MMFPSE: Zombies Addon

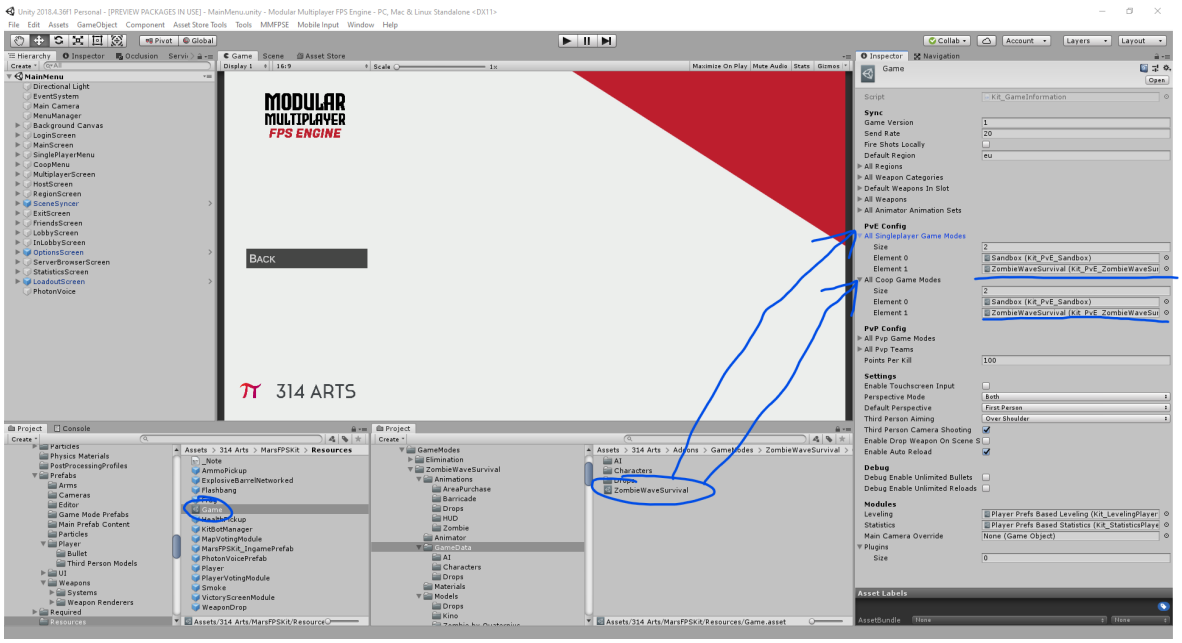
Version 1.0.0.0

Requirements

This version of the Zombies addon requires at least MMFPSE 0.7.4.0

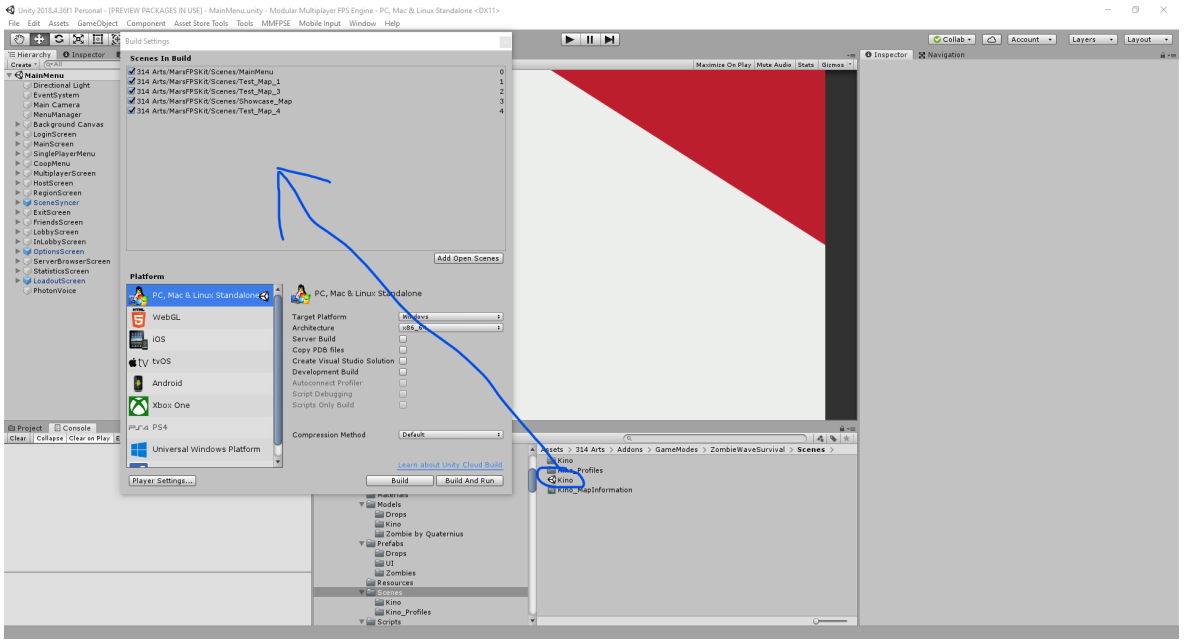
Getting Started

To add the game mode into the game, open the "**Game**" file (**314 Arts/MarsFPSKit/Resources**) and drag the "**ZombieWaveSurvival**" file (**314 Arts/Addons/GameModes/ZombieWaveSurvival/GameData**) into the "**All Singleplayer Game Modes**" and "**All Coop Game Modes**" array:



You can remove the “Sandbox” game mode if you want, but make sure not to leave an entry that says “Null” or “None” behind.

