UNITY INTRODUCTION EX03

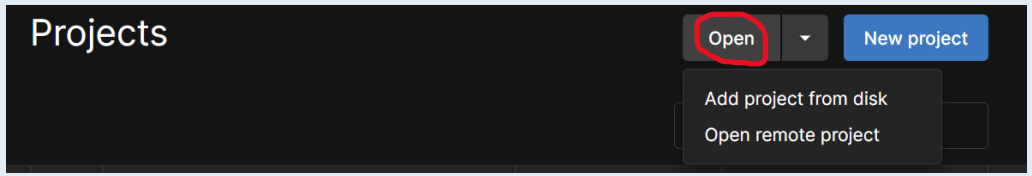
Corso Realtà Virtuale 2023/2024

susanna.brambilla@unimi.it

WITH UNITY V2022.3.5

SET UP CHARACTER

1. Download the Alien folder you can find in Ex03-02 Github folder and open create a new project from Unity Hub:



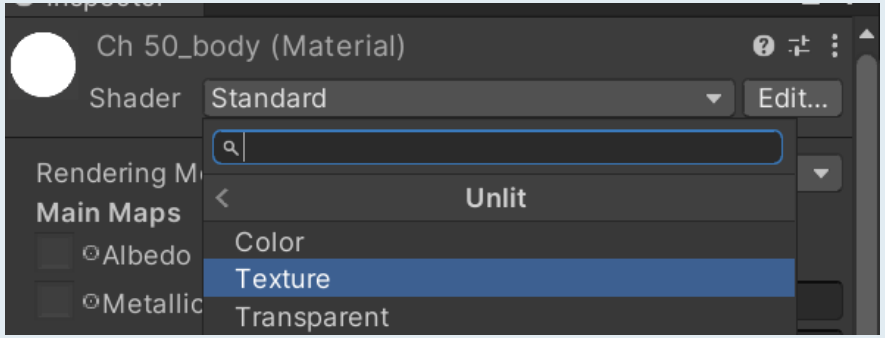
2. Drag the Alien.fbx, and its .png textures into the Assets folder

3. Select the .fbx and select Rig tab in the Inspector and Animation Type -> Humanoid, then Apply

4. In Materials tab select Use Embedded Materials in Location and click Extract Materials button, then Apply

ASSIGN TEXTURES

1. Select the first generated material *Ch50_body* and in Shader select Unlit > Texture:



2. Select the *Ch50_body* .png texture by dragging it into texture space:

3. Repeat for *Ch50_body1* material

CONTROL CHARACTER: SET UP

1. Add the Alien in the Hierarchy, and Add Component:
    1. Rigidbody and set the Mass to 60
    2. Capsule Collider and Edit Collider in order to contain the character

2. In Project window, click on the right mouse button > Create > C# Script and rename it ‘Player Controller’ and open it

3. Add a reference to character’s Rigidbody component with:

```
private Rigidbody rb_player;
```

4. Get the component in Start() with:

```
rb_player = GetComponent<Rigidbody>();
```

CONTROL CHARACTER: MOVEMENT

1. Add a public float variable for movement speed and call it speed (N.B. public allows to modify the variable in the Inspector):

```
public float speed;
```

2. Add a private Vector3 variable for movement direction:

```
private Vector3 movementInput;
```

3. Initialize the movementInput vector in the Update function:

```
movementInput = new Vector3(Input.GetAxis("Horizontal"), 0f, Input.GetAxis("vertical"));
```

4. We want to use the movementInput vector as the velocity of the rigidBody:

```
rb_player.velocity = movementInput;
```

CONTROL CHARACTER: MOVEMENT

5. To set the speed of the movement we must multiply the x and z values of the Vector3 for the speed variable; the speed variable can be changed in the Inspector:

```
movementInput = new Vector3(Input.GetAxis("Horizontal")*speed, 0f, Input.GetAxis("Vertical")*speed);
```

N.B. All the physics calculations should be done in the FixedUpdate. In FixedUpdate the update is done at a fixed framerate:

- Let's create a FixedUpdate function
- Move the movement of the rigidBody inside it

```
private void FixedUpdate()
{
    rb_player.velocity = movementInput;
}
```

