

# GDD Microjuego

## Enlaces de descarga (GDD, Github, Demos):

Github: [https://github.com/Patala2004/Microjuego\\_Unity](https://github.com/Patala2004/Microjuego_Unity) (Links de descarga de demos y GDD en el github)

Demos .exe: En el github **VER README PARA DESCARGA**

## Escenas

El microjuego consiste en una escena llamada "SampleScene". La escena contiene todo el juego.

## Assets y GameObjects

El juego tiene varios **GameObjects**.

- Player: La nave que controla el jugador. Con ella se mueve por el mapa y dispara balas.
- Gun: La "pistola". Se encarga de la generación de las balas al principio del juego y del pooling de ellas
- Bullet: Durante el juego se generan GameObjects correspondientes a las balas.
- Camara: Elige que parte del juego se renderiza en la pantalla
- UI: 'User Interface'. Tiene todo el contenido de la interfaz del usuario, incluyendo un contador de puntos y un indicador de vidas
- PauseMenu: Un menú de pausa que aparece cuando pausas el juego
- DeadScreen: Un menú que aparece cuando mueres
- Background1: Un conjunto de las tiles que conforman el fondo del juego.
- EnemySpawner: Se encarga de instanciar y gestionar el pooling de los meteoritos
- ActiveBulletFolder: No hace nada de por si. A las instancias de 'Bullet' se le pone esto como padre por orden.
- PauseManager: Se encarga de gestionar el pausado y reanudado del juego.

## Assets:

- Sprites:
  - 'Ship': Sprite de la nave del jugador
  - 'Space': Sprite para el fondo
  - 'Meteor': Sprite para los meteoritos
  - 'Energy Ball': Sprite para los indicadores de escudos / vidas
  - 'Bullet': Sprite para las balas

- 'Arrow Alert': Sprite para las alertas de spawn de enemigos
- Prefabs:
  - Arrow\_Alert: Flecha para efectos de la UI de avisos de dirección de meteoritos.
  - Background 1: Conjunto de tiles, colliders y scripts que se crean un fondo que sigue al jugador
  - BG\_Tile: Prefab de una única tile que se usó para generar el fondo de Background 1
  - Bullet: Prefab para las balas. Contienen un collider y un Rigidbody2D al igual que un script para el movimiento. Este prefab es usado por el arma del jugador a la hora de disparar
  - Meteorite: Meteorito estándar
  - Smaller meteorite: Meteorito pequeño que aparece tras la destrucción de un meteorito grande

## Mecánicas:

**Movimiento:** El jugador se puede mover por todo el mapa. Para esto puede rotar su nave, acelerar la nave y desacelerar la nave.

**Seguimiento del fondo:** Mientras la nave se mueve por el mapa el fondo la va siguiendo de tal forma que no se note el movimiento del fondo y parezca que se están generando nuevas casillas

**Generación de Meteoritos:** Durante la partida aparecen meteoritos. Estos meteoritos aparecen en un círculo alrededor del jugador y siempre volando en su dirección. De esta forma el jugador tiene que reaccionar a meteoritos de todas las direcciones. La generación de meteoritos se gestiona usando pooling -> se generan todos los meteoritos al principio de la partida y luego el EnemySpawner se encarga de moverlos, añadirles velocidad y rotarles hacia el jugador, al igual de encargarse de que los meteoritos "destruidos" o muy alejados del jugador vuelvan a la "habitación de meteoritos".

**Disparos:** El jugador puede disparar balas hacia enfrente suyo. Con esto puede intentar destruir los meteoritos. Las balas se gestionan mediante un sistema de 'pooling', usando el gameobject 'Gun'. Al principio de la partida se generan todas las balas debajo del jugador y luego el arma se encarga de añadirles velocidad cuando haya falta dispararlas al igual de volver a guardar las balas cuando se alejen mucho del jugador. La mecánica de disparo cuenta con un cooldown/enfriamiento para evitar que se pueda hacer un rayo láser de balas.

**Varios tipos de meteoritos:** Hay dos tipos de meteoritos. Los meteoritos grandes y los meteoritos pequeños. Al destruir un meteorito grande con una bala se transforma en dos meteoritos pequeños.

**Colisión con meteoritos:** El jugador es castigado si se choca con meteoritos. El jugador al principio de la partida cuenta con 5 escudos, los cuales pierde cada

vez que colisiona con algún meteorito. Si vuelve a colisionar cuando se le han agotado los escudos termina la partida y salta la 'DeadScreen'.

**Pausado:** El juego tiene una mecánica de pausado y reanudación al igual que una pantalla de pausado que indica como hacer esto.