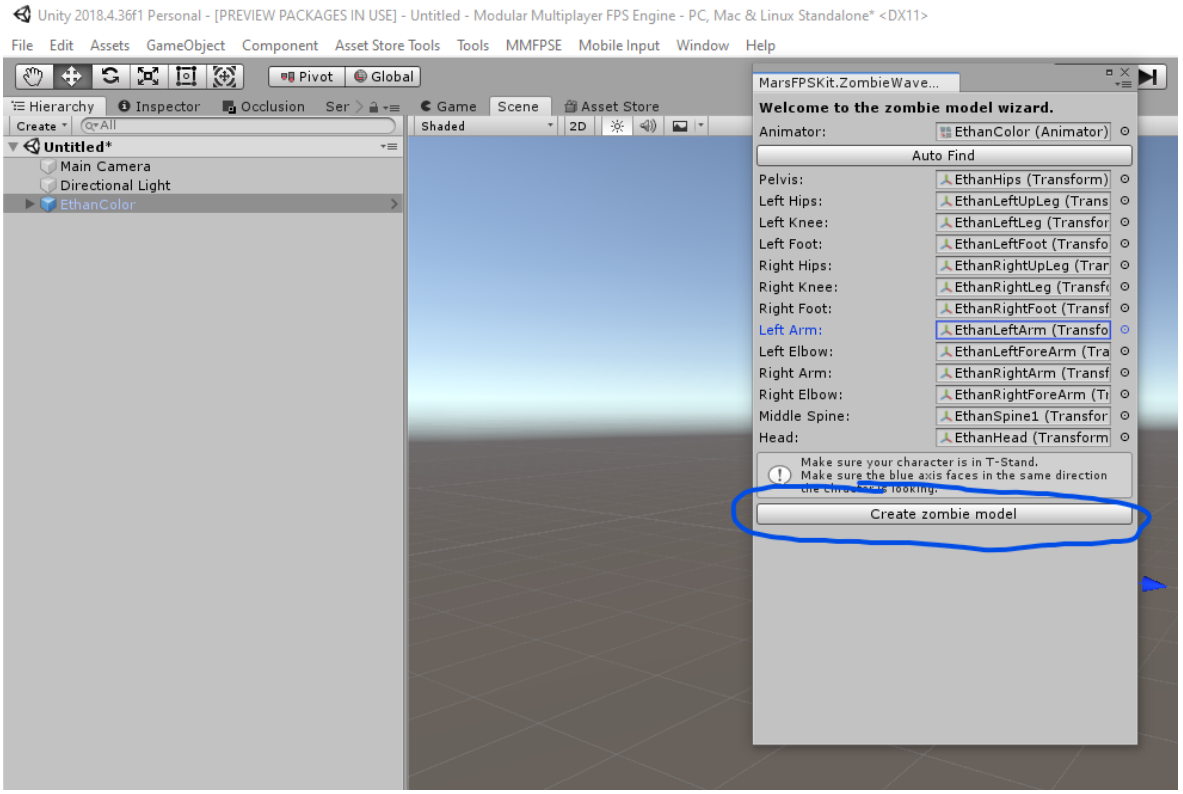
Then you just need to add the default map to the build settings.
Open the Build menu by clicking on “**File -> Build Settings**” in the top bar.
Drag the “**Kino**” scene (**314 Arts/Addons/GameModes/ZombieWaveSurvival/Scenes**) into the window, like this:



Adding a new zombie

For adding a new zombie “skin”, there is a wizard. On the top bar click on

“**MMFPSE/Zombies/Setup new zombie model**”. This screen is very similar to the Unity Ragdoll wizard. Assign your new model’s animator and assign all the bones to it.
Then click “**Create zombie model**”:



The wizard will save a **prefab** and an **asset** file.

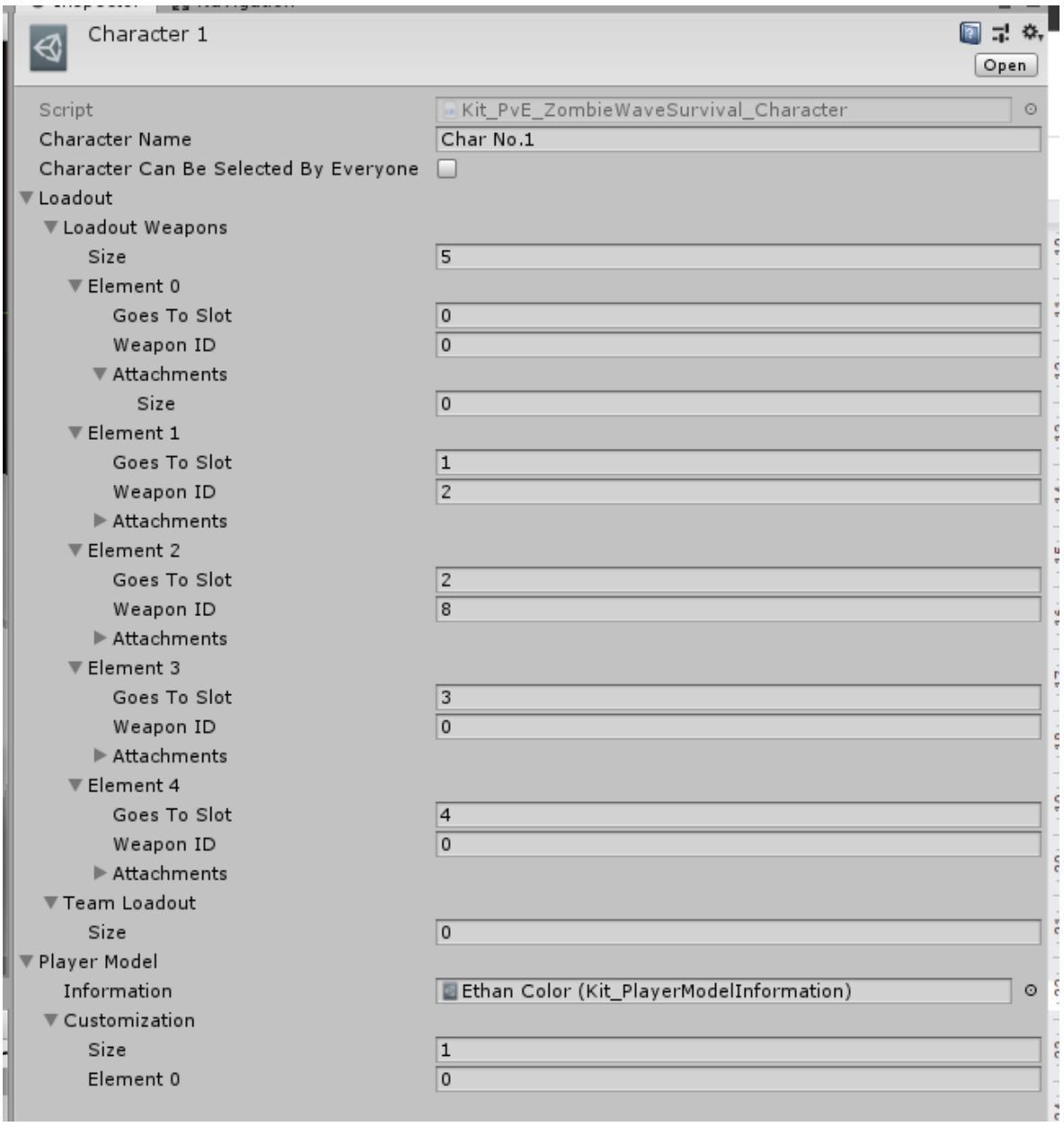
To put the zombie in game, you simply have to assign it (**the prefab**) to the “**Zombie Prefabs**” array in the “**ZombieWaveSurvival**” file (314 Arts/Addons/GameModes/ZombieWaveSurvival/GameData).
Then the skin will be used in the configured waves (between “Spawn After Wave” and “Spawn Until Wave”).

