ULTIMATE HORROR SYSTEM

SCRIPTS:

INTERACTIVE OBJECT

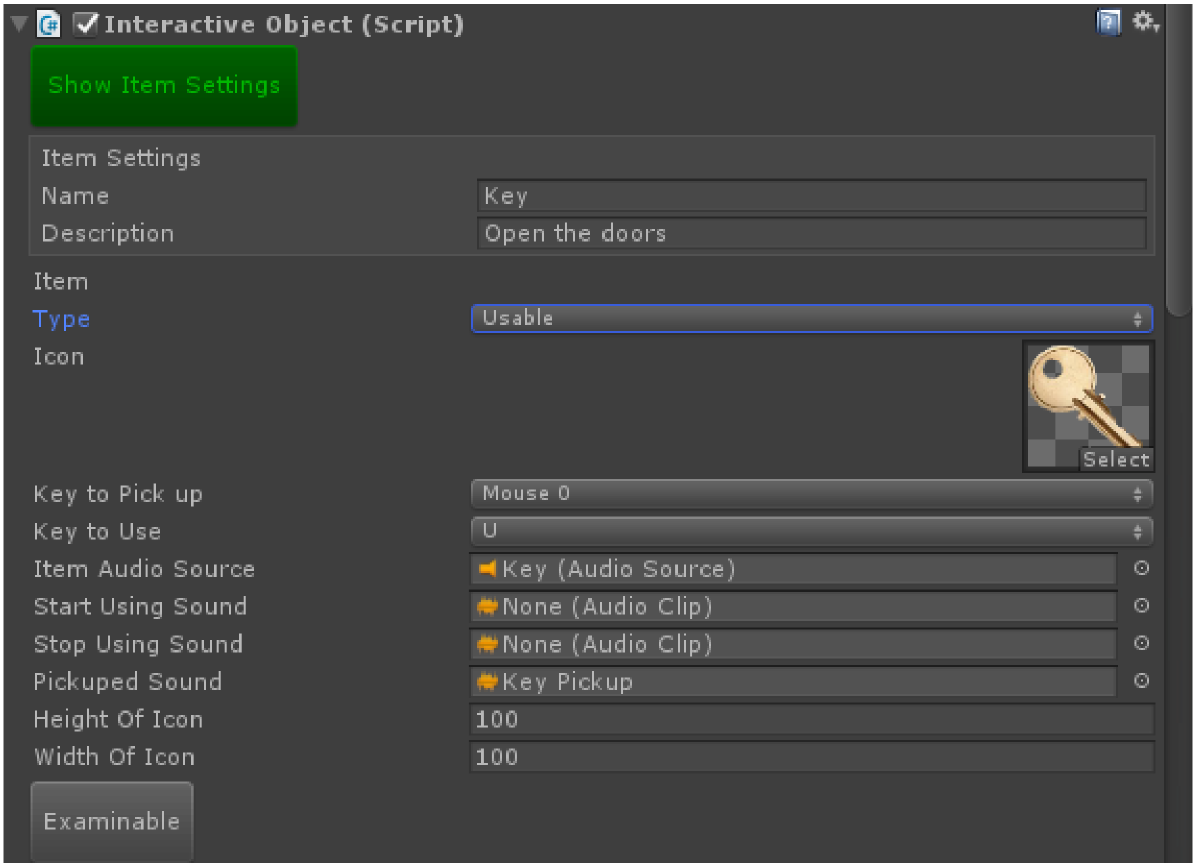
  - ○ **Name (string):** The name of item **(obligatory field)** it's used to save the item in the inventory, and used if you want to show item name on UI .

- **Show Item Settings (GUI Button): Do you want to show the name and description of the item when the player looks it?.**

  - **Description (string):** The description of the item.

- **Type (enum) :** The type of the item:

  - **Usable (enum Value):** You can pickup it, store it in the inventory and use it. Example: A key, a flashlight, a camera.



☑ Interactive Object (Script)

Item Settings

Name: Key

Description: Open the doors

Item

Type: Usable

Icon

Key to Pick up: Mouse 0

Key to Use: U

Item Audio Source: Key (Audio Source)

Start Using Sound: None (Audio Clip)

Stop Using Sound: None (Audio Clip)

Pickuped Sound: Key Pickup

Height Of Icon: 100

Width Of Icon: 100

- **Icon (Sprite):** The icon showed in the inventory when you pickup it.
- **Key To Pickup (KeyCode):** The key you have to press to pickup it.
- **Key To Use (KeyCode):** The key you have to press to use it.
- **Item Audio Source (AudioSource):** The audio source that will play the sounds of this item.
- **Start Using Sound (AudioClip):** The sound played when the player starts to use it.
- **Stop Using Sound (AudioClip):** The sound played when the player stops of use it.
- **Pickuped Sound (AudioClip):** The sound played when the player picks up this item.
- **Height Of Icon (float):** The height value assigned to the Rect Transform of the icon in the inventory
- **Width Of Icon (float):** The width value assigned to the Rect Transform of the icon in the inventory

- **Animations**

Legacy animations for the item for the differents states. (Idle, Walk, Run,Shoot if it is a weapon, etc)

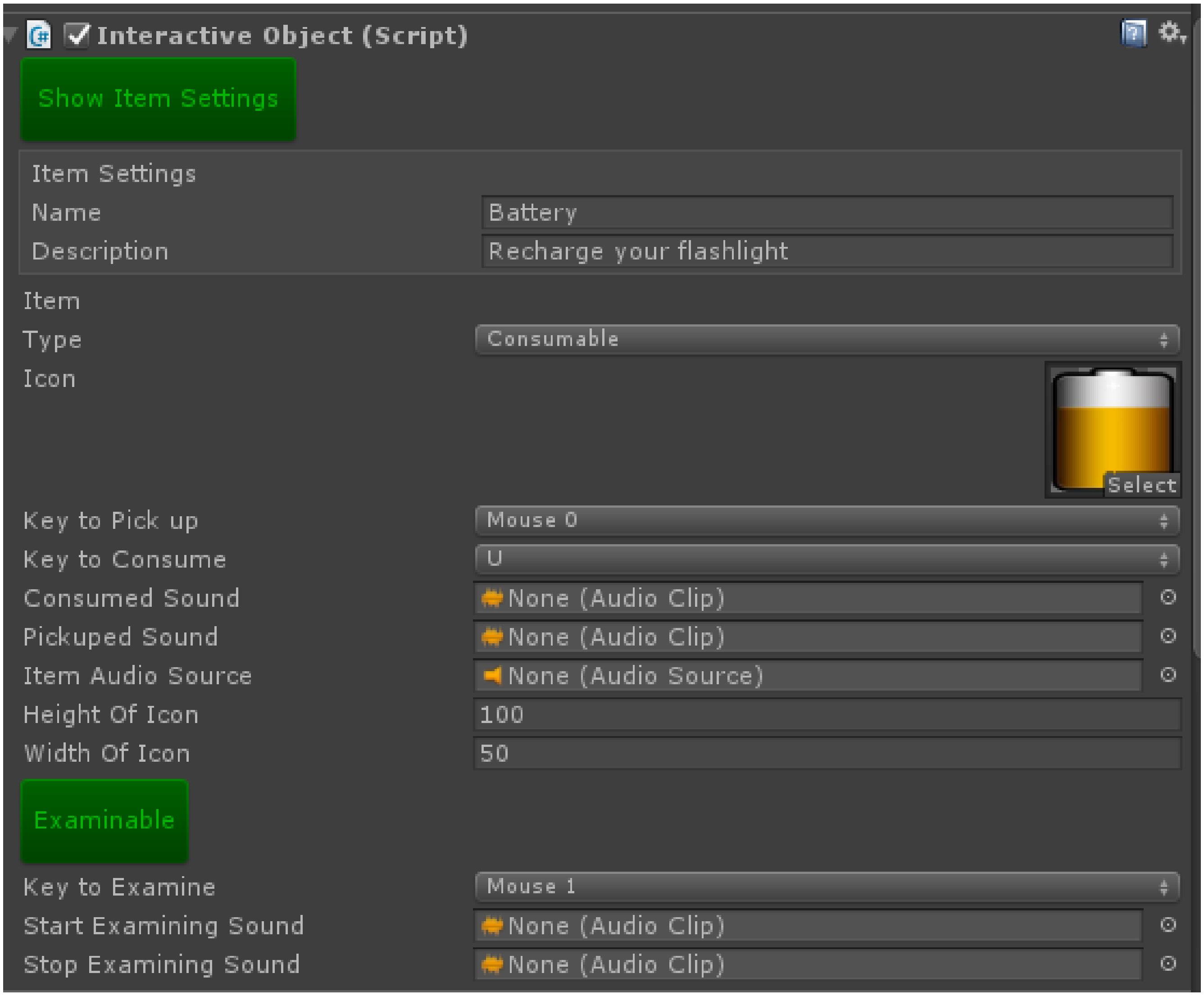
- **Examinable (GUI Button):** This item can be examined?.

