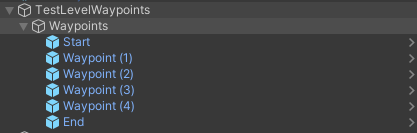
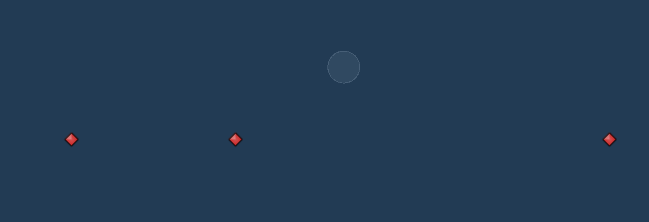
## **Level Creation –**

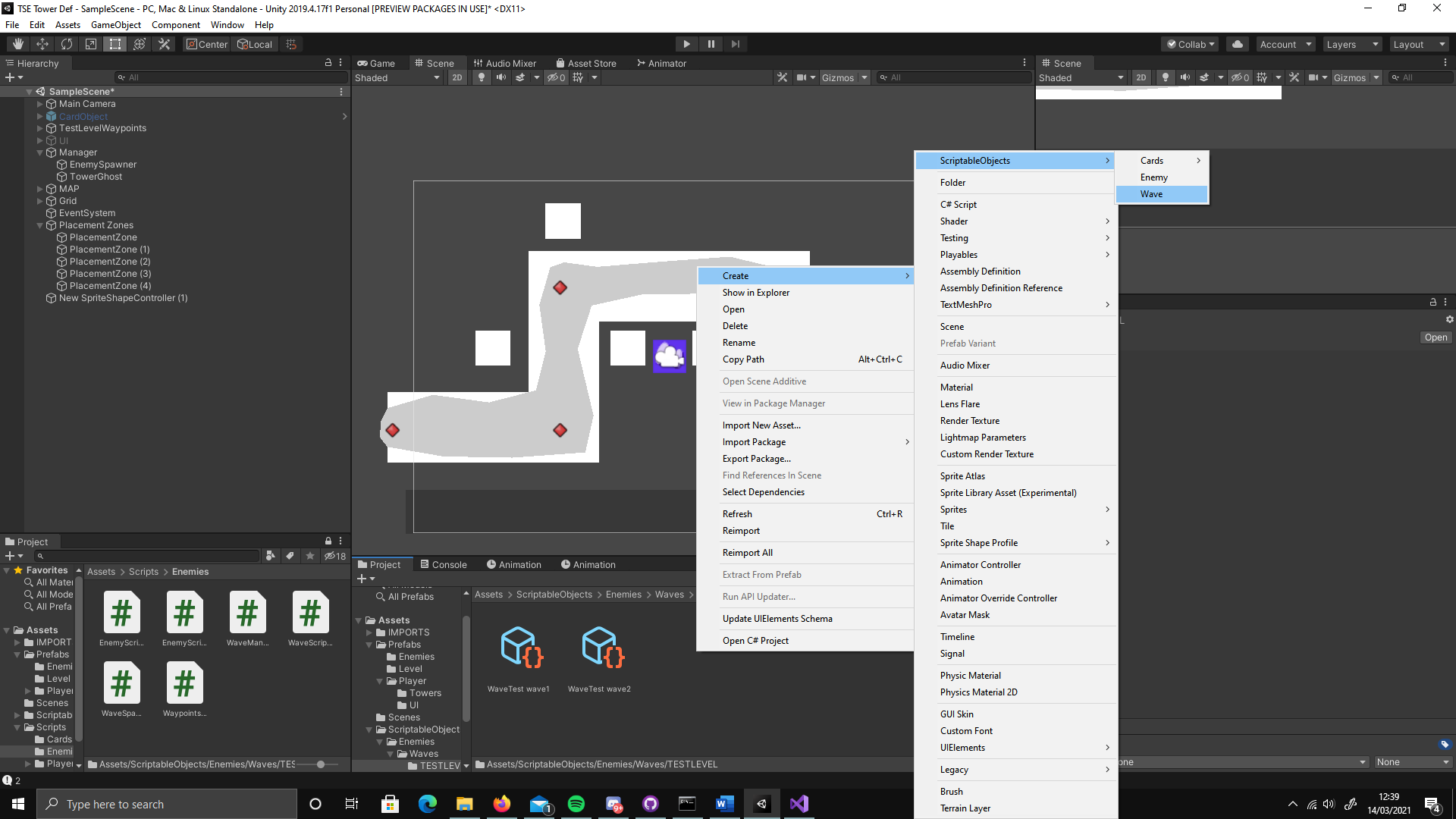
**Waypoints –**

The prefab “Level Waypoints” can be used to create a series of empty objects shown to the right. Adding additional waypoint prefabs between Waypoint(1) and end will add additional points for the enemies to path to, as shown lower right.