The settings file that was created by the wizard has no attacks and sounds assigned to it. For reference on what you can put here, take a look at the “**Ethan Zombie Fast/Slow**” which you can find in “314 Arts/Addons/GameModes/ZombieWaveSurvival/GameData/AI”.

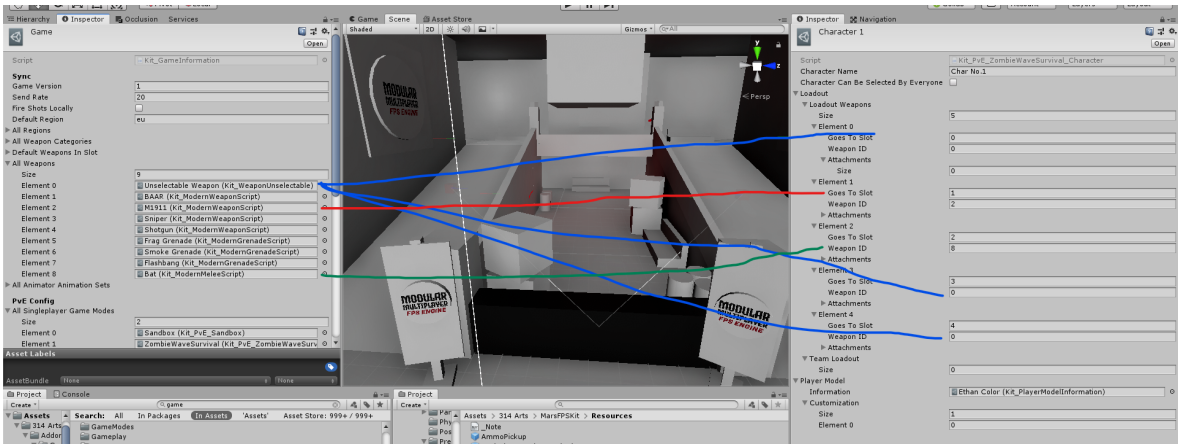
If you are unsure what a variable means, you can hover over it with your mouse and a tooltip will appear explaining what it means.

Adding a new character

A “Zombie” player character consists of a **spawn loadout** and a **Player Model** (guide on how to create this is in the main Kit documentation):



The Loadout numbers can be found in the “Game” file:



Please do note that you should have at least one weapon **that is not a "Kit_WeaponUnselectable"** (The "Unselectable Weapon" in the screenshot).
This "Unselectable Weapon" is a **placeholder** that can not be used until it is replaced by another weapon during gameplay.

The "**Character Name**" is the string that is displayed in the pre-game lobby.
If "**Character Can Be Selected By Everyone**" is checked, this character can be selected by more than one player.

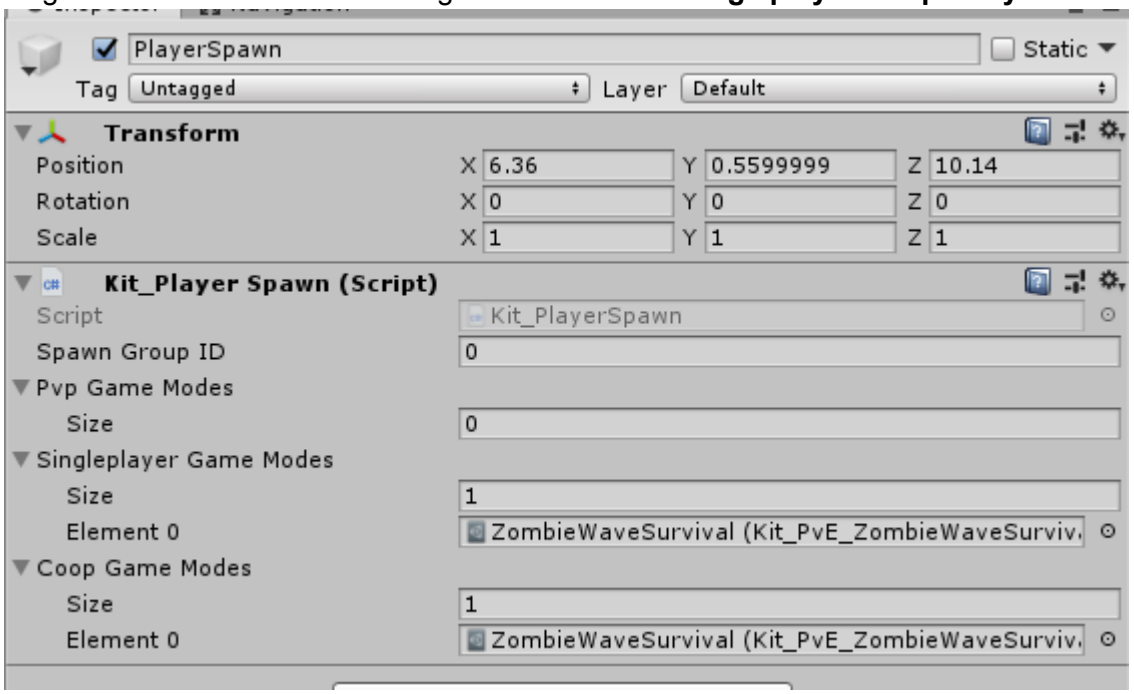
Ideally you should have as many characters as you allow players or at least one character that has this option checked.

Adding a new map

Adding a new map to the Zombie game mode is very similar to the main kit.

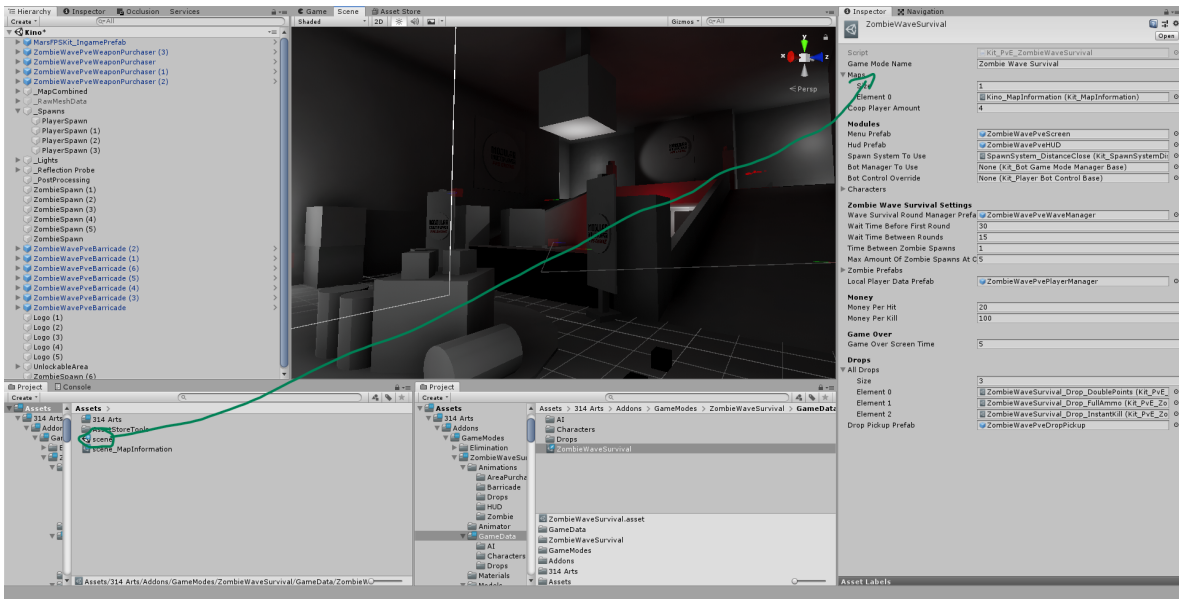
First, set up your scene as a kit map by using the "**MMFPSE/Scene/Setup**" tool in the top bar. This will save the scene if not already done and add the ingame prefab.

