**Level Hierarchy And Systems**

I have restructured the prefabs folder in order to make creating/editing levels much easier(hopefully). In this document i will explain the NEEDED hierarchy for each level, as well as how each piece works for a better understanding.

**Table of contents:**

Hierarchy; pg1; info on each part of a level (summary)

Prefab Organization; pg 3; info on the new organization system.

Info To Know; pg 6 ; in depth explanation on how to use certain objects/scripts

Update:

Balls must be inside their respective puzzles.

New feature:

Enable Objects when puzzle is finished.

In Order to enable an object when the puzzle is finished, it must have the tag “EnableMe” and it must be inside of its respective puzzle. This object will normally be a bridge or stairs, and does not have to be disabled before starting the game. It will be disabled through script so it will be easy to make sure it is in the correct spot.

**Hierarchy**

This part is important due to how certain scripts gather information, thus it is best to follow this structure in order to allow for easier creation of each level. Below is a picture of how each scene hierarchy should look.



**CameraRig and SteamVR** - This is the player

**WwiseGlobal** - This is the audio manager

**PathManager** - This is what allows felix to fly

**Memory fragment** - takes player to next level?

**Felix’s dad** - This is felix, or at least what holds him together

**PuzzleArea** - Hold a puzzle together

**Environment** - where everything else should go; This is mostly for making everything look nice and easy to get to in the hierarchy.

**CameraRig and SteamVR in depth -**



**SteamVR -** its best to keep this directly above or below the cameraRig, this must be in the scene and must be its own entity.

**CameraRig -** The picture(left) shows what should be inside the camera rig (minus felix… That needs to get deleted but i already took the picture sooo oh well).

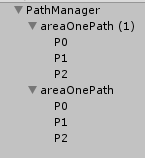
Its best to just not touch anything inside of this unless absolutely necessary.

**hipsY/Feet** - more info in “Info To Know” Section.

**WwiseGlobal -**

Just don’t touch unless you are noam. All i know is it makes the sound work and it needs to be in the scene.

**PathManager -**



Most info on this is in the “Info To Know” section.

**Note: This can be ignored and left as for now**

**AreaPath** - Allows for easier organization, just acts as a folder

**P0** **- Px** - each of these act as node points, the order from top to bottom tells felix where to go and when.

**Note: You can have as many AreaPath objects and Px objects as you like. An areaPath is NOT limited to 3 node points.**

This can be disabled for functionality testing along with felix’s dad

**Memory fragment -**

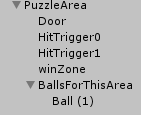
Not entirely sure, but its supposed to be in each level lol. Im pretty sure this is supposed to teleport the player to the next level.

**Felix’s Dad -**

This is felix, please do not fuck with him…. I need to make a few fixes due to changes made to the prefab. You should keep him in the level and just disable him and the PathManager for testing.

**PuzzleArea -**

**Important:** Everything that is part of this puzzle should be in here, that includes any grav rings or platforms in order to keep things organized. **NOTE: the tag BouncePlatform should only be used on objects that NEED to be hit in order to finish the puzzle.**



**PuzzleArea -** This holds an instance of a puzzle, has a script attached

**Door -** The door to be destroyed after the puzzle has been completed

**HitTrigger -** anything that has to be hit by the ball

**WinZone -** Last thing the ball has to hit in order to complete the puzzle

**BallsForThisArea -** Just holds the balls and makes organization easier

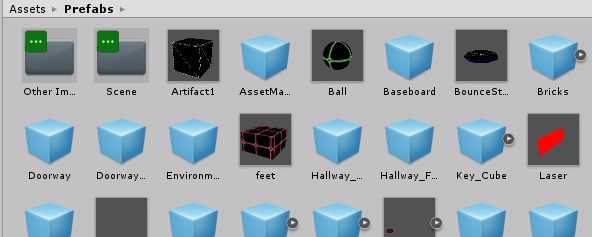
* **HitTrigger -** This can be ANYTHING that the ball has to hit or pass through.
  + As many as you want in a PuzzleArea
    - If you dont want any of these, make sure nothing in the puzzle has the tag “BouncePlatform”
  + **NOTE:** these must be in order of first to last in the hierarchy.
  + **I.e. the ball must go in this order to beat the puzzle**
    - HitTrigger0 -> HitTrigger1 -> winZone
* **WinZone -**
  + Only one per PuzzleArea
  + Must be a trigger
* **Balls -** 
  + All balls for a puzzle must be kept in the PuzzleArea parent, for best practice you should just keep them in their respective “BallsForThisArea” parent.
  + **NOTE: BALLS HAVE TO BE UNDER THEIR RESPECTIVE PUZZLEAREA FOR THINGS TO WORK. BALLS CANNOT BE PLACED OUTSIDE BECAUSE IT WILL CAUSE ERRORS.**

