

# Microjuego Asteroids GDD

Alexander Zerpa, c200339

## ESCENAS

Solo existe una escena, *MainScene* en donde se encuentran todos los *GameObjects* y en donde ocurre la entidad del juego.

## ASSETS Y GAMEOBJECTS

### GameObjects:

*Space*: Sprite simple que representa el fondo del área del juego.

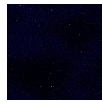


Fig. 1: Space sprite

*Ship*: Sprite simple con colisiones y lógica, representa al jugador.



Fig. 2: Ship sprite

### *BulletSpawner*:

Anidado dentro de *Ship*, usado para el spawn de *Bullet*.

*Canvas*: Contiene los elementos de UI.

*Score*: Texto que indica la puntuación del jugador.

*PauseButton*: Se usa para pausar el juego.

*MainCamera*: Usada para el renderizado del juego.

*EventSystem*: Autogenerado por Unity.

### Prefabs:

*Bullet*: Spawneado por *BulletSpawner*, tiene colisionador y logica propia.

*Meteor*: Principal enemigo del juego, al ser impactado por *Bullet* es reemplazado por dos *MeteorSmall*.

*MeteorSmall*: Igual que *Meteor* solo que mas pequeño y con logica de movimiento manejada por script.



Fig. 3: Meteor sprite

## MECÁNICAS

### Movimiento

El jugador se mueve controlando la aceleración de la nave y su rotación.

*Looping*: Si el jugador se desplaza fuera de la pantalla este volverá a entrar por el lado contrario.

### Disparo

El jugador puede disparar una bala en la dirección en la que la nave apunta, esta bala se ira desplazando por el espacio y destruirá el asteroide con el que colisione.

### Asteroides

Durante el juego se spawnearan asteroides en la parte de arriba de la pantalla, estos irán cayendo y serán el principal obstáculo para el jugador.

*Colisión*: Si un asteroide colisiona con el jugador este pierde.

*Separación*: Cuando un asteroide es impactado por una bala, este se divide en dos mas pequeños que se mueven perpendicular al vector de movimiento con la bala.

Estos asteroides mas pequeños serán destruidos la siguiente vez que sean impactados.

### Score

Cada vez que el jugador destruye un asteroide este gana un punto para el score.

### Pausa

El jugador tiene la habilidad de pausar o reanudar el juego en cualquier momento.

## SCRIPTS

### Bullet

Componente del *GameObject Bullet*. Se encarga de:

*Tiempo de vida*: Después de un cierto tiempo el objeto es destruido.

*Movimiento de la bala*: Actualiza al posición de la bala cada frame.

*Destrucción de asteroides*: Destruye el *GameObject* con el que colisiona si este es un asteroide, en el caso de los asteroides grandes también se encarga de crear los dos pequeños y calcular su vector inicial.

*Score*: Actualiza el score cada vez que destruye un asteroide.

### *EnemySpawner*

Componente de *MainCamera* se encarga de crear los asteroides en la parte de arriba de la pantalla y destruirlos después de un tiempo específico.

### *Meteor*

Componente del *Prefab MeteorSmall* se encarga de actualizar la posición cada frame y de destruir el *GameObject* después de un tiempo determinado.

### *PauseManager*

Componente de *PauseButton* pausa el juego cuando el jugador presiona el botón designado o hace click en el *GameObject*.

### *Player*

Componente de *Ship*. Se encarga de:

*Movimiento del jugador:* Aplica aceleración y rotación según los inputs del jugador.

*Disparo:* Crea el objeto *Bullet* y define su dirección.

*Posición de la nave:* Evita que la nave salga de la pantalla transportándola al lado contrario.

*Reinicio:* Reinicia el score y el la escena cuando colisiona con un asteroide.

## INTERACCIÓN

*Rotación:* El jugador puede rotar la nave presionando las flechas del teclado. *Izquierda* contra reloj y *derecha* viceversa

*Aceleración:* Presionando flecha *arriba* se gana aceleración en la dirección en la que apunta la nave y con flecha *abajo* en la dirección contraria.

*Disparo:* Al presionar la tecla *espacio* la nave dispara una bala en la dirección en la que apunta.