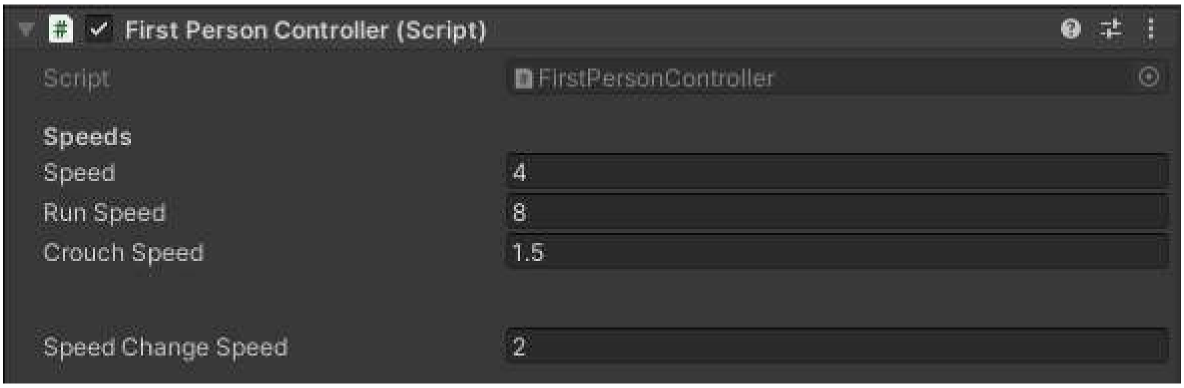
sdf first person controller

by:sdf studios

This is the documentation for using the sdf first person controller which covers the basics of controller scripting, recommendations, and so on. If you encounter errors, have questions or have tips on how to improve the controller, you can write to us at sdfstudiosdev@gmail.com

First person Controller

Speeds

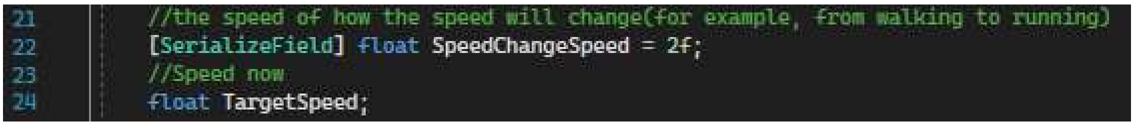


First Person Controller (Script)

Script	FirstPersonController
Speeds	
Speed	4
Run Speed	8
Crouch Speed	1.5
Speed Change Speed	2

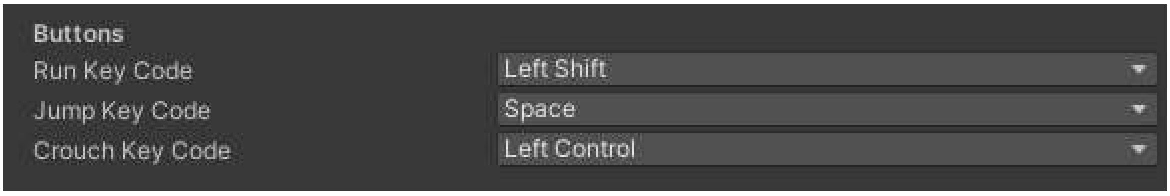
These variables determine the speed of the player during a certain action.

Speed Change Speed - This variable affects how long the speed changes.
The speed does not change abruptly. It will gradually increase as in reality. If you start running, the controller will not immediately run at the set speed, but will gradually accelerate until it reaches the run speed. If you want this smooth speed change you can set this variable to 0 . to get the speed that the player currently has, you need to get the value of TargetSpeed.



```
    //the speed of how the speed will change(for example, from walking to running)
    [SerializeField] float SpeedChangeSpeed = 2f;
    //Speed now
    float TargetSpeed;
```

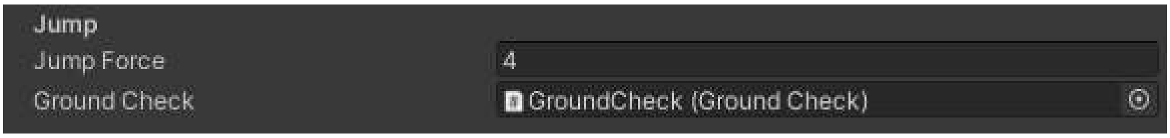
Buttons



Buttons	
Run Key Code	Left Shift
Jump Key Code	Space
Crouch Key Code	Left Control

indicates which button must be pressed in order for the player to perform the desired action