**Environment -**

Anything for the environement should be thrown in here. It will allow for easier navigation. Feel free to make empty objects inside of this to allow for easier navigation for you guys as well.

**Note:** this is not necessary for the game to work properly, but it does make life easier in the long run.

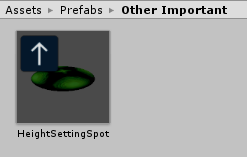
**New Prefab Organization**



*New organization folders: Scene, and Other Important.*

Anything outside of the two new folders are things i wasn’t sure what to do with, they can stay where they are for now.

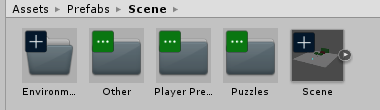
**Inside Other Important Folder -**

Things in this folder are necessary for certain code, These are not to be deleted and not to be placed in the scene. 

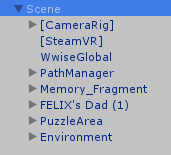
Currently, this only contains one thing but if prefabs are made that meet this criteria; they should be put in this folder.

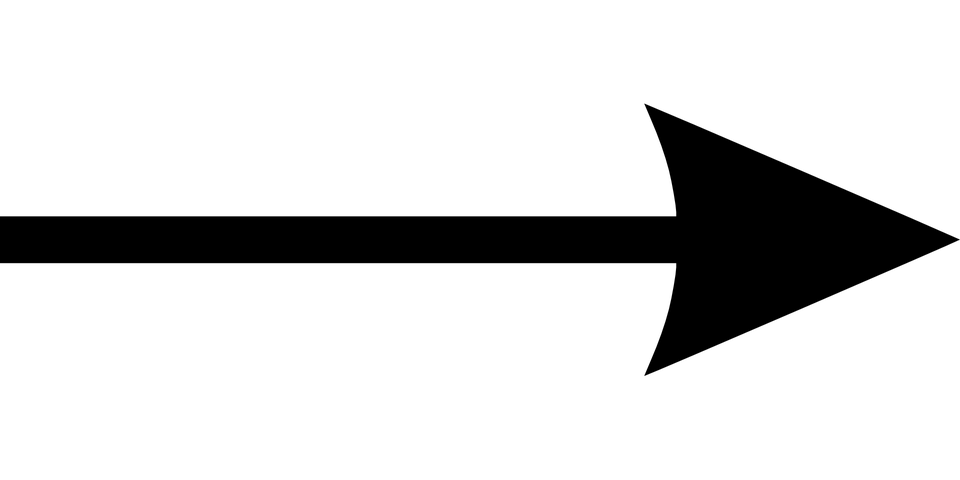
**Inside Scene Folder -**

Inside the scene folder is four other folders and a prefab. The prefab is a “quick start” to making a level. It includes all of the essentials in one neat package.



* Scene Prefab
  + The prefab on the far right is like a quick start to any level.
  + You can drag and drop it directly into the scene and the level will have all the necessary components for a level.
  + **NOTE: YOU MUST BREAK THIS PREFAB IN ORDER FOR THE LEVEL TO WORK**
    - All of these components are held in a parent named scene. Just take all the objects out of the scene object and delete “scene”





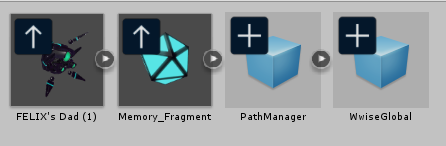
*Scene prefab Drag out all children working scene(delete scene obj)*

**Inside Environment Folder -**

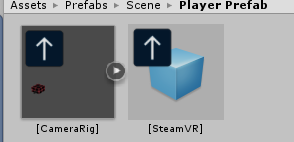
This is currently empty, but this can be used to store any environmental prefabs such as floors, walls etc.

