LEVER PUZZLE SYSTEM V1.0
DOCUMENTATION

Contents

MATTHEW RUKAS – VOLUMETRIC GAMES
WWW.VOLUMETRIC-GAMES.COM
VOLUMETRICGAMES@GMAIL.COM

Introduction

Thank you for purchasing the “Lever Puzzle System”. This is a Raycast based system where the aim is to pull the levers in the correct order, once the limit is reached you can press a button to test if the order is correct. Each of the levers has it’s own number that you can set, and the “LeverController” script controls the order you want. It’s very easy to setup and even easier to add more switches.

The asset includes:
- Raycast system that detects the switches and buttons independently, so an infinite number of levers can be added as well as multiple versions of switches in the same scene.
- Add as many switches as are needed
- Limit the number of levers you can pull based on the value given
- Buttons, indicator lights and animated switches included
- Scripts which can be easily modified
- Sound effects for the system

FAQ

Q). How do I Import the asset?

A). Go to the Unity asset store and visit your “Download manager”. Download the asset if not already downloaded and click “Import”, import all required features of the asset for your use. It should have appeared in your project under “Lever Puzzle System”.

Q). Before you start / Why isn’t my character moving in the demo scene?

A). I was advised by Unity to delete the “**Standard Assets**” > “**Characters**” from the asset, so please make sure to right click in the “**Project Panel**” > “**Import Package**” > “**Characters**” so the FPSController will be updated for you to use!

Q). Is there an example of this asset working?

A). Yes, you can open the “LeverDemo” to see the lever asset in action or use this scene as your initial base of your project.

Q). How can I manually setup this asset?

A). See the manual setup instructions on “Page 2”.

Q). I’m having trouble getting the interaction to work. What can I do?

A). Make sure your Interactive objects have the “LayerMask” at the top right of the inspector as “Interact”, and the appropriate tags are added to each. See setup for more details.

Q). How can I add more levers to the set I already have?

A). Just duplicate a lever you already have, make sure it has the appropriate tags. Edit the “LeverController” > “LeverOrder” to incorporate 6 numbers, and set pulls to 6. Refer to the “Extending the lever system” for more information.

Q). How can I add more than one lever set to my scene?

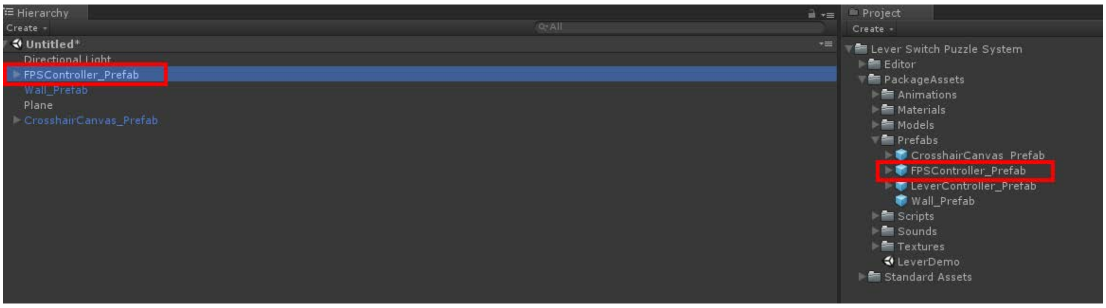
A). Check the “Extending the keypad system”.

Q). Do you want to organise the lever elements into a empty GameObject?

