DarkTree FPS System

Manual document

v.1.2

Content:

- **1. Introduction**
- **2. Layers setting**
- **3. Getting started**
- **4. FAQ**
- **5. Components overview**
    - *Weapon.cs*
    - *WeaponSettingSO.cs*
    - *FpsController.cs*
    - *WeaponManager.cs*
    - *Sway.cs*
    - *FootstepSound.cs*
    - *HitFXManager.cs*
    - *UseObjects.cs*
    - *GetCollisionTag.cs*
    - *ObjectHealth.cs*

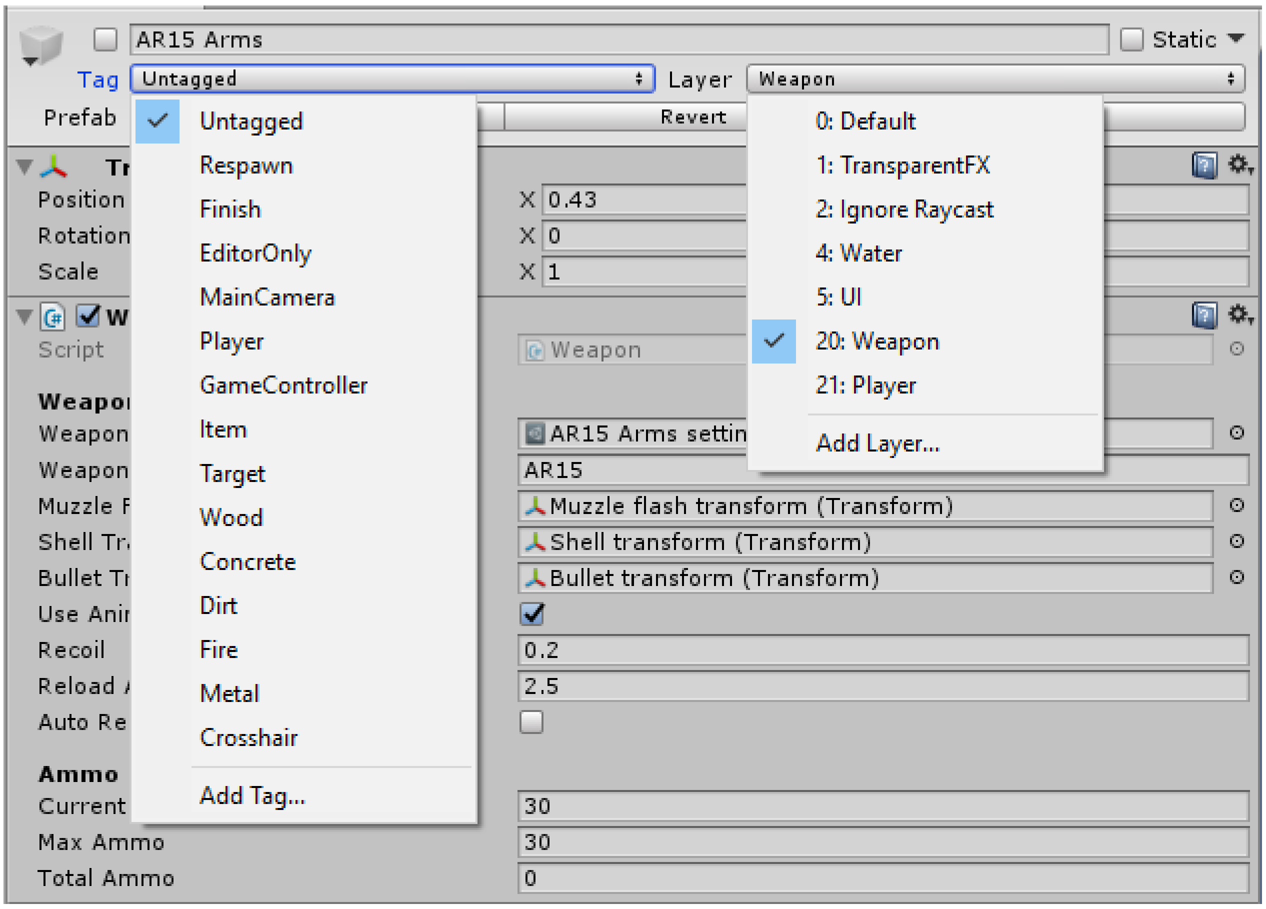
Email: darktreedevelopment@gmail.com Nick Kucher 2019

Introduction

DarkTree FPS System is a powerful, simple and fast way to create weapons for your FPS game. All the scripts are commented in the important parts of the code so you can modify system according to your purposes if you need.

Main script of the system for each weapon is the Weapon.cs. Weapon.cs provides shooting, reloading, hitting and other main functionalities to make your weapon work. Each Weapon.cs must have scriptable object with a weapon settings. System have ready effects and sounds for your weapon set in the 'GamePrefab'. I strongly recommend you to use 'GamePrefab' before you get how system works. Using 'GamePrefab' is guaranteed way that all the features will work out of the box.

Layers setting (IMPORTANT)


AR15 Arms
Static
Tag Untagged
Layer Weapon
Prefab
Revert
Position X 0.43
Rotation X 0
Scale X 1
Script Weapon
Weapo
Weapon AR15 Arms settin
Weapon AR15
Muzzle F Muzzle flash transform (Transform)
Shell Tr Shell transform (Transform)
Bullet Tr Bullet transform (Transform)
Use Anir ✓
Recoil 0.2
Reload 2.5
Auto Re ☐
Ammo
Current 30
Max Ammo 30
Total Ammo 0

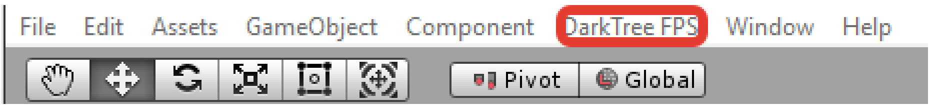
Email: darktreedevelopment@gmail.com Nick Kucher 2019

Getting started with the Weapon Wizard.

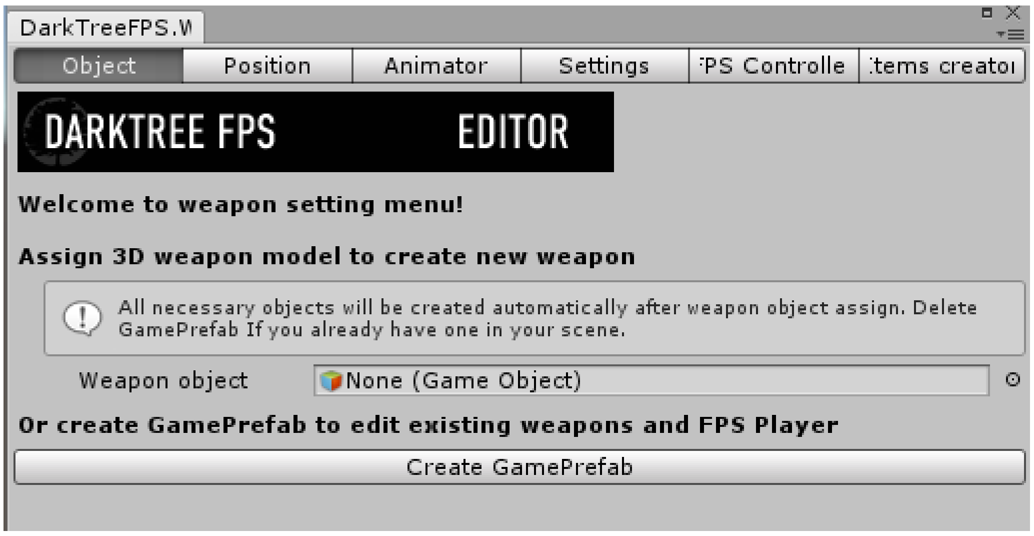
Setting a new weapon guide

1. Create new empty scene

2. Go to the DarkTree FPS → Weapon Wizard.



3. Drag your weapon 3d model to the 'Weapon object' field or expand project view and find your object with click on the field.