Jump

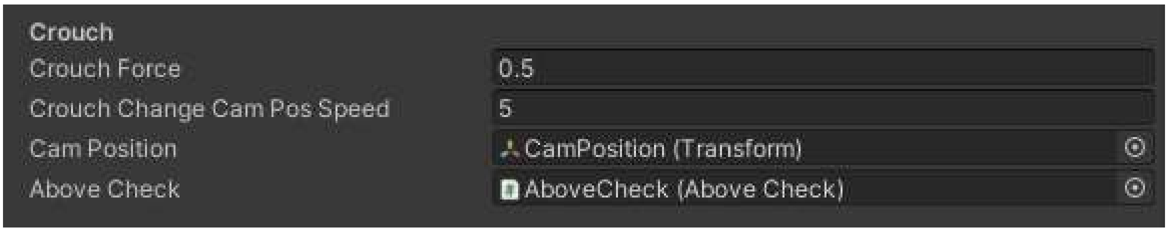


Jump	
Jump Force	4
Ground Check	GroundCheck (Ground Check)

Jump Force - jump power. the more it costs the higher the player will jump

Ground Check -link to a script that checks if there is ground above the player so that the player does not jump an unlimited number of times

Crouch



Crouch	
Crouch Force	0.5
Crouch Change Cam Pos Speed	5
Cam Position	CamPosition (Transform)
Above Check	AboveCheck (Above Check)

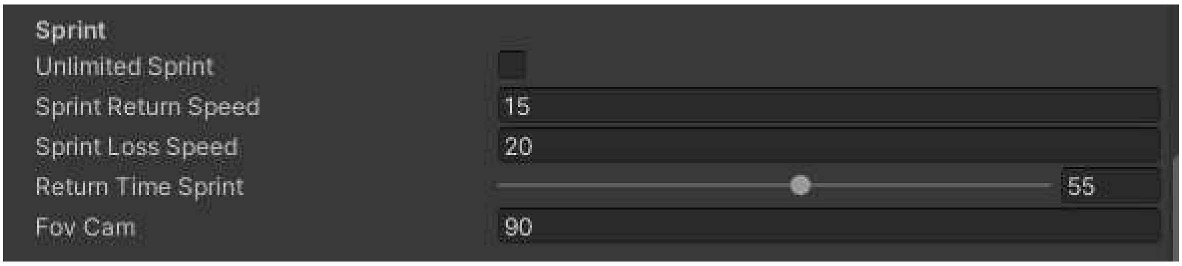
Crouch Force - a variable that determines how far the player's camera goes down. she also wags at the collider

Crouch Change Cam Pos Speed - works the same as SpeedChangeSpeed. Depends on how fast to let go or raise the player's camera.

Cam Position - Camera start position

Above Check - link to a script that checks if there is any object above the player so that he does not stand up when there is something above him and does not get stuck

Sprint



Sprint	
Unlimited Sprint	□
Sprint Return Speed	15
Sprint Loss Speed	20
Return Time Sprint	55
Fov Cam	90

