

GDD Microjuego: Asteroids

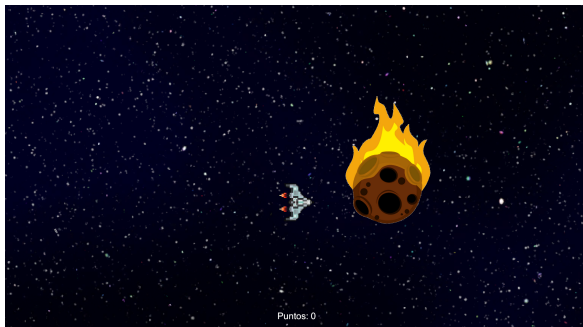
Título del juego: Asteroids

Plataforma: PC

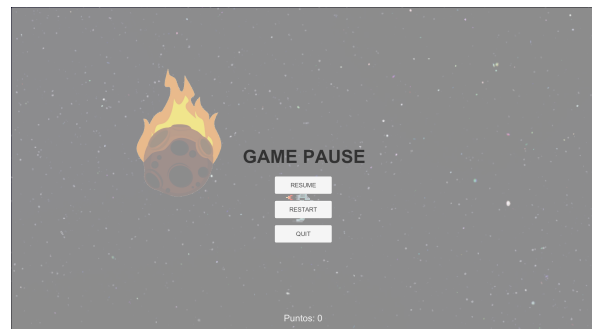
Creado y documentado por: Randy Rubio Sosa y Laura Granero Martín

1. Escenas

Es un videojuego de un único nivel con una sola escena. El juego se desarrolla en el mismo plano, y tiene un ligero menú de opciones (reanudar, reiniciar y quitar).



Pantalla de juego en acción.



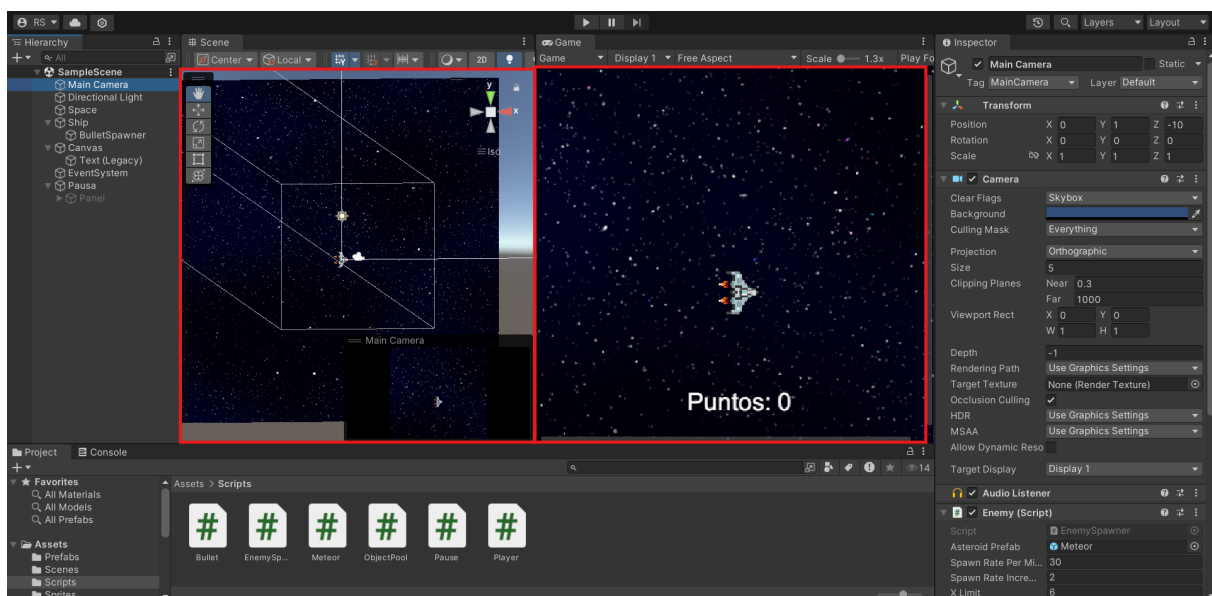
Menú de opciones (juego en pausa).