  1. **Key To Examine (KeyCode):** The key you have to press to examine it.

  2. **Start Examining Sound (AudioClip):** The sound played when the player starts to examine it.

  3. **Stop Examining Sound (AudioClip):** The sound played when the player stops of examine it.

Consumable (enum Value): You can pickup it, store it in the inventory and consume it (only one time).
Example: A battery, a bullet, a health potion.



Item Settings

Name	Battery
Description	Recharge your flashlight

Item

Type	Consumable
Icon	
Key to Pick up	Mouse 0
Key to Consume	U
Consumed Sound	None (Audio Clip)
Pickuped Sound	None (Audio Clip)
Item Audio Source	None (Audio Source)
Height Of Icon	100
Width Of Icon	50
Key to Examine	Mouse 1
Start Examining Sound	None (Audio Clip)
Stop Examining Sound	None (Audio Clip)

- **Icon (Sprite):** The icon showed in the inventory when you pickup it.
- **Key To Pickup (KeyCode):** The key you have to press to pickup it.
- **Key To Consume (KeyCode):** The key you have to press to consume it.
- **Item Audio Source (AudioSource):** The audio source that will play the sounds of this item.
- **Consumed Sound (AudioClip):** The sound played when the player consumes it.
- **Pickuped Sound (AudioClip):** The sound played when the player picks up this item.

    - **Height Of Icon (float):** The height value assigned to the Rect Transform of the icon in the inventory

    - **Width Of Icon (float):** The width value assigned to the Rect Transform of the icon in the inventory

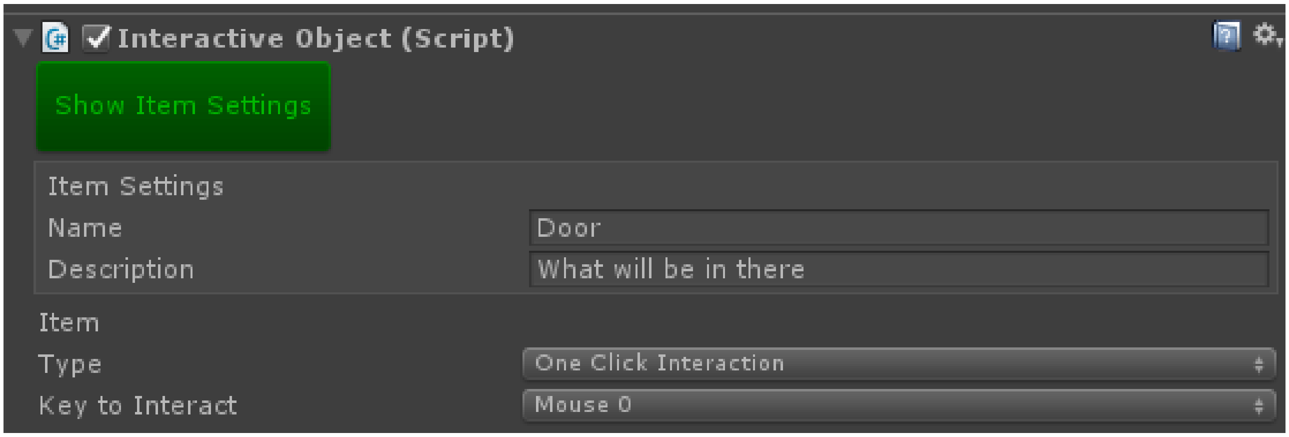
    - **Examinable (GUI Button):** This item can be examined?.

        1. **Key To Examine (KeyCode):** The key you have to press to examine it.

        2. **Start Examining Sound (AudioClip):** The sound played when the player starts to examine it.

        3. **Stop Examining Sound (AudioClip):** The sound played when the player stops of examine it.

- **One Click Interaction (enum Value):** You can click on it.
  Example: A door, a lamp.



☑ Interactive Object (Script)

Show Item Settings

Item Settings

Name: Door

Description: What will be in there

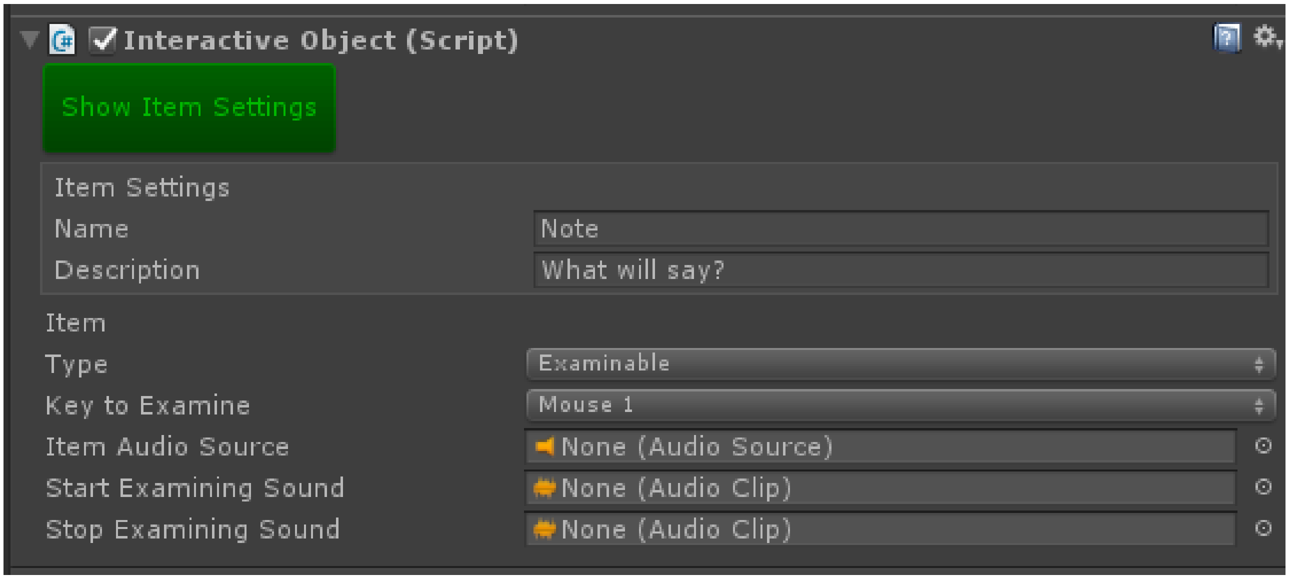
Item

Type: One Click Interaction

Key to Interact: Mouse 0

    - **Key To Consume** (**KeyCode**): The key you have to press to interact with it.

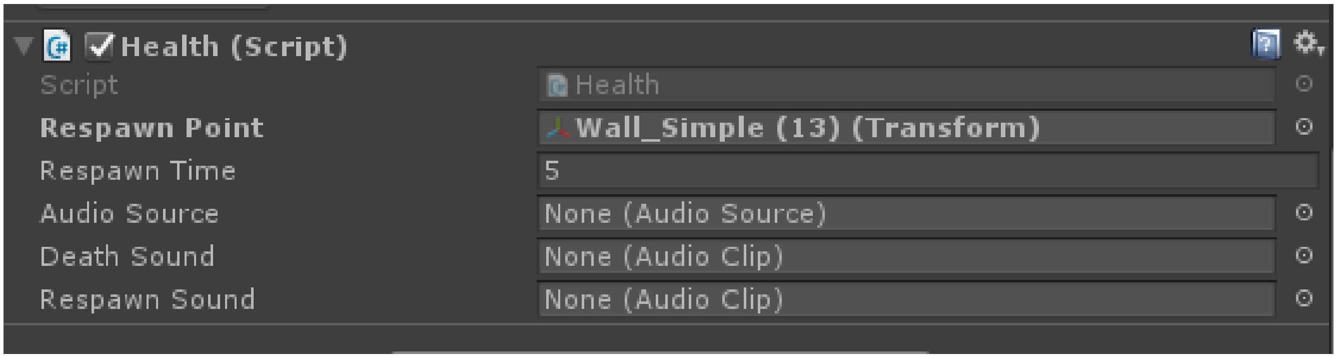
- **Examinable (enum Value):** You can only examine it.
  Example: A note, a damaged object.