Unlimited Sprint - checks if the player can run without restrictions

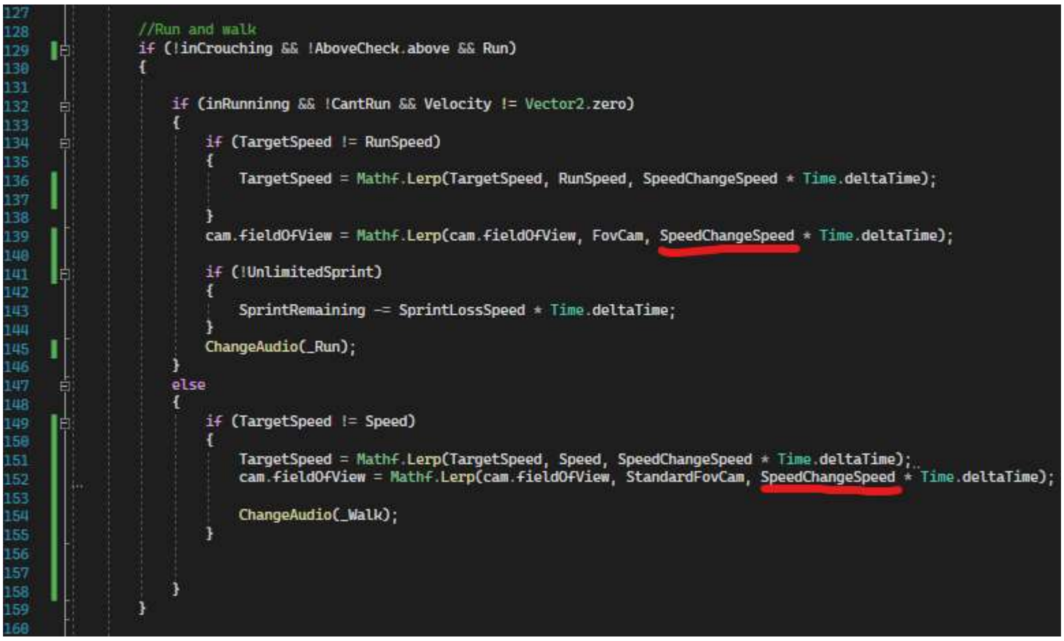
Sprint Return Speed - the sprint return speed depends on this variable (the larger the value of the variable, the longer the sprint will return)

Sprint Loss Speed - this variable determines the speed of sprint loss during its use (the higher the value, the less the player can run)

Return Time Sprint - depends on what moment access to the sprint will return (the less the variable costs, the longer you need to wait for the return of access to the sprint)

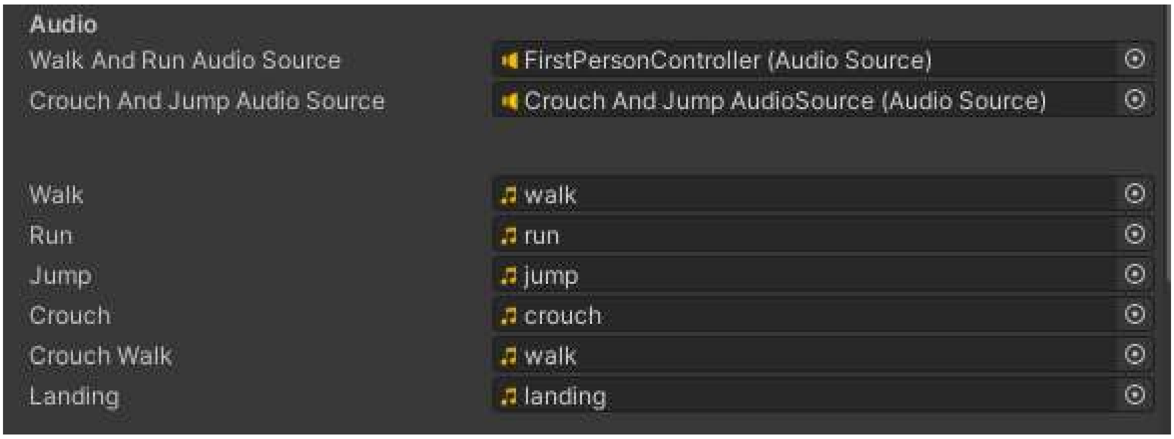
Fov Cam - What will be the Field of view of the camera while running

What will be the Field of view of the camera while running
Recommendation: the speed of changing the Field of view of the camera depends on the SpeedChangeSpeed variable. If you want the player to change the Field of view faster without changing the speed of the speed change, you can create a separate variable and change the speed change speed to this variable where it is highlighted in the picture



```
//Run and walk
if (!inCrouching && !AboveCheck.above && Run)
{

    if (inRunninng && !CantRun && Velocity != Vector2.zero)
    {
        if (TargetSpeed != RunSpeed)
        {
            TargetSpeed = Mathf.Lerp(TargetSpeed, RunSpeed, SpeedChangeSpeed * Time.deltaTime);

        }
        cam.fieldOfView = Mathf.Lerp(cam.fieldOfView, FovCam, SpeedChangeSpeed * Time.deltaTime);

        if (!UnlimitedSprint)
        {
            SprintRemaining -= SprintLossSpeed * Time.deltaTime;
        }
        ChangeAudio(_Run);
    }
    else
    {
        if (TargetSpeed != Speed)
        {
            TargetSpeed = Mathf.Lerp(TargetSpeed, Speed, SpeedChangeSpeed * Time.deltaTime);
            cam.fieldOfView = Mathf.Lerp(cam.fieldOfView, StandardFovCam, SpeedChangeSpeed * Time.deltaTime);

            ChangeAudio(_Walk);
        }


    }
}
```

