Here, I will be writing all the steps I took, for posterity. Some of these steps are not related to scene management, but I added them regardless.

## Creating the project

1. Create a new project using the “Universal 2D” template.
   1. Uncheck “Connect to Unity Cloud”.

## Setting up the first scene

*Note*: I will refer to the first scene as “SampleScene”, though you don’t necessarily have to name yours that.

1. Add the player.
   1. Create an empty object.
   2. Create a sprite object as a child of the empty object.
      1. Set the sprite to anything. Here, I just used a simple “Square”, because I didn’t have any assets on me.
   3. Back to the empty parent, add a “Rigidbody 2D” component.
      1. Set its Body Type to “Kinematic”, and check the “Simulated” checkbox.
   4. Still on the empty parent, add a “Box Collider 2D”.
      1. Make sure “Is Trigger” is **unchecked**.
      2. The rigidbody and the collider combined will let the player collide with objects around the scene.
2. Add the door.
   1. Create an empty object.
   2. Create a sprite object as a child of the empty object.
      1. Set the sprite to anything. Again, I used a “Square”, but changed the colour and dimensions. Because the sprite is its own separate child, changing the dimensions doesn’t change the dimensions of the parent.
   3. Back to the empty parent, add a “Rigidbody 2D” component.
      1. Set its Body Type to “Static”.
   4. Still on the empty parent, add a “Box Collider 2D” component.
      1. Check the “IsTrigger” checkbox, so the player can walk into the door, instead of bumping against it.
      2. Scale the collider to the dimensions of the sprite.
3. Add a “Scripts” folder in the file hierarchy.
   1. Create a new MonoBehaviour script inside to handle the scene changing.
      1. I called mine “TriggerSceneChange”, though you can name yours to whatever naming scheme sounds best.
      2. Make sure that it’s a MonoBehaviour script; otherwise, it can’t be a component of the parent door object.
      3. Attach this script to the door parent.
   2. *Note*: I have another script to handle player movement, but I’ve been told that your game already has player movement, so I won’t touch upon it here. The script will still be given to you to access for posterity.
4. After the player and the door have been set up completely…
   1. In the file hierarchy, create a “Prefabs” folder.
   2. Click and drag the player from the hierarchy into the folder. This will create a copy, or “prefab”, of the player for later use. The player icon in the hierarchy should turn blue; this means that it’s now a prefab.
   3. Do the same with the door.

## Setting up the second scene

1. Create a new scene in the “Scenes” folder in the file hierarchy. Here, I’ve named mine “NextScene”, so I will refer to the second scene as such throughout the rest of this tutorial.
2. Add the “Player” prefab and the “Door” prefab to the hierarchy.
   1. While selecting the “Door” prefab, inside the Inspector, write “SampleScene” in the script component’s “Scene Name” parameter, instead of “NextScene”. This will let us go back to the original scene.
3. With the “NextScene” scene open, go to File → Build Profiles. In the window that opens, go to the Scene List tab, and click the “Add Open Scenes” button.
   1. *Note*: During gameplay, scenes can only be loaded from the Scene List. Neglecting to add scenes will throw exceptions.

## Testing your work

1. Go back to “SampleScene”, and start the game. If everything works, you should be able to move the player into the door and load “NextScene”. Then, you should be able to enter that scene’s door and load “SampleScene”.