El jugador puede desplazar la nave con las teclas A y D (izquierda y derecha), W y S (arriba y abajo) o con las flechas del teclado. Para disparar, debe presionar la barra espaciadora; para pausar el juego y acceder al menú, la tecla Escape. La escena se desarrolla en un espacio infinito, por lo que el jugador no tiene que preocuparse por chocar contra los bordes de la pantalla. Además, hay un contador de puntos en la parte inferior de la pantalla.

El juego se desarrolla en un espacio 2D con una mecánica sencilla: destruir los asteroides y evitar ser impactado por ellos.

La escena consta de los siguientes elementos: la cámara principal, el espacio, la nave, el marcador de puntos y el menú de pausa.

A nivel de producción, podemos decir que el videojuego realmente tiene 2 escenas: la de juego y la de director. Mediante la escena de director podremos visualizar y colocar los objetos de forma más precisa, cambiando tamaños y posiciones.



La escena de la izquierda es la escena de director. La de la derecha, la escena de juego. En dicha escena podremos testear las modificaciones más recientes de una manera rápida.

2. Assets y GameObjects

• Prefabs

Asteroids utiliza 3 prefabs: balas (Bullet), asteroides (Meteor) y fragmentos de asteroides (Meteor Variant). Estos se emplean como prefabs por ser los elementos más dinámicos del juego. Usan scripts para controlar su movimiento, reaparición y colisiones con otros objetos. Además, se destaca el uso de Rigidbody y Box/Capsule Collider.

Rigidbody y Box/Capsule Collider serán elementos clave en el videojuego. Rigidbody nos permitirá aplicar fuerzas físicas a nuestros objetos, como gravedad, empujes, velocidades, rotaciones y velocidades angulares. Esto aportará un movimiento más realista al juego (por ejemplo, el movimiento esperado de una nave en el espacio). Los colliders también serán piezas fundamentales, ya que en ellos basaremos muchas de nuestras mecánicas. Controlan las colisiones entre objetos y las reacciones que estas desencadenan.

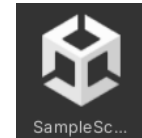
Los prefabs de nuestro juego se basan principalmente en el funcionamiento de estas herramientas.

• Scenes

Esta carpeta almacena la escena única del videojuego.

• Scripts

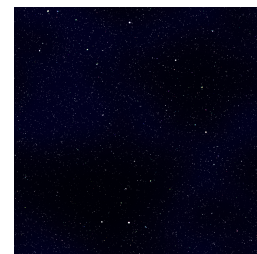
- Bullet: evita colisiones con la nave, controla el movimiento y eliminación de la bala (implementa la técnica de pooling). Gestiona las colisiones con asteroides, crea instancias de fragmentos de asteroide, incrementa la puntuación y actualiza el texto correspondiente. Es utilizado por el GameObject Bullet.
- Player: maneja el movimiento, rotación y velocidad de la nave. Controla la traslación para simular un espacio infinito. Crea instancias de balas, ajustando su posición y dirección. Gestiona colisiones con asteroides y reinicia la partida en caso de impacto. Es utilizado por el GameObject Ship.
- Meteor: gestiona la desactivación de los asteroides. Al colisionar con una bala, reinicia el tiempo de desaparición del meteorito. Se aplica a los GameObjects Enemy y Enemy_fr.
- EnemySpawner: invoca asteroides en el espacio y aumenta gradualmente su tasa de aparición. Implementa la técnica de "pooling" y ajusta el movimiento de los asteroides al reaparecer tras su invocación inicial. Este script está colocado en el GameObject Main Camera.
- Pause: detiene el juego y realiza las acciones de reanudar, reiniciar o salir del juego. Este script es utilizado por el GameObject Pausa.
- ObjectPool: implementa la técnica de pooling para varios objetos. Está situada en la Main Camera. Es utilizada para el ahorro de recursos a la hora de generar balas, asteroides y fragmentos. Este script preguntará sobre qué GameObjects se realizará el pooling y también qué numero de objetos tendrá cada agrupación.