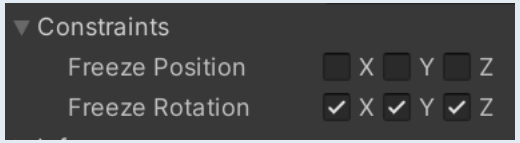
CONTROL CHARACTER: MOVEMENT

6. In the Hierarchy, add a plane with: click right mouse button > Create > 3D object > Plane, and set its position in (0, 0, 0)

7. Add the PlayerControl script as component of the Alien object

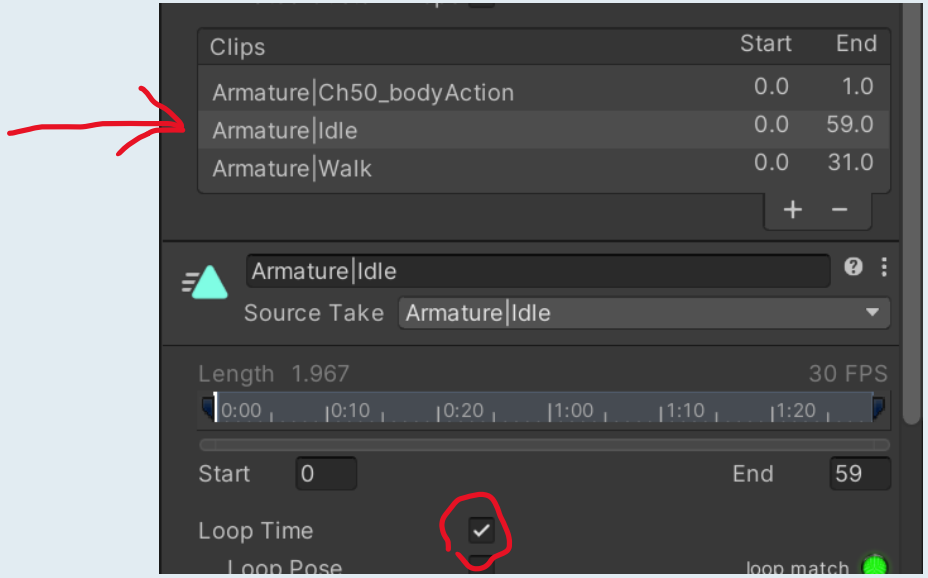
8. Press play and try to move your character with arrows/WASD

9. Add rigidbody constraints to prevent your character from rotating:



ANIMATION 1/3

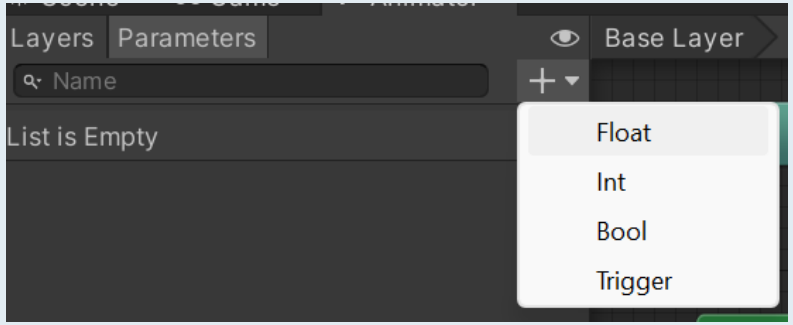
1. In the Animation tab, select the Idle animation in Clips and check Loop Time:



2. Repeat for the Walk animation and Apply

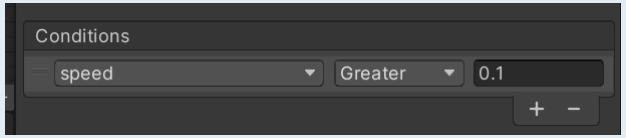
ANIMATION 2/3

3. In the Project window right click to add a New Animation Controller
4. Double click on the icon and open it
5. Drag the *Idle* and *Walk* Alien.fbx animations inside the window
6. Select the Idle state and click with the right mouse button > Make Transition and link it to Walk state
7. Click with the left mouse button on the transition from Idle to Walk and on the left select Parameters
8. Click on + > float and rename the parameter ‘speed’:

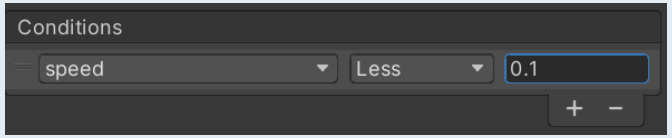


ANIMATION 3/3

9. In the right panel, add a new condition with + in Conditions, select speed Greater 0.1:

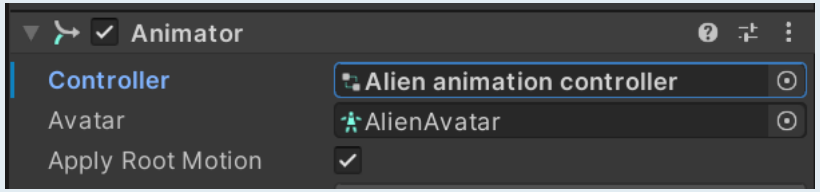


10. Select the Walk state and click with the right mouse button > Make Transition, and link it to the Idle state
11. Click with the left mouse button on the transition from Walk to Idle and on the left select Parameters
12. In the right panel, add a new condition with + in Conditions, select speed Less 0.1:



CONTROL CHARACTER: ANIMATION

1. Drag the Alien animation controller into Controller field in Alien Animator component



2. To add the animation to the movement we need an Animator variable:

```
private Animator an_player;
```

3. Get the character's Animator component in Start():

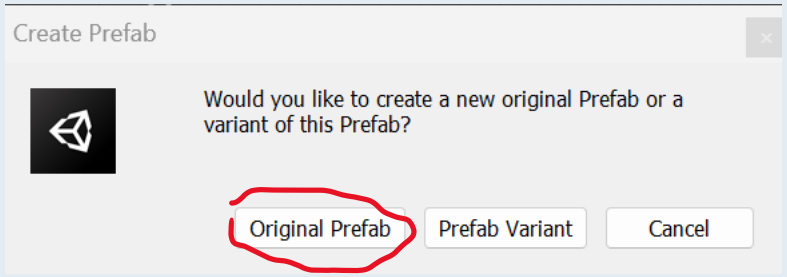
```
an_player = GetComponent<Animator>();
```

4. In FixedUpdate() set the 'speed' parameter as the magnitude of the rigidbody velocity

```
an_player.SetFloat("speed", rb_player.velocity.magnitude);
```

CREATE A PREFAB AND A SPAWN BUTTON

1. Drag the Alien GO in the scene into the Project panel and create an Original Prefab

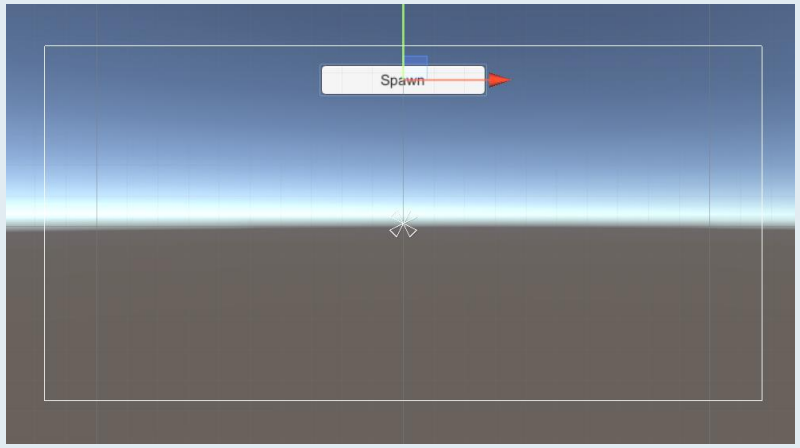


2. Delete the Alien in the scene by clicking on it in the Hierarchy and press [canc]
3. Create a new Canvas in the Hierarchy: click right mouse button > UI > Canvas
4. In Canvas component of Canvas select:
    1. Render Mode : Screen Space – Camera
    2. Render Camera : Main Camera

CREATE A PREFAB AND A SPAWN BUTTON

5. Select the Canvas and: click right mouse button > Create > UI > Button to create a button as Canvas's children

6. Select the Button object and clicking on the arrow in the left you will see its Text children, select it
7. In the Inspector, change Text from 'Button' to 'Spawn'
8. Move the Button on the top area of the Canvas



SPAWN A PREFAB

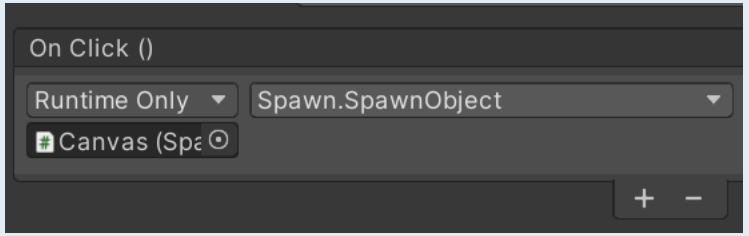
1. Create a new script in the project panel and call it ‘Spawn’
2. Declare a public GameObject variable to indicate the object to spawn:

```
public GameObject objectToSpawn;
```

