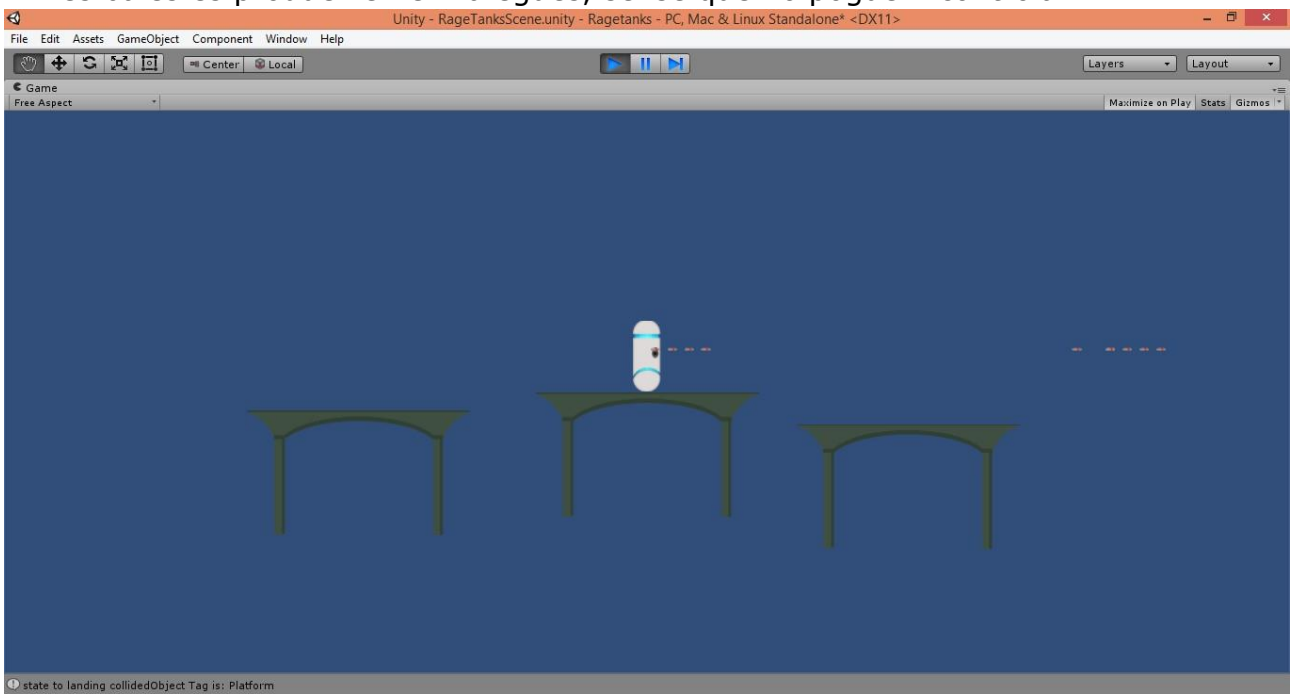
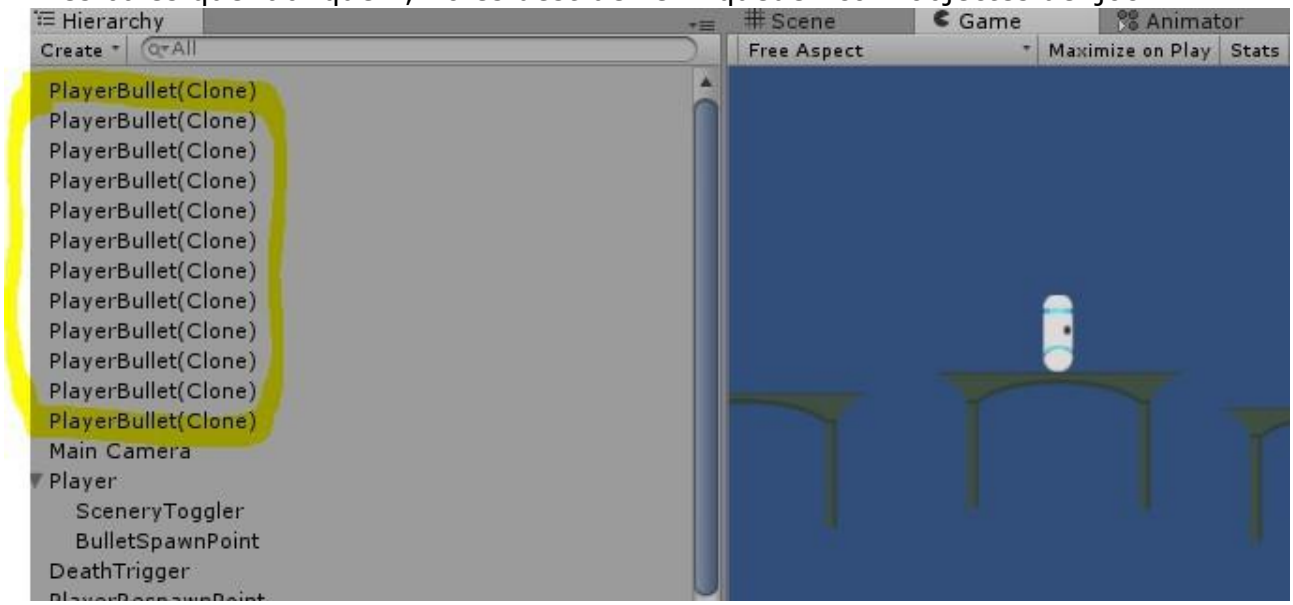
# 9. Últims detalls de firingWeapon