Now you need to add **player spawn**(s) to it, simply add the Kit_PlayerSpawn script and drag the "**ZombieWaveSurvival**" game mode to the **Singleplayer/Coop array**:



The next step is to **add at least one zombie spawn** to it, see the section on Zombie spawns.

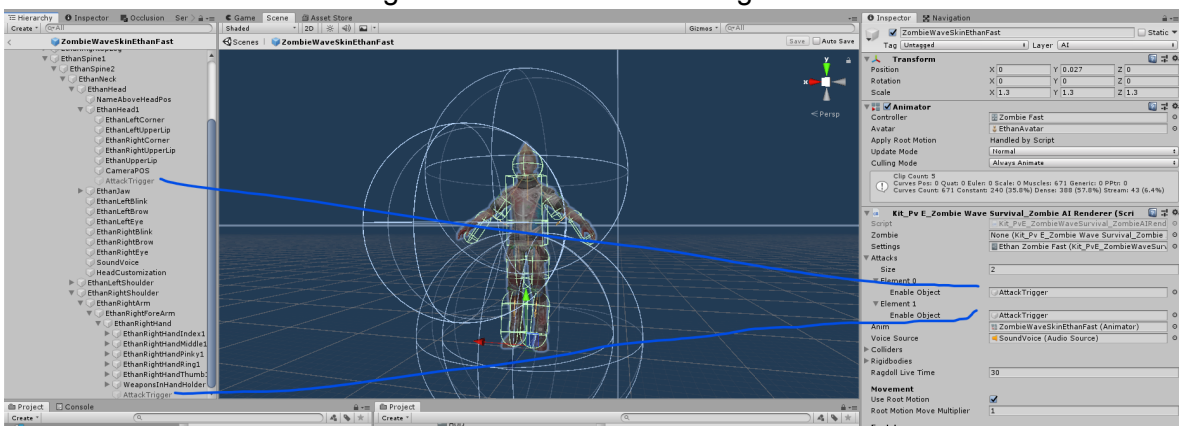
Now you can add your map to the "**ZombieWaveSurvival**" game mode file:



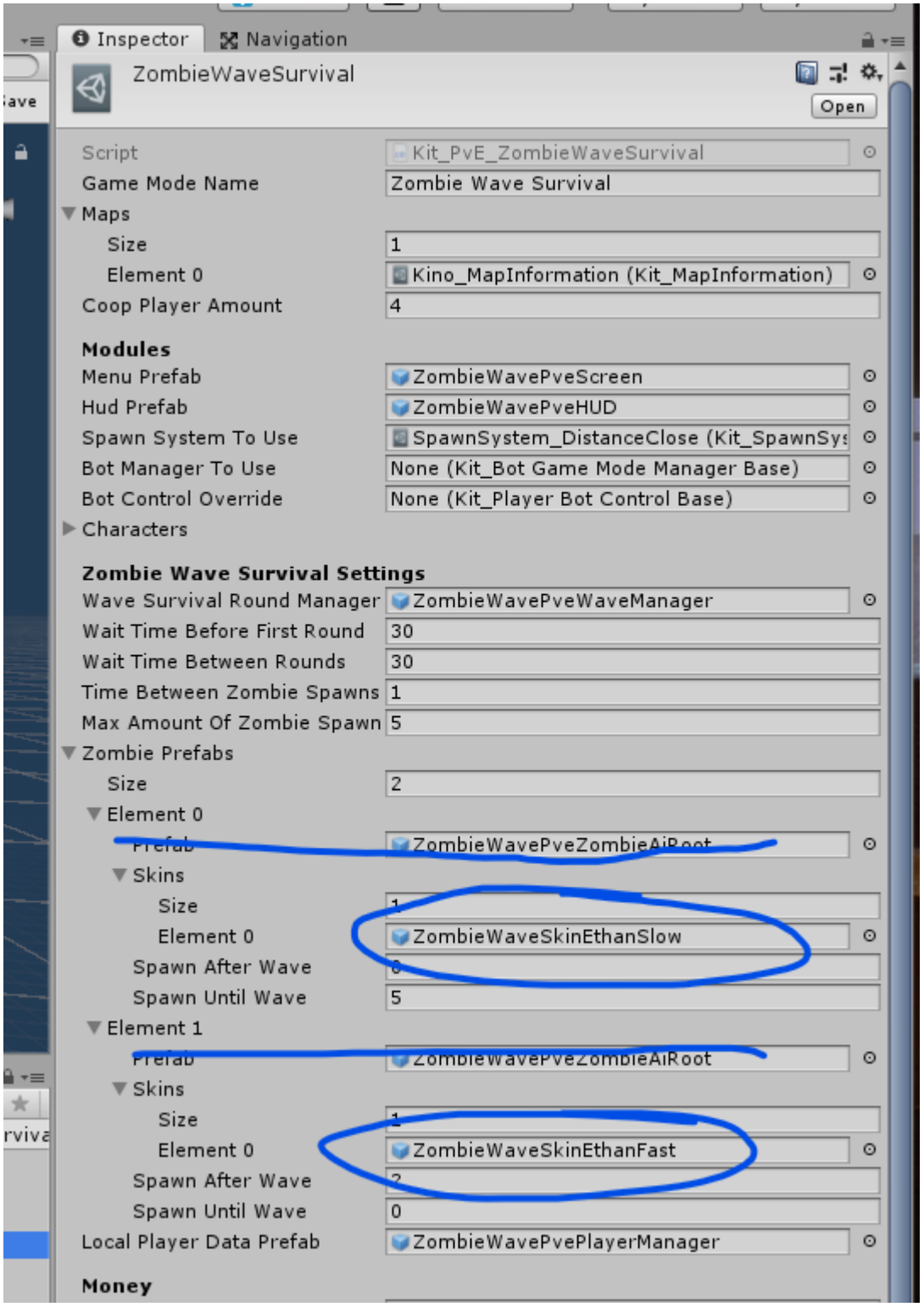
Simply drag its **_MapInformation** file (this was created by the Scene Setup tool) to the "**Maps**" array in the game mode.

