Door Object Overview:

This document is designed to provide an explanation of the Door Object and its Door component for designers to be able to add them properly to room objects when designing rooms. Use this document with the Room Object Overview document. Also there will be a Building Door objects document which will go through the process of making Doors from Scratch.

Inspector Overview:

Below is a figure (Figure 1) and table (Table 1) showing and describing the different values that can be set on a door component.

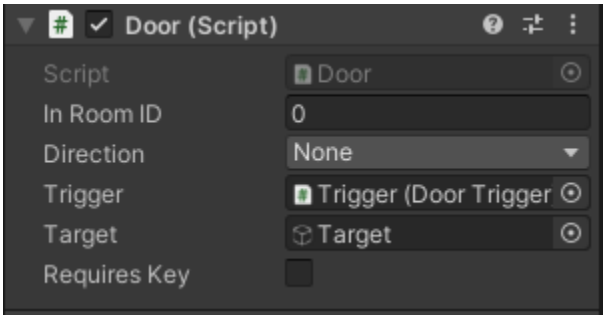

Figure 1: Inspector view of a door component. Values seen here are the defaults

Table 1: Detailed explanation of values seen in the inspector

Field	Explanation
In Room ID (Int)	This is an integer that acts as a distinguishing value that is only used when two or more doors from one room lead to the same other room. If this is the only door from the room this object is placed in that leads to a different room then this value can be left as is. So if all of the doors in a room lead to different rooms, then they can all have a value of 0, it does not matter. However, if for example, two doors in a room both lead to the same room then those two doors need different values for this field, for example 0 and 1. Then in the other room's prefab variant you need to setup the doors so that the 0 and 1 value doors match, so that the player will come through the correct door. Just remember in most cases this value does nothing
Direction (enum)	This is used to determine in which direction the connecting room should be spawned. So for example if this door leads to a room

	that is to the right it needs a value of East. If down: South, Etc. THIS NEEDS TO BE SET TO THE CORRECT VALUE
Trigger (Custom Door Trigger Type)	This is a reference to an object that handles OnTriggerEnter2D callbacks. This should already be setup so it should not be adjusted by the room designer. (See Table 2 for more details)
Target (GameObject)	This is an empty game object that is used for its position. The purpose of this object is to track a position that the player should move to when entering this door. So it is usually placed just in front of the door. This should be set up, so it should not be changed. If you want to change the position move the referenced object (See Figure 2 and Table 2)
Requires Key (Bool)	This is just whether or not the door requires a key to open. If checked the door will not open unless the player holds a key. In the future this will change to an enum that has None, Key, Boss Key. This will allow for different keys to be required.

Door Object Hierarchy Overview:

Here you can find a Figure (Figure 2) showing a default door's object Hierarchy as well as a table (Table 2) describing some of the objects seen therein. The Trigger, Target, and EnemyDoorBlocker should be a part of every door object. These objects are critical to the correct function of a door. However, the values on these objects can be changed, see Table 2 for details. The Left and Right objects are used for sprite renderers and box colliders.

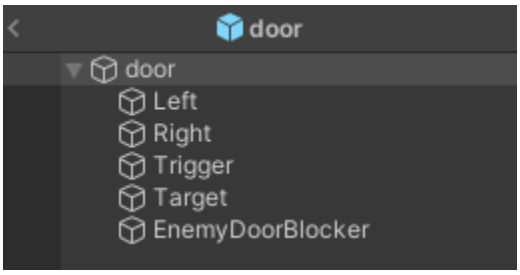

Figure 2: Image of the Door objects Hierarchy

Table 2: Descriptions of each object in the Door object Hierarchy

Object Name	Explanation
Door	The Root object, it contains the Door component as well as some other components that allow the door to operate properly. The only other component on this object that should need adjustment is the animator, and only when creating new doors with different animations.
Left and Right objects	These are objects that hold the sprite renderers and the colliders for those sprites. The animator on the root Door object changes the sprites here and the state of the collider. Feel free to create different amounts of these to fit your door needs, and put whatever default sprite you want on the corresponding sprite renderers. The Box colliders should disabled by default and have the “Is used by composite” box checked.
Trigger	This object has a Door Trigger component and a box collider on it that is used to signal a room transition. The collider should be oriented to cover the width of the door and cover the back part of the door. When the player walks through the door and touches this collider it signals to the Game Controller that the player needs to get moved to a different room.
Target	This is just an empty game object that is used for its position. It should be positioned in front of the door and centered to be in the middle of the door. This object is used in room transitions to determine where the player controller should move to. So you want it to be far enough in front of the door so that the player is not pushed by the door closing behind it. The default position works well for this. Also it needs to be centered along the middle axis of the entire door because this position is used to determine the position of connected rooms. And so if it is offset to the left or right, connected rooms will be offset as well.
EnemyDoorBlocker	This is a GameObject that has a box collider on it. It is used to block enemies from going into the door. It is really only needed for passages, as passages have no blocking collider and enemies could just walk out of the rooms. However, it is easier if all of the door objects have them. The collider should be positioned so that is covers the front part of the door and maybe a bit in front of it.