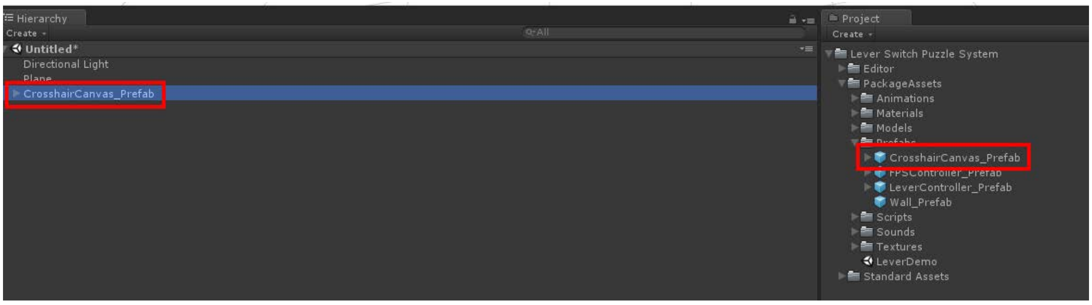
A). When you create an empty GameObject, please make sure that the X, Y, Z coordinates of the empty GameObject are all 0, 0, 0 before dragging the parts into the GameObject.

Manual Setup – Initial #1

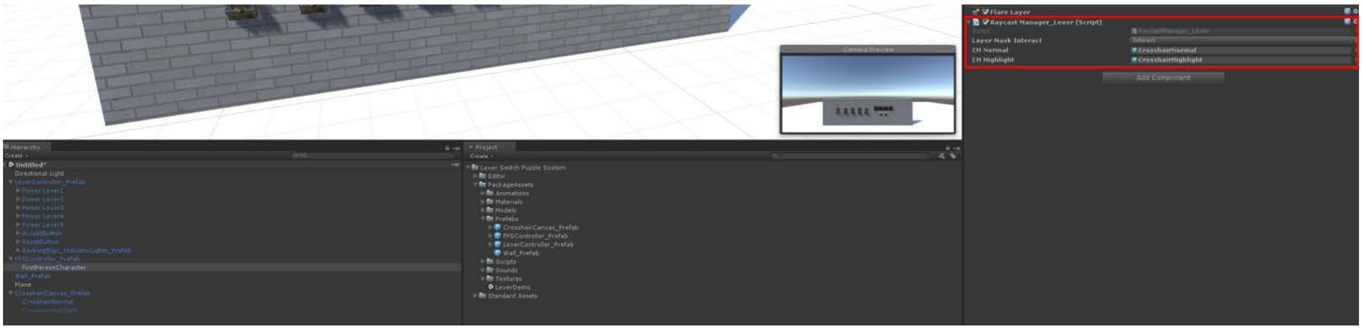
1. When starting your new project please import the “**Characters**” Standard assets or any FPSController you wish. Add an “**FPSController**” to your scene. You can drag this into the hierarchy or the main scene. Please navigate to the scripts folder and add the “**RaycastManager_Lever**” script to your “**FirstPersonCharacter**” or “**MainCamera**”



2. Now drag the “**CrosshairCanvas_Prefab**” from the “**Prefabs**” folder into the hierarchy. **NOTE: Only drag this prefab into the hierarchy, not the scene.**

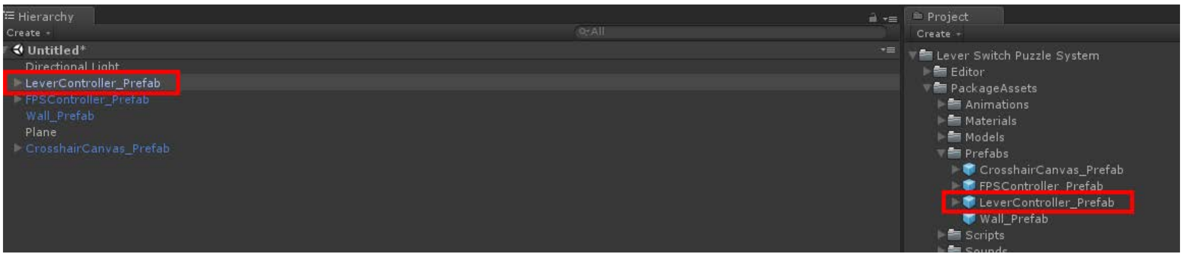


3. Open the “**CrosshairCanvas_Prefab**” parent and you will find “**CrosshairNormal**” and “**CrosshairHighlight**” add those to the “**FPSController**” > “**FirstPersonCharacter**” “**RaycastManager_Lever**” crosshair slots in the inspector. Also make sure the “Layer Mask Interact” is set to “Interact”. **NOTE: If this isn’t available in the dropdown you will need to create it at the top right of the Unity inspector, by choose the “Layer” dropdown and choosing “Add Layer”.**