This can be removed entirely if it is not useful.

**Inside Other Folder -**

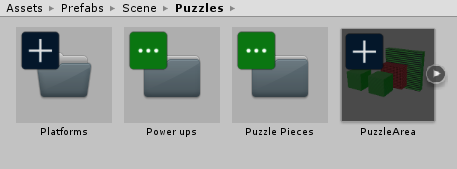


* Parts of the Scene prefab that are moved here so they have a home. This way if something is accidentally deleted, you only have to go to this folder and put it into the level. Or if my scene prefab isn’t to your liking.

**Inside Player Prefab Folder -** 

This contains the Camera Rig and the SteamVR prefabs. Again, for simplicity and modularity in terms of how the designer prefers to create levels.

**Inside Puzzles Folder -**



* The prefab in the photo (far right) is a basic puzzle area, the same as the one used in the Scene prefab.
  + It contains:
    - Puzzle Area
    - Two triggers
    - One winzone
    - One door
    - And One ball inside of a parent BallsForThisArea
* The other folders are for modularity.

**Puzzle Pieces -**

* Contains every piece of the prefab for PuzzleArea
* No duplicates in this folder, just a base object.

**Power Ups -**

* This is a folder dedicated to any prefab that is used to change how the ball acts.
  + Currently contains both Grav Rings

**Platforms -**

* Currently empty
* Used to hold any prefabs that the ball has to hit. This includes platforms, triggers, winzones, etc.
* **Note that this is for final prefabs, such as those with the proper scripts/settings as well as a model.**

**Info To Know**

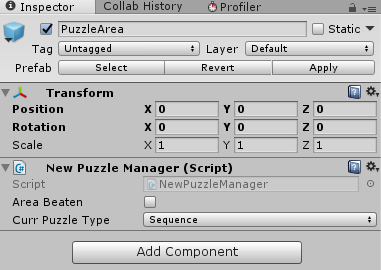
This part covers the do’s and don’ts of fine tuning objects in the inspector menu.

Also, it will cover the important aspects of each of the items inside and how to properly use them.

**Puzzle Pieces**

The need to know info about each piece of a puzzle.

**PuzzleArea -**



**Name: Puzzle Area(x)** (where x is a number)

**GameObject: Empty**

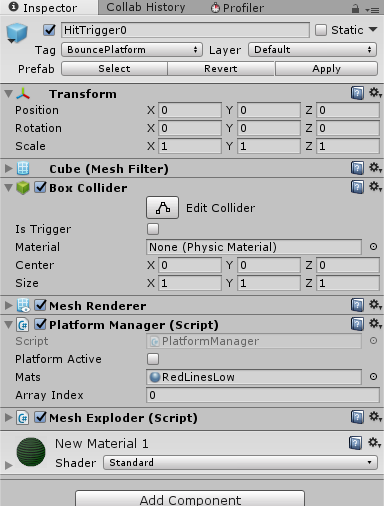
**Tag: Untagged**

**Script(s): New Puzzle Manager**

**New Puzzle Manager**

* Area Beaten
  + Should be false
* Puzzle Type
  + Must be Sequence
  + More options will be - added later

