

# DOORS PRO

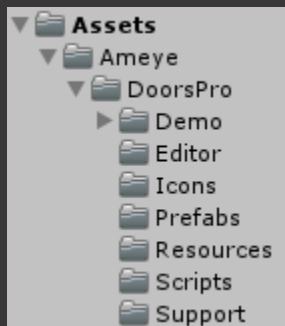
Welcome!

This is the documentation file of the asset '*Doors Pro*'. Here you'll find a guide to get you started as well as answers to questions you might have. If you still have questions after reading this guide, contact me. The contact information is at the end of the file.

## SETUP PROCESS

### Importing the files

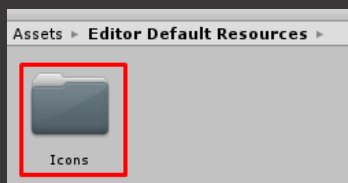
When you've downloaded and imported Doors Pro, your project window will look like this:



### Setting up tags and icons

First of all, we're going to move some folders so that the icons are displayed nicely!

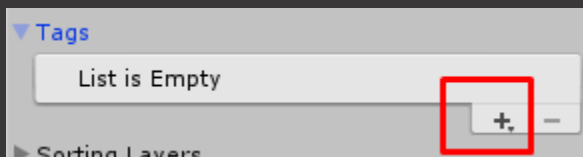
- 1) Create a folder called 'Editor Default Resources' and put it directly under the Assets root folder. So this would be Assets/Editor Default Resources.
- 2) Find the folder 'Icons' under Assets/Ameye/DoorsPro/Icons and move that folder into the Editor Default Resources folder.



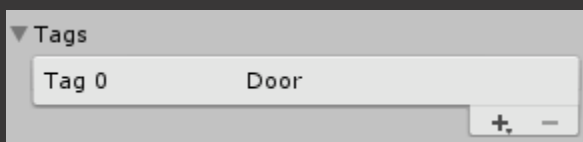
### 3) Re-start Unity

Secondly, you'll need to create the tag 'Door'. You can do this by following these steps:

1. Click Edit>Project Settings>Tags and Layers
2. Click on the little arrow next to 'Tags' and create a new tag by clicking on the plus icon

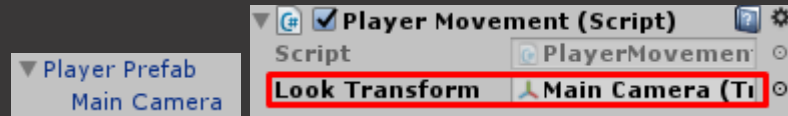


3. Name the new tag 'Door'



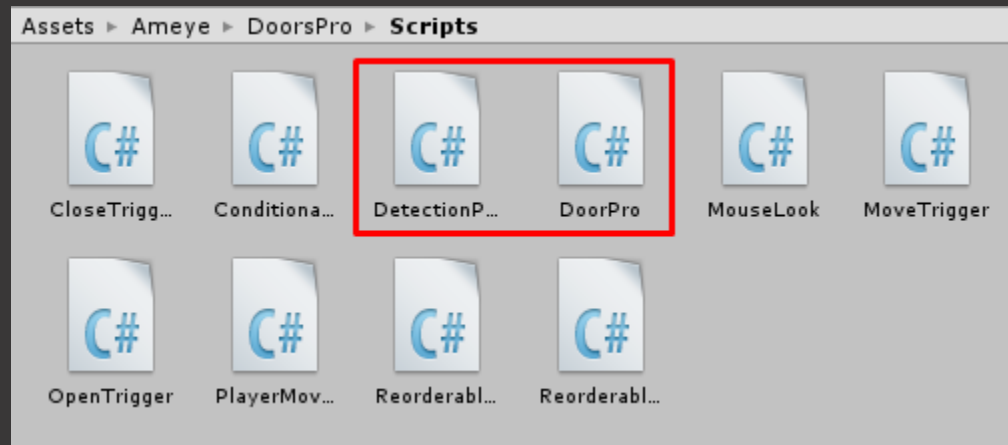
Creating a player

When starting with the Doors Pro tool you'll need to have a player in your scene that can move and look around. If you don't have a functioning player, there is a 'Player Prefab' included in the project files under Assets>Ameye>DoorsPro>Prefabs>Player Prefab. If you don't want to use the prefab but you still need the scripts to create a player, there is a 'PlayerMovement' script and a 'MouseLook' script. You can find them under Assets>Ameye>DoorsPro>Scripts.