Audio

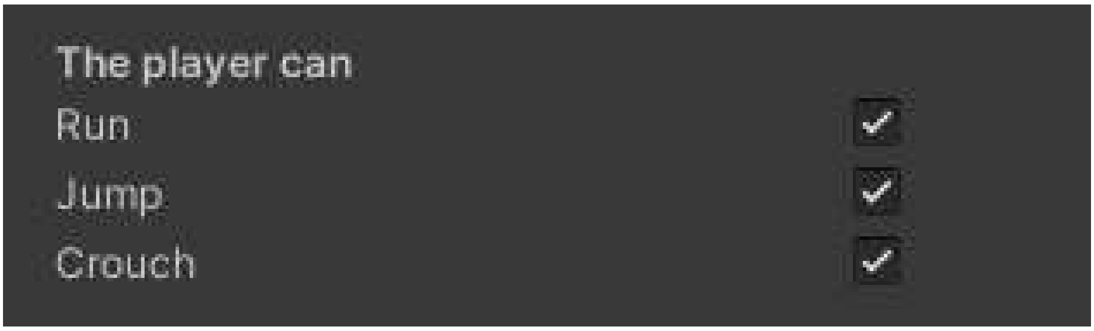


Audio

Walk And Run Audio Source	FirstPersonController (Audio Source)
Crouch And Jump Audio Source	Crouch And Jump AudioSource (Audio Source)
Walk	walk
Run	run
Jump	jump
Crouch	crouch
Crouch Walk	walk
Landing	landing

Here you can find the necessary links to the AudioSource and the audio clips that will be played with a certain action.