Item Settings	
Name	Note
Description	What will say?
Item	
Type	Examinable
Key to Examine	Mouse 1
Item Audio Source	None (Audio Source)
Start Examining Sound	None (Audio Clip)
Stop Examining Sound	None (Audio Clip)

- **Key To Examine (KeyCode):** The key you have to press to examine it.
- **Start Examining Sound (AudioClip):** The sound played when the player starts to examine it.
- **Stop Examining Sound** (AudioClip): The sound played when the player stops of examine it.

HEALTH



Script	Health
Respawn Point	**Wall_Simple (13) (Transform)**
Respawn Time	5
Audio Source	None (Audio Source)
Death Sound	None (Audio Clip)
Respawn Sound	None (Audio Clip)

- **Respawn Point (Transform):** The position where the player will spawn after die.

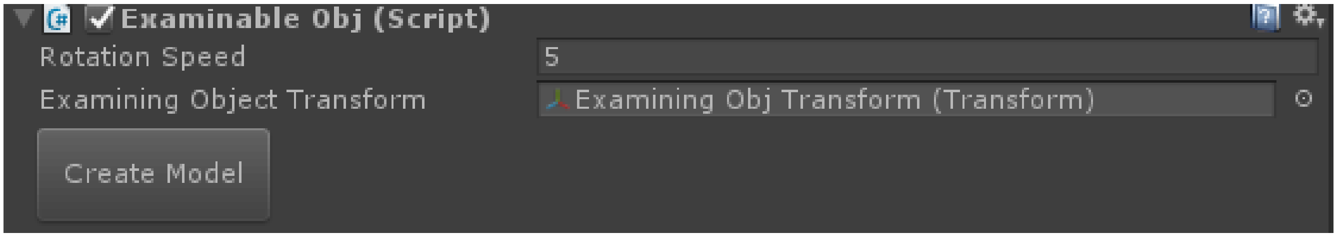
- **Respawn Time (float):** The time between the player death and respawn.

- **Audio Source (AudioSource):** The audio source that will play the sounds of this script.

- **Death Sound (AudioClip):** The sound played when the player dies.

- **Respawn Sound (AudioClip):** The sound played when the player respawns.

EXAMINABLEOBJ



Rotation Speed 5

Examining Object Transform Examining Obj Transform (Transform)

Rotation Speed (float): The speed that the object will rotate

Examining Object Transform (Transform): The position where will be the object that the player is examining. This transform should be inside of the player camera.

Create Model (GUI Button): When you press it, it will duplicate the object to which this is added and will add it to the Examining Object Transform, so when the player is examining this object, the model created previously will be activated, to have a better look of this.

PlayerData

It is a static class, (it is not added to a gameobject), it contains information about the player like the player camera, if the player has flashlight, what gameobject is using the player, what gameobject is examining the player. You can use this class to get information about the player quickly and develop your own game of horror!.

For example: there is a variable in this class called “IsExamining”, (true while the player is examining a GameObject), you could play a music while the player is examining something.

To access to a script of Easy Horror System you need add:

```
using AxlPlay;
```

Because the scripts are in that namespace.

And now:

```
void Update() {
        // the player is examining a gameobject
        if(PlayerData.IsExamining) {
                // do something
        }
}
```

UIEFFECTS

This is only to do a fade in and fade out effect in the UI.

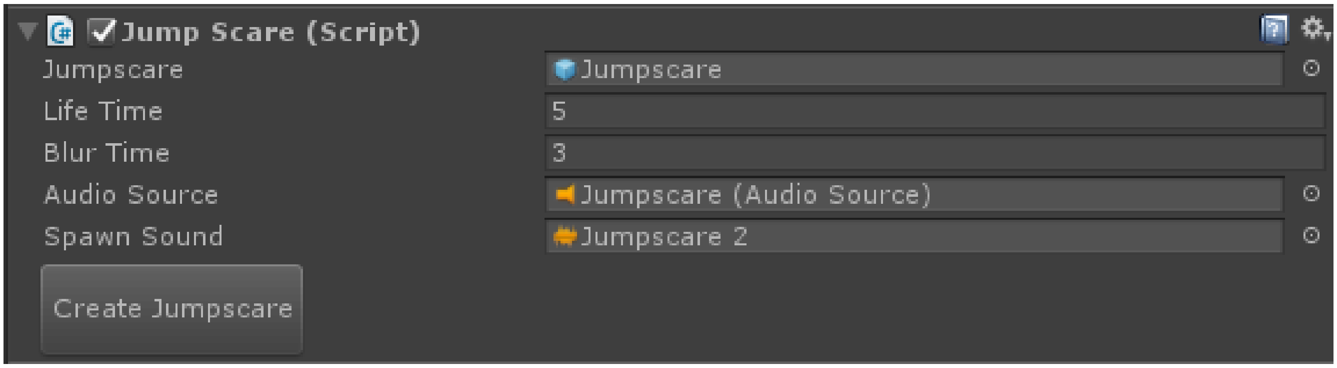
LOCK

This is a static class it is not attached to a gameobject.

It contains a enum with the **Lock Types**.

Lock Types are used by the **Door** and by the **Key**.

JUMPSCARE



☑ Jump Scare (Script)

Jumpscare: Jumpscare

Life Time: 5

Blur Time: 3

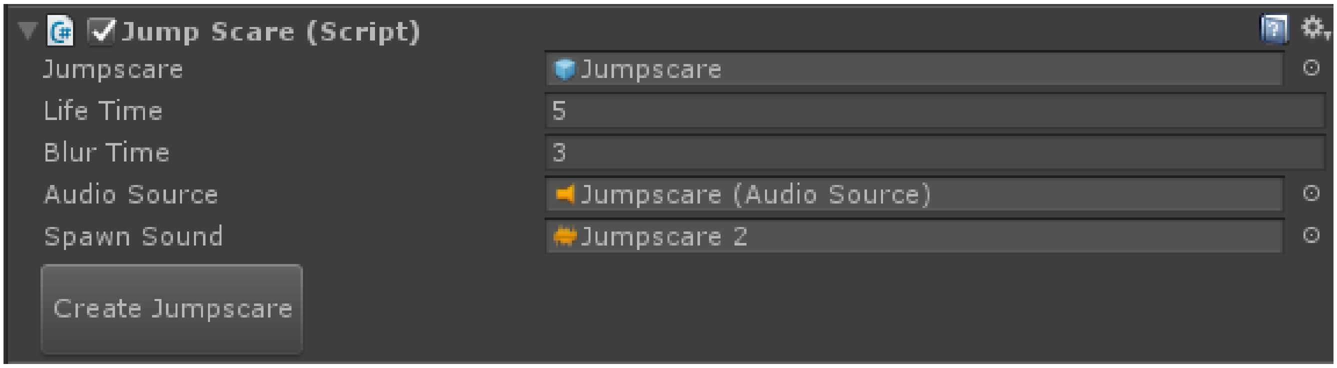
Audio Source: Jumpscare (Audio Source)

Spawn Sound: Jumpscare 2

In the scene “**Demo 1**” in the GameObject called “**Horror System**” is the example of this script.

- **Jumpscare (GameObject):** The gameobject (in the scene) that will be the jumpscare.
- **Life Time (float):** The time that the **Jumpscare** will be active.
- **Audio Source (AudioSource):** The audio source that will play the **Spawn Sound**.
- **Spawn Sound (AudioClip):** The sound played when the **Jumpscare** spawns.

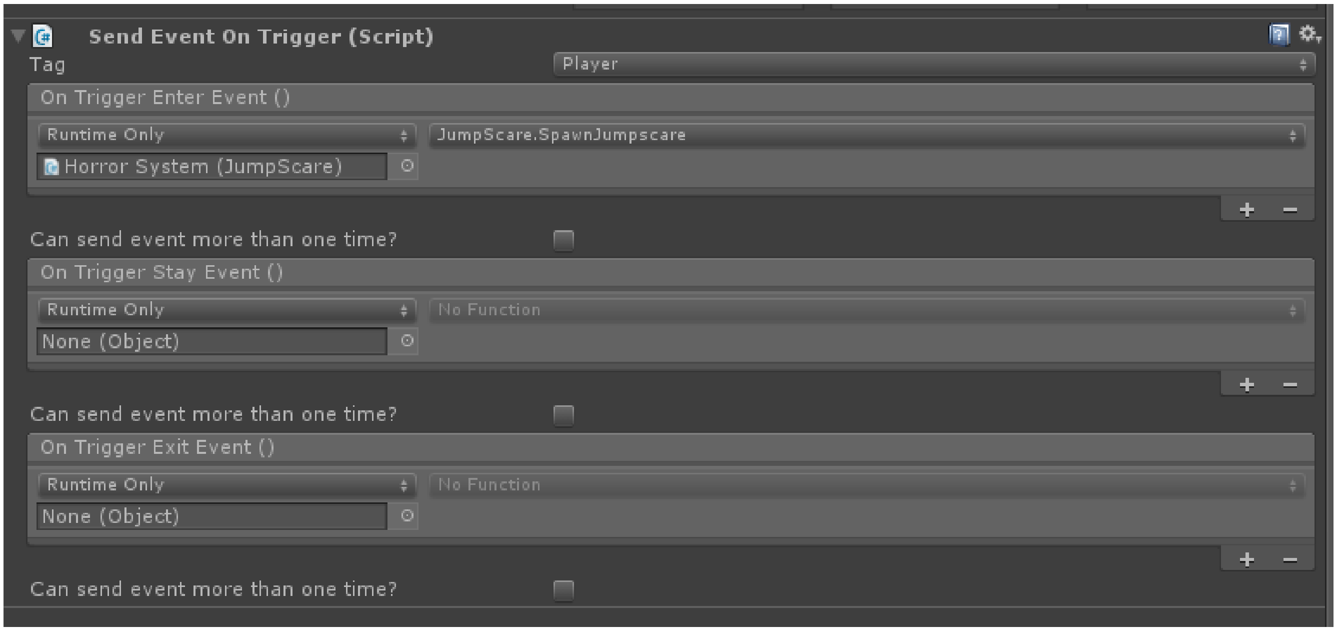
The **Jumpscare** is moved by the animator controller and it need Apply Root Motion to **Jumpscare**.