Zombie Attacks

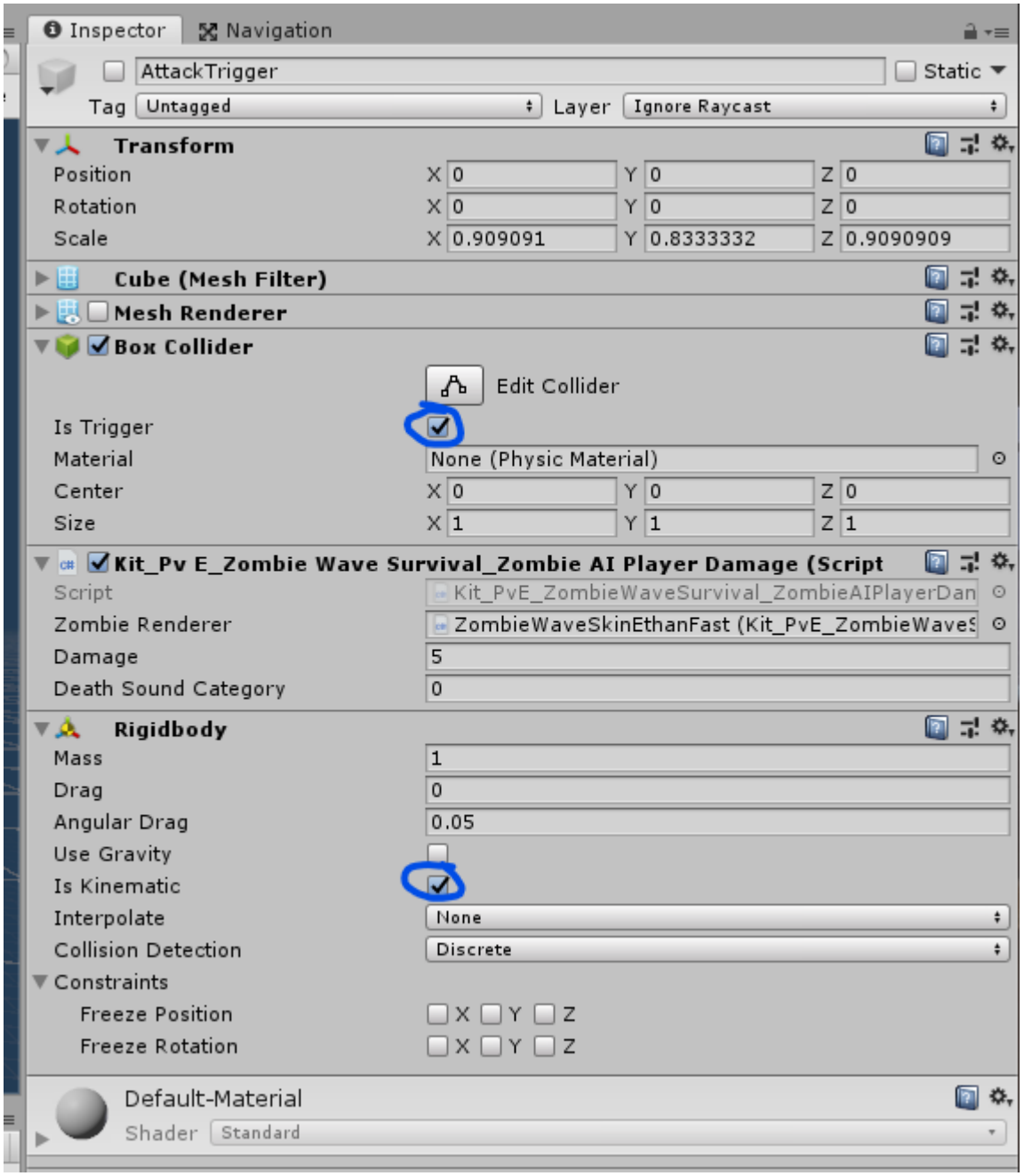
To apply damage to a player upon an attack, a Game Object gets enabled during the duration of the attack as configured in the "Zombie Settings" file:



The triggers for this are part of the Zombie prefab that you assigned in the "**Skins**" array. NOT part of the "Prefab" (ZombieWavePveZombieAiRoot):



An attack trigger consists of a trigger (any form of collider with "**IsTrigger**" set to **true**), the damage script (**Kit_PvE_ZombieWaveSurvival_ZombieAIPlayerDamage**) and a **kinematic Rigidbody**:



(The Mesh Filter and Mesh Renderer are just there to easier visualize how the trigger looks, they are not necessary for the function).

Important:
The zombie attack trigger's layer should be "**Ignore Raycast**". Otherwise you will hit it with bullets.

Zombie Footsteps

Zombie Footsteps work via Animation events.

The event is called

Footstep

in the

Kit_PvE_ZombieWaveSurvival_ZombieAIRenderer

script. It takes an int as a parameter which tells us which foot to use. The script must be on the same object as the zombie's animator. If you use the wizard, this is done automatically.

The feet are configured in the same script under the

Foot Steps

array.
If you use the wizard to set your zombie model up, they are automatically created.

The sounds for the zombie are stored in his Settings file under

Footstep Sounds

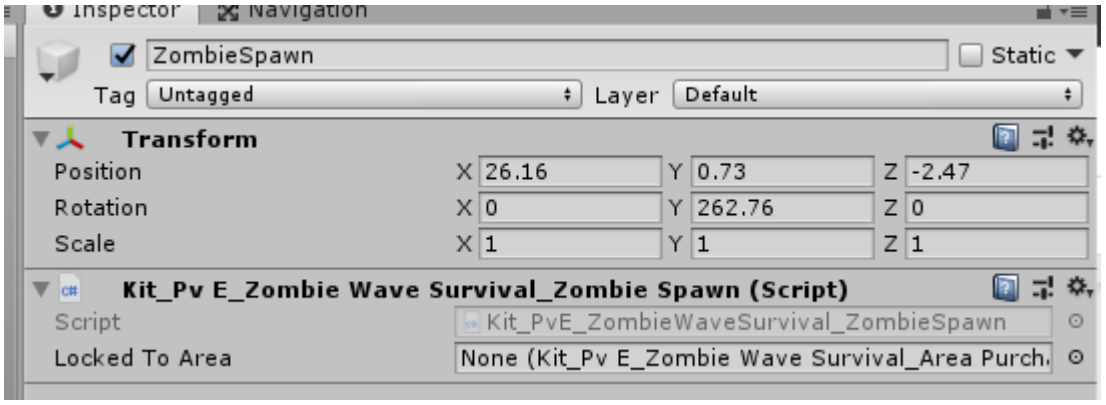
This is a dictionary that maps a string (the **Tag of the object** you are walking on to a **ZombieFootstep**, which simply includes an array of sounds to play.

