Making a movement control asset compatible with an online game project; short final project report

Ilia Chichkanov

Learning assignment
Online Game Programming, Risto Koskenkorva
22.05.2022
Game Development

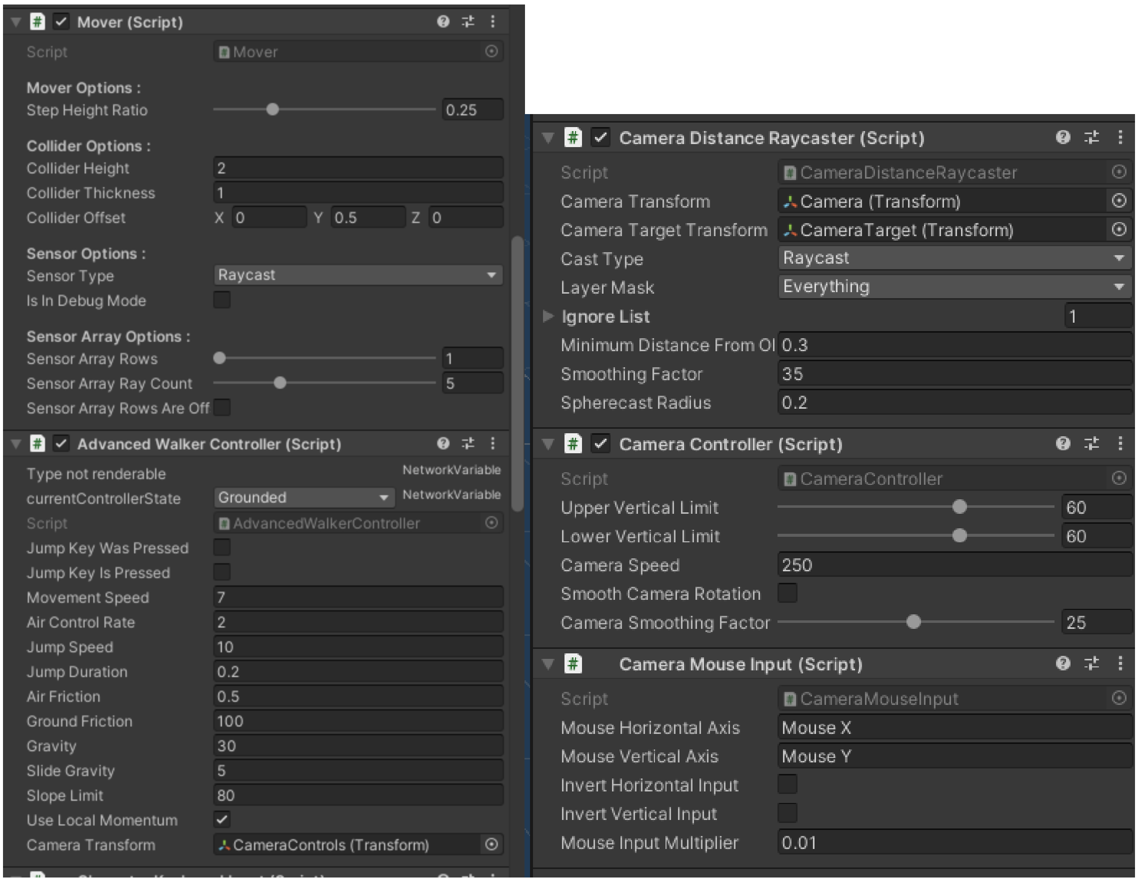
Jyväskylän ammattikorkeakoulu
University of Applied Sciences

Contents

1 Player prefab asset

As a base for the player prefab in the project, the asset from the Character Movement Fundamentals package was used. It includes all required scripts and files needed for the project, such as movement controls, animations, audio and 3D models. The key challenge to make the asset compatible with an online game project was to make its scripts working on the client's side of the game. Therefore, movement, audio and animation scripts were revised.

2 Movement controls



Mover (Script)
Script: Mover
Mover Options :
Step Height Ratio: 0.25
Collider Options :
Collider Height: 2
Collider Thickness: 1
Collider Offset: X 0 Y 0.5 Z 0
Sensor Options :
Sensor Type: Raycast
Is In Debug Mode: □
Sensor Array Options :
Sensor Array Rows: 1
Sensor Array Ray Count: 5
Sensor Array Rows Are Off: □

Advanced Walker Controller (Script)
Type not renderable — NetworkVariable
currentControllerState: Grounded — NetworkVariable
Script: AdvancedWalkerController
Jump Key Was Pressed: □
Jump Key Is Pressed: □
Movement Speed: 7
Air Control Rate: 2
Jump Speed: 10
Jump Duration: 0.2
Air Friction: 0.5
Ground Friction: 100
Gravity: 30
Slide Gravity: 5
Slope Limit: 80
Use Local Momentum: ☑
Camera Transform: CameraControls (Transform)