☑ Jump Scare (Script)

Jumpscare: Jumpscare
Life Time: 5
Blur Time: 3
Audio Source: Jumpscare (Audio Source)
Spawn Sound: Jumpscare 2

The **Jump Scare** script also need to be called to work. I call it with the script **SendEventOnTrigger**, that calls a method when a gameobject with certain tag enters by the trigger.



Send Event On Trigger (Script)

Tag: Player

On Trigger Enter Event ()

Runtime Only — JumpScare.SpawnJumpscare

Horror System (JumpScare)

Can send event more than one time? □

On Trigger Stay Event ()

Runtime Only — No Function

None (Object)

Can send event more than one time? □

On Trigger Exit Event ()

Runtime Only — No Function

None (Object)

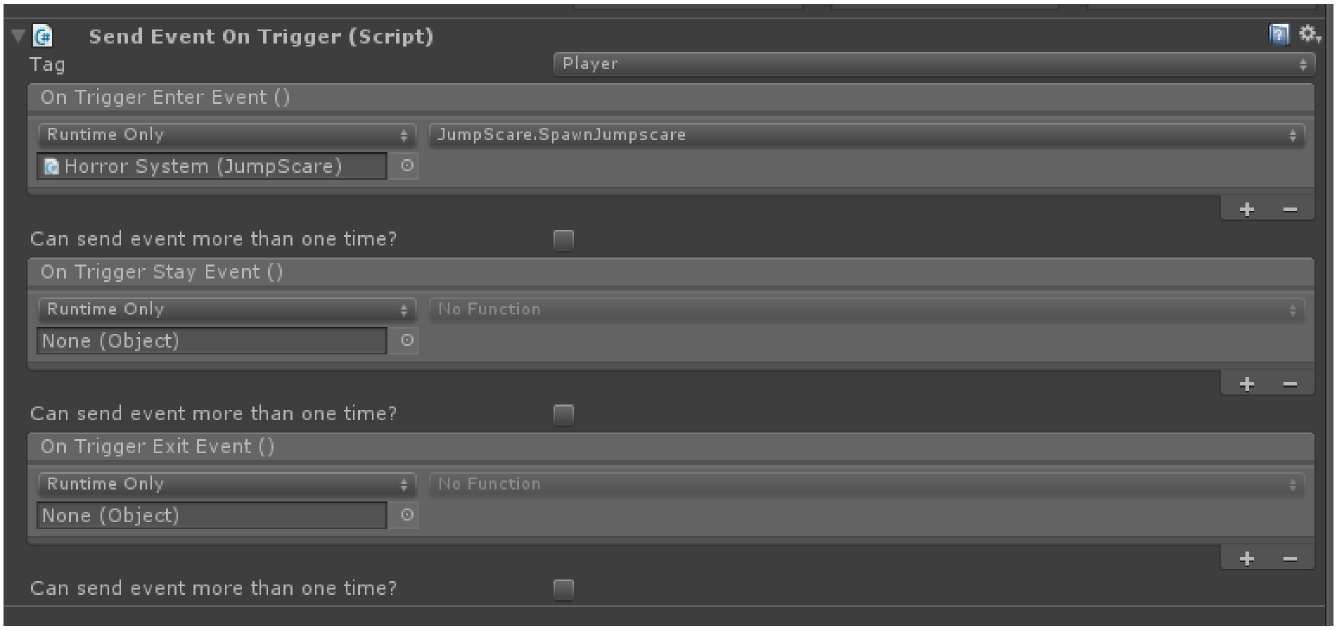
Can send event more than one time? □

SENDEVENTONTRIGGER

It will call a method when a gameobject with certain **Tag** enters by the trigger, stay in the trigger, or exit from the trigger.

- **Tag (Tag):** The tag of the object that can interact with the trigger.
- **On Trigger Enter Event (Unity Event):** The method called when a gameobject with **Tag** enters by the trigger
- **Can send event more than one time (bool):** If it is false the method will be called one time and not will be called more, though the gameobject with **Tag** enters by the trigger.
- **On Trigger Stay Event (Unity Event):** The method called when a gameobject with **Tag** stay in the trigger
- **Can send event more than one time (bool):** If it is false the method will be called one time and not will be called more, though the gameobject with **Tag** stays in the trigger.
- **On Trigger Exit Event (Unity Event):** The method called when a gameobject with **Tag** exit from the trigger
- **Can send event more than one time (bool):** If it is false the method will be called one time and not will be called more, though the gameobject with **Tag** exit from the trigger.

EXAMPLE 1:



Send Event On Trigger (Script)

Tag: Player

On Trigger Enter Event ()

Runtime Only — JumpScare.SpawnJumpscare

Horror System (JumpScare)

Can send event more than one time? □

On Trigger Stay Event ()

Runtime Only — No Function

None (Object)

Can send event more than one time? □

On Trigger Exit Event ()

Runtime Only — No Function

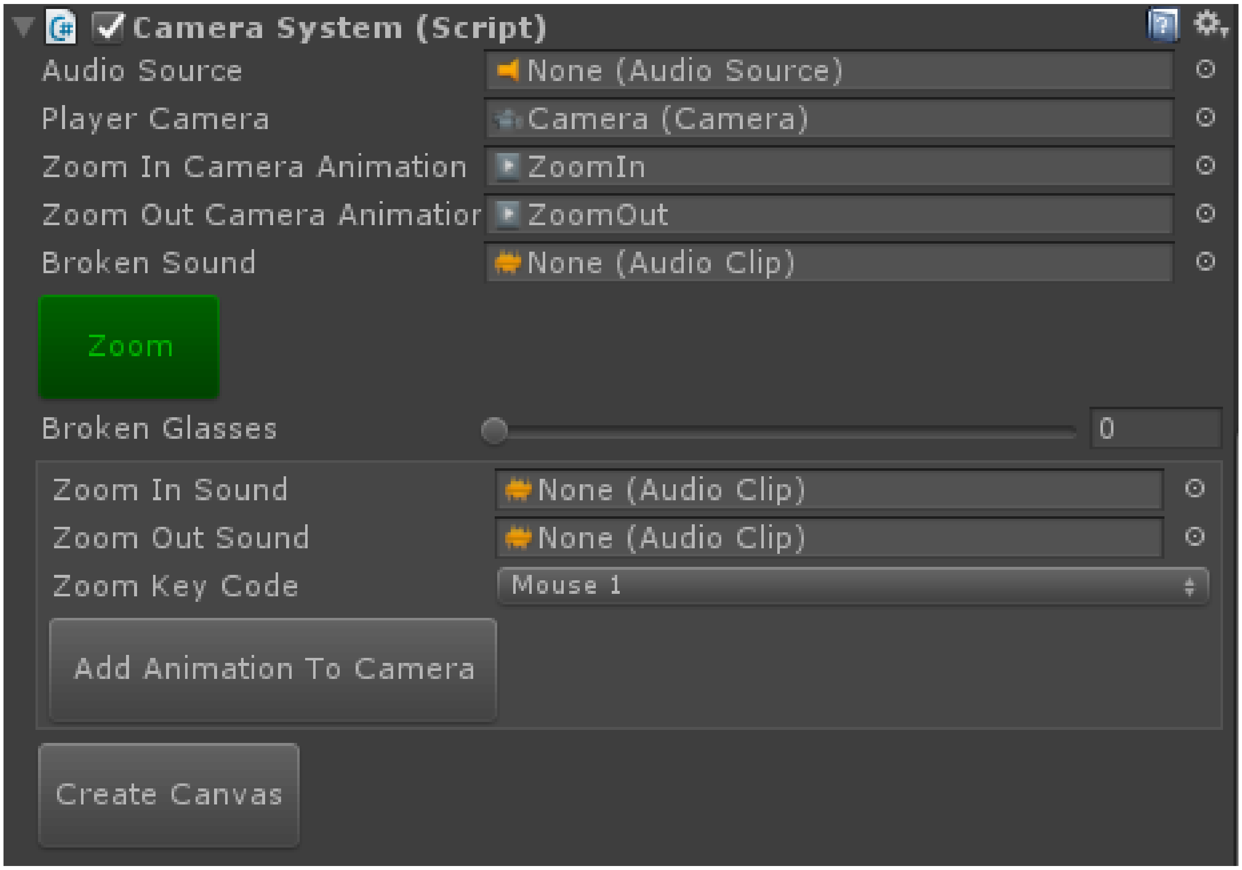
None (Object)

