Making an animated door

Make an animated+lockable door

Animators are useful for many applications beyond animating the player character. They are also a great way to animate parts of your environment of architecture. Lets use the animator to make a lockable/unlockable door.

This tutorial combines previous topics into a single tutorial:

- Keyframe animation
- Animator state machines
- if statements
- Yarn commands

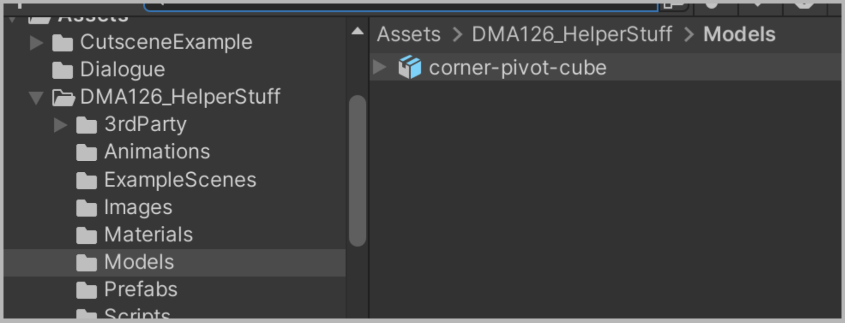
The steps below are a little bit general, but I've also included videos you can follow along with.

Step 1 make your door & set up animator

Video instructions: _door_1_set_up_animation.mp4

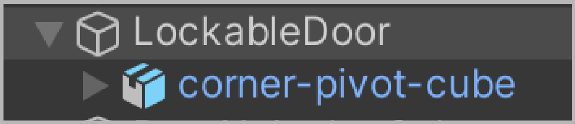
Set up the door object

- Start with an empty game object and name 'LockableDoor'
- Find the model DMA126_HelperStuff / Models / corner-pivot-cube model and drag into the scene,



This is just a resized cube, with its pivot at its lower left corner so it rotates around the hinge, rather than the center of the shape.

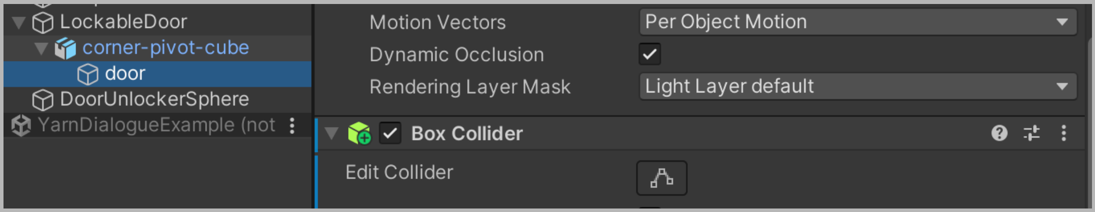
- Child this model to your empty game object, and Set its position to (0,0,0)



LockableDoor
corner-pivot-cube

[More about parent/children/pivots]()

- Add a box collider to the 'door' child object of the model

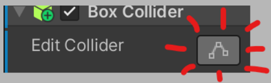


LockableDoor
corner-pivot-cube
door
DoorUnlockerSphere
YarnDialogueExample (not

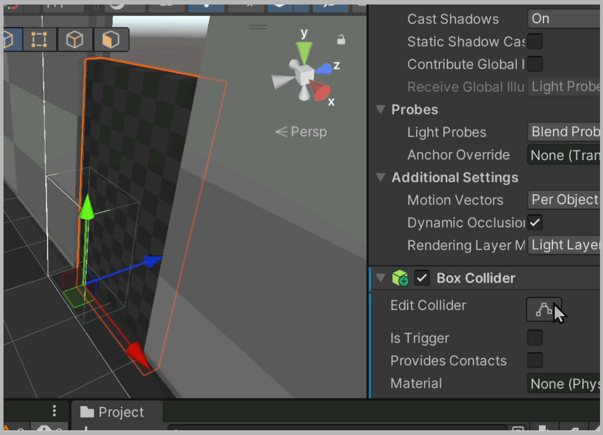
Motion Vectors	Per Object Motion
Dynamic Occlusion	✓
Rendering Layer Mask	Light Layer default

Box Collider
Edit Collider

TIP: You can use the edit collider button to more visually edit the box collider to conform to the door.