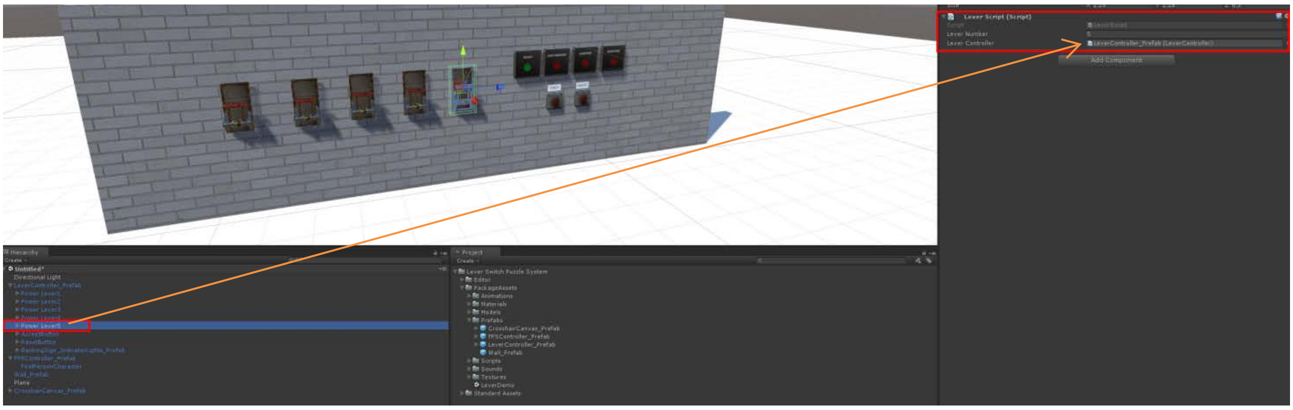


Raycast Manager_Lever (Script)
Script: RaycastManager_Lever
Layer Mask Interact: Interact
CH Normal: CrosshairNormal
CH Highlight: CrosshairHighlight

4. Add “**LeverController_Prefab**” to your hierarchy or scene from the assets “**Prefabs**” folder.



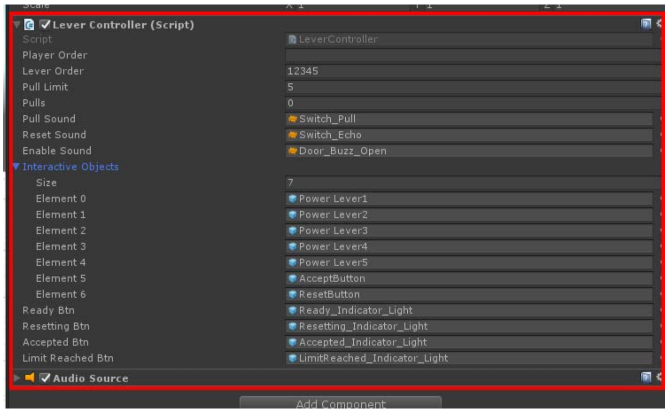
5. **OPTIONAL**: You will want to add a floor to your scene, if not already.
6. You can position the “**LeverController_Prefab**” anywhere in the scene, it will move as one.



Lever Number 5

Lever Controller LeverController_Prefab (LeverController)

7. Select the "**LeverController_Prefab**"
   PlayerOrder: Leave this blank, it will auto populate as you press the levers
   LevelOrder: This is the code order you wish to complete the lever puzzle
   PullLimit: Add the number of digits that are in the string of "LevelOrder"
   Pulls: Leave at "0" by default, this will be incremented as the levers are pulled
   PullSound: You can add your own sound or use the provided "Switch_Pull" sound effect
   ResetSound: You can add your own sound or use the provided "Switch_Echo" sound effect.
   EnableSound: You can use your own or use the provided "Door_Buzz_Open" sound effect.
   InteractiveObjects: This is an array, based on how many levers, and the two buttons (Reset and accept). Example: If you have 5 levers, and those 2 recommended buttons you'll need 7 array size and add those to it.
   ReadyBtn: You need to add if not already the "Ready_Indicator_Light" which will be found in the "BackingSign_Indicator_Light_Prefab" under the appropriate folder.