The red diamonds shown below are what represents the Waypoint empties, and do not appear in the actual game, only the editor, allowing for clearer representation of the path being formed.

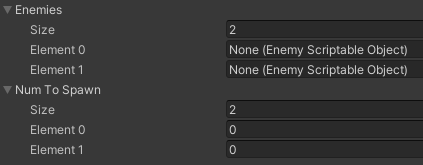
It is advised to create a level map which then has the waypoints following a path set I n the artwork.

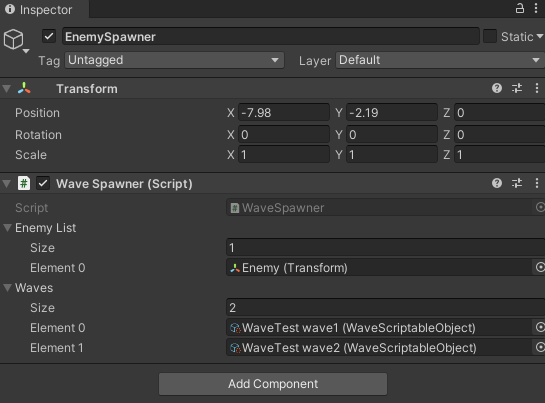
### **Tower Placement –**

 The prefab object “Placement Zones” provides an empty to store the zones and a single placement zone which can be duplicated to cover a map. The zones are tagged as “Placeable Zone” to work in the tower placement coding.

### **Enemy Waves -**

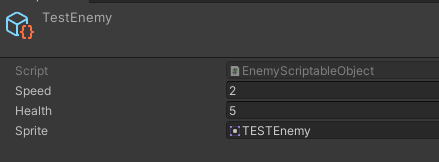
To create a wave, create a scriptable object by right clicking in the project window, Create, Scriptable objects, Wave. (Right)

The scriptable object will have 2 fields, Enemies Size and Num to Spawn size. Both should have the same size. The enemies fields take an enemy scriptable object, this will be what is spawned by the script using this object, the Num To Spawn sets how many of each unity to spawn, applying the position to the matching place in the Enemies, I.E. Enemies[1] spawns Num To Spawn[1].

 Once Waves are created they can be applied to the Enemy Spawner script under manager, the Enemy (Transform) object is essentially an empty base for the enemy stats and sprites to be applied to. Every wave should be added in order here.

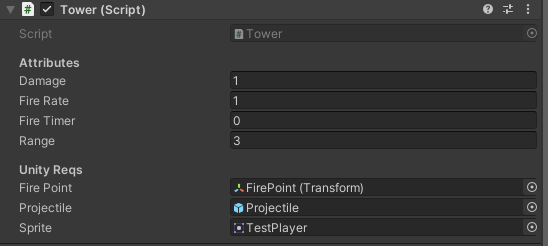
## **Objects**

### **Enemies –**

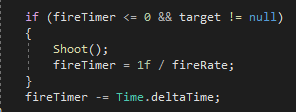
Creating a new enemy is done in the same way as a wave, create a Scriptable object Enemy and fill in the fields as necessary.

(This system may be updated to support more complex enemies.)

### **Towers –**

New towers currently require a new script or additional coding on the “projectile” script to allow for additional On-Hit effects. The 2 current projectiles are BASIC and EXPLOSIVE, with a single variable (explosion radius) deciding whether or not the effect occurs.

The Tower prefab object currently has 7 visible fields.

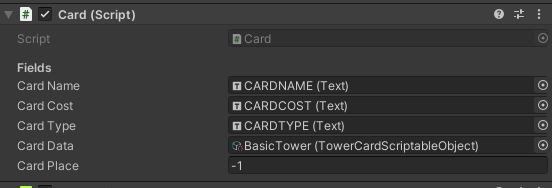
ATTRIBUTES  
Damage – The damage done to enemies hit.  
Fire Rate – The amount of shots per second. 1 second / fire rate.  
Fire Timer - The timer to count how long until the next shot, primarily visible for debugging. If set to a value there will be a delay of that many seconds until the first shot.  
Range – how many distance units the tower can fire over.

UNITY REQS  
Fire point – Where the projectile will fire from on the tower, on the test models it is the centre.  
Projectile – The projectile being fired by the tower, currently what applies special effects such as explosion.  
Sprite – The sprite being used to represent the tower.

NONE OF THESE FIELDS CAN BE LEFT BLANK.

### **Cards –**

Cards Utilise scriptable objects again since they will be reused so frequently. The “Card” fields can be filled out as necessary to reflect the card, name, what it does and some card art. The standard stats will actually affect gameplay, currently the cost of the tower will effect currency needed to play the card and object to Make is a prefab of a created tower.

The card prefab will have all values from the card scriptable object applied to it through the deck script using the Card Data field. The fields shown in the image on the right are the text fields which will be filled, these are prefilled in the prefab so do not require altering.