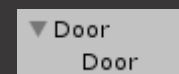


If you'd like to use these, add the Main Camera in your scene as a child of the player and position it like if the camera were the eyes of the player. Add the 'MouseLook' script to the camera and the 'PlayerMovement' script to the player. Then click on the player and under the 'Player Movement' script component click and drag the camera game object in the hierarchy onto the 'Look Transform slot'. Now that the player is fully set up we can set up the door tool itself.

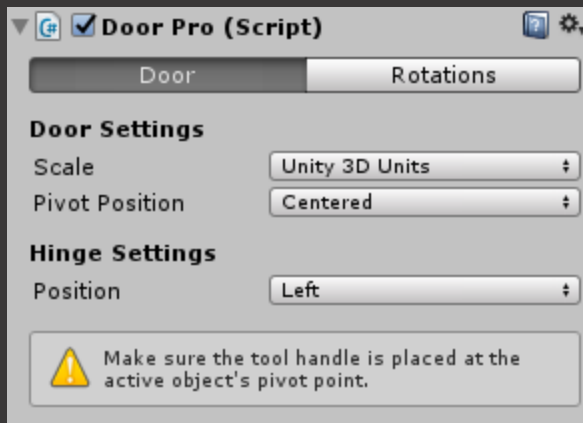
## The door script



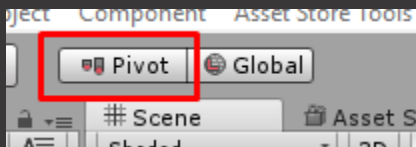
The door tool itself comes with 2 main scripts, 'DoorPro' and 'DetectionPro'. You can find them under Assets>Ameye>DoorsPro>Scripts. Drag the detection script onto the player you just created and then drag the door script onto every single door in your scene. When you drag the door script onto your door, it will automatically create a hierarchy around the door model. There will be a parent object added to your door model and later on also some children.



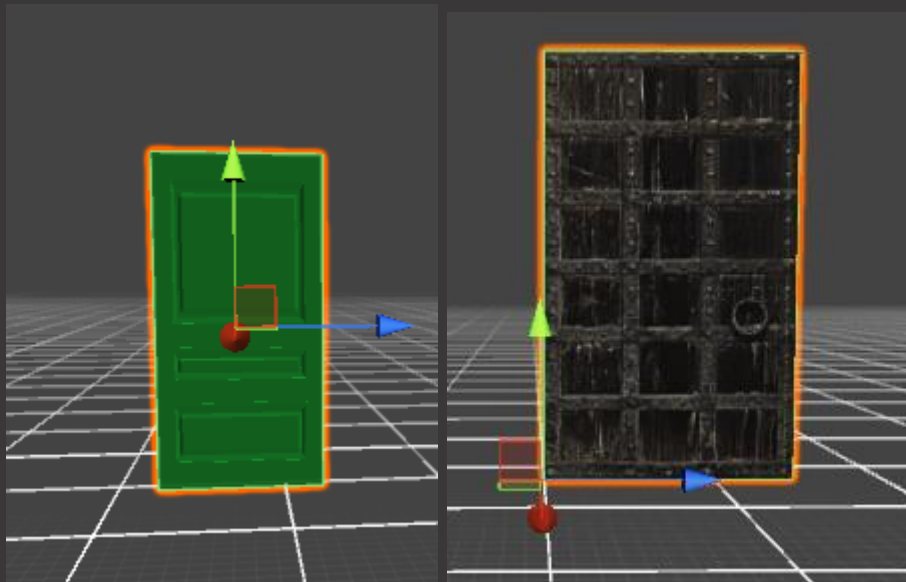
Click on the arrow next to the parent object and then on your door model itself (the child). Here you'll see the 'DoorPro' script component and its UI.