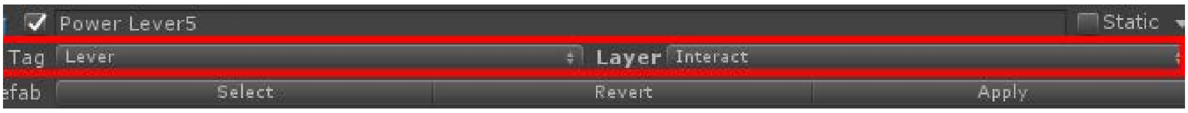


☑ Lever Controller (Script)

Script	LeverController
Player Order	
Lever Order	12345
Pull Limit	5
Pulls	0
Pull Sound	Switch_Pull
Reset Sound	Switch_Echo
Enable Sound	Door_Buzz_Open
Interactive Objects	
Size	7
Element 0	Power Lever1
Element 1	Power Lever2
Element 2	Power Lever3
Element 3	Power Lever4
Element 4	Power Lever5
Element 5	AcceptButton
Element 6	ResetButton
Ready Btn	Ready_Indicator_Light
Resetting Btn	Resetting_Indicator_Light
Accepted Btn	Accepted_Indicator_Light
Limit Reached Btn	LimitReached_Indicator_Light

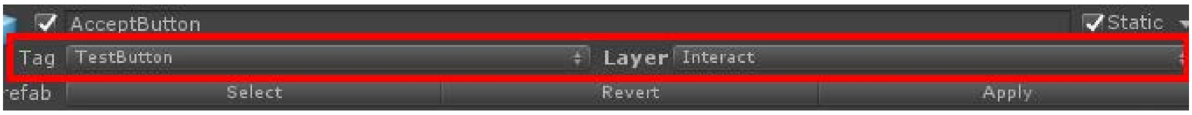
☑ Audio Source

8. Select any of the levers in the "**LeverController_Prefab**" and make sure it has the appropriate "LayerMask" and tag. LayerMask = "**Interact**" and Tag = "**Lever**".
   NOTE: You may need to create a new user layer and call this "**Interact**". It should then populate the other levers.

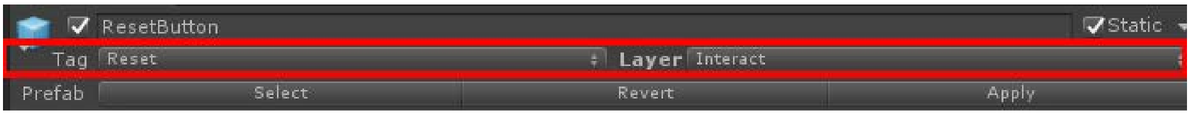


☑ Power Lever5 □ Static
Tag Lever Layer Interact

9. Similarly with the Reset and accept buttons, that require the "TestButton" and "Reset" tag.



☑ AcceptButton ☑ Static
Tag TestButton Layer Interact



☑ ResetButton ☑ Static
Tag Reset Layer Interact

Final Notes

Your asset is setup and ready to use in your scene, please remember a few things.

There are 4 indicator lights: **Ready**, **Limit Reached**, **Accepted**, **Resetting**. Each turn red and green depending on the circumstance.

Ready – Green when you can use the switches, red when not.
Limit Reached – Green when you have pulled the switches the maximum number of times, meaning you need to press the test button or reset. Red when you can use the switches as normal.
Accepted – Green when you've pressed the test button and the combination is correct after the limit was reached.
Resetting – Green when you either got the code wrong after testing the order or you reset it manually, a short delay is added before you can reuse the switches again.