Welcome to weapon setting menu!

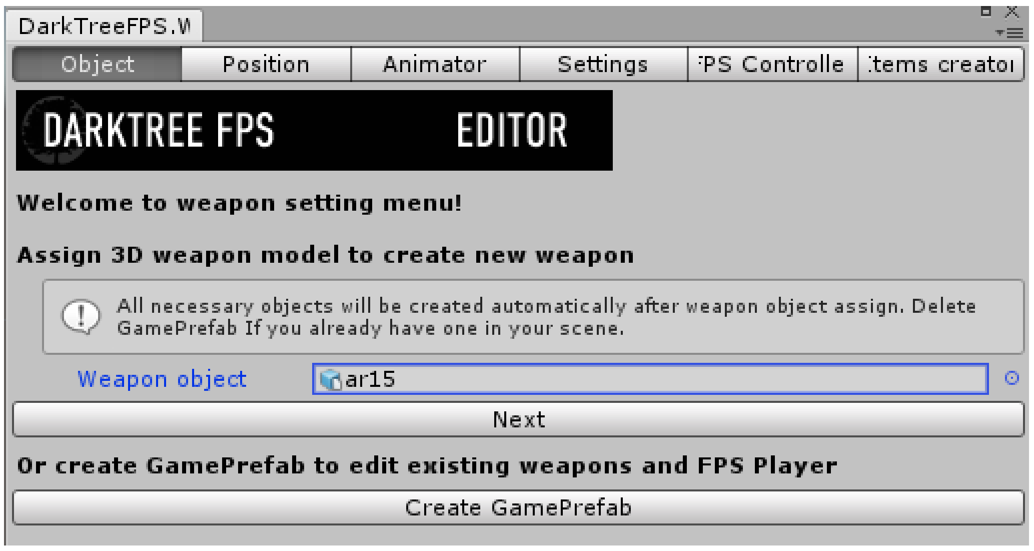
Assign 3D weapon model to create new weapon

All necessary objects will be created automatically after weapon object assign. Delete GamePrefab If you already have one in your scene.

Weapon object: None (Game Object)

Or create GamePrefab to edit existing weapons and FPS Player

Press 'Next'



Welcome to weapon setting menu!

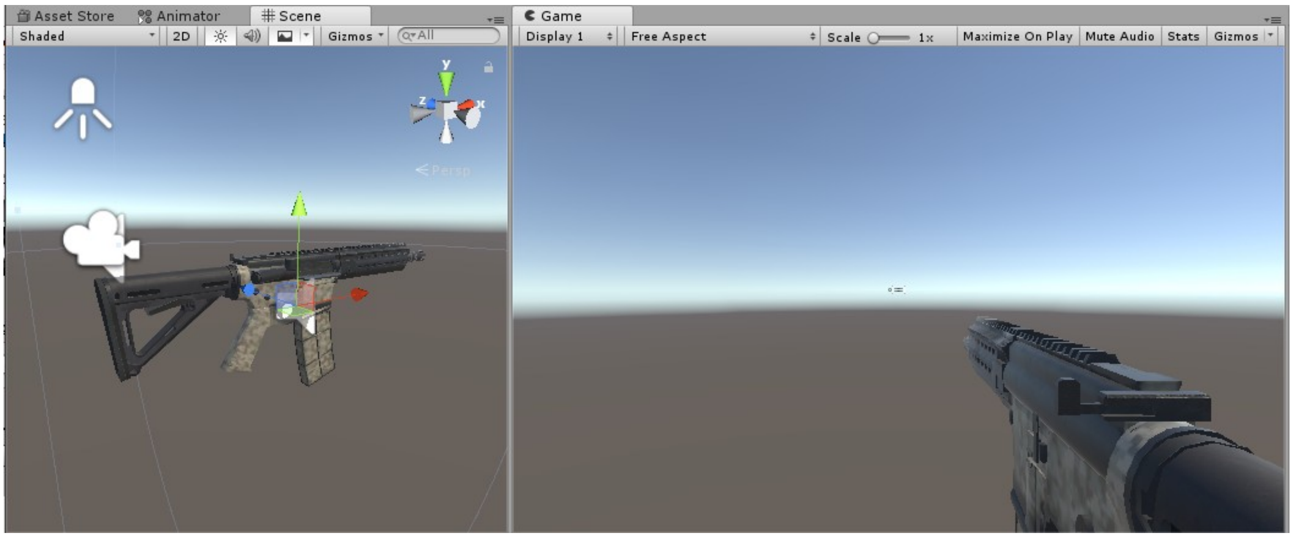
Assign 3D weapon model to create new weapon

All necessary objects will be created automatically after weapon object assign. Delete GamePrefab If you already have one in your scene.

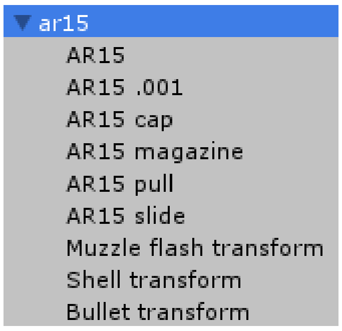
Weapon object: ar15

Or create GamePrefab to edit existing weapons and FPS Player

4.1 Move and scale your weapon for the proper view in the Game window



4.2 Click on ‘Create helpers for weapon’. Three empty gameobject will added to your weapon.



ar15
- AR15
- AR15 .001
- AR15 cap
- AR15 magazine
- AR15 pull
- AR15 slide
- Muzzle flash transform
- Shell transform
- Bullet transform

Z axis is the forward direction of each gameobject so make sure that every gameobject look in the right direction.

Muzzle flash transform – transform where a muzzle flash effect will be spawned.

Shell transform – transform where a shell object will be ejected on shot.

Bullet transform – transform where is a bullet start position. Forward axis must be aligned with the weapon muzzle to work proper.