Encara que ja poguem disparar, si ens fixem en l'execució del joc hi podem trobar alguns defectes:

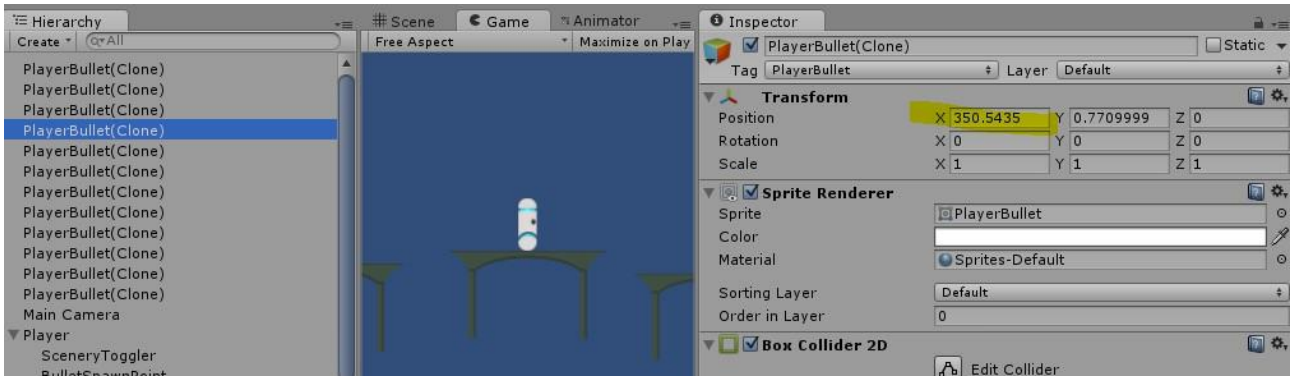
1. Les bales es produeixen en ràfegues, sense que ho poguem controlar:




2. Les bales que fabriquem, no es destrueixen i queden com objectes del joc ...




... i no deixen de moure's en la direcció en que s'han disparat:




## 9.1 Evitar les ràfegues incontrolades

Anem a implementar un sistema per establir un retard entre tret i tret. Si s'intenta disparar abans que passi un temps mínim, s'ignorarà el tret. Per això utilitzarem la funció **checkIfAbortOnStateCondition**.

Per associar un retard a l'estat **firingWeapon**, implementarem un sistema més general que permetrà associar-ne un a cadascun dels estats.