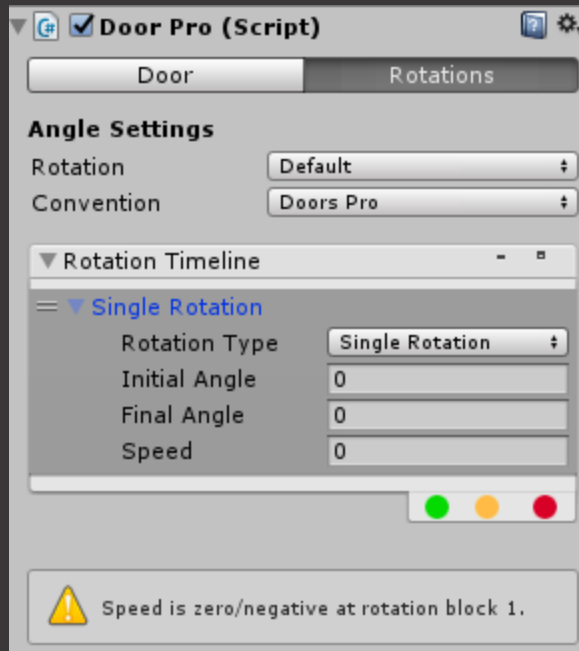
The door script has 2 tabs, the *Door Tab* and the *Rotations Tab*. In the door tab, you need to select what type your door is. Select whether or not the scale correctly matches with Unity3D units, and also select if the pivot position is centered or correctly positioned. To see whether the pivot position is centered or not, make sure the tool handle is placed at the active object's pivot point. You can do this by toggling this:



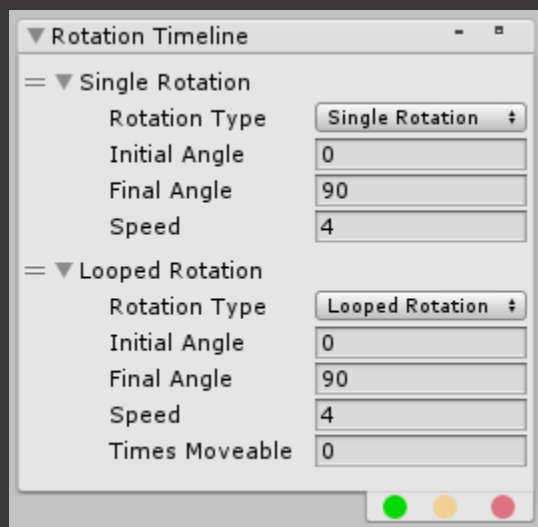
If you selected 'pivot' and the warning on the Door Pro script disappears, look at your door, if the pivot point appears at the center, you need to select 'Centered' in the door settings menu, and if the pivot point seems to be where the hinge should be, select 'Correctly Positioned'. (Centered and correctly positioned respectively in the images below)



Now that the door settings are done, let's move on to the rotations themselves!



In the rotations tab you can control how the door actually behaves. First let's look at the rotation timeline. The rotation timeline will control how your door behaves over time and is made up of several list elements or 'rotation blocks'. Initially there will be no rotation blocks added and the timeline will say 'No rotations added'. You also have 3 buttons on the bottom right, the green button will add a single rotation, the yellow button will remove the selected rotation and lastly the red button will clear the whole rotation timeline. When you add a rotation block, a new list element will appear with a number of options. Usually you'll just need one rotation block but sometimes you might need 2, 3 or even more. Let's take a look at some rotation blocks in depth shall we.