Box Collider
Edit Collider

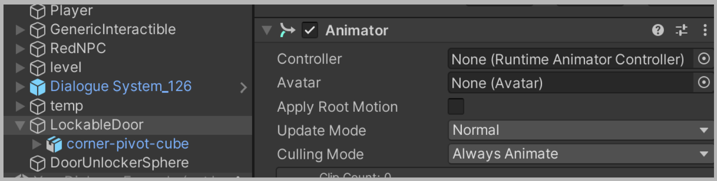


Cast Shadows	On
Static Shadow Cas	
Contribute Global I	
Receive Global Illu	Light Probe
Probes	
Light Probes	Blend Prob
Anchor Override	None (Tran
Additional Settings	
Motion Vectors	Per Object
Dynamic Occlusio	✓
Rendering Layer M	Light Laye
Box Collider	
Edit Collider	
Is Trigger	
Provides Contacts	
Material	None (Phys

Create an open animation

[Refer to the previous tutorial on keyframe animation]()

- Add an animator to the toplevel 'LockableDoor' object.



Player
GenericInteractible
RedNPC
level
Dialogue System_126
temp
LockableDoor
corner-pivot-cube
DoorUnlockerSphere

Animator

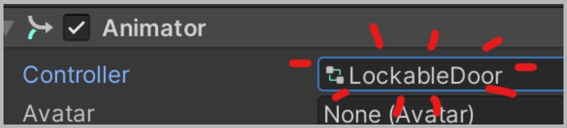
Controller	None (Runtime Animator Controller)
Avatar	None (Avatar)
Apply Root Motion	
Update Mode	Normal
Culling Mode	Always Animate

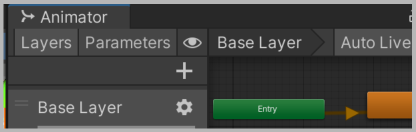
- Add a keyframe at start for child 'corner-pivot-cube' with rotation (0,0,0)
- Add keyframe a little later with the child rotated to something like (0,-110, 0)
- Select the animation clip in your project and disable looping

Set up the animator

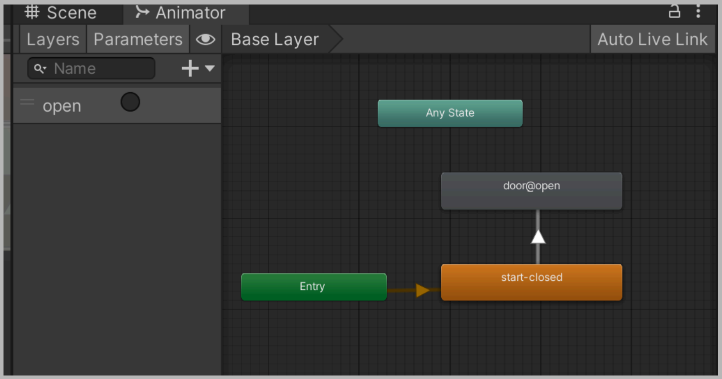
Refer to the previous tutorial on scripting animation

- Open 'animator' window by double clicking on the *Controller* of the animator component

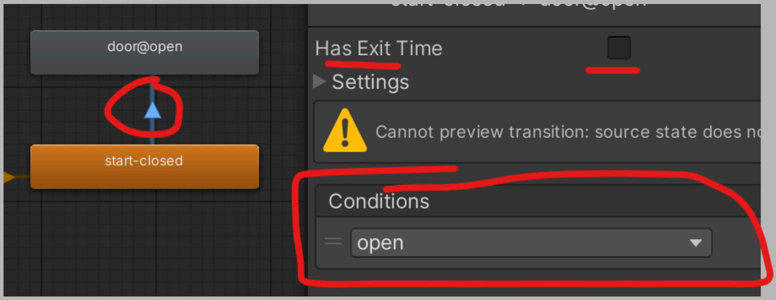




- Create a new animator state called 'start-closed'
- Create a new trigger called 'open'



- Create a transition from 'start-closed' to your open animation with 'open' as the condition, and uncheck 'has exit time'



Variations/Suggestions

There's no reason it has to be a literal door. It could be a giant mouth that opens, a more elaborate mechanical contraption, or a creature that needs to be coaxed out the way.

A complicated ship door

King Zora, slowly scooting away to reveal a passage in Zelda Ocarina of Time

Step 2 create a script 'LockableDoor' to control opening/closing/locking

We will write a script, implementing the *interface* IPlayerInteractible (adding some specifically named public methods to the script). Doing this will let our player interact with the door in the same it interacts with YarnNPC and GenericActionInteractibles (white text appears letting up know what action can be taken, pressing space to take the action)

Here is an official tutorial on C# interfaces, and one from Brackeys

Video Instructions: _door_2_create_script.mp4

Here is the complete script.