After you have done with the helpers positioning click ‘Next’

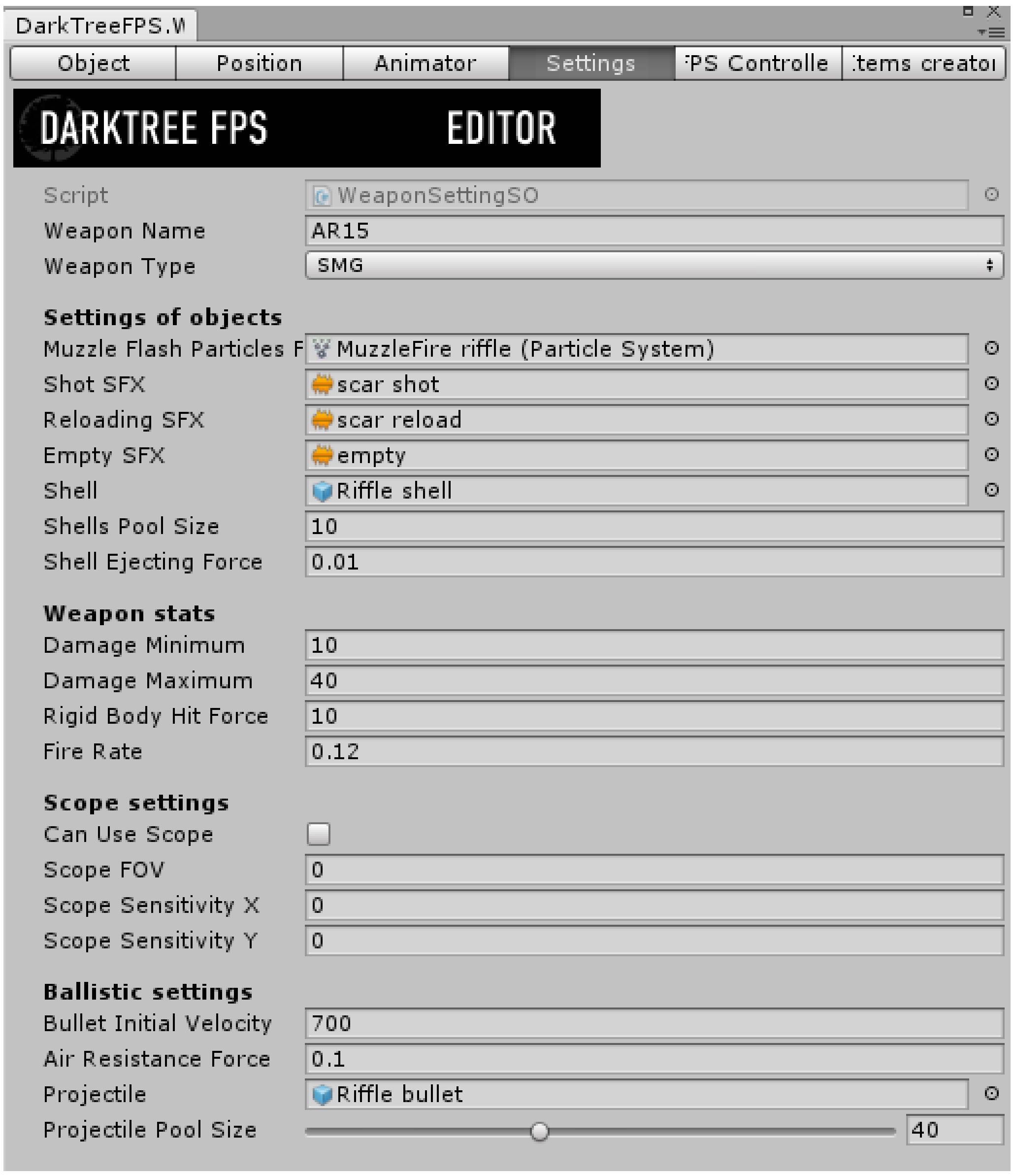
5. If you have animations for your weapon press ‘Use animation’ and assign animations to the inspector fields. Aim position animation must have only one frame. If you creating sniper riffle, aim position must be outside camera FOV.

If you have not any animations at now you can assign them later with the animation controller. The animation controller will have name of weapon you creating.

Press ‘Create animator’. **The animator must exist on the weapon gameobject for proper work!**

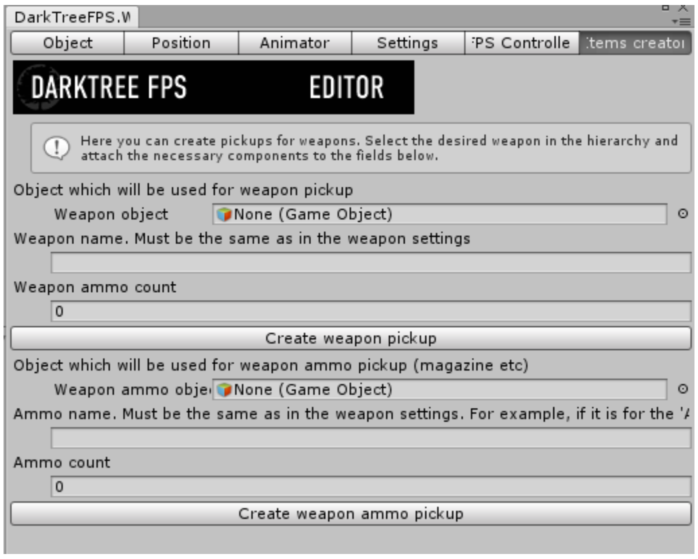
6. Go to the Settings tab and create new scriptable object for the weapon. And set variables as you need. Package contains ready sample effects so you can apply them from project.

Each variable have editor tooltip in the inspector view if you have to get more information about it. Also documentation includes component overview on the last pages.



Script	WeaponSettingSO
Weapon Name	AR15
Weapon Type	SMG
Settings of objects	
Muzzle Flash Particles F	MuzzleFire riffle (Particle System)
Shot SFX	scar shot
Reloading SFX	scar reload
Empty SFX	empty
Shell	Riffle shell
Shells Pool Size	10
Shell Ejecting Force	0.01
Weapon stats	
Damage Minimum	10
Damage Maximum	40
Rigid Body Hit Force	10
Fire Rate	0.12
Scope settings	
Can Use Scope	☐
Scope FOV	0
Scope Sensitivity X	0
Scope Sensitivity Y	0
Ballistic settings	
Bullet Initial Velocity	700
Air Resistance Force	0.1
Projectile	Riffle bullet
Projectile Pool Size	40

7. Go to the Items creator tab and take a look a the it fields.



Here you can create pickups for weapons. Select the desired weapon in the hierarchy and attach the necessary components to the fields below.

Object which will be used for weapon pickup

Weapon object: None (Game Object)

Weapon name. Must be the same as in the weapon settings

Weapon ammo count: 0

Object which will be used for weapon ammo pickup (magazine etc)

Weapon ammo obje: None (Game Object)

Ammo name. Must be the same as in the weapon settings. For example, if it is for the '

Ammo count: 0

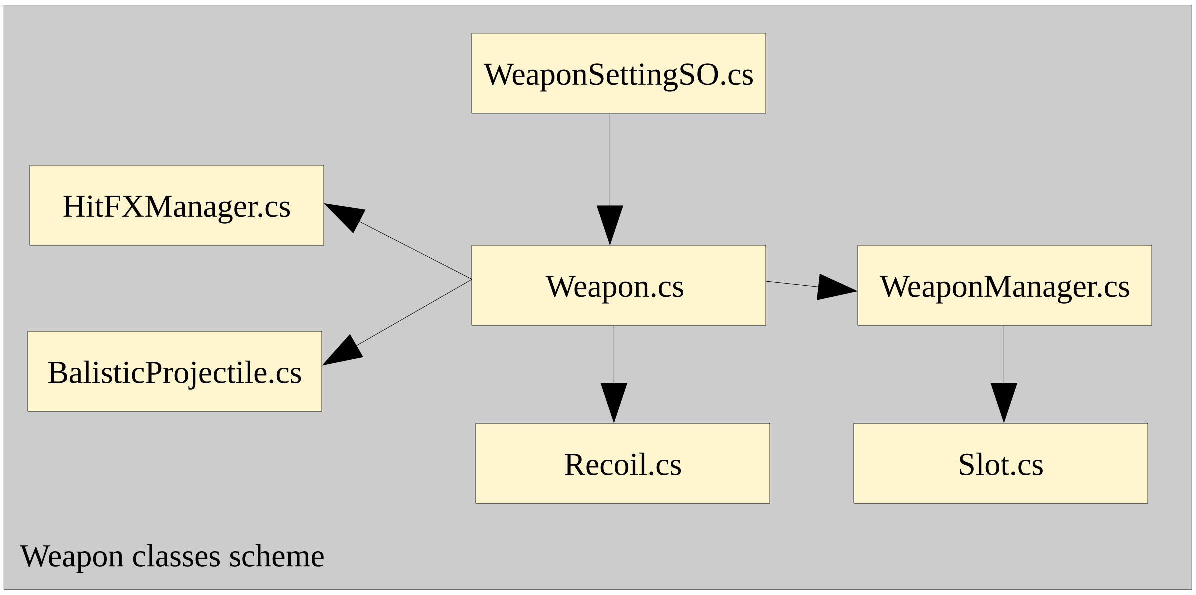
Apply weapon models (without arms) to the Weapon object field. Apply weapon magazine to Weapon ammo object field. Set weapon name as in the Weapon component and add number of ammo to each ammo count field. Press buttons to create pickup objects. They will be created in scene view. Edit colliders of the objects, and save them to prefabs.