Sprite Asteroide



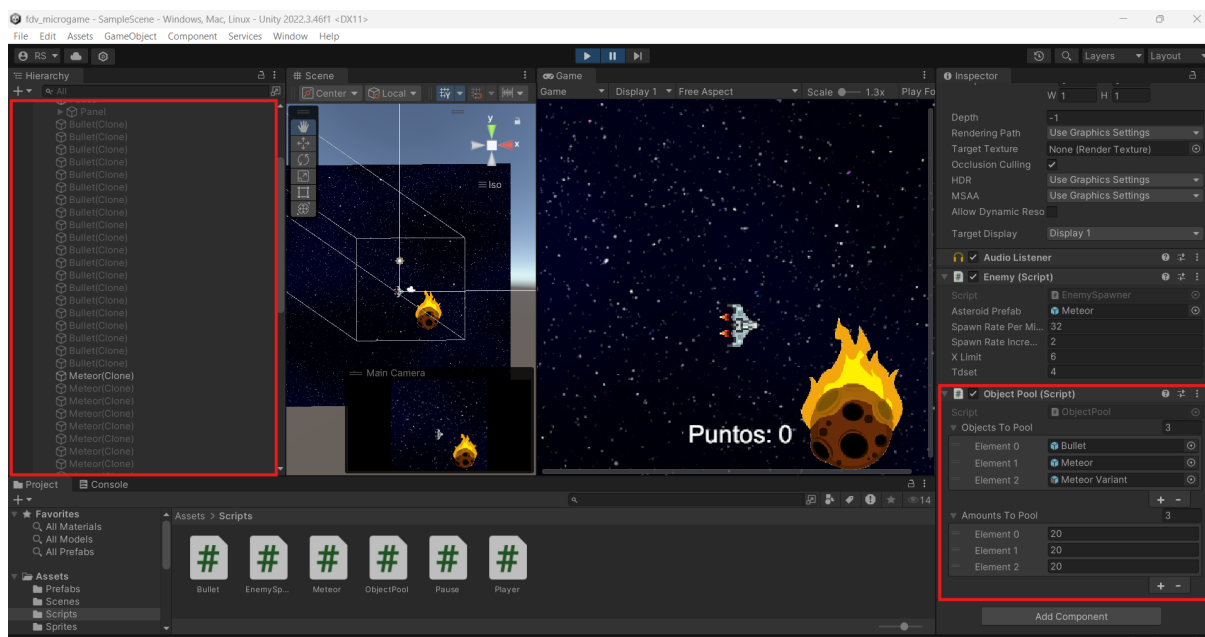
Sprite Nave



Sprite Espacio

3. Mecánicas

- **Movimiento:** el desplazamiento se controla mediante fuerzas del motor, mientras que el giro utiliza "transform", ambos gestionados por scripts. La nave se mueve de forma kinética, mientras que los asteroides y sus fragmentos se rigen por la gravedad.
- **Disparos:** se efectúan al pulsar la barra espaciadora. Las balas siguen una trayectoria recta y solo colisionan con asteroides y sus fragmentos. Tras la colisión, tanto la bala como el objetivo desaparecen. Si no hay colisión, la bala avanza hasta salir de la pantalla y desaparece.
- **Aparición de asteroides:** surgen aleatoriamente en la parte superior de la pantalla. Su movimiento está controlado por la gravedad. Desaparecen ya sea por colisión con una bala o al salir de la pantalla, tras un tiempo determinado. Existe la posibilidad de que los asteroides se fragmenten, dividiéndose en asteroides más pequeños. Con el paso del tiempo, aumenta la frecuencia de aparición de los asteroides.
- **Sistema de vida y pausas:** el jugador no dispone de vidas extra. Al colisionar con un asteroide, la partida se reinicia automáticamente. Además, el jugador puede pausar el juego y acceder a un menú de opciones.
- **Pooling:** Se emplea un conjunto limitado de objetos, inicialmente desactivados en la escena, que se activan según las necesidades del juego o del jugador. Tras su uso, estos objetos se desactivan en lugar de eliminarse, quedando disponibles para futuras llamadas. Esta técnica optimiza el rendimiento del juego al reducir la carga de trabajo de la CPU.



