**🡪 MICROJUEGO: ASTEROIDS**

**1. Escenas**

* **Main Scene :** es el escenario único donde se desarrolla toda la acción. Contiene la nave, la cámara, los botones de pausa y el panel

**2. Assets y GameObjects**

**Principales en la escena:**

* **Player:** GameObject controlado por el jugador. Tiene un Rigidbody y colisionadores para detectar impactos. El objetivo es que la nave dispare balas a los meteoritos. 🡪 Player.cs
* **Bullet (Prefab)**: es la bala disparada por el jugador. Es un Prefab instanciado dinámicamente desde el jugador. 🡪  Bullet.cs.
* **EnemySpawner:** son los meteoritos que van cayendo del cielo. Si hay colisión con la nave, entonces se reinicia el juego. 🡪 EnemySpawner.cs.
* **UI Canvas**
  + Contiene:
    - **Puntos**: ir guardando la puntuación del juego.
    - **Botón Pausa**: al darle al botón pausas el juego y aparece el panel en e que se podrá renaudar el juego.
    - **Panel Pausa**: para parar el juego, contiene un botón para poder continuar jugando.
* **PauseManager**: se encarga de pausar el juego y poder renaudarlo. 🡪PauseManager.cs.

**3. Mecánicas**

* **Movimiento del jugador**
  + Controlado por teclado:
    - Flechas / WASD para girar y avanzar.
    - Nave se mueve con empujes
* **Disparo**
  + Con la tecla **Espacio** el jugador dispara balas.
  + Las balas se instancian en la posición del cañón del jugador (gun).
  + Se mueven en dirección de como la nave esta posicionada y desaparecen tras cierto tiempo o al colisionar con un meteorito.
* **Enemigos**
  + Spawneados automáticamente cada ciertos segundos.
  + Van cayendo cada x segundos de diferentes posiciones.
  + Al recibir un disparo, son destruidos y aumentan la puntuación. Si colisionan con la nave, el juego se reinicia.
* **Puntuación (Score)**
  + Se incrementa al destruir enemigos.
  + Se muestra en pantalla mediante UI Text.
* **Pausa**
  + Puede activarse con la tecla **ESC** o con el **botón Pausa** en la interfaz.
  + Al pausar:
    - Se congela el tiempo .
    - Aparece el panel de pausa con la opción de reanudar.
  + Al reanudar, todo vuelve al estado normal.

**4. Scripts**

* **Player.cs**
  + Gestiona movimiento y rotación del jugador.
  + Controla el disparo de balas.
  + Detecta colisiones.
  + Bloquea entrada si PauseManager.IsPaused es verdadero.
* **Bullet.cs**
  + Define dirección y velocidad de la bala.
  + Gestiona la vida útil del proyectil: destrucción tras tiempo o impacto.
  + Al colisionar con un enemigo:
    - Destruye enemigo y bala.
    - Actualiza puntuación en el marcador.
* **EnemySpawner.cs**
  + Temporiza el spawn de enemigos en la escena.
  + Usa Time.time / deltaTime para espaciar las oleadas.
* **PauseManager.cs**
  + Controla el estado de pausa del juego.
  + Escucha la tecla **ESC** para pausar el juego.
  + Gestiona UI de pausa (panel y botones).
  + Congela físicas en pausa.

**5. Interacción**

* **Controles del jugador**
  + Flechas / WASD → mover nave.
  + Espacio → disparar.
  + ESC → pausar/reanudar.
* **Interacción UI**
  + Botón Pausa (UI) → activa el panel de pausa.
  + Botón Reanudar (UI) → cierra el panel y continúa el juego.