Can send event more than one time? □

RESULT OF THIS:

When a gameobject with the **Tag** “**Player**” enters by the trigger the method “**SpawnJumpscare**” will be called but it only will happen one time, no matter how many times the gameobject enters by the trigger.

CAMERASYSTEM



Camera System (Script)

Audio Source: None (Audio Source)

Player Camera: Camera (Camera)

Zoom In Camera Animation: ZoomIn

Zoom Out Camera Animatior: ZoomOut

Broken Sound: None (Audio Clip)

Broken Glasses: 0

Zoom In Sound: None (Audio Clip)

Zoom Out Sound: None (Audio Clip)

Zoom Key Code: Mouse 1

In the prefabs folder there is a example of a camera.

- **Audio Source (AudioSource): The audio source that will play the sounds of the camera.**

- **Player Camera (Camera): The Camera of the player.**

- **Zoom In Camera Animation (AnimationClip):** To do zoom in, the camera needs an animation that substracts the field of view. In the Resources folder there is an animation ready.

- **Zoom Out Camera Animation (AnimationClip):** To do zoom out, the camera needs an animation that adds the field of view. In the Resources folder there is an animation ready.

- **Broken Sound (AudioClip):** The sound played when the camera breaks.

- **Zoom (GUI Button):** The camera can do zoom?.

    - **Broken Glasses (Image[]):** The images of broken glasses. When the camera breaks the **Camera System** will choose a random broken glass, (if you have more than one) and display it in the canvas.

  - **Zoom In Sound (AudioClip):** The sound played when player do zoom in with the camera.

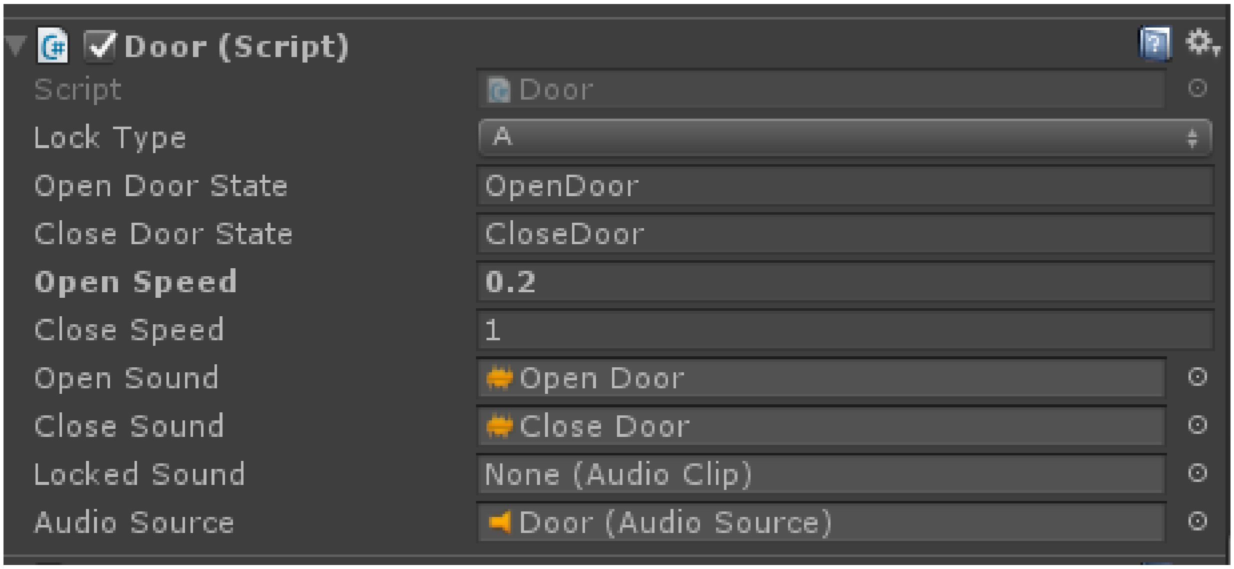
  - **Zoom Out Sound (AudioClip):** The sound played when player do zoom out with the camera.

  - **Zoom Key Code (KeyCode):** The key that the player needs to press to do zoom in or zoom out

  - **Add Animation to Camera (GUI Button)**: If you press it the Camera System will add to your camera an Animation component with the Zoom In Camera Animation and Zoom Out Camera Animation.

- **Create Canvas (GUI Button):** If you press it the **Camera System** will create a Canvas with the UI of the camera, called "**Camera Canvas**" and you can modify it.

DOOR



☑ Door (Script)

Script	Door
Lock Type	A
Open Door State	OpenDoor
Close Door State	CloseDoor
Open Speed	0.2
Close Speed	1
Open Sound	Open Door
Close Sound	Close Door
Locked Sound	None (Audio Clip)
Audio Source	Door (Audio Source)

In the prefabs folder there is a example of a camera.

- **Lock Type (enum):** This variable is also in **KeySystem**.
  Example: If the **Lock Type** is "**A**" you need a **Key** with the **Lock Type** "**A**" to open this door.
- The **Lock Type** can be "**Free**" and the door don't need a key to be open.

- The doors are open with an animation clip that rotates it. In the Animations folder there is an Open Door animation and Close Door animation.

- **Open Door State (string):** The name of the state in the animator controller that has the animation of open door.

- **Close Door State (string):** The name of the state in the animator controller that has the animation of close door.

- **Open Speed (float):** The speed at which the animation Open Door will run.

- **Close Speed (float):** The speed at which the animation Close Door will run.

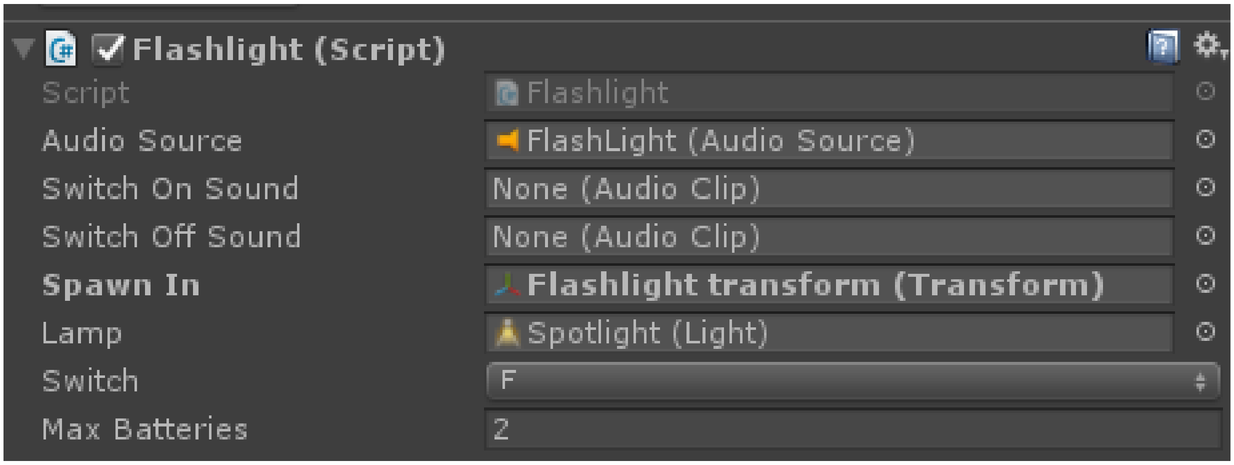
- **Open Sound (AudioClip):** The sound played when the player open the door.

- **Close Sound (AudioClip):** The sound played when the player close the door.

- **Locked Sound (AudioClip):** The sound played when the player try to open the door but the door needs a key or needs a key of other **Lock Type**.