8. If you followed guide right you can press play and test your weapon. If something goes wrong don’t worry, just look at the editor console to find out what components may be missed or take a look at the FAQ. If you have questions contact me at email : darktreedevelopment@gmail.com.

How it is work.

Short components overview

### ---	Weapon classes	---



Main class of the Weapon object is Weapon.cs. Weapon.cs grab data from WeaponSettingSO.cs scriptable object and provides interaction, shooting, reloading, animation and other behaviors.

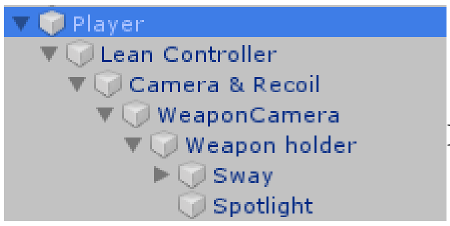
Weapon push BalisticProjectile object and get hit information returned from it's class. Weapon class send hit information to the HitFXManager.cs. Hit manager controls different effects something like bullet hit particles (for different surfaces type), decals, and hit sound effects, and play them if it is needed.

WeaponManager holds all Weapon objects inside player hierarchy and turn them on/off.

Slot made to store picked weapons just like inventory slots. Slots are controlled by WeaponManager.

--- Player Hierarchy ---

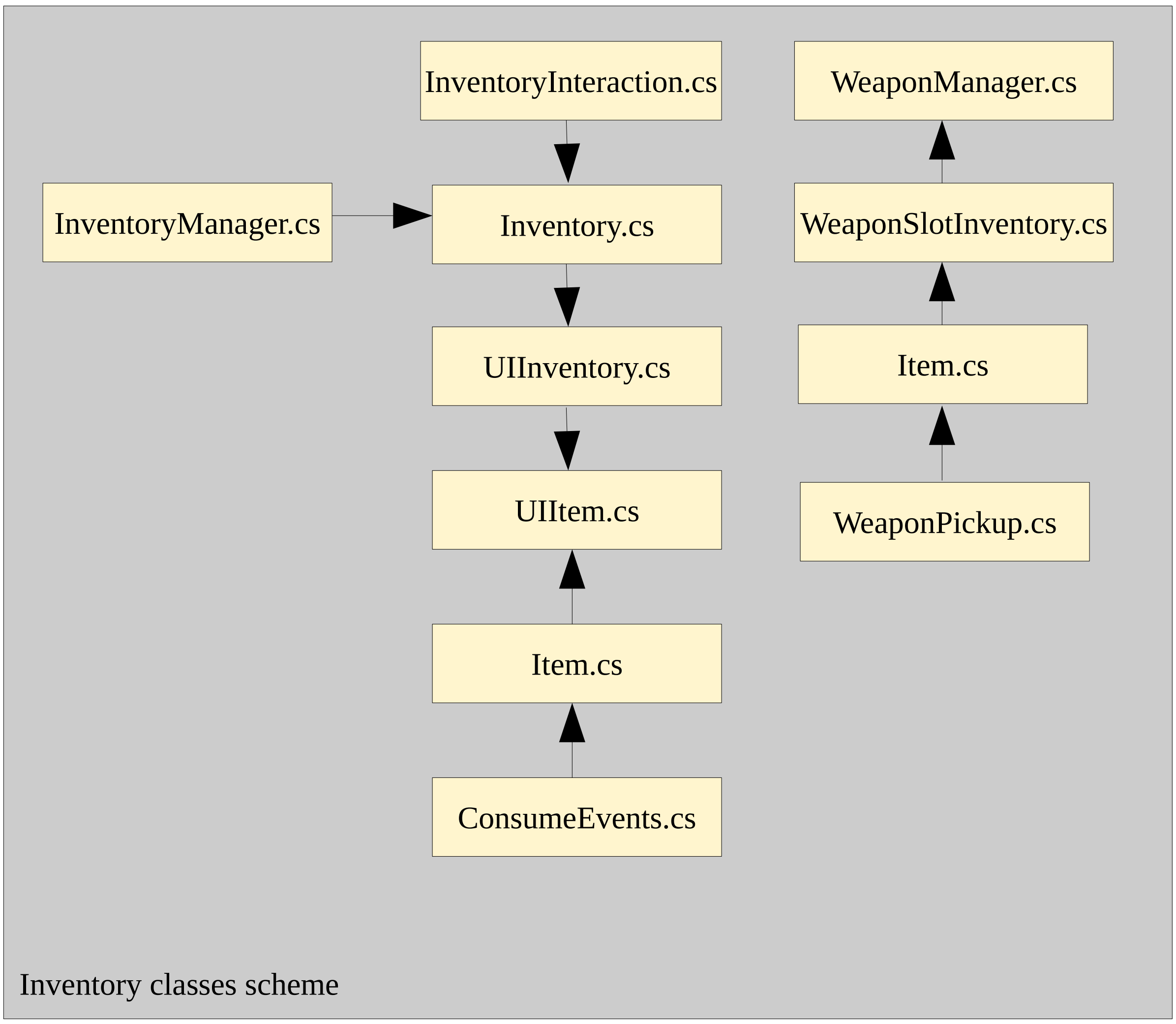
First, look at the hierarchy screenshot bellow.



As you can see, the player object contains from other objects. Each component in the hierarchy plays own role.

The Player hold FPSController.cs, PlayerStats.cs, GetCollisionTag.cs components.
Lean Controller hold Lean component.
Camera & Recoil – Main camera and Recoil script.
WeaponCamera – Camera which render only Weapon layer objects (separated from main rendering)
WeaponHolder – hold Animator component with Walk, Run, Hide/Unhide weapon animations and states.

--- Inventory ---



Inventory.cs is main wrap around inventory components. This class provides set of methods to remove, pickup, store, use, check items.

UIInventory create and hold all UIItem components. **UIItem** is class for drawing and interaction between player and item. UIItem also contains reference to Item.

Item store information about item and have events to decide what it should to do in case of event call. For example, we have **ConsumeEvents** class which add some value to player stats (satiety, thirst, health). ConsumeEvents create listener to an Item component, so when we use Item we also call listener and it method.

InventoryManager controls cases when we can open inventory and when we not.