### 9.1.1: Crear les dades necessàries.

A l'script PlayerStateController, després de l'enumeració d'estats declarem el vector **stateDelayTimer**:

```
// Vector per gestionar temporitzacions en cada estat public static float[]
stateDelayTimer = new float[(int)playerStates._stateCount];
```

Es tracta d'un vector de tants elements com estats (per això utilitzem **_stateCount**) on carregarem valors per gestionar les temporitzacions associades a cada estat, si les necessitem.

**(Aquest vector quedarà disponible per gestionar temporitzacions per qualsevol estat del joc, si ho necessitem en algún moment).**

### 9.1.2: Inicialitzar les dades.

A l'script **PlayerStateListener**, el el mètode **Start()** hi escribim el següent codi per donar un valor inicial:

```
PlayerStateController.stateDelayTimer
            [(int)PlayerStateController.playerStates.firingWeapon] = 0.0f;
```

**(Recordem que la funció Start() es crida una sola vegada, quan apareix en escena l'objecte que té associat l'script).**

### 9.1.3: Implementar el retard.

En el mètode **onStateChange**, en les accions associades a **firingWeapon**, afegim aquest codi

```
// Establir temps a partir del qual es pot tornar a disparar
PlayerStateController.stateDelayTimer[(int)PlayerStateController.
                              playerStates.firingWeapon] = Time.time + 0.25f;
```

**Time.time** conté el temps en segons (amb decimals) des de l'inici del joc. Si es fan diverses consultes de **Time.time** en el mateix frame, s'obté el mateix valor.

**Veure http://docs.unity3d.com/ScriptReference/Time.html i**
**http://docs.unity3d.com/ScriptReference/Time-time.html i**
**http://answers.unity3d.com/questions/9757/timetime-explanation.html**

En aquesta instrucció hem agafat el temps actual més un quart de segon i l'hem carregat en el vector de gestió de les temporitzacions.

Finalment, en el mètode **checkIfAbortOnStateCondition**, en la comprovació corresponent a **firingWeapon** escribim:

```
// Ignorar si no ha passat prou temps
if(PlayerStateController.stateDelayTimer
      [(int)PlayerStateController.playerStates.firingWeapon] > Time.time)
      { returnVal = true; }
```

D'aquesta manera, el programa ignorarà l'acció de disparar fins que passin aproximadament 0,25 segons des de l'ultim tret realitzat.

## EXPERIMENT

Executar el joc tal com està en aquest moment, amb els últims canvis.

Esperimentar amb diferents valors en el camp **Mass** del **RigidBody2D** del projectil. Veure l'efecte d'aquests canvis sobre el moviment del projectil.

## 9.2 Destruir els projectils

Implementarem la manera d'eliminar les bales, en el moment que xoquin amb algun altre objecte del joc o quan hagi passat un cert temps des que han estat disparades.

### 9.2.1: Destruir el projectil quan hagi passat un segon.

A l'script **PlayerBulletController**, gestionarem un "temporitzador" de la vida de la bala.

Definim el camp `private float`

```
selfDestructTimer = 0.0f;
```

Com a última instrucció del mètode **launchBullet**, afegim la instrucció

```
//Establir moment d'autodestrucció
selfDestructTimer = Time.time + 1.0f;
```

de manera que, després de disparar una bala, selfDestructTimer contingui "l'hora" de dintre d'un segon.

Finalment sobreescriurem el mètode **Update()** amb el codi que destruirà l'objecte bala, si ha passat ja un segon des que va ser disparada.

```
void Update()
{
//Destruir l'objecte si ha consumit el temps
      if(selfDestructTimer> 0.0f)
      { if(selfDestructTimer<Time.time)
                  Destroy(gameObject);
      }
}
```

**(Recordem que la funció Update() es crida una vegada per frame).**

## EXPERIMENT

Executar el joc tal com està en aquest moment, amb els últims canvis.
Comprovar que els objectes projectil es destrueixen passat un segon.

### 9.2.2: Destruir el projectil quan col·lisioni amb un objecte.

Com que l'objecte Bullet té un **BoxCollider2D** que és trigger, pot detectar col·lisions i prendre la iniciativa d'actuar. Ja s'ha vist que Unity genera un event que pot ser capturat pel mètode **onTriggerEnter2D**.