- **Audio Source (AudioSource):** The audio source that will play the sounds of the door.

FLASHLIGHT



☑ Flashlight (Script)

Script	Flashlight
Audio Source	FlashLight (Audio Source)
Switch On Sound	None (Audio Clip)
Switch Off Sound	None (Audio Clip)
Spawn In	Flashlight transform (Transform)
Lamp	Spotlight (Light)
Switch	F
Max Batteries	2

In the prefabs folder there is a example of a Flashlight.

- **Audio Source (AudioSource):** The audio source that will play the sounds of the **FlashLight**.

- **Switch On Sound (AudioClip):** The sound played when the player turn on the **FlashLight**.

- **Switch Off Sound(AudioClip):** The sound played when the player turn off the **FlashLight**.

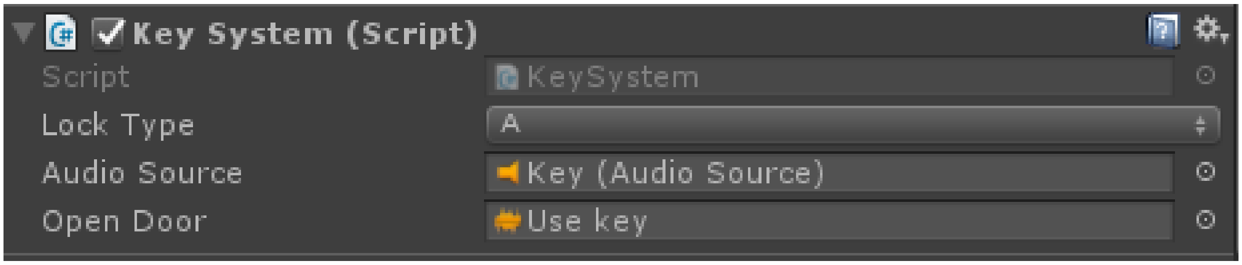
- **Spawn In (Transform):** The position where will be the **FlashLight** while the player is using it.

- **Lamp (Light):** The light of the **FlashLight**.

- **Switch (KeyCode):** The key that the player need to press to switch the **FlashLight**.

- **Max Batteries (float):** The maximum amount of batteries that can has the **FlashLight**.

KEYSYSTEM



Key System (Script)

Script: KeySystem

Lock Type: A

Audio Source: Key (Audio Source)

Open Door: Use key

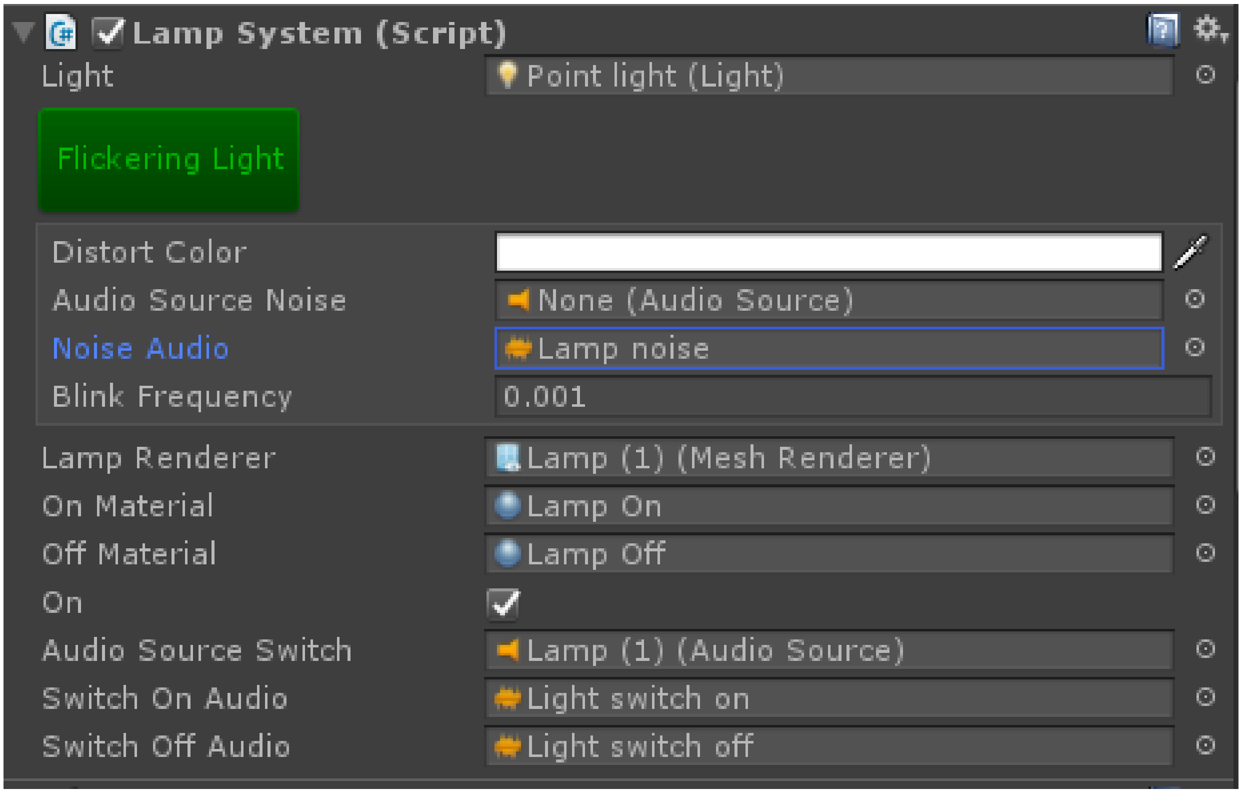
In the prefabs folder there is a example of a Key.

- **Lock Type (enum):** This variable is also in **Door**.
  Example: If the **Lock Type** is “**A**” this key can open doors with the same **Lock Type**. The **Lock Type** of the **Door** can be “**Free**” and the door don't need a key to be open.

- **Audio Source (AudioSource):** The audio source that will play the sounds of the **Key**.

- **Open Door (AudioClip):** The sound played when the player use the **Key** to open a door.

LAMPSYSTEM



Light: Point light (Light)

Distort Color:

Audio Source Noise: None (Audio Source)

Noise Audio: Lamp noise

Blink Frequency: 0.001

Lamp Renderer: Lamp (1) (Mesh Renderer)

On Material: Lamp On

Off Material: Lamp Off

On: ☑

Audio Source Switch: Lamp (1) (Audio Source)

Switch On Audio: Light switch on

Switch Off Audio: Light switch off

In the prefabs folder there is a example of a Lamp.

- **Light (Light):** The light of the lamp.
- **Flickering Light (GUI Button):** The lamp will flicker?.
    - **Distort Color (Color):** The color that the **Light** will turn.
    - **Audio Source Noise (AudioSource):** The audio source that will play the **Noise Audio**.
    - **Noise Audio (AudioClip):** The sound played while the **Light** is on.
    - **Blink Frequency (float):** How often will the **Light** blink?.
- **Lamp Renderer (Renderer):** The mesh renderer of the Lamp.
- **On Material (Material):** The material of the lamp when it is on.

- **Off Material (Material):** The material of the lamp when it is off.

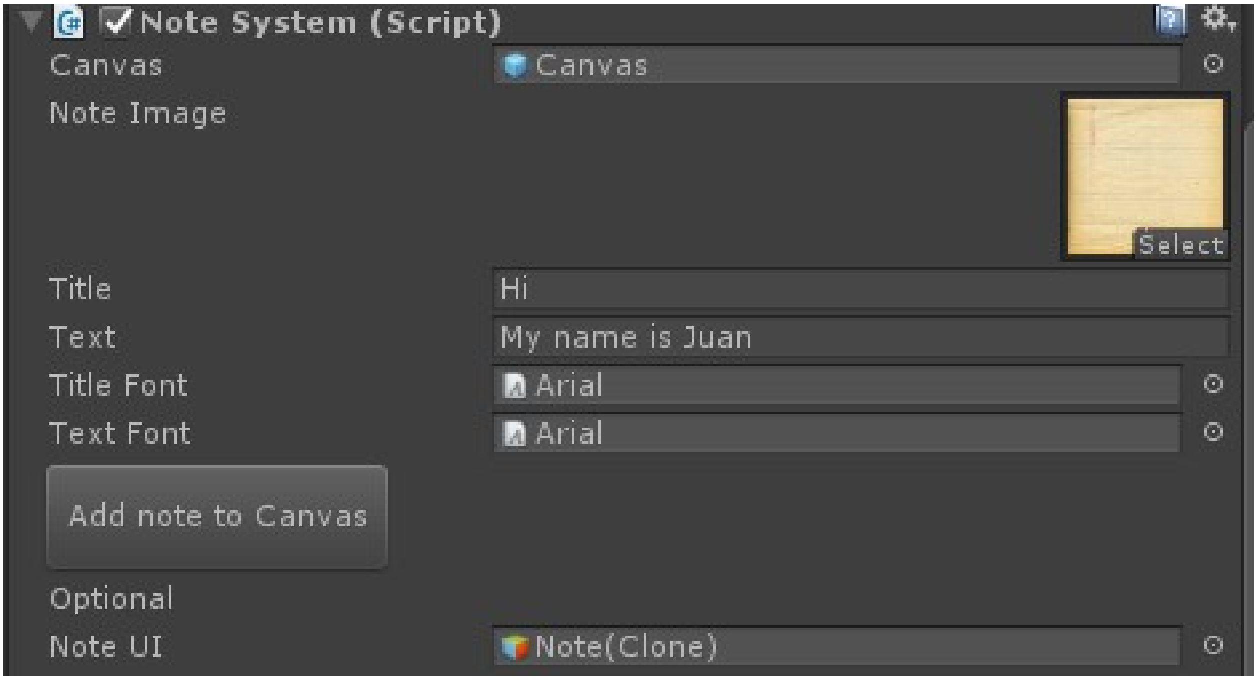
- **On (bool):** The lamp will start on?.