**Puntuación:** El jugador gana un punto cada vez que destruye un meteorito. Estos puntos luego aparecen en la pantalla.

**Sistema de alertas:** La nave cuenta con un radar que avisa al jugador de la dirección desde la que se aproximan meteoritos

## Scripts:

**Background\_Tile\_Manager.cs:** Detecta la colisión del jugador con los bordes del fondo / mapa para mover el fondo hacia la posición del jugador una cantidad exacta para que no se note el movimiento del fondo.

**Bullet\_Movement.cs:** Se encarga del movimiento de las balas. Cada frame mira si ha sido disparada (si tiene velocidad), y si la tiene calcula la siguiente posición de esta. Tras alejarse cierta distancia del jugador el script también reinicia la bala. El script también ofrece una función pública para que otros objetos puedan reiniciar la bala.

**Camera\_Follow.cs:** Se encarga de seguir al jugador y de añadir efectos de velocidad. Para esto sigue la posición del jugador, con algo de retraso si el jugador se mueve muy rápido. También hace zoom-out dependiendo de la velocidad de la nave, para añadir mayor efecto de velocidad. Para no interrumpir el screen-shake este script deja de calcular el seguimiento si ese efecto esta activo.

**Enemy\_Spawner.cs:** Se encarga de la generación de meteoritos. Al principio de la partida este script genera todos los meteoritos grandes y pequeños y los guarda muy alejados del jugador. Si el jugador se acerca demasiado este script detecta una colisión y cambia de lugar mucho antes de que los meteoritos estáticos puedan entrar en la cámara. Durante la partida este script "genera" un meteorito cada cierto intervalo de tiempo. La generación ocurre calculando un ángulo al azar alrededor del jugador (Entre 0 y 360 grados). Luego calcula la posición donde tiene que mover al meteorito para que esté a ese ángulo del jugador y rota el meteorito para que le mire al jugador. Después le añade velocidad al meteorito y llama a las alertas de meteoritos, haciendo que le salga al jugador una indicación de la dirección desde donde se aproxima. Cuando se destroza a un meteorito este script recibe el ángulo, la velocidad y la posición del antiguo meteorito grande, con los cuales calcula los nuevos ángulos, la nueva velocidad y las nuevas posiciones de los 2 meteoritos pequeños que "genera" (no los genera, usa pooling) en su lugar.

**Gun.cs:** Al principio de la partida genera todas las balas y las guarda debajo del jugador. Luego, teniendo en cuenta un intervalo de enfriamiento, "dispara" una

bala cada vez que el jugador presiona click izquierdo. Para hacer esto simplemente añade velocidad a una de las balas y le pasa la posición y la rotación del jugador.

**Meteorite\_Collider.cs:** Se encarga de las colisiones de los asteroides. Si colisiona con un jugador llama a la función que le quita una vida (o llama a la pantalla de muerte de no tener más) y reinicia al meteorito (Sistema de pooling). Si colisiona con una bala mira si es un meteorito grande, en cuyo caso indica al generador de enemigos que tiene que mover a dos meteoritos pequeños antes de reiniciarse, o si es un meteorito pequeño, en cuyo caso se reinicia. En ambos casos también indica al objeto UI que tiene que actualizar la puntuación.

**Meteorite\_Movement.cs:** Avanza el meteorito basado en una rotación y velocidad variable.

**Pause\_Manager.cs:** Al principio de la partida esconde la pantalla de pausa. Pausa el juego y enseña la pantalla de pausa si pulsas ESC cuando el juego está en marcha o reanuda el juego y esconde esa pantalla si está pausado. Para hacer esto pausa la funcionalidad del sistema del Deltatime.

**Player\_Movement.cs:** Se encarga de calcular el movimiento de la nave y de rotar, acelerar y/o decelerar la nave dependiendo de los Inputs del usuario.

**Screen\_Shake.cs:** Gestiona la llamada a la función pública que inicia el efecto del screenshake, el cual es llamado tras una colisión con un meteorito, y del seguimiento al jugador durante la duración de este.

**UI.cs:** Al principio de la partida genera las flechas que se usan para el sistema de aviso a meteoritos y las gestiona mediante pooling. Se encarga también del sistema de alertas del radar de la nave, “generando” una flecha con la aparición de un meteorito y escondiéndola tras un corto intervalo de tiempo. También se encarga de la actualización de las puntuaciones y de las vidas, escondiendo un escudo por cada colisión y llamando al efecto de screenShake en cada colisión. En caso de muerte (colisión sin escudos) este script llama a la pantalla de muerte.

## Interacción:

El jugador puede interactuar con el juego mediante una serie limitada de inputs con los cuales puede realizar una serie de actividades clasificadas de la siguiente forma

- Movimiento del jugador:
  - Rotar la nave -> (Botones A y D del teclado)
  - Acelerar la nave -> W
  - Desacelerar la nave -> S
- Disparo:
  - Disparar bala -> Click
- Pausado del juego:
  - Pausado y reanudado -> ESC