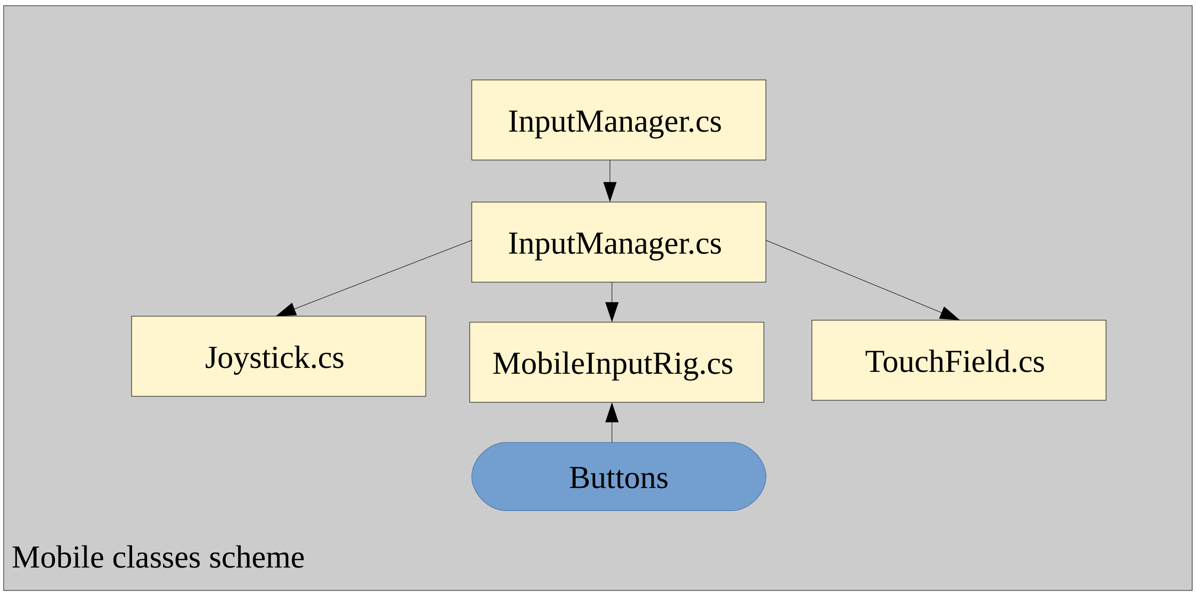
InventoryInteraction is bridge for interaction panel gameobject which being shown when we click on the item we choose. After use button click it will get item, and will use it with inventory methods.

Weapon slots are in fact not part of inventory code. It's separated code which draw only picked weapon object and work with WeaponManager. See code to get more information how it works because it's pretty simple piece.

And more about **ConsumeEvents**… ConsumeEvents create listener for an item on start. Because we can't have scene references in prefab. I decided to do not create consumable methods in inventory and made this class because it's more suitable case for my opinion.

Inventory is very simple. You can easily create your custom events if you can scripting just invoke events in method call. If you want to ask something what I've not described here, contact me at email.

--- Mobile ---



Mobile input contains of components above. Joystick transmits movement input to the FPSController and TouchField camera look. Some buttons directly use FPSController methods but the main are with MobileInputRig. MobileInputRig provide weapon interaction methods. See class to get more information about it.

FAQ

Q: I have assigned aim animation to the controller but nothing happens when I press aim key.
A: Make sure that weight of aim layer is set to 1

Q: My bullets (shells) goes in wrong direction or they are to big in the game view.
A: Check your helpers. First, forward direction for each transform is Z-axis in local mode. For bullet transform, you must set direction of your transform to forward.
About shell transform. Play with shell transform position and shell object scale to get suitable result. Shells are on default layer so they will be rendered in different from main camera perspective because they have different FOV.

Q: I can’t use my consumable item.
A: Consumable item must have ConsumeEvents component to automatically create use event on start!

Q: I use 2019 Unity version and can enter play mode due errors in the console.
A: See DarkTree FPS thread on the Unity Forum. I have solution for that

Components overview

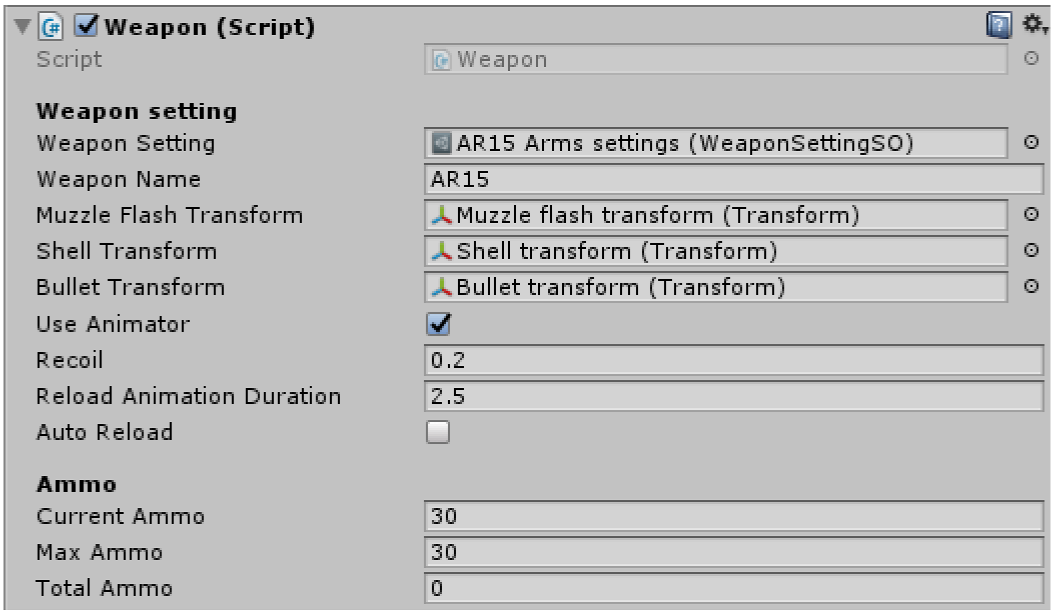
All the FPS System classes are in the “DarkTreeFPS” namespace.

There is the complete components overview with the main variables, methods, classes description.

Warning: Editor classes are not included in the components overview because it’s not a part of gameplay core of the system. If you have questions and suggestions about editor part of the system contact me at email.

1. Weapon.cs

Main class for each weapon. It holds private transforms and utility settings and it’s own WeaponSettingSO scriptable object. The main shooting logic and weapon behaviors are in the Weapon script;



Weapon (Script)