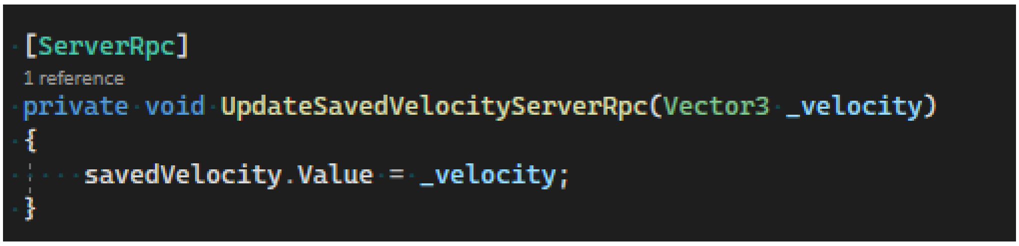
Camera Distance Raycaster (Script)
Script: CameraDistanceRaycaster
Camera Transform: Camera (Transform)
Camera Target Transform: CameraTarget (Transform)
Cast Type: Raycast
Layer Mask: Everything
Ignore List: 1
Minimum Distance From Ol: 0.3
Smoothing Factor: 35
Spherecast Radius: 0.2

Camera Controller (Script)
Script: CameraController
Upper Vertical Limit: 60
Lower Vertical Limit: 60
Camera Speed: 250
Smooth Camera Rotation: □
Camera Smoothing Factor: 25

Camera Mouse Input (Script)
Script: CameraMouseInput
Mouse Horizontal Axis: Mouse X
Mouse Vertical Axis: Mouse Y
Invert Horizontal Input: □
Invert Vertical Input: □
Mouse Input Multiplier: 0.01

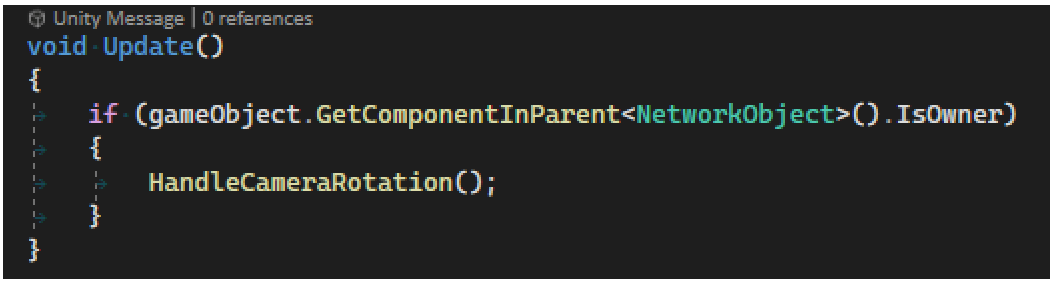
Character movement controls is quite advanced and consists of several scripts linked to each other. Thus, Advanced Walker Controller responsible for transforming player input into speed and player states (Grounded, Sliding, Falling, Rising, Jumping). Mover gets savedVelocity variable form the script and makes the player moving accordingly (it also handles all physics, collision detection and ground detection).

In Advanced Walker Controller the value savedVelocity was changed to be a network variable. The value is controlled by the server and clients are required to send ServerRpc to change the value.



```
[ServerRpc]
1 reference
private void UpdateSavedVelocityServerRpc(Vector3 _velocity)
{
    savedVelocity.Value = _velocity;
}
```

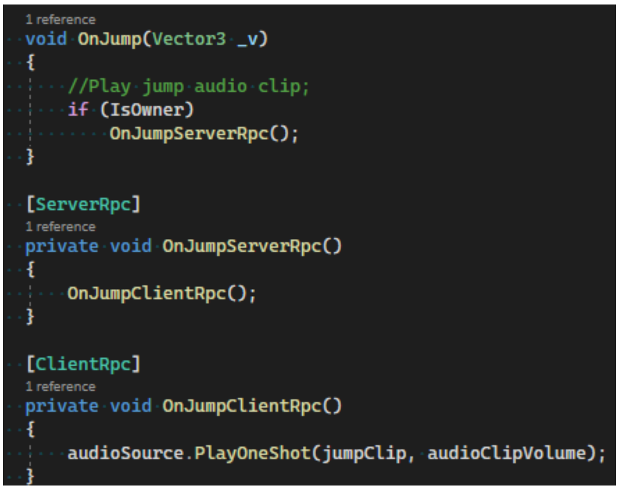
Camera Controller script rotates a player based on the mouse input. The only required change for Camera Controller was to add IsOwner check into the Update() method the following way:



```
Unity Message | 0 references
void Update()
{
    if (gameObject.GetComponentInParent<NetworkObject>().IsOwner)
    {
        HandleCameraRotation();
    }
}
```

3 Animations and audio

In order to make animations working, Netcode only requires the Network Animator script to be attached to a game object. However, regarding audio, Audio Control script had to be changed. Thus, IsOwner checks and combinations of ServerRpc and ClientRpc were added to PlayFootstepSound(), OnLand() and OnJump() methods the following way.



```
1 reference
void OnJump(Vector3 _v)
{
    //Play jump audio clip;
    if (IsOwner)
        OnJumpServerRpc();
}

[ServerRpc]
1 reference
private void OnJumpServerRpc()
{
    OnJumpClientRpc();
}

[ClientRpc]
1 reference
private void OnJumpClientRpc()
{
    audioSource.PlayOneShot(jumpClip, audioClipVolume);
}
```

References

https://assetstore.unity.com/packages/tools/physics/character-movement-fundamentals-144966

https://github.com/AliasNe0/OGP-Final-Project