

Intro to Scripting part 4, 5

Parenting

KeyCode

Prefabs

Instantiate

Transformations

Collisions

Tags

CAVE testing

Teleport

Collisions

Dynamic parenting

Constraints

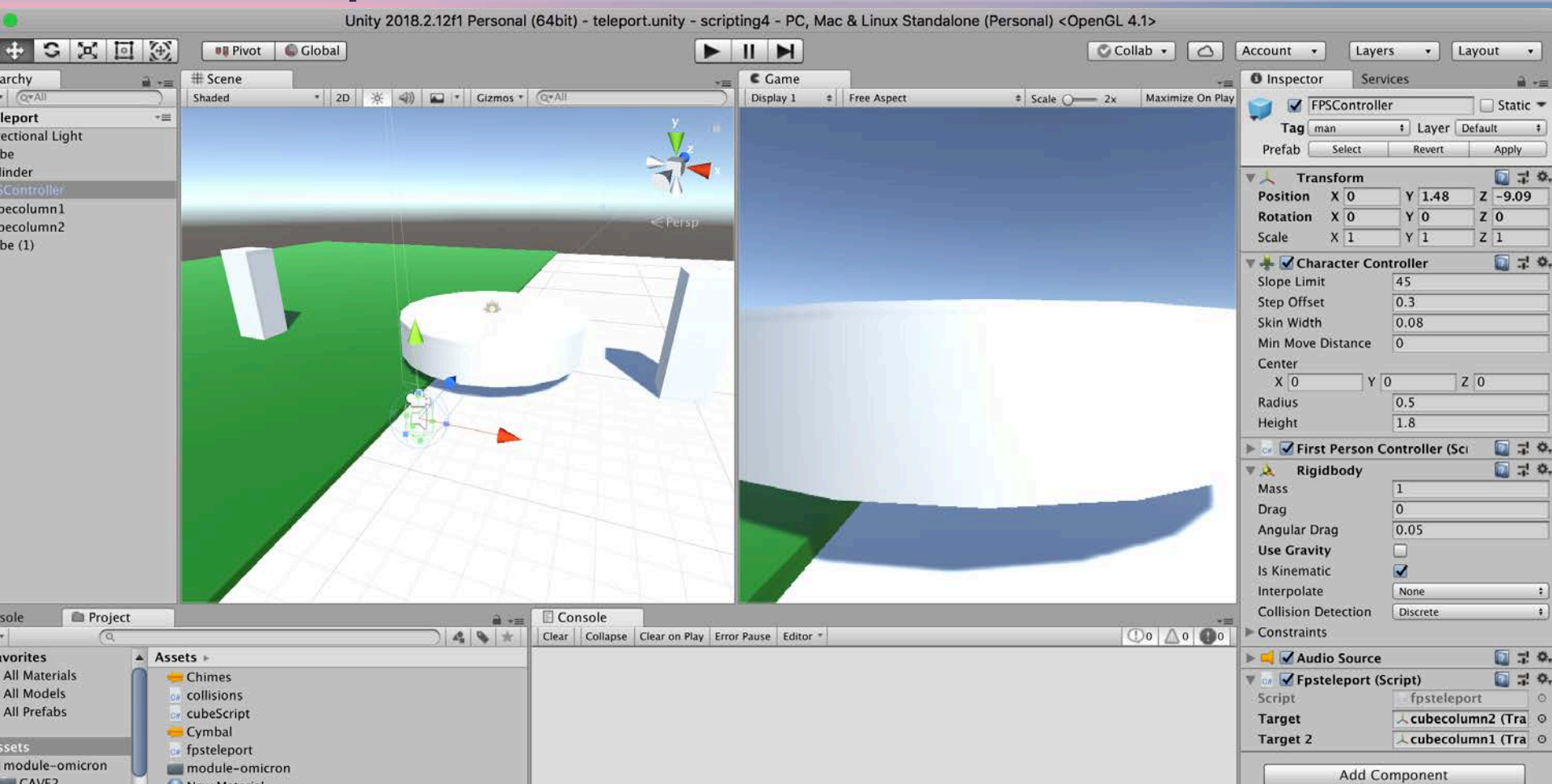
Basic teleport

Unity 2018.2.12f1 Personal (64bit) - teleport.unity - scripting4 - PC, Mac & Linux Standalone (Personal) <OpenGL 4.1>

