo técnicas de programación mejoran directamente la jugabilidad.

En cuanto a lo aprendido, quedó claro que pequeñas variaciones en parámetros como el ángulo, la velocidad o la curvatura de las balas pueden cambiar radicalmente la experiencia del jugador. El patrón recto, el radial y el estrellado comparten la misma base técnica (instanciación de balas y transformaciones), pero generan sensaciones muy distintas. También resultó enriquecedor trabajar con rotaciones y funciones trigonométricas, ya que estas matemáticas se convierten en herramientas creativas para construir figuras y trayectorias en pantalla. Finalmente, la capacidad de separar cada patrón en scripts individuales facilitó la organización del proyecto y permitió cambiar entre fases del jefe de manera más ordenada.

Conclusión

El desarrollo de este jefe estilo bullet hell shooter me permitió aplicar de forma práctica los conocimientos de Unity adquiridos hasta el momento. Los tres patrones de disparo muestran una progresión clara de dificultad y variedad. Gracias al uso del Time Manager, el jefe cambia de fase en momentos definidos, lo que aporta estructura al combate. Además, el object pooling y la interfaz de usuario enriquecieron la experiencia al garantizar un rendimiento fluido y una presentación clara de la información. En conjunto, esta actividad representó un reto técnico y creativo: diseñar ataques visualmente atractivos, optimizar el manejo de múltiples objetos en pantalla y mantener una jugabilidad desafiante pero justa para el jugador. La experiencia fue una oportunidad para reforzar la relación entre programación, matemáticas y diseño de videojuegos, pilares esenciales en el desarrollo de un bullet hell.

Repositorio: <https://github.com/jorgeGrVI/BulletHell-Shooter>

Video Demo: <https://www.youtube.com/watch?v=2nRMwH3QkoI>