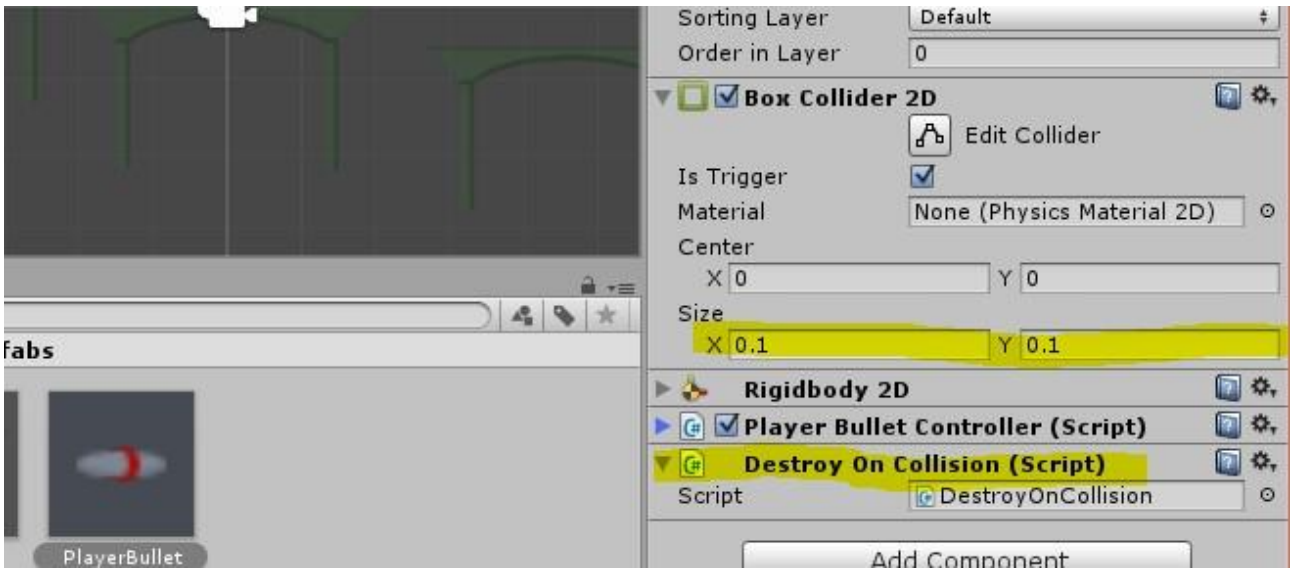
Així que creem un nou script **DestroyOnCollision**, que vincularem al Prefab PlayerBullet:

```
using UnityEngine; using
System.Collections;

public class DestroyOnCollision : MonoBehaviour
{ void OnTriggerEnter2D(Collider2D hitObj)
      {
            DestroyObject(gameObject);
      }
}
```

Últimes precaucions:

- Assegurar-se que s'ha vinculat l'script amb el Prefab de la bala.
- Revisar i ajustar els paràmetres del BoxCollider2D. Si la seva mida és massa petita, el sistema pot no ser capaç de gestionar correctament les col·lisions.




**EXPERIMENT**

Executar el joc tal com està en aquest moment, amb els últims canvis.
Disparar sobre una plataforma i comprovar que el projectil desapareix en col·lisionar.

## Activitats

1. Implementar una segona arma del Player, amb projectils que li surtin de la part superior amb trajectòria parabòlica. **(Fer-ho en una branca independent del projecte, per no barrejar l'exemple principal amb els exercicis).**

**Suggeriments**: nou estat **firingMortar**. Aplicar forces sobre el projectil de manera similar al salt del Player. Generació i eliminació de projectils com la de PlayerBullet. Es pot utilitzar un sprite diferent del de la bala normal.

2. Fer un llistat amb els conceptes de introduits en aquest document, indicar la pàgina i explicar què són i per a què s'utilitzen.