3. Create a new function called *SpawnObject* and instantiate a new object with:

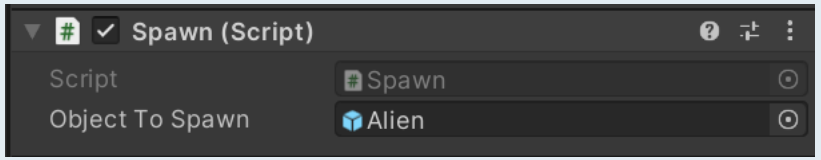
```
public void SpawnObject()
{
    GameObject newObject;
    newObject = Instantiate(objectToSpawn, new Vector3(0, 0, 0), new Quaternion(0, 0, 0, 0));

}
```

4. Attach the script to the Canvas object
5. Select the Button object and, in Button’s On Click () properties select + and Canvas > Spawn > SpawnObject

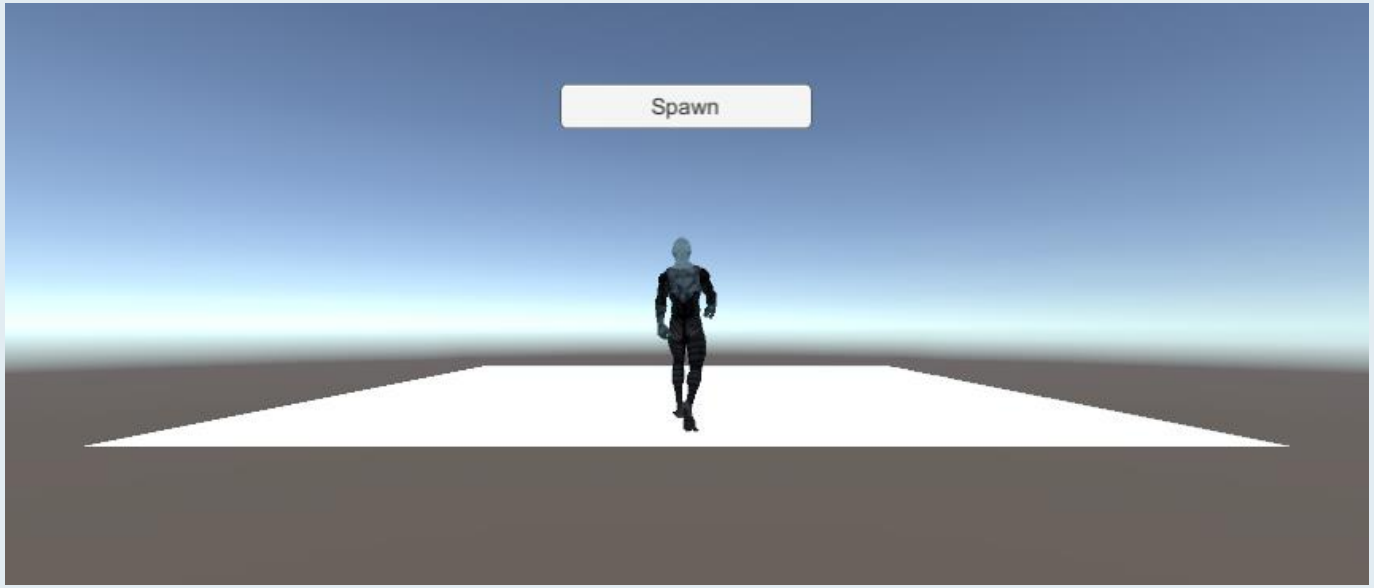


SPAWN A PREFAB

6. Select Canvas object and, in Spawn component, drag the Alien prefab in Object to Spawn field



7. Press Play:



ADD CONTROLS AND REFINE

Controls:

- New documentation for player input actions: Unity - Scripting API: Input (unity3d.com)

Refine:

- Rigidbody: Unity - Scripting API: Rigidbody (unity3d.com)
  - Add force (to jump): Unity - Scripting API: Rigidbody.AddForce (unity3d.com)
- Quaternion (to change rotation): Unity - Scripting API: Quaternion (unity3d.com)
- …