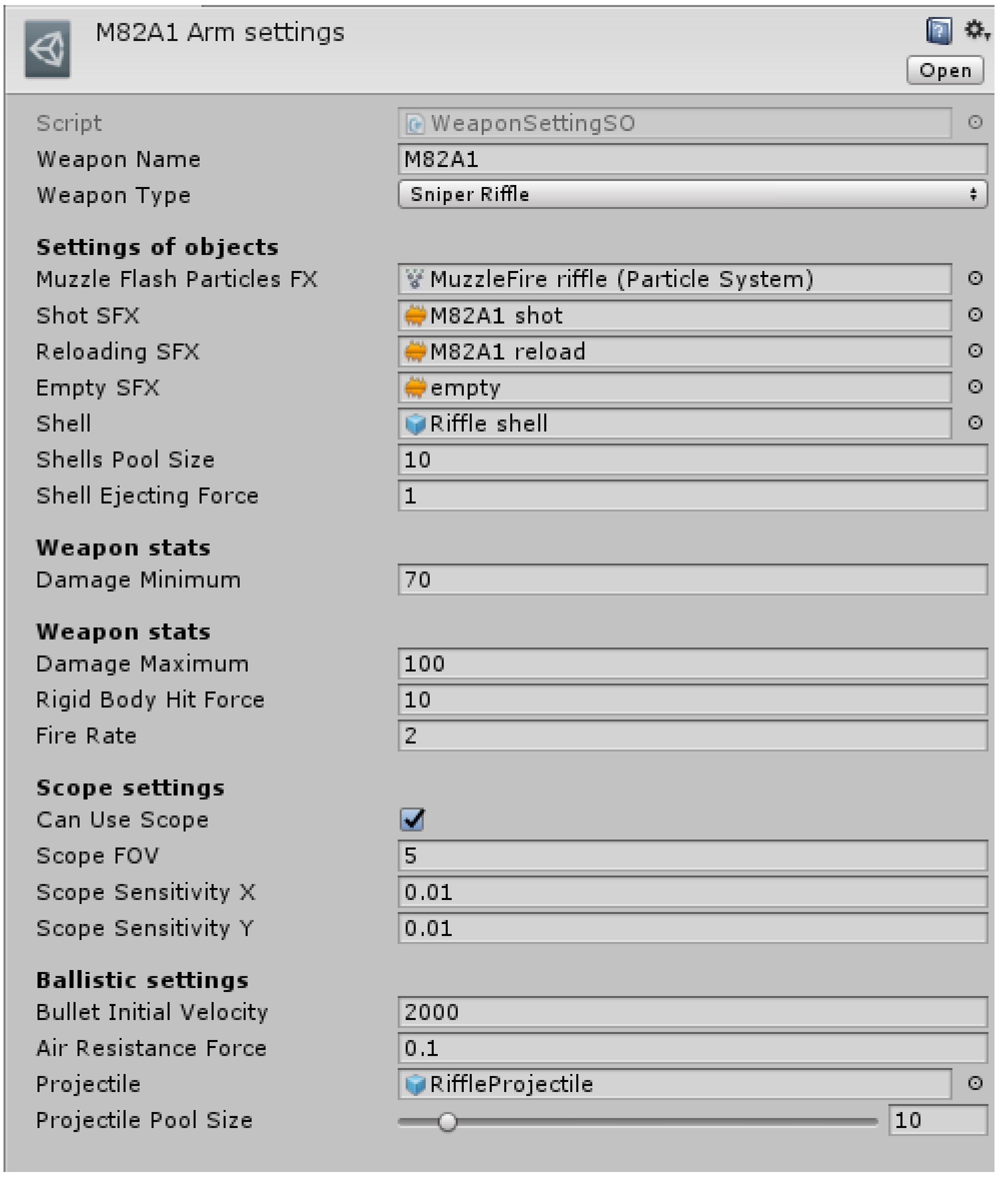
Script	Weapon
Weapon setting	
Weapon Setting	AR15 Arms settings (WeaponSettingSO)
Weapon Name	AR15
Muzzle Flash Transform	Muzzle flash transform (Transform)
Shell Transform	Shell transform (Transform)
Bullet Transform	Bullet transform (Transform)
Use Animator	✓
Recoil	0.2
Reload Animation Duration	2.5
Auto Reload	☐
Ammo	
Current Ammo	30
Max Ammo	30
Total Ammo	0

Weapon setting	
Weapon Setting	Weapon setting scriptable object that contains all the weapon parameters.
Weapon Name	Field used by pickup system as the key of object to enable.
Muzzle Flash Transform	Empty gameobject used to represent transform of muzzle flash effect. Must be on the end of weapon’s muzzle.
Shell Transform	Empty gameobject used to represent transform of shell ejecting point.
Bullet Transform	Empty gameobject used to represent transform of bullet spawning point. Must be in the middle of weapon’s muzzle. Think about where is the bullet start in the real weapon.
Use Animator	Should weapon use animator?

Recoil	Fake recoil applied to the camera at the shoot moment.
Reload Animation Duration	Duration of your weapon's reloading animation. Used to sync reloading states between each other.
Auto Reload	Should weapon automatically reload after out of ammo?
Ammo	
Current Ammo	An ammo in weapon magazine.
Max Ammo	Maximum amount of an ammo in weapon magazine;
Total Ammo	Total ammo amount available for reloading.

2. WeaponSettingSO

Class that contains settings for each individual weapon.



M82A1 Arm settings

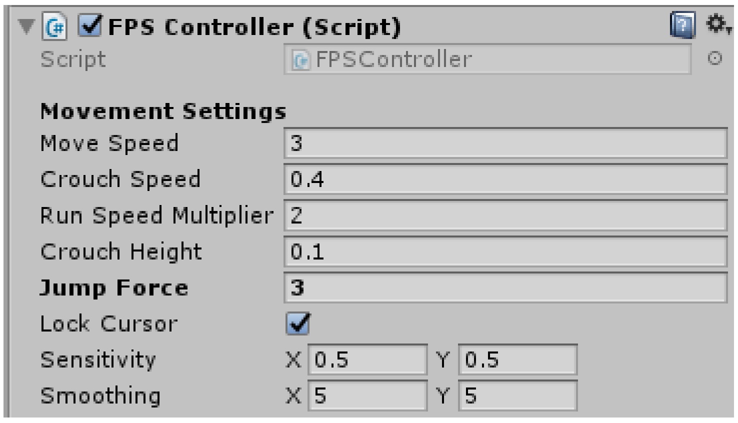
Open

Script	WeaponSettingSO
Weapon Name	M82A1
Weapon Type	Sniper Riffle
Settings of objects	
Muzzle Flash Particles FX	MuzzleFire riffle (Particle System)
Shot SFX	M82A1 shot
Reloading SFX	M82A1 reload
Empty SFX	empty
Shell	Riffle shell
Shells Pool Size	10
Shell Ejecting Force	1
Weapon stats	
Damage Minimum	70
Weapon stats	
Damage Maximum	100
Rigid Body Hit Force	10
Fire Rate	2
Scope settings	
Can Use Scope	✓
Scope FOV	5
Scope Sensitivity X	0.01
Scope Sensitivity Y	0.01
Ballistic settings	
Bullet Initial Velocity	2000
Air Resistance Force	0.1
Projectile	RiffleProjectile
Projectile Pool Size	10

Weapon Name	Weapon name
Weapon Type	Weapon type used to trigger some specific events for

	different type of weapons.
Settings of objects	
Muzzle Flash Paricles FX	Muzzle flash particle system
Shot SFX	Shot audioclip
Reloading SFX	Reloading audioclip
Empty SFX	Empty sound audioclip
Shell	Shell 3d model
Shells Pool Size	Maximum num of shell objects will be spawned on start
Shell Ejecting Force	Force which will be applied to the shell on shot.
Weapon stats	
Damage Minimum	Minimal amount of damage in damage range
Damage Maximum	Maximal amount of damage in damage range
Rigid Body Hit Force	Force that will be applied to rigidbody on bullet hit
Fire Rate	Shots per second (1=1sPs, 0.5=2sPs, 3=0.33sPs etc)
Scope settings	
Can Use Scope	Can aim with sniper scope
Scope FOV	FOV which will be applied on aim
Scope Sensitivity X	Modified mouse look sensitivity for comfortable aim
Scope Sensitivity Y	Modified mouse look sensitivity for comfortable aim
Ballistic Settings	
Bullet Initial Velocity	Force that will be applied to bullet rigidbody on shot
Air Resistance Force	Force that slow down bullet in opposite direction
Projectile	Projectile object with applied BalisticProjectile script
Projectile Pool Size	Same as the name

