[the academy_of_code]

Roblox Summer

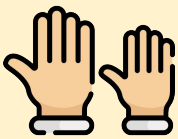
Lesson links:

www.theacademyofcode.com/handouts

Lesson 1 - Introduction to Basics

Learning Outcomes:

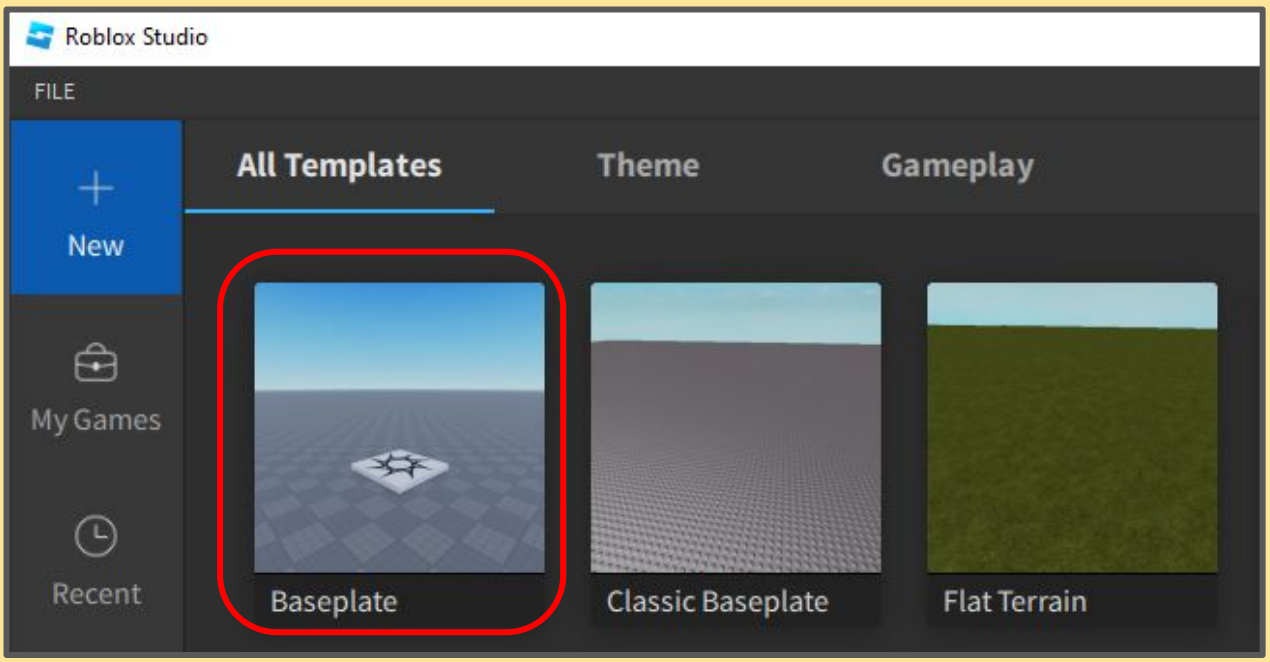
- Learn the basics of how to navigate Roblox
- Begin working in the Roblox environment



REMEMBER: Raise your hand. We love to help!

Let's Create our Project

1. Let's start by opening **Roblox Studio**. You should see a menu screen to select a starting template. Choose the baseplate option



Learning to Move!

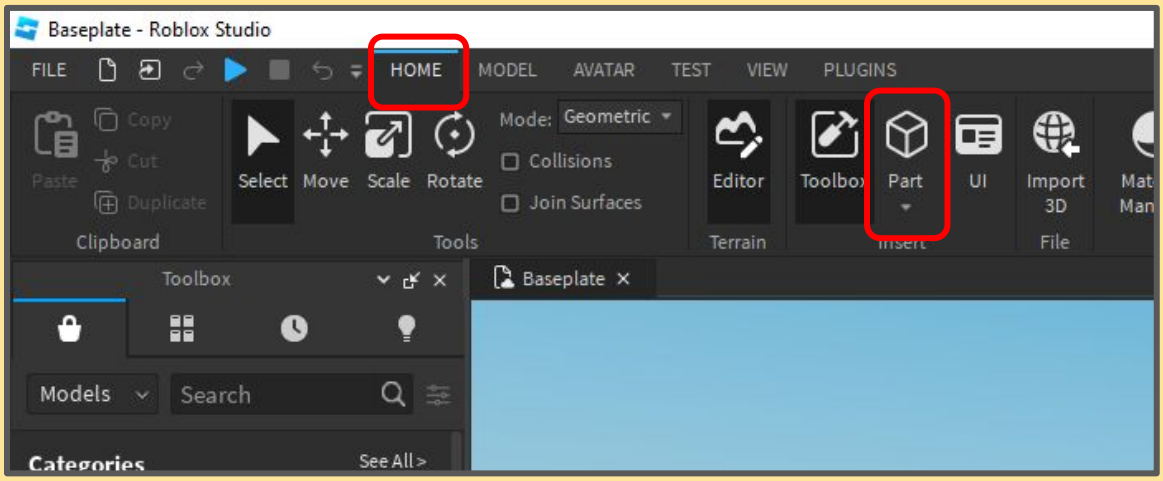
The most important thing to learn first is how to move around the roblox world!