Zombie Spawns

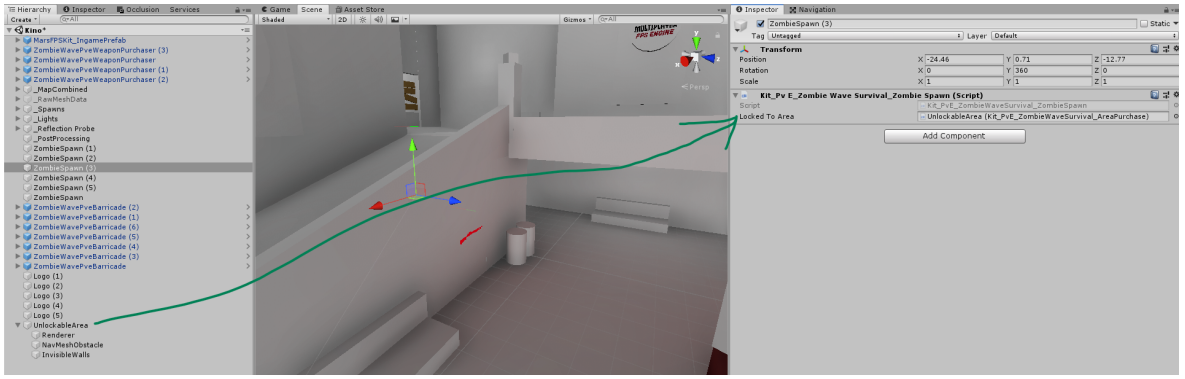
A zombie spawn simply is a game object with the

Kit_PvE_ZombieWaveSurvival_ZombieSpawn

script:

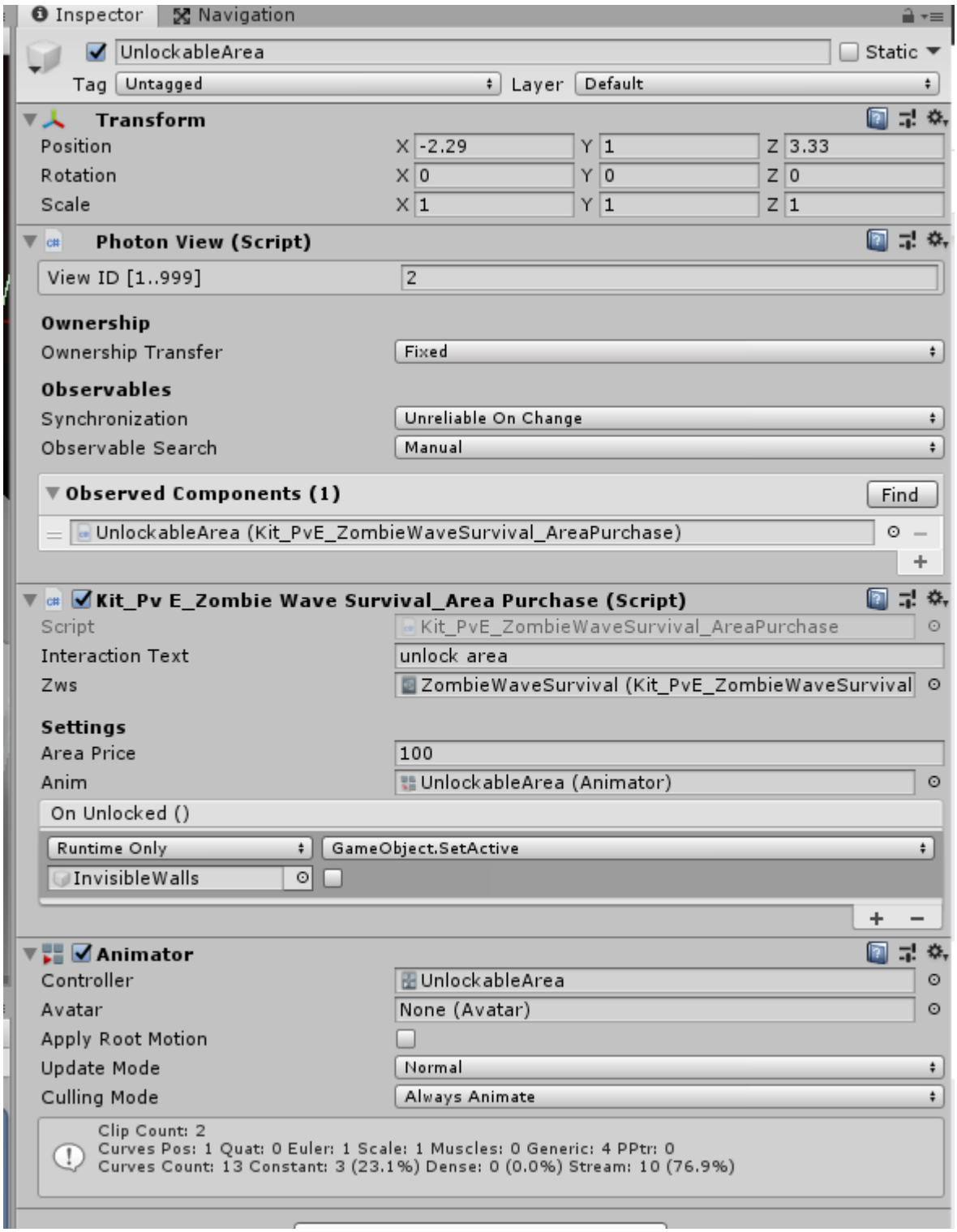


If a spawn is in an area that has to be unlocked first, assign that area to the "**Locked To Area**" variable:



Unlockable Areas

An unlockable area consists of a **Photon View**, the script (**Kit_PvE_ZombieWaveSurvival_AreaPurchase**) and an **Animator**.
You will also need **Invisible walls** to block access to the area before it was unlocked and a **NavMeshObstacle** to stop the Zombie AI from walking in it:



The **NavMeshObstacle** is by default controller via the animator, but it can also be disabled via the “OnUnlocked()” event.
The **Invisible walls** are by default controlled via the event but they can also be controlled via animator.