Los objetos se crean de una manera más dinámica, cuando son destruidos o se alejan mucho, se desactivan. Esto le facilita el trabajo a la CPU. En la parte de la derecha podemos ver cómo seleccionamos los objetos a los que les queremos aplicar este método de optimización.

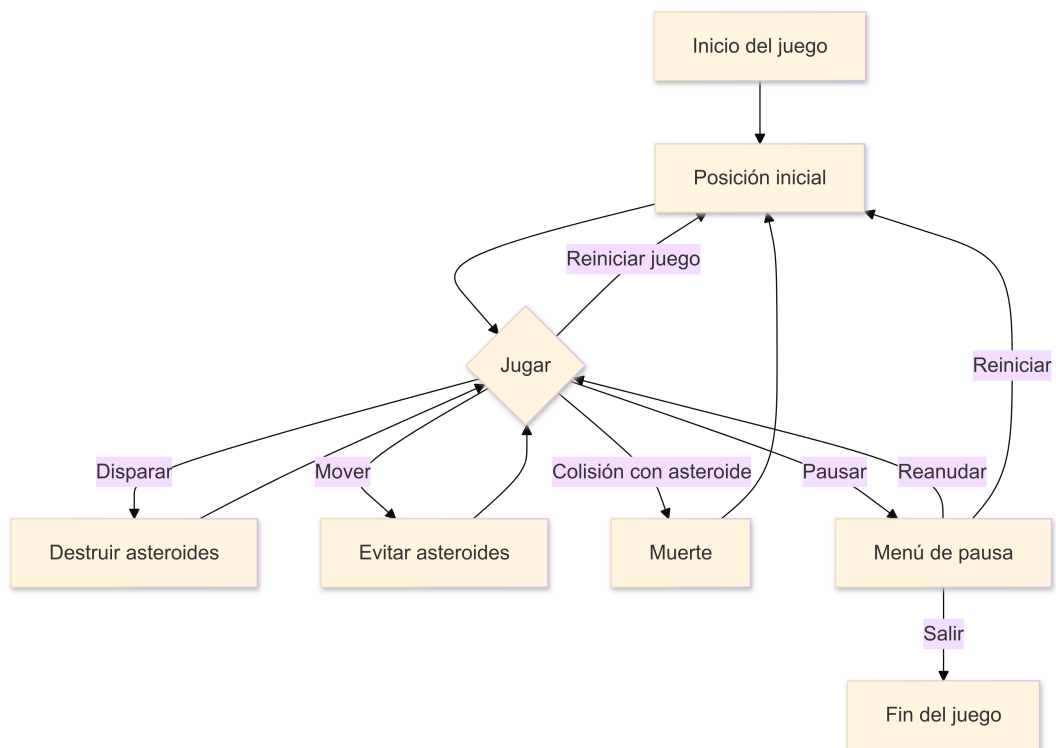


Diagrama de juego. En este diagrama se detallan las fases por las que pasa el jugador a medida que transcurre el juego.

4. Interacción

- Movimiento: Flechas direccionales y teclas WASD.
- Disparos: Barra espaciadora.
- Pausa y menú: Tecla Escape. El menú ofrece tres opciones: reanudar, reiniciar o salir de la partida. También se puede reanudar el juego presionando Escape nuevamente.
- Interacción con enemigos: El contacto directo con los asteroides resulta en la pérdida y reinicio inmediato de la partida.

5. Recursos

Enlace al repositorio del juego:

https://github.com/andy-913/fdv_microgame