3. FPS Controller


FPS Controller (Script)
Script FPSController

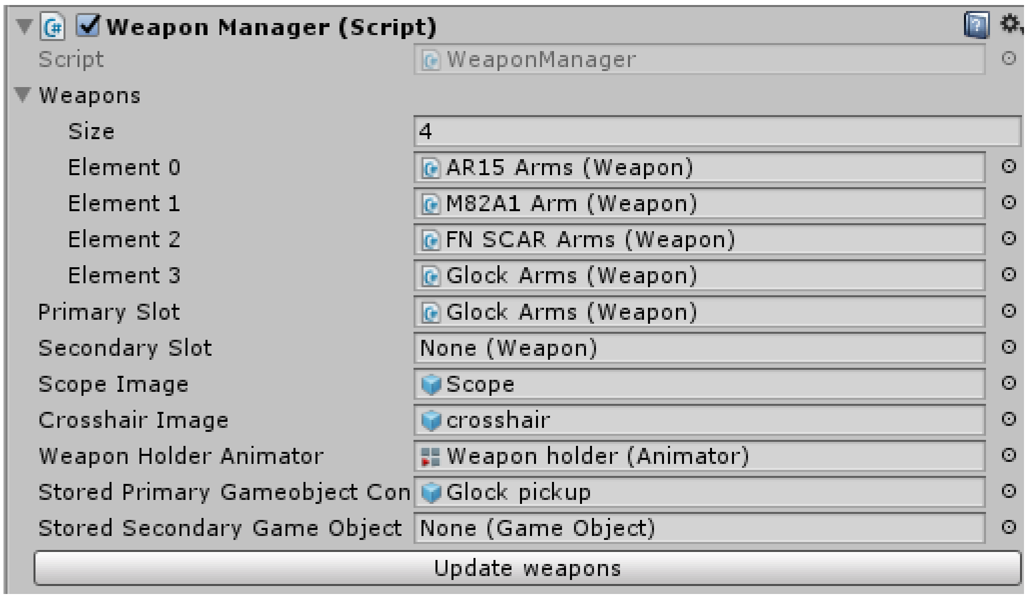
Movement Settings
Move Speed 3
Crouch Speed 0.4
Run Speed Multiplier 2
Crouch Height 0.1
Jump Force 3
Lock Cursor ✓
Sensitivity X 0.5 Y 0.5
Smoothing X 5 Y 5

Movement Settings	
Move Speed	Character's walk speed in units per second (uPs next)

Crouch Speed	Character’s crouch movement speed in uPs
Run Speed Multiplier	Multiply Move Speed on own number
Crouch Height	Height of the Character when crouch
Jump Force	Force applied to Character when jump
Lock Cursor	Lock and hide cursor on play
Sensitivity (Vector2)	Mouse sensitivity for each individual axis.
Smoothing (Vector2)	Mouse smoothing for each individual axis.

4. WeaponManager

(WIP | may be rewritten in the future updates for better functionality)



Weapon Manager (Script)

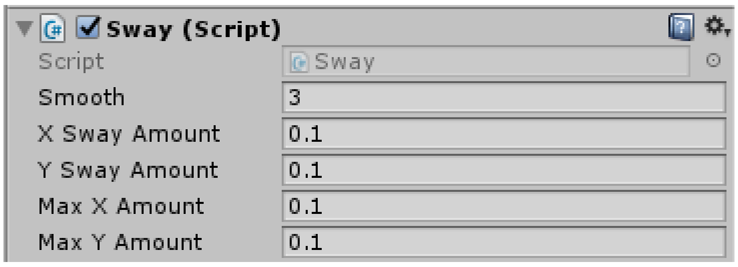
Script	WeaponManager
Weapons	
Size	4
Element 0	AR15 Arms (Weapon)
Element 1	M82A1 Arm (Weapon)
Element 2	FN SCAR Arms (Weapon)
Element 3	Glock Arms (Weapon)
Primary Slot	Glock Arms (Weapon)
Secondary Slot	None (Weapon)
Scope Image	Scope
Crosshair Image	crosshair
Weapon Holder Animator	Weapon holder (Animator)
Stored Primary Gameobject Con	Glock pickup
Stored Secondary Game Object	None (Game Object)

Update weapons

Weapons [array type of (Weapon)]	Holds all the weapon objects available in the scene. One of the most important fields of the system. All pickup actions uses the array to find suitable item to enable or add ammo.
Primary Slot	The virtual slot that keep picked weapon
Secondary Slot	The virtual slot that keep picked weapon
Scope Image	Canvas image used for scope drawing.
Crosshair Image	Canvas image used for crosshair drawing
Weapon Holder Animator	An animator put on the ‘WeaponHolder’ gameobject that contains animations for weapons transition
Stored Primary Gameobject	The gameobject picked by player. When player drops the item object drops as the instance of the weapon used by player
Stored Secondary Gameobject	The same as description above
Update weapons [Button] – press to collect all the weapons available in the scene to Weapons array.	

5. Sway

Class used for weapon sway effect.



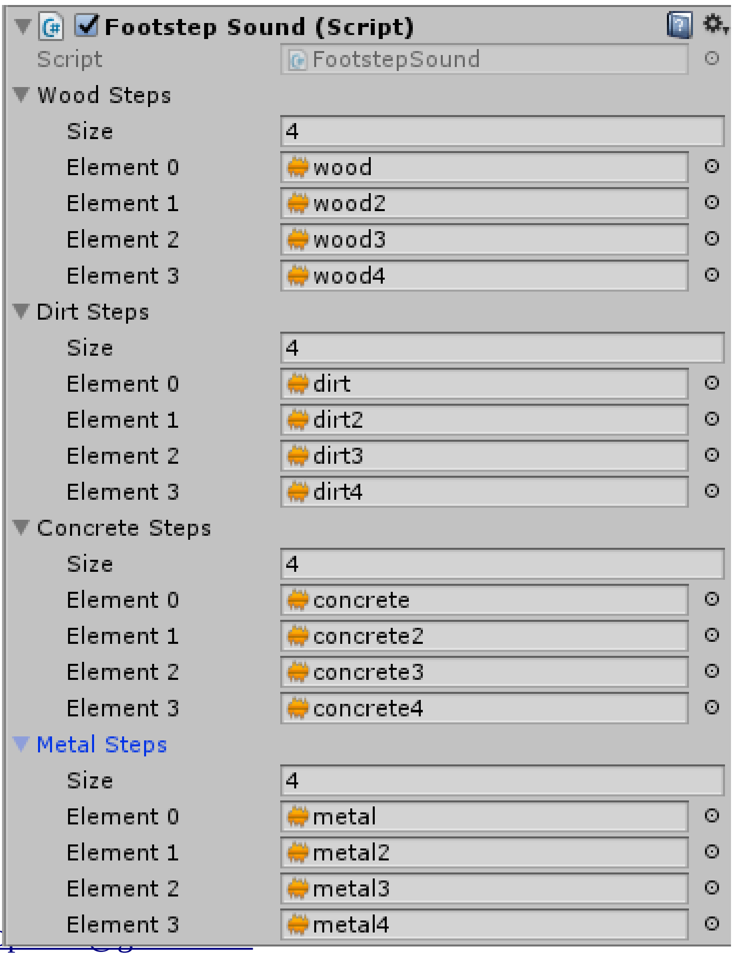
Sway (Script)