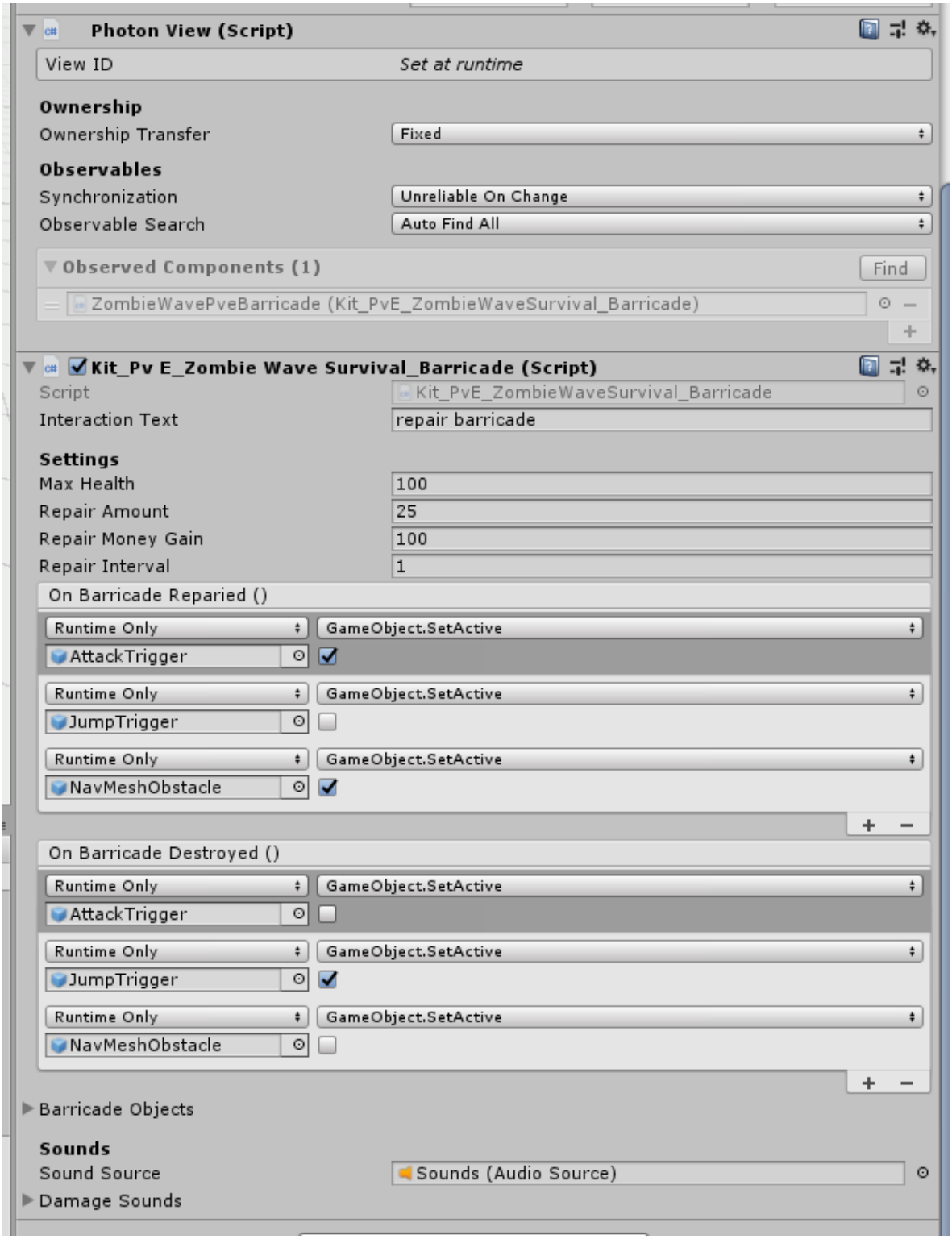
The **animator** has a bool called “**unlocked**” set to it via the script. Use this to switch to an “Open” / “Unlocked” state in the Animator Controller.

To be able to interact with the area, you need a collider/trigger under/on (in the hierarchy)

it, that is **not** in the “IgnoreRaycastAndPickup” and “PlayerRoot” layer.

Barricades

This is the barricade script:



It requires a PhotonView and that it is added under "**Observed Components**". Photon can handle that itself now, but just make sure it is there.

The default prefab is called "**ZombieWavePveBarricade**". You can use this or use it as a reference point.

Interaction text is what is displayed to the user after the "Press [F] to: "

Max Health is the start health and maximum health of the barricade

Repair Amount is the amount of health that gets restored to the barricade each repair interval

Repair Money Gain is how much money a player gains for repairing a barricade (per interval)

Repair Interval is the amount in seconds between repairs while holding F

On Barricade Repaired() gets fired once the barricade goes from 0 health to more than 0. Here you should enable the attack trigger, disable the jump trigger and enable theNavMeshObstacle

On Barricade Destroyed() gets fired once the barricade has reached 0 health from zombie attacks. Here you should disable the attack trigger, enable the jump trigger and disable the NavMeshObstacle

The "**Barricade Objects**" are the visual objects that make up the barricade, the wooden planks, so to say.

They should be ordered by element #0 being the one that gets destroyed first and the last one is the one that gets destroyed last. This is important for the repair process

The barricade objects have sounds they can play when being **destroyed/repaired** via the sound source (see below).

They also have an animator that plays a "**Destroy**" animation and a "**Repair**" animation at the given times.

