Camera and Controls recipe

Contents

Setup a simple level

1. Create a simple level.
2. Set spawn location position to (0,14,0).
3. Create a part-> block. Rename to part1. Anchor it. Set position to (19,7,0) and size to (12,13,16). Orientation to (0,90,0).
4. Duplicate part2 and rename to part2. Set size to (12,15,18) and position to (40,8,0).
5. Duplicate part2 and rename to part3. Set size to (12,9,18) and position to (62,5,0).
6. Duplicate the spawn location and rename to SpawnLocation2 and position to (82,14,0).
7. Play and see.

Add camera script

1. In workspace, under StarterPlayer->StarterPlayerScripts add a local script and rename to 'CameraScript'.
2. In 'CameraScript', add code below.

```
local players = game:GetService("Players")

local runSvc = game:GetService("RunService")


local player = players.LocalPlayer

local camera = workspace.CurrentCamera


local camDepthOffset = 24
local htOffset = 2


local function UpdateCam()
    local char = player.Character
    if char then
            local root = char:FindFirstChild("HumanoidRootPart")
            if root then
                    local rpos = root.Position + Vector3.new(0,htOffset,0)
                    local cpos  = Vector3.new(rpos.x, rpos.y, camDepthOffset)
                    camera.CFrame = CFrame.lookAt(cpos,rpos)
            end
    end
end


runSvc:BindToRenderStep("Sidescrollcam", Enum.RenderPriority.Camera.Value+1, UpdateCam)
```

3. Basically, we are storing player and camera in variables. Then we are binding UpdateCam() function to RenderStep so that it can execute at every rendering. In UpdateCam() we are first getting HumanoidRootPart, storing it in variable 'root'. To root position (rpos) we are adding height offset. We are ensuring camera is at root.position in x and y but camDepthOffset in z axis. And it is always looking at rpos. This way camera can move left, right, up or down to giving 'side scrolling effect'.
4. Play and see. Player can move on the block and camera will move side to side with him. But when user presses 'w' or 's' key player can move towards or away from camera, but camera will stay put.

Add player control script

1. To ensure the player can go only from side to side, we need to add some code.
2. To StarterPlayerScripts add one more local script, rename to 'ControlScript'.
3. Add code to 'CoontrolScript' as below.

```
local player = game.Players.LocalPlayer
local runSvc = game:GetService("RunService")
local conAcSvc = game:GetService("ContextActionService")
local jumping = false
local leftVal, rightVal = 0, 0

local function onLeft(acNm, ipSt)
    if ipSt == Enum.UserInputState.Begin then
        leftVal = 1
    elseif ipSt == Enum.UserInputState.End then
        leftVal = 0
    end
end

local function onRight(acNm, ipSt)
    if ipSt == Enum.UserInputState.Begin then
        rightVal = 1
    elseif ipSt == Enum.UserInputState.End then
        rightVal = 0
    end
end

local function onJump(acNm, ipSt)
    if ipSt == Enum.UserInputState.Begin then
        jumping = true
    elseif ipSt == Enum.UserInputState.End then
        jumping = false
    end
end

local function onUpdate()
```

```
    if player.Character and player.Character:FindFirstChild("Humanoid") then
        if jumping then
            player.Character.Humanoid.Jump = true
        end
        local moveDir = rightVal - leftVal
        player.Character.Humanoid:Move(Vector3.new(moveDir,0,0),false)
    end
end


runSvc:BindToRenderStep("Control", Enum.RenderPriority.Input.Value, onUpdate)


conAcSvc:BindAction("Left",onLeft, true, "a", Enum.KeyCode.Left, Enum.KeyCode.DPadLeft)
conAcSvc:BindAction("Right",onRight, true, "d", Enum.KeyCode.Right, Enum.KeyCode.DPadRight)
conAcSvc:BindAction("Jump",onJump, true, " ", Enum.KeyCode.Space, Enum.KeyCode.DPadUp)
```

4. Basically, we have variables to store player, run service and context action service. Variable for left or right or jump. In onLeft() function we are setting leftVal based on whether ‘a’ key is pressed or not. Similarly onRight() and onJump(). onUpdate() is making character jump by setting value on Humanoid.jump or Humanoid:Move. Finally, we are binding all functions to run service or context action service.

Add coins and ability to collect them and UI

1. Use this link - https://create.roblox.com/store/asset/2683656699/Collectable-Coin - and GetModel and drag drop it to 3d view floating on top of one block. Delete ‘Leaderboardscript’ and ‘FloatingCoinscript’. Set the scale of ‘CoinMesh’ to (7,7,7).
2. See the code in CollectCoinScript and SpinningCoinScript.
3. Add intValue variable to workspace and rename to ‘Points’.
4. In the CollectCoinScript, remove the line player.leaderstats … and instead put workspace.Points.Value = workspace.Points.Value + 1
5. Change script.Parent:Remove() to script.Parent:Destroy().
6. Play and see, when coin it collected in workspace, value of points must increase and coin should vanish after playing a sound.
7. Now, duplicate the collectiblecoin and place it floating on each block.

8. Add a ScreenGUI to starterGUI. To screenGUI add frame, put appropriate color, size, transparency. Add textLabel to frame. Put appropriate text color, size, font and background transparency.
9. Add and script to textLabel and in script add code below.

```
local function updatepoints()

    script.Parent.Text = "Points: " .. tostring(workspace.Points.Value)

end


workspace.Points.Changed:Connect(updatepoints)
```

10.

Play

1. Play and see.
2. Gamer must be able to run to left / right like we do in ‘Mario platform game’ and collect coins.