- **Audio Source Switch (AudioSource):** The audio source that will play the switch audio.

- **Switch On Audio (AudioClip):** The sound played when the Light turn on.

- **Switch Off Audio (AudioClip):** The sound played when the Light turn off.

NOTESYSTEM



Note System (Script)

Canvas: Canvas

Note Image

Title: Hi

Text: My name is Juan

Title Font: Arial

Text Font: Arial

Optional

Note UI: Note(Clone)

In the prefabs folder there is a example of a Note.

- **Canvas (GameObject):** The canvas where will be added this note.

- **Note Image (Sprite):** The note image.

- **Title (string):** The title of the note (it can be modified later directly in **Canvas**).

- **Text (string):** The text of the note (it can be modified later directly in **Canvas**).

- **Title Font (string):** The font of the **Title** (it can be modified later directly in **Canvas**).

- **Text Font (string):** The font of the **Text** (it can be modified later directly in **Canvas**).

- **Add note to Canvas (GUI Button):** If you hit it, will be created a note and added to **Canvas**.

<u>**Other way to create a note:**</u>

You can create the note yourself and add it to **Note UI,** and you don't have to fill the previous variables**:**

- **Note UI (GameObject):** The note in the canvas.

PLAYER

Climb Speed: At what speed the player will climb a ladder.

Grab

Grab Icon: The icon showed to tell to the Local player that he can grab that item.

Move To Offset: When you grab an item should it be bringed to the front of the player Camera? Or grab it from his position.

Grab From Distance: At what distance can the player grab an item.

Start Weapons: The weapons that will have the player at the start of the game.

Camera

Ragdoll Die Camera: When the player dies is activated this camera to show his ragdoll body dying.

Mini map Camera: The camera that renderizes the mini map world

P Camera: The camera that is inside the player, (The local camera).

AudioSource: To play sounds like footstep, jump and land.

Footsteps Sounds and Land Sounds: They are played depending on the layer that is staying the player, and you should add 2 in footsteps sounds.

Camera Anim: The camera animator to play animations like sniper sight breathing.

FPS View: The gameobject reference to apply the tilt rotation to it.

Weapon Recoil, Cam Kick Back, Weapon Kick Back: All them are references to apply different rotations and positions to make a more realistic gun.

Crouch Walk Speed: At wich speed the player will move crouched.

Jump Force: Force applied to the player in Y direction when jump:

Walk Speed: The normal speed of the player:

Run Speed:

Look Sensitivity: The mouse sensitivity to look around.

Weapon Base: To put weapons inside of it

Model Weapon Base: To put weapons inside of another player's body

Gravity: The gravity of this player

Head Bob: Should the camera have head bobbing when player walks.

Headbob Amount X: The amount in horizontal position that will bob the camera

Headbob Amount Y: The amount in vertical position that will bob the camera

Eye Height Racio: The base height of the head bob.

Walk Distance To Play Step: How often (in distance) will play footstep sounds

Stick To Ground Force: Gravity with will stick to ground when jump.

VR PLAYER

Add this script to your player when you want to use it on Google Cardboard VR

Movement Type: Feet Or Blueetoth Controller, Should the player move with real walking or using a blueetoth controller connected to your cellphone.

Head Base: The gameobject where is attached the head motion script of Google Cardboard

Ground Tag: The tag of the ground.

AudioSource: AudioSource To play movement sounds.

Footsteps Sounds: For better sound add 2 differents footsteps sounds.

Jump Sound:

Touch Ground Sound: Land Sound

Speed Factor: Walk Speed

Walk Min Time:

Delta Derivative:

Derivative Threshold:

Jump Speed: Jump Force.

AI DOLL

Model Renderer: The mesh object

Doll Behaviour: There are completely differents dolls abilities and gameplays.

Dead Time To Kill: How much time (only for the first doll behaviour) you can't look elsewhere before be killed by the doll

Jump Scare 3D:

Audio Source: AudioSource to play sound like screams and spawn sounds.

Kill Sound: The scream played when you are killed

Dissapear Sound: Sound played when the doll dissapear.

Spawn Sound: Sound played when the doll appears.

Min Random Spawn: Doll will be spawned in a random time between this seconds and **Max Random Spawn**

Time Spawned: How much time will be active the doll

Max Spawns: How much spawns can do the doll if is infinite let it on zero.

Spawn Distance From Player: Distance that the doll keep when spawn.

Respawn Time: When the doll kills the player, the doll waits this time to start spawning again.

AI MANAGER

AI Enemies To Spawn: Would you like to spawn the enemies from the manager?, you could have them already actives in the scene.

Spawn Points: The random spawn points.

AI PLAYER:

AI Damager: The hand trigger to hit enemy.

Model: Gameobject where is attached the animator.

Attack Interval: Attack rate in seconds

Min Distance To Attack: The distance between your enemy to attack

GAME MANAGER:

Jumpscares 2D: If you want to show some 2D jumpscares you've to add them in this list.

Jumpscares 2D AudioSource: AudioSource to play the screams.

Items Prefabs: All the items needs to be attached to this list for the save & load system.

Examinable Object Transform: When examining item the position reference.

Spawn Points: Spawn Points of the player to spawn when death.

PICKUP SYSTEM

Start Items: Initial player items.

Max Items In Inventory: How much items can the player pick up.

Inventory Items Icons: In Canvas. To show item icons.

Player Controller: The player controller script.

Pick Distance: At wich distance can the player pick up.

Default_Crosshair: Normal crosshair

Interact_Crosshair: Crosshair when pointing some.

Description: Description UI Panel when pointing some.

Item Name Txt: Name Info in Descriptive Panel.

Description Txt: Item Info in Descriptive Panel.

Interactive Objects: Layer of the objects that can be picked up

RECHARGABLE

Add this script to items that work with batteries (flashlight, camera).

Max Batteries: Maximum batteries that can hold this item.

StartBatteries: The batteries that brings the item.

WEAPON

Shell Prefab: Should be in resources folder. If you don't want a shell let it empty

Shell Pos Base: Where will be spawned the shells.

Shell Force: Force applied to rigidbody

Shell Torque: Torque applied to rigidbody

Also you can modify the random variables to add it a different look with every shoot.

Infinite Ammo: Recommended for AI Players.

Weapon Type: To match it with his magazine ammo type.

Weapon Shoots: How many shots will fire with each one (example a shotgun fires 6 shots a pistol only 1)

Distance With Camera: The position Z of the Weapon Base when you use it.

Use Sight 2D: It is a sniper?.

Sight Breath: Should the sight has an animation like sniper breathing movement?.

Sight 2D Camera Fov: Zoom applied when sighting.

Max Camera Fov: Max Zoom allowed.

Max Time To Explote: How much time can you sostain the breath? Before play an animation and release the breath.

Recoil

KB: Is the random kick back force applied to the weapon and camera when shoot horizontally and vertically.

Return Speed: is the speed to return to initial position before kick back force

Aim Recoil: The random recoil that has the raycast thrown by the weapon shot (for example the different position of each shot in a shotgun).

Recoil: The recoil of the weapon made by code with Quaternion. Lerp

Camera Shake On Shoot: If also the camera should has kick back recoil or only the weapon.

Sway

Sway the weapon when turning around.

Tilt: Tilt weapon when moving.

Crosshair

The crosshair size depending of every state of the weapon (example it is more unprecise if you are running)

Effects in hit. (example bullet hole) depending of the layer hitted.