Inspector Services

- FPSController Static
 - Tag: man Layer: Default
 - Prefab: Select Revert Apply
- Transform
 - Position: X 0 Y 1.48 Z -9.09
 - Rotation: X 0 Y 0 Z 0
 - Scale: X 1 Y 1 Z 1
- Character Controller
 - Slope Limit: 45
 - Step Offset: 0.3
 - Skin Width: 0.08
 - Min Move Distance: 0
 - Center: X 0 Y 0 Z 0
 - Radius: 0.5
 - Height: 1.8
- First Person Controller (Sci)
- Rigidbody
 - Mass: 1
 - Drag: 0
 - Angular Drag: 0.05
 - Use Gravity:
 - Is Kinematic:
 - Interpolate: None
 - Collision Detection: Discrete
- Constraints
- Audio Source
 - Fpsteleport (Script)
 - Script: fpsteleport
 - Target: cubecolumn2 (Tra)
 - Target 2: cubecolumn1 (Tra)

Add Component



The screenshot displays the Unity 2018.2.12f1 Personal interface. The main window is split into a Scene view on the left and a Game view on the right. The Scene view shows a 3D environment with a white character controller on a white grid floor, a white cylinder, and a green plane. The Game view shows the character controller from a first-person perspective. The Inspector panel on the right shows the following components and their settings:

- FPSController** (Static): Tag: man, Layer: Default.
- Transform**: Position: X 0, Y 1.48, Z -9.09; Rotation: X 0, Y 0, Z 0; Scale: X 1, Y 1, Z 1.
- Character Controller**: Slope Limit: 45; Step Offset: 0.3; Skin Width: 0.08; Min Move Distance: 0; Center: X 0, Y 0, Z 0; Radius: 0.5; Height: 1.8.
- First Person Controller (Sci)** (disabled).
- Rigidbody**: Mass: 1; Drag: 0; Angular Drag: 0.05; Use Gravity: ; Is Kinematic: ; Interpolate: None; Collision Detection: Discrete.
- Constraints** (disabled).
- Audio Source**: Fpsteleport (Script). Script: fpsteleport; Target: cubecolumn2 (Tra); Target 2: cubecolumn1 (Tra).

The Hierarchy panel on the left shows the scene's object structure, including a Directional Light, a Cylinder, and several columns. The Console panel at the bottom shows the current state of the game, including the FPS and character controller settings.

Basic teleport

Create two vertical cubes and position them on different platforms

The cubes will serve as teleport portals

Rename cubes:

- teleport1
- teleport2

Add game tags to each cube (ex. “column”, “mango”)

Add material to each platform to visually differentiate the locations

Create new script “fpsteleport” and assign it to FPSController

Basic teleport

```
public class fpsteleport : MonoBehaviour {  
    public Transform target = null;  
  
    private void OnTriggerEnter(Collider other)  
    {  
  
        if(other.gameObject.tag=="column")  
            {  
                this.transform.position = target.position;  
            }  
    }  
    void Update ()  
    {  
    }  
}
```

Basic teleport

Check "Is Trigger" button on • teleport1 object in Inspector.

Check "Is Trigger" button on • teleport2 object in Inspector.

Add destination location value to target variable (drag and drop teleport 2 object to variable field in Inspector)

Basic teleport

Add variable

```
public Transform target2 = null;
```

if statement for teleport from location 2 back to location 1

add two variables to identify if the teleport has been made

```
bool firstJump = false;
```

```
bool secondJump = false;
```

teleport

```
public class fpsteleport : MonoBehaviour {  
    public Transform target = null;  
    public Transform target2 = null;  
    bool firstJump = false;  
    bool secondJump = false;  
  
    private void OnTriggerEnter(Collider other) {  
        if(other.gameObject.tag=="column" && firstJump==false && secondJump==false )  
            {  
                this.transform.position = target.position;  
                firstJump = true;  
            }  
    }  
}
```

teleport

```
if (other.gameObject.tag == "mango" && firstJump==false && secondJump==false )  
{  
    this.transform.position = target2.position;  
    secondJump = true;  
} }
```