Script	Sway
Smooth	3
X Sway Amount	0.1
Y Sway Amount	0.1
Max X Amount	0.1
Max Y Amount	0.1

Smooth	Smoothness of sway movement
X Sway Amount	Sway amount for specified axis
Y Sway Amount	Sway amount for specified axis
Max X Amount	Max X Sway amount for specified axis
Max Y Amount	Max Y Sway amount for specified axis

5. FootstepSound

The script contains four arrays with step sound effects. You can modify script to add new sound effects. (WIP | may be rewritten in future updates)

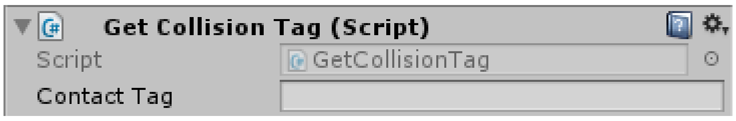


Footstep Sound (Script)

Script	FootstepSound
Wood Steps	
Size	4
Element 0	wood
Element 1	wood2
Element 2	wood3
Element 3	wood4
Dirt Steps	
Size	4
Element 0	dirt
Element 1	dirt2
Element 2	dirt3
Element 3	dirt4
Concrete Steps	
Size	4
Element 0	concrete
Element 1	concrete2
Element 2	concrete3
Element 3	concrete4
Metal Steps	
Size	4
Element 0	metal
Element 1	metal2
Element 2	metal3
Element 3	metal4

6. GetCollisionTag

The script receives collision data from the controller rigidbody and transmits it to the FootstepSound component.

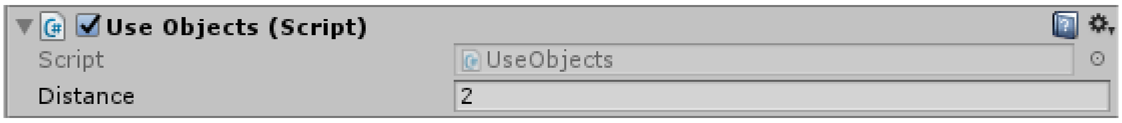


Get Collision Tag (Script)

Script	GetCollisionTag
Contact Tag	

7. UseObjects

Class provides interaction between player and items. (WIP | may be rewritten in future updates)

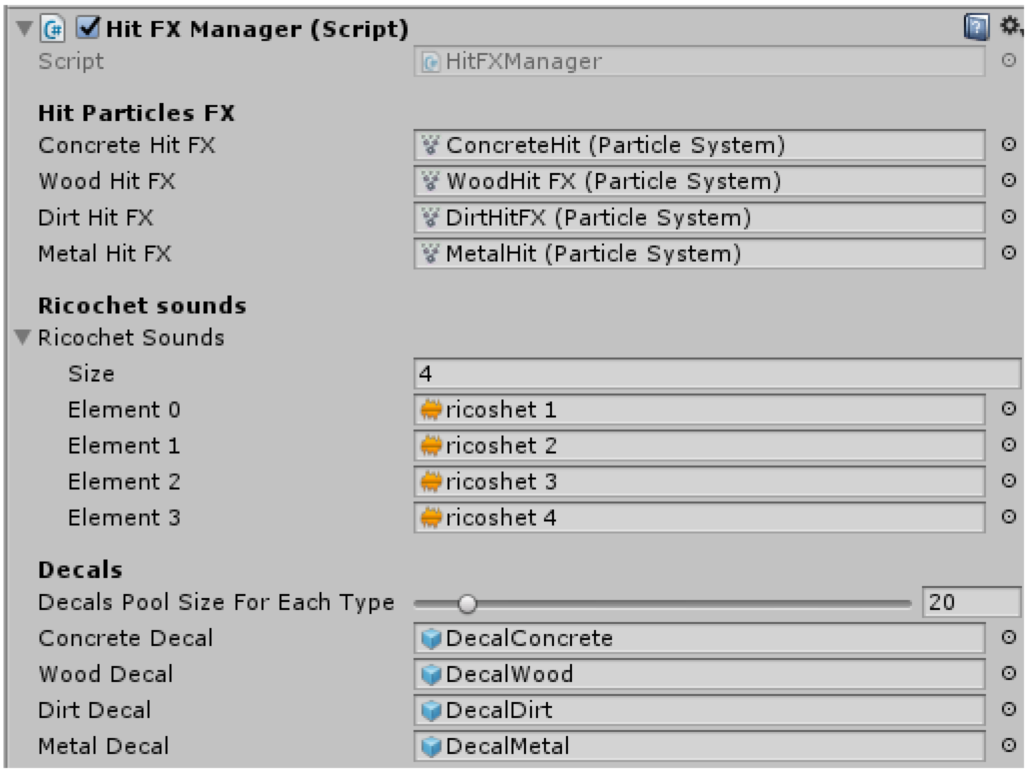


Use Objects (Script)

Script	UseObjects
Distance	2

8. Hit FX Manager

If you want to customize your effects you should to take a look at the Hit FX Manager. It creates effects instances on Start() and the Weapon script took them to work with.



Hit FX Manager (Script)

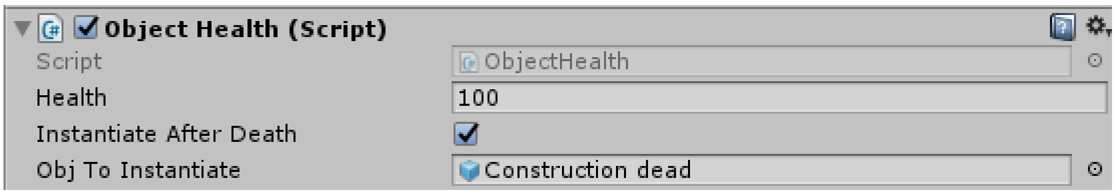
Script	HitFXManager
Hit Particles FX	
Concrete Hit FX	ConcreteHit (Particle System)
Wood Hit FX	WoodHit FX (Particle System)
Dirt Hit FX	DirtHitFX (Particle System)
Metal Hit FX	MetalHit (Particle System)
Ricochet sounds	
Ricochet Sounds	
Size	4
Element 0	ricoshet 1
Element 1	ricoshet 2
Element 2	ricoshet 3
Element 3	ricoshet 4
Decals	
Decals Pool Size For Each Type	20
Concrete Decal	DecalConcrete
Wood Decal	DecalWood
Dirt Decal	DecalDirt
Metal Decal	DecalMetal

Hit Particles FX	
Concrete Hit FX	Particle system used for bullet hit FX

Wood Hit FX	Particle system used for bullet hit FX
Dirt Hit FX	Particle system used for bullet hit FX
Metal Hit FX	Particle system used for bullet hit FX
Ricochet sounds	
Ricochet Sounds [array]	Array of an audioclips to play a different impact sounds on bullet hit
Decals	
Decals Pool Size For Each Type	How many decal instances should be created on start for each type of the hit surface?
Concrete Decal	Decal for concrete hit FX
Wood Decal	Decal for wood hit FX
Dirt Decal	Decal for dirt hit FX
Metal Decal	Decal for metal hit FX

9. ObjectHealth

You can destroy an objects using ObjectHealth script.


Object Health (Script)

Script	ObjectHealth
Health	100
Instantiate After Death	✓
Obj To Instantiate	Construction dead

Health	Object health
Instantiate After Death	If true object will instantiate 'Obj To Instantiate' gameobject after death. If false object will destroy
Obj To Instantiate	An object which will be instantiated after death