Damage

Set the damage values depending on which part of the body was hitted

Cartridge Ammo: Magazine ammo (example a glock has 15 bullets).

Start Ammunition: How much ammunition will have at the start of the game (example 90 bullets).

Shoot Interval: Fire rate (every . Seconds).

Reload sounds

Called by **AnimationsSounds** script. Animation Events

Aiming Down

Aim Position: The position of the weapon base when aiming down.

Ads Speed: At wich speed will be the weapon aimed down.

Fov Ads: Change field of view of the camera when aimed down. At **Fov Ads Speed.**

Running

Run position: The position of the weapon base when running.

Run Pose Pos Speed: At wich speed the weapon base will lerp to the new position.

TORCH

Spawn In: The item base reference position where will be the torch while using it. (Default should be Item Base GameObject inside the player and camera :)

Flame: The Fx (Particle System) Effect.

SWITCH

Switch Rotate: The switch button to rotate when turn lights.

Rot To Angle: Up to wich angle will rotate.

Speed Rot: How fast will rotate.

Lamps: The lamps (lamps should have a **LampSystem** script attached) that will turn on/off with this switch.

LOCKER

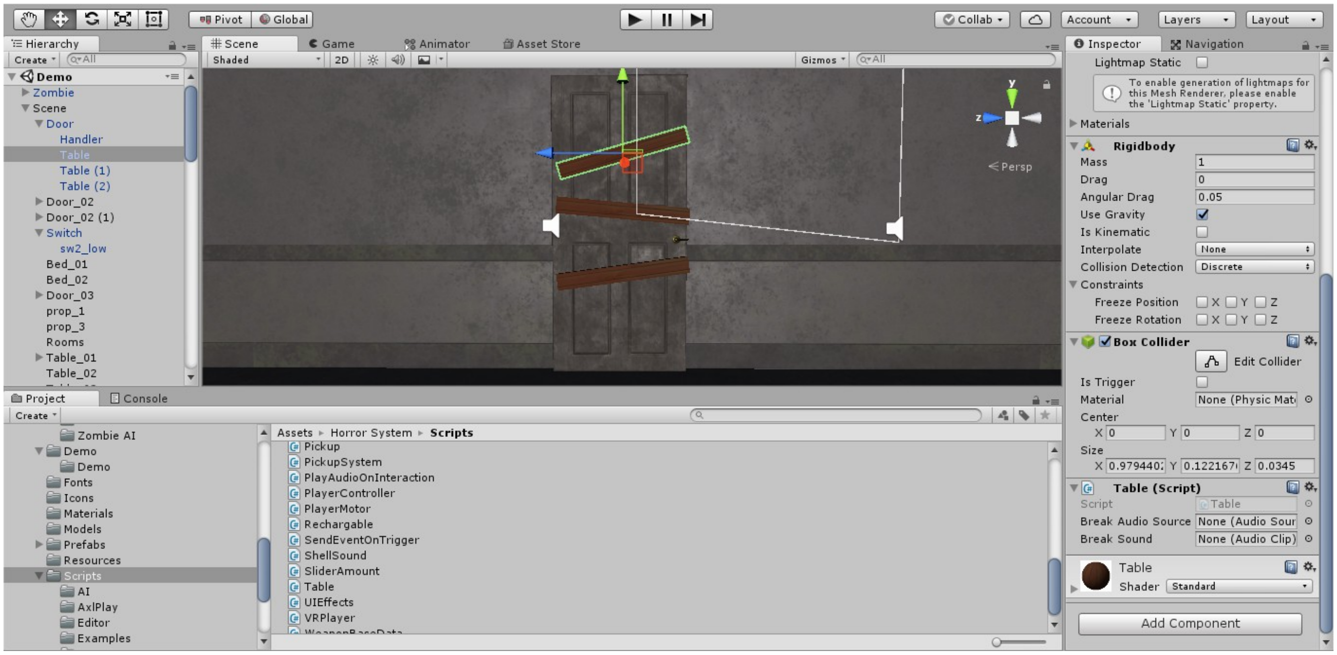
The door of the locker needs to have a **Door** script attached to open and close it, the locker is used in Horror Games to hide from zombies and enemies.

Inside Locker: The reference position and rotation to put into the player when enter the locker (to look outwards the locker)

Rot Speed: When the player enters into, at wich speed will the player rotate to match the Inside Locker rotation.

TABLE

If you want to make a blocked door by tables, you should add your table mesh into the door, and add to the tables the script **TABLE** it will add a Rigidbody and a Collider, these are needed to check for a shoot of a weapon and come off. Also you must add all these tables to your door script



Lightmap Static □

To enable generation of lightmaps for this Mesh Renderer, please enable the 'Lightmap Static' property.

Materials

Rigidbody
Mass 1
Drag 0
Angular Drag 0.05
Use Gravity ☑
Is Kinematic □
Interpolate None
Collision Detection Discrete
Constraints
Freeze Position □ X □ Y □ Z
Freeze Rotation □ X □ Y □ Z

☑ Box Collider
Is Trigger □
Material None (Physic Mat
Center X 0 Y 0 Z 0
Size X 0.979440: Y 0.122167(Z 0.0345

Table (Script)
Script Table
Break Audio Source None (Audio Sour
Break Sound None (Audio Clip)

Table
Shader Standard

CREATE YOUR OWN ITEM

I recommend you to read the documentation of **Interactive Object** because it is the base of all the items.

1- Add to your item the script **Interactive Object**.
2- Select the type of the item.

The type of the item will change how it will work and the methods that we will use.

USABLE:

In the folder Easy Horror System/Scripts/Examples there is a script called “**UsableItemExample**” with the methods that we are going to create.

1- Create a new script.

Now, the next depends on what you want to create. All the items work with specifics methods called by **HUDSystem** and **InventorySystem**.

Those methods for this type of item are:

Pickuped: Called when the player pickup this item.

StartUsing: Called when the player starts using this item.

StopUsing: Called when the player stops using this item.

StartLooking: Called when the player looks directly and closely this item.

StopLooking: Called when the player stops looking directly and closely this item.

Also you can create an IEnumerator with these methods.

To see more examples you can look the script: **Flashlight** in (Easy Horror System/Scripts/Items) and **CameraSystem**.

ONE CLICK INTERACTION:

In the folder Easy Horror System/Scripts/Examples there is a script called "**OneClickInteractionExample**" with the methods that we are going to create.

1- Create a new script.

Now, the next depends on what you want to create. All the items work with specifics methods called by **HUDSystem** and **InventorySystem**.

Those methods for this type of items are:

Interaction: Called when the player interact with this item.

StartLooking: Called when the player looks directly and closely this item.

StopLooking: Called when the player stops looking directly and closely this item.

Also you can create an IEnumerator with these methods.

To see more examples you can look the script: **Door** in (Easy Horror System/Scripts/Items) and **LampSystem**.

EXAMINABLE:

In the folder Easy Horror System/Scripts/Examples there is a script called "**ExaminableItemExample**" with the methods that we are going to create.

1- Create a new script.

Now, the next depends on what you want to create. All the items work with specifics methods called by **HUDSystem** and **InventorySystem**.

Those methods for this type of items are:

StartExamining: Called when the player starts examining this item.

StopExamining: Called when the player stops of examine this item.

StartLooking: Called when the player looks directly and closely this item.

StopLooking: Called when the player stops looking directly and closely this item.

Also you can create an IEnumerator with these methods.

To see more examples you can look the script: **ExaminableObj** in (Easy Horror System/Scripts).

CONSUMABLE:

In the folder Easy Horror System/Scripts/Examples there is a script called "**ConsumableItemExample**" with the methods that we are going to create.

1- Create a new script.

2- Add the following line:

using AxlPlay;

Now, the next depends on what you want to create. All the items work with specifics methods called by **HUDSystem** and **InventorySystem**.

Those methods for this type of items are:

StartLooking: Called when the player looks directly and closely this item.

StopLooking: Called when the player stops looking directly and closely this item.

Pickuped: Called when the player pickup this item.

Consume: Called when the player consume this item.

In this method we need to write the following line once that the item has been consumed.

```
NotificationCenter.DefaultCenter.PostNotification(this, "ItemConsumed");
```

Also you can create an IEnumerator with these methods.

To see more examples you can look the script: **Battery** in (Easy Horror System/Scripts/Items).

THIS IS ALL FOR THE PDF DOCUMENT.

NOTIFICATION CENTER DOCUMENTATION

I WILL BE ADDING VIDEO TUTORIALS THERE OF EASY HORROR SYSTEM.

IF YOU HAVE ANY QUESTION OR NEED SUPPORT LET ME KNOW: axlplay@gmail.com