**HitTrigger -**

**Name: HitTrigger(x)** (where x is a number)

**GameObject: Some Mesh**

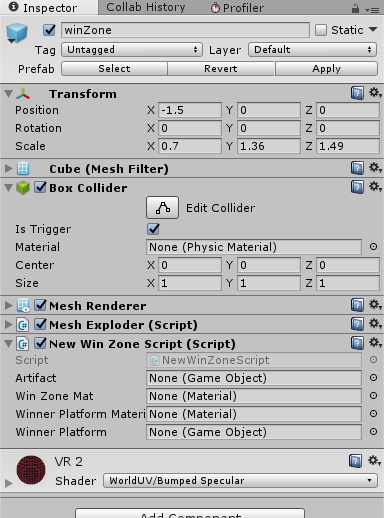
**Tag: BouncePlatform**

**Script(s): Platform Manager ; Mesh Exploder**

**Other Components:**

* **Mesh Filter**
* **Collider**
* **Mesh Renderer**
* **Platform Manager**
  + Platform Active should be left as false.
  + Mats is the material the object switches to after it is hit. The material in the Mesh Renderer is what it switches back too if the puzzle isnt done correctly.
  + Array Index - Ignore it, i don’t think it does anything anymore lol.
* **Mesh Exploder** 
  + Makes the platform go **BOOM** after the puzzle is completed. If it doesn’t go **BOOM**, then this script isnt on it.
* **Mesh Filter**
  + If you change the mesh, the shape is changed. **Do not make changes to the prefab.**
  + If changes are needed, use a non prefabbed version of this. Any new models can be saved as a prefab in the “Platforms” folder under Puzzles.
* **Collider**
  + **This can be any type of collider**
  + This can either be a trigger or not, the code will see it either way.
  + Suggestion: don’t use it as a trigger if you want the ball to bounce off.
  + **Stay away from mesh colliders if at all possible, they totally suck for optimization**

**Win Zone -**

**Name: winZone**

**GameObject: Some Mesh**

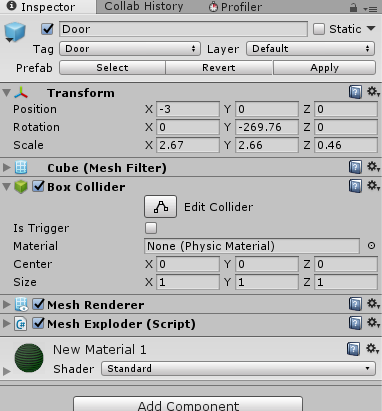
**Tag: Untagged**

**Script(s): New Win Zone Script ; Mesh Exploder**

**Other Components:**

* **Mesh Filter**
* **Mesh Renderer**
* **Collider**
* **Collider**
  + **This can be any type of collider**
  + Must be a trigger, the code will not accept it if it is not.
  + **Stay away from mesh colliders if possible**
* **Mesh Exploder**
  + Make it go boom when a ball hits it (in the proper order of course)
* **New Win Zone Script**
  + **Artifact slot**
    - Can be none, if it is not then there must be a winner platform.
    - This will instantiate the object that is slotted once a ball hits the winzone
  + **Win Zone Mat**
    - Material the winzone changes into after being beaten.
    - Just leave it as none if the winzone is supposed to blow up anyways.
  + **Winner Platform Material**
    - The material the winner platform changes into. (may be useless now)
  + **Winner Platform**
    - Where the artifact will spawn. This can be an existing platform that is changed into the “canTeleport” Laye.
    - Just leave it as none, pretty sure its not used anymore anyways.

**Door -**

**Name: Door** (feel free to add numbers)

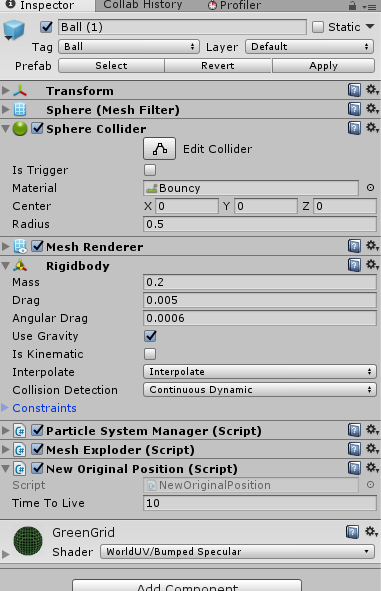
**GameObject: Some Mesh**

**Tag: Door**

**Script(s): Mesh Exploder**

**Other Components:**

* **Collider**
* **Mesh Filter**
* **Mesh Renderer**
* **Collider**
  + **NOT A TRIGGER**
  + **Stay away from mesh colliders if possible.**

**Ball -**

**Name:Ball (x)**

**GameObject: Sphere**

**Tag: Ball**

**Script(s): New Original Position; Mesh Exploder; Particle System Manager**

**Other:**

* **Mesh Filter**
* **Sphere Collider**
* **Mesh Renderer**
* **RigidBody**
* **New Original Position**
  + **Time to live**
    - Amount of time in seconds the ball stays alive.
    - This can be changed to make puzzles more difficult/easier
* **Particle System Manager**
  + **Dont touch, but is needed to cause particle effect when the ball hits something**
* **RigidBody**
  + **Use Gravity = true**
  + **Collision Detection = continuous dynamic**
  + This object needs a rigidbody in order to have physics