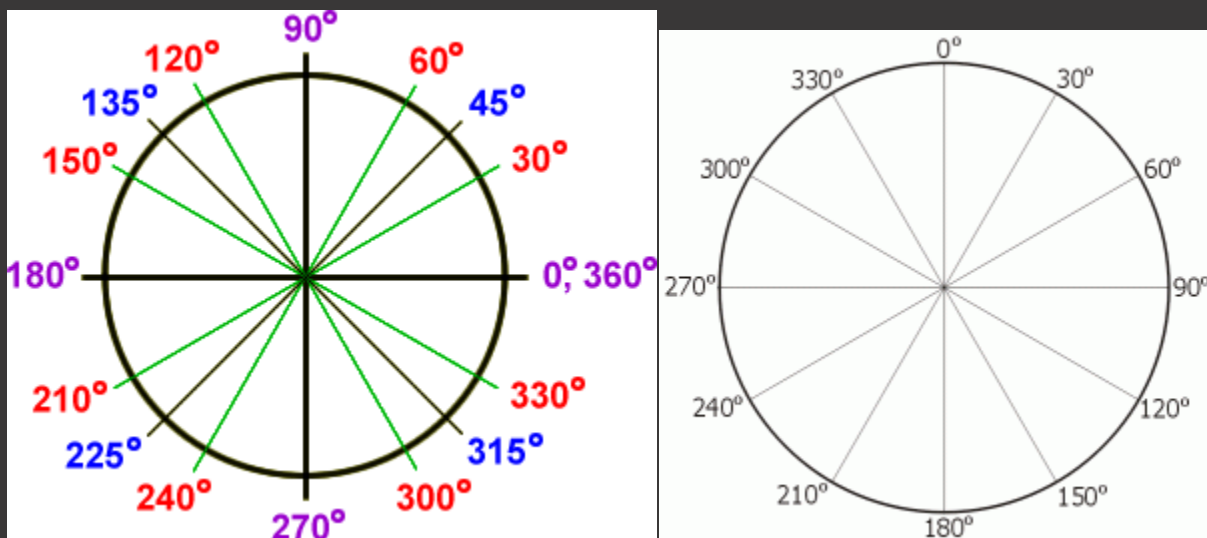
Rotation blocks can be single rotations or looped rotations, you can change which one you want at the top of each block. A single rotation block is a rotation that will only be performed one single time. The single rotation block has several settings. First, you have the option to switch between a single and a looped rotation. You also have 2 angle fields, one to control the initial angle, and one to control the final

angle. In the section below, there is some more information on how these angles are interpreted. Lastly you have a speed variable where you can control how fast the door rotates.

A looped rotation block controls a rotation that gets performed a number of times. This is what you normally would expect a door to do. The settings are mostly the same as with a single rotation block, but now there is also a *Times Moveable* variable where you can set how many times you want the door to rotate. When you set this variable to zero, the door will be able to open and close for an infinite amount of times.

Rotation blocks will be performed chronologically, so if the example image above, the door will first rotate from 0 to 90 degrees, and will then rotate back and forth from 90 degrees to 0 degrees for an infinite amount of time. Rotation blocks are of course not performed automatically, but they are triggered by trigger zones.

Under the rotation block, you have the angle settings. There are 2 angle conventions that you can switch between. The default convention, and the Unity convention respectively. If following the Unity convention, when an object rotates  $+90^\circ$ , it rotates  $90^\circ$  clockwise ( $-90^\circ$  would be  $90^\circ$  counterclockwise) while with the default convention, when you input  $90^\circ$  the door will rotate counterclockwise. You can switch between these two whenever you'd like. Here are some visuals to illustrate this, the first picture is how my tool handles rotations, and the second one is how Unity handles rotations.

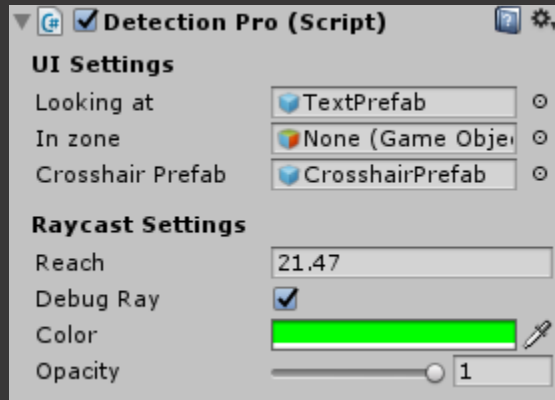


The other settings is about what will happen when performing a rotation that's over 180 degrees. By default Unity will always pick the shortest way when rotating to certain angles, but this is of course not always what you'd want. There is an option in my tool to select which rotation method you prefer.

### The detection script

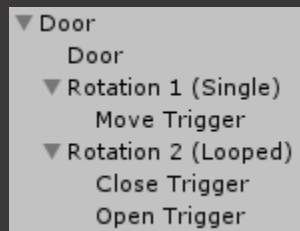
The detection script is located on the player object and casts invisible rays to determine whether or not the player is looking at the door. You'll find 2 sections: the UI settings and the Raycast settings. In the UI Settings section you'll find different settings where you enter text prefabs to show to the player when certain events occur. For example you can display a certain text when the player is looking at a certain object, or you can display text when the player is in a trigger zone. In the Raycast Settings section you

have the option to make the ray appear in play-mode (will be displayed in scene view), this is of course for debugging purposes. You can pick the color of the ray and it's opacity. You can also select how far the player is able to 'reach', you wouldn't want the player to be able to open doors that are hundreds of meters away.