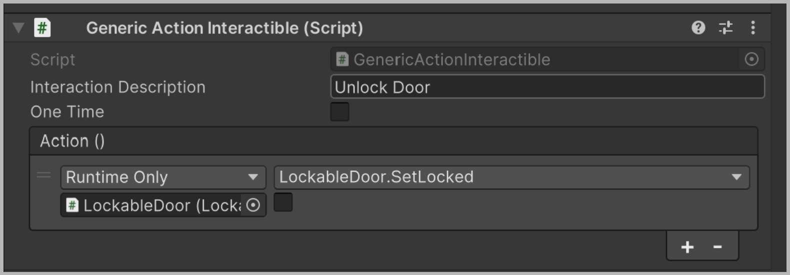
```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class LockableDoor : MonoBehaviour, IPlayerInteractible
{
    Animator animator;
    public bool isLocked = false;

    void Start()
    {
        animator = GetComponent<Animator>(); //get the attached animator
    }

    //This method 'SetLocked' will let us unlock from TriggerZones, Yarn Commands, etc...
    [YarnCommand("SetLocked")] //<- - - This line is necessary to execute this code from yarn.
    public void SetLocked(bool nowLocked)
    {
        this.isLocked = nowLocked;
    }

    public bool GetInteractionAllowed()
    {
        //Always show the interaction message (lock interaction will just do nothing)
        return true;
    }

    public string GetInteractionDescription()
    {
        if (isLocked)
        {
            //show 'Locked' for the interaction message if the door is locked.
            return "Locked";
        }
        else
        {
            //show 'Open door' for the interaction message if the door is not locked.
            return "Open door";
        }
    }

    public void Interact()
    {
        //only play the open animation if the door is unlocked.
        if (isLocked == false)
        {
            animator.SetTrigger("open");
        }
    }
}
```

Using the door in your game

Video Instructions: _door_3_controlling_lock_state.mp4

GenericActionInteractible / Trigger Zone.

How to set up a trigger zone
How to set up a generic action interactible


Generic Action Interactible (Script)

Script: GenericActionInteractible
Interaction Description: Unlock Door
One Time: ☐
Action ()
Runtime Only | LockableDoor.SetLocked
LockableDoor (Lock: ☐

Yarn command

Please check the Official documentation for Yarn commands (calling Unity C# code from a Yarn script), but basically:
You need this before the method you want to call (this is already set up in the completed example script)

```
[YarnCommand("SetLocked")]
```

To execute the command from the yarn script, you would use this line to unlock

```
<<SetLocked LockableDoor false>>
```

You could also use the DoCustomCommand126 helper if you wanted to avoid editing/creating more scripts.

Challenges/additions

- Create a more interesting "door" and open animation
- Add a "rattle" animation if the door is locked, and the player tries to open anyway
- Make the door re-closeable.
- Play sounds when door is open/closed/rattled/unlocked
- Only open the door if multiple conditions are met, (e.g. have collected all of a particular collectible)
- Could you unlock a door in another scene? (hint: you probably need to use static variables)

Here is the start of a script that you could use as a starting point for unlocking a door after multiple collectibles.
It's designed to be attached to your player, and "collect" certain objects on trigger collisions.
It's actually just a lightly modified version of the roll-a-ball controller.

```
public class CollectibleUnlocker : MonoBehaviour
{
    private int count = 0;
    public TextMeshProUGUI pickUpCountText;

     //Change this to decide how many collectibles are needed
    int nPickupsRequired = 2;

    //You need to add another variable here of type LockableDoor (for the simplest solution)
    //...

    void Start()
    {
        UpdatePickupCount();
    }


    void OnTriggerEnter(Collider other)
    {
        //You might want to change this condition,
        //either using a different tag from "PickUp",
        //checking other.name == "something"
        //or other.GetComponent<SomeScript>() != null
        //depending on your collectible
        if (other.gameObject.CompareTag("PickUp"))
        {
            other.gameObject.SetActive(false);
            count = count + 1;
            UpdatePickupCount();
        }
    }

    void UpdatePickupCount()
    {
        if (count >= nPickupsRequired)
        {
            //unlock here!
        }

        if (pickUpCountText != null)
        {
            pickUpCountText.text = "Count: " + count.ToString();
        }

    }
}
```