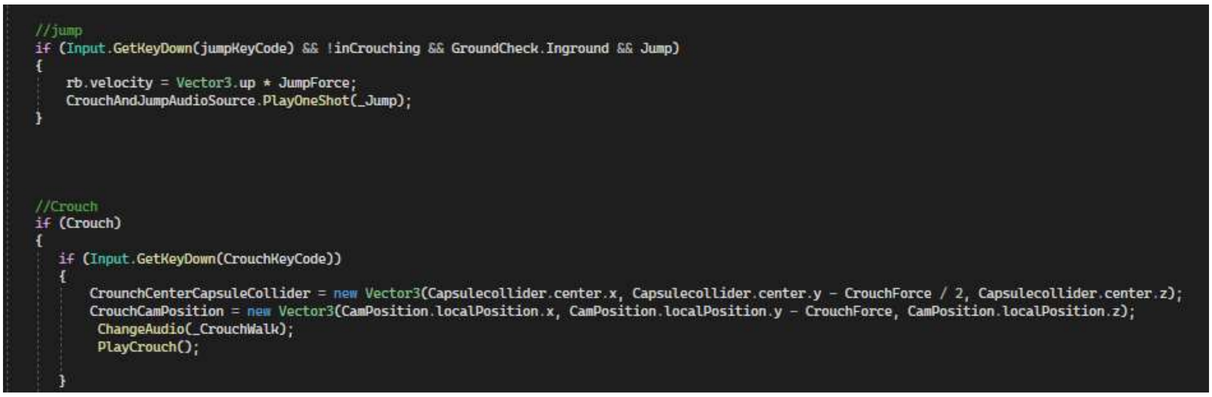
The player can



The player can

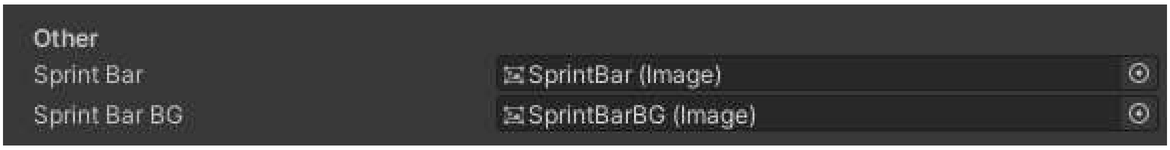
Run	☑
Jump	☑
Crouch	☑

If the player should not do any of these actions, you can uncheck
Recommendation: If you do not plan for the player to do any action in the game completely, then you can remove the part of the code that is responsible for this action. Part of the code of each action is commented out so that you understand that this particular part of the code is responsible for a certain action.



```
//jump
if (Input.GetKeyDown(jumpKeyCode) && !inCrouching && GroundCheck.Inground && Jump)
{
    rb.velocity = Vector3.up * JumpForce;
    CrouchAndJumpAudioSource.PlayOneShot(_Jump);
}



//Crouch
if (Crouch)
{
   if (Input.GetKeyDown(CrouchKeyCode))
   {
       CrounchCenterCapsuleCollider = new Vector3(Capsulecollider.center.x, Capsulecollider.center.y - CrouchForce / 2, Capsulecollider.center.z);
       CrouchCamPosition = new Vector3(CamPosition.localPosition.x, CamPosition.localPosition.y - CrouchForce, CamPosition.localPosition.z);
        ChangeAudio(_CrouchWalk);
        PlayCrouch();

   }
```

Other



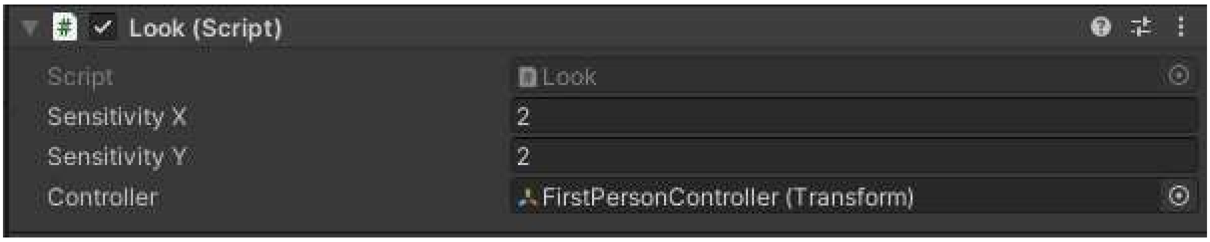
Other

Sprint Bar: SprintBar (Image)

Sprint Bar BG: SprintBarBG (Image)

Used to store data. We recommend not touching private variables. each variable will be commented out explaining what it is responsible for.

Look



Look (Script)

Script: Look

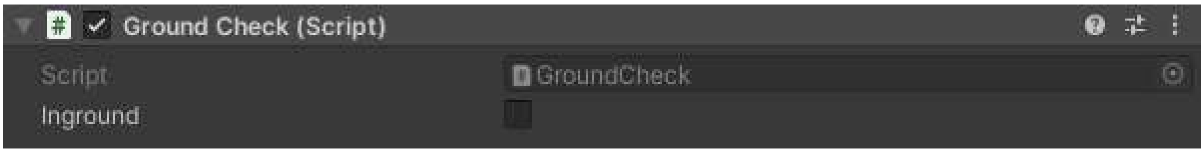
Sensitivity X: 2

Sensitivity Y: 2

Controller: FirstPersonController (Transform)

The script is on the camera. Used to turn the player
Recommendation: If you need an inverted rotation, set a negative value for the sensitivity