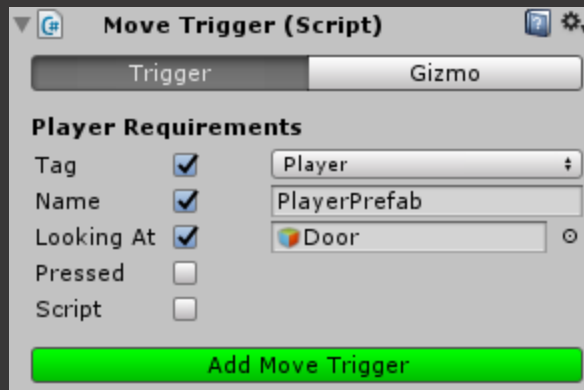
## Trigger zones

After you dragged the door script onto your door, there was automatically created a hierarchy. After you added some rotation blocks, you'll notice that for every rotation block, there is a trigger zone added to that hierarchy.



Trigger zones are zones in the scene that control which rotation gets performed and when it gets performed. Every rotation block gets controlled by at least one trigger zone. If the rotation is a single rotation, one trigger zone is added by default (a move trigger), and when the rotation is a looped rotation, two trigger zones are added by default (an open trigger and a close trigger), you can of course add as many as you'd like. Single rotations only need 1 trigger, a move trigger. A looped rotation requires 2 trigger zones, one to control the 'open' rotation and another to control the 'close rotation'.

Every trigger zones has a trigger script on it, let's take a closer look. Open, close and move triggers all work the same, but as an example, let's look at a move trigger.

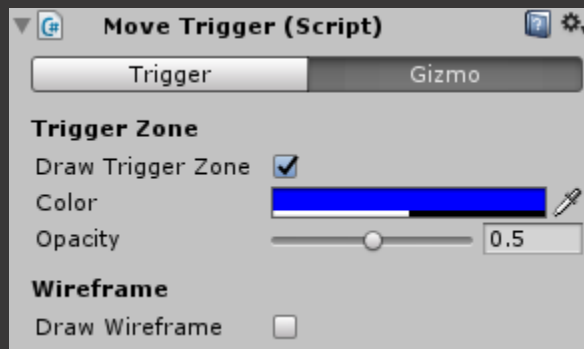


For every trigger zone, you can add requirements that the player should meet in order to be able to open/close the door (perform the rotation block linked to the trigger zone). You have 5 possible requirements and you can combine them as you'd like. When you want to set up a requirement, just click on the toggle and an input field will appear. The tag and name requirements are pretty self-explanatory. The 'looking-at' requirement, has an object field where you can put a gameobject that the player has to look at in order to open the door. The 'pressed' requirement has a field where you need to input a keycode, you can see these codes here:

<https://docs.unity3d.com/ScriptReference/KeyCode.html>

The 'script' requirement has also a string field where you can enter the name of the script that the player has to have on him in order to open the door.

Trigger zones with player requirements like these are very handy. You might need the player to press the key 'E' before the door will open, or you might have a script called *Key* that you only add to players that you want to be able to move the door. You can have no requirements so the door will just automatically open when the player enters the trigger zone, or you can activate all of them and make it really hard for the player to meet all the requirements. Trigger zones are boxes that you can move around and you can change the dimensions. You have a button to add even more trigger zones so you could have several zones control the same rotation. You can visualize these trigger zones in the *gizmo* tab.



You can pick the color, the opacity and you can draw a wireframe around the trigger zone box. After you chose the rotations in the rotation timeline module and you set up the trigger zones how you wanted, you're ready to go.

## FREQUENTLY ENCOUNTERED ISSUES

*Will be updated*

## CONTACT

If you have any questions about the asset, you encounter bugs or issues, you have feature requests, ideas, critiques, poems you've written, life questions,... You can contact me through:

- E-mail: [alexanderameye@gmail.Com](mailto:alexanderameye@gmail.Com)
- Skype: alexander.ameye