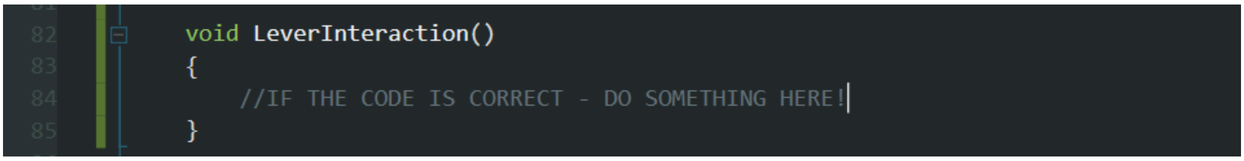
NOTE: You can test the system is working by selecting the "**LeverController_Prefab**" in the hierarchy and testing the game outside of maximize on play. This allows you to see the inspector whilst playing the game. You will then see the "**PlayerOrder**" increment as you pull each lever. Test each button and switch to make sure your asset is working correctly.

If you have any problems, don't hesitate to send an email!

Extending the Switch System

I want to do something different when the code is valid:

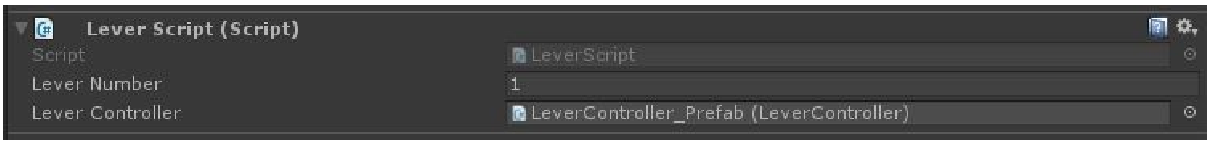
In the “**LeverController**” script you can find line 82 which has a function called “**LeverInteraction()**” In this function can add your own interaction, for the possibility of opening a door or secret passage. It’s dependant on the game and the setting it’s being used. It could switch on a generator for powering a building. You may want to add a boolean if you have multiple switches sets of levers in the same scene to detect which interaction you’d like. If you feel any sort of confusion about this, please don’t hesitate to send me an email!



```
82      void LeverInteraction()
83      {
84          //IF THE CODE IS CORRECT - DO SOMETHING HERE!
85      }
```

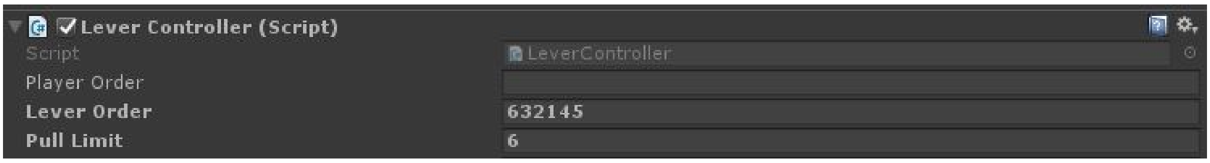
Adding more than 5 levers to a setup?

Just add another lever by duplicating one you already have. Give the lever another number. If you have added your 6th lever, make this number 6. Do this in the “**LeverScript**” by changing the “LeverNumber” value in the inspector.



Lever Script (Script)	
Script	LeverScript
Lever Number	1
Lever Controller	LeverController_Prefab (LeverController)

Increment the pulls and order number on the “**LeverController**” GameObject, dependant on how many levers you have. For Example; If you have 6 levers, make sure the “**LeverOrder**” has 6 numbers: 1,2,3,4,5,6 and the pull limit also has “6” in it.



☑ Lever Controller (Script)	
Script	LeverController
Player Order	
Lever Order	632145
Pull Limit	6

How to add another lever set to the same scene?

Duplicate the entire “**LeverController**” parent GameObject in your scene and reposition wherever you want. Check all scripts are linked and it should be working fine.

Contact

If you have any problems with the pack, or have some ideas for new features you'd be interested in, please feel free to contact me.

Email: volumetricgames@gmail.com

Website: http://www.volumetric-games.com