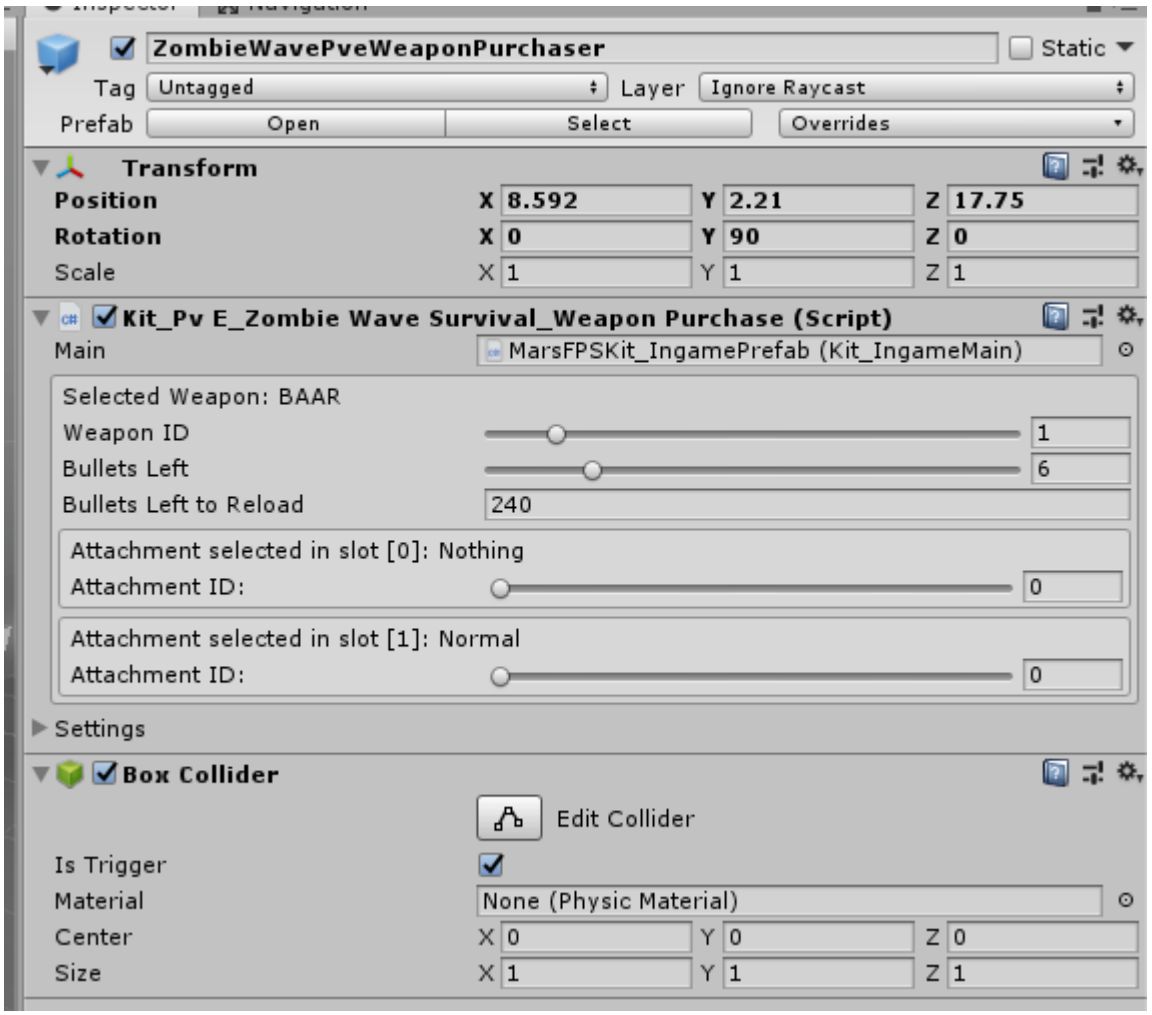
The **Sound Source** is used for all sounds of this barricade. Damage, Destroy and Repair sounds.

The **Damage Sounds** get played every time the barricade is hit by a zombie.

For reference use the default prefab as mentioned above.

Weapon purchase

To turn an object into a weapon purchase point, simply add the **Kit_PvE_ZombieWaveSurvival_WeaponPurchase** script to it as well as a trigger somewhere, like this:



There is also a prefab for this called
ZombieWavePveWeaponPurchaser
On the script, you can configure which weapon you can buy there, with how much ammo and with which attachments it is bought.
Under **settings** you can configure the **price** for the **weapon** and the **ammo**.

FAQ

How to change how many zombies spawn each wave?

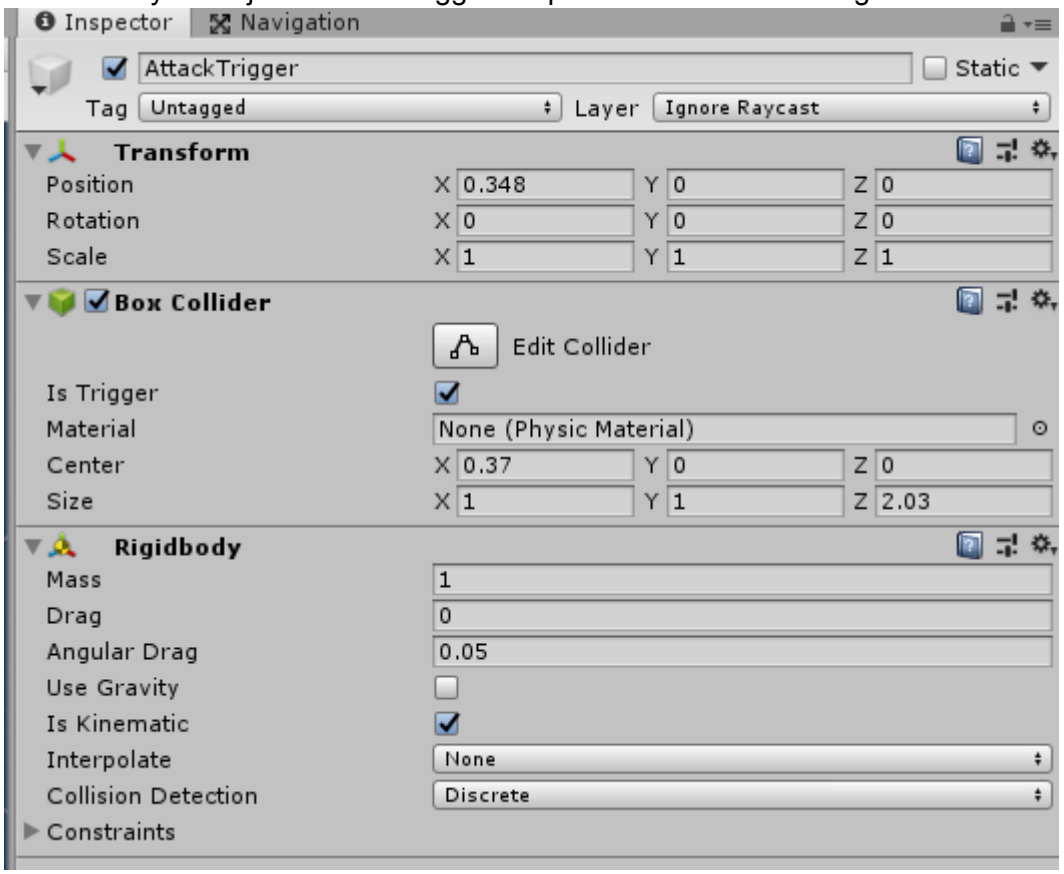
Because the wave survival mode supports infinite waves, the amount of zombies is defined with a formula.

To change the formula, you have to open the
Kit_PvE_ZombieWaveSurvival_WaveManager
script. The amount of zombies is defined in:
public static int GetAmountOfZombies(int round, int players)
The default is:

return (players * 5) * round;
You can change this to anything you like.

My zombie triggers don’t work!

This is most likely caused by the lack of a Rigidbody.
The trigger system is part of the physics system and only functions properly with a rigidbody.
Add one to your object with the trigger script on it with these settings:



I hit the attack triggers! (Concrete impact on zombie triggers)

Make sure that the layer of the triggers is set to “**Ignore Raycast**” (see picture above)

My zombies don’t move!

Make sure that you have a NavMesh baked.
Every time you make bigger changes to your map you need to re-bake the NavMesh.

Need Help?

Do you have a problem that you cannot solve using this document? You can seek help and/or join our community on our Discord here!
Be sure to have your Invoice ID ready to confirm your purchase of MMFPSE (if not already done) and the Zombies Addon.

Changelog

1.0.0.0

Initial Release