Basic teleport

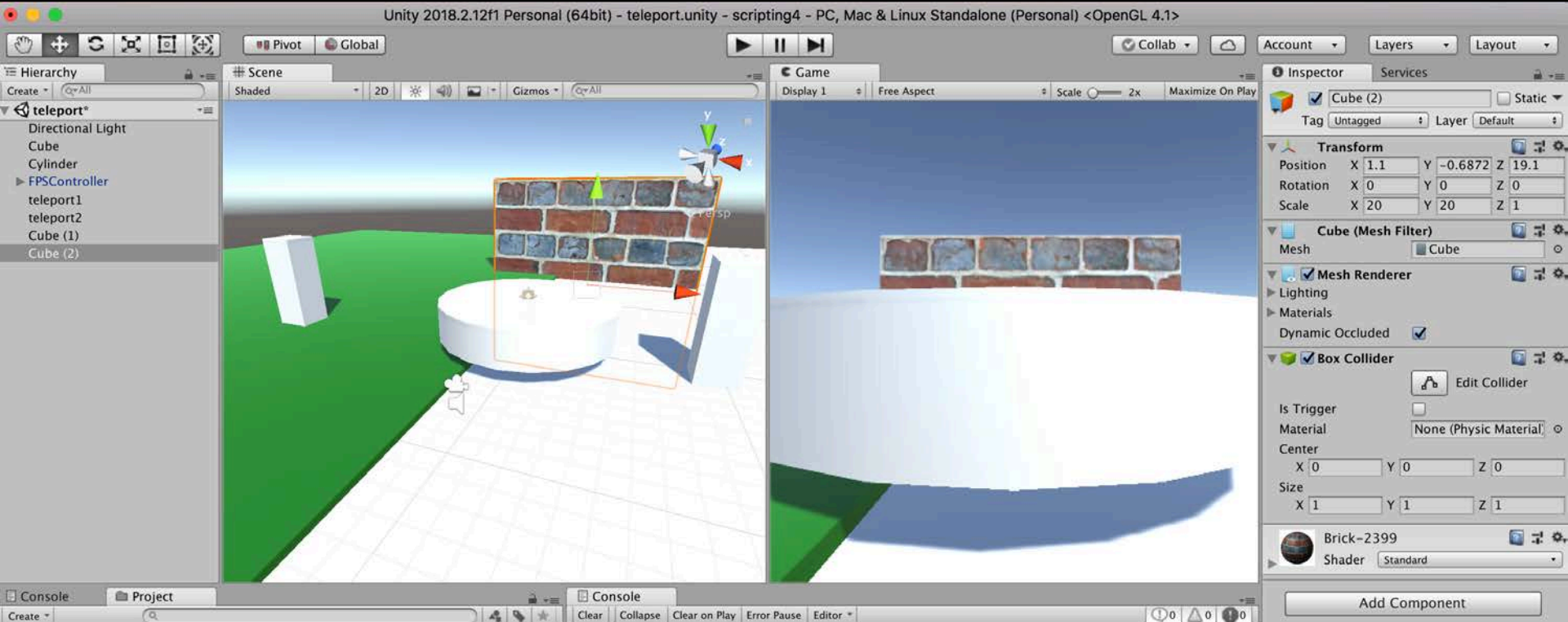
Add function `OnTriggerExit` to set both jump variables to their initial pre-jump state

teleport

```
private void OnTriggerExit(Collider other)
{
    if (other.gameObject.tag == "column")
    {
        secondJump = false;
    }
    if (other.gameObject.tag == "mango")
    {
        firstJump = false;
    } }

void Update () {
}}
```

Teleport, collisions, constraints



Teleport, collisions, constraints

CAVE2 Testing 2

Environment –Geometry- Textures/materials

Interaction

Navigation

Start thinking about:

- Audio (FXs, bg, narration, music, etc.)
- Special FXs (particles, environmental, magic, transformations, etc.)
- Interaction and navigation details

Teleport, collisions, constraints

Teleport

Constraints within geometry

Change material/color

Add force

i/o mouse buttons, keyboard

Collisions

Play sounds

Emit particles

Transformations: Rotating/scaling/moving objects