GroundCheck

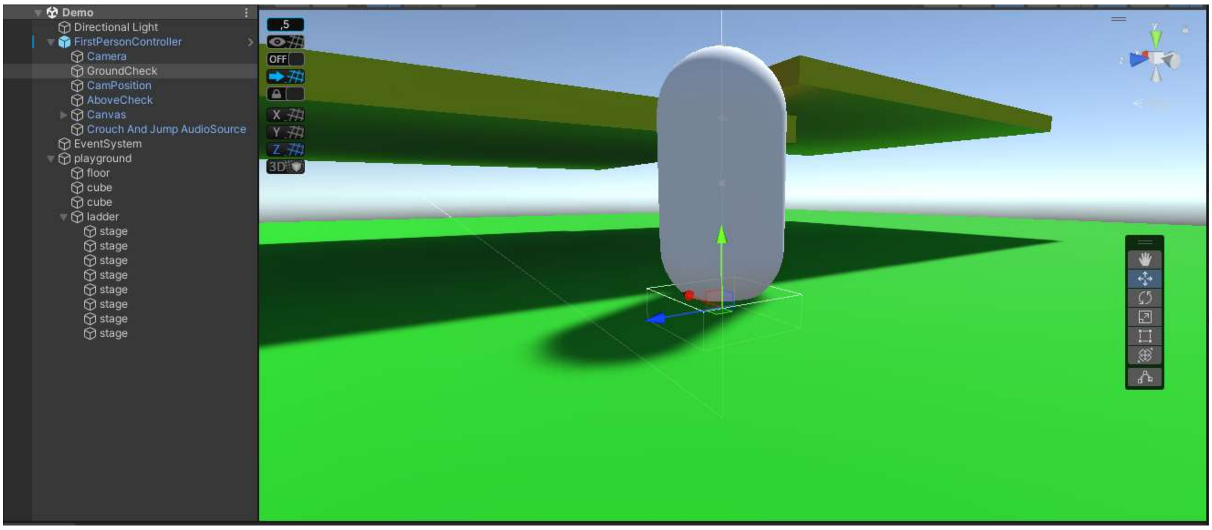


Ground Check (Script)

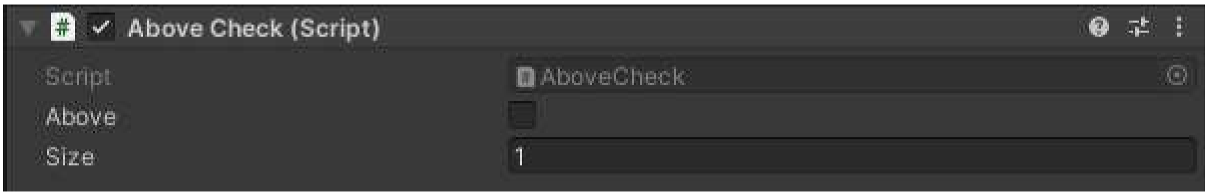
Script: GroundCheck

Inground: □

Used to jump. it checks if the player is above the ground. if yes then the bool variable inground is true. if the player has jumped then inground is false to prevent the player from doing infinite jumps. All this is checked thanks to the collider.



AboveCheck



Above Check (Script)

Script AboveCheck

Above □

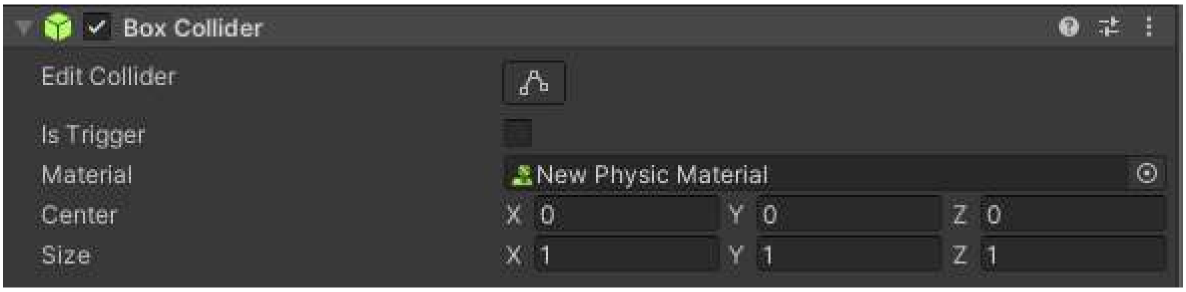
Size 1

Used to check if there is an object above the player during Crouch. If there is anything, the variable Above is true and does not allow the player to stand up until there is nothing above the player. if the player has nothing, then the variable is false and the player will be able to stand up. raycast is used.

Size - the length of the raycast

Recommendation

To prevent the player from sticking to the walls, put new physic material in material to the mesh colliders.



Box Collider

Edit Collider

Is Trigger □

Material New Physic Material

Center X 0 Y 0 Z 0

Size X 1 Y 1 Z 1

This is the end of the documentation. This asset can be used for commercial purposes. If you have any questions, you can write to us at sdfstudiosdev@gmail.com