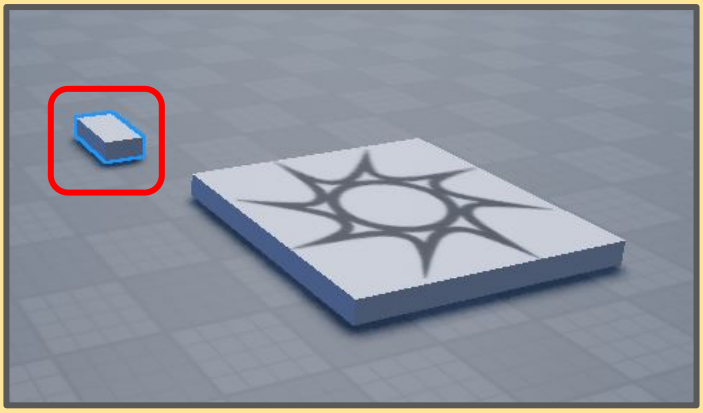
1. Practise turning the camera around by holding right click and moving the mouse around.

2. Practise moving around the world using WASD or the arrow keys

Adding your first part

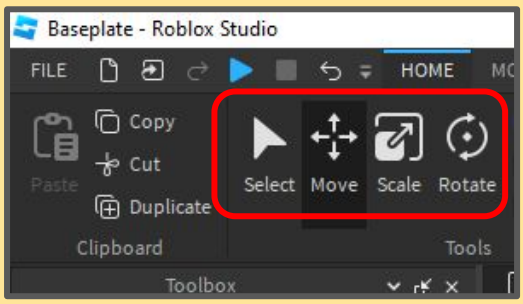
1. At the top left part of the studio you should see the “Part” option. If you can’t see it make sure you have “Home” selected first. If you press this button it will make a block appear in your world



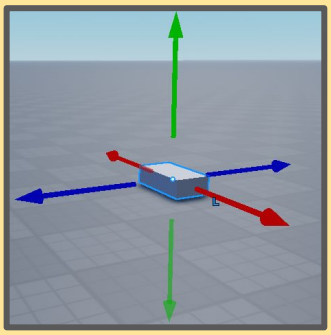


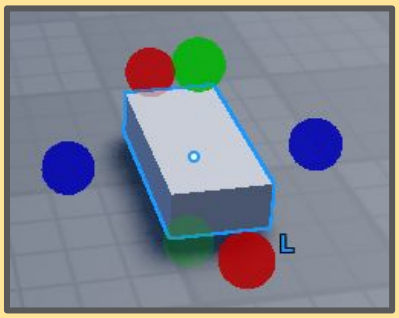
2

Now we are going to work with moving and editing the piece around. At the top left you should see a list of options as shown below. We are going to look at each one



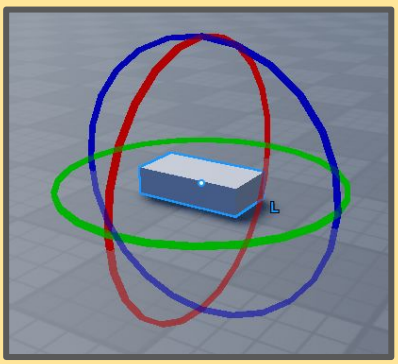
First let's look at the **move**. The roblox world is in 3D which means you can move it in 3 directions. Up-Down, Left-Right and Forward-Backward



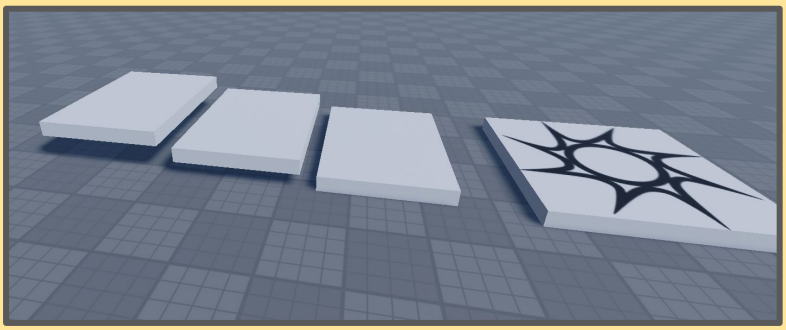


Next we have the **scale**. This is similar to move but instead of moving it it will make the block bigger or smaller in that direction

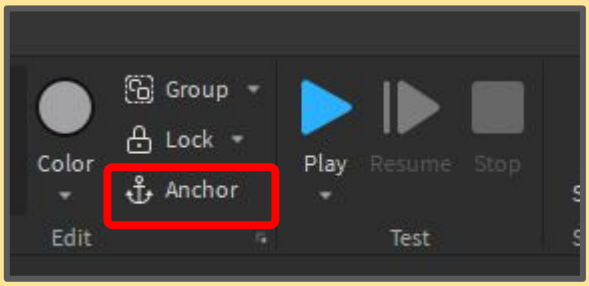
Finally we have **rotate**. This rotates the block in the same directions as previous 2 options



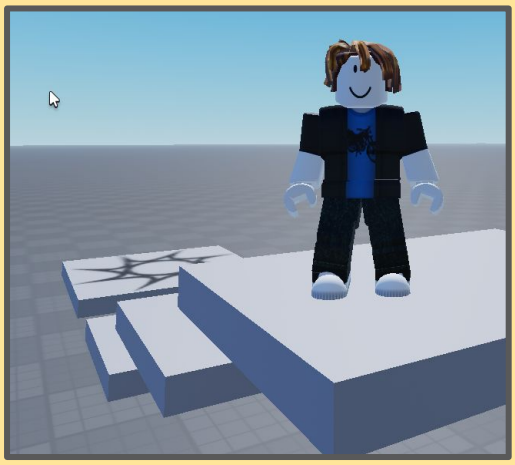
3 Using what you have learned, create a set of stairs in front of the starting block.



4 If you were to try and play the roblox world now all the blocks would fall down. If you want them to stay in the air you have to “**Anchor**” them. To do this, click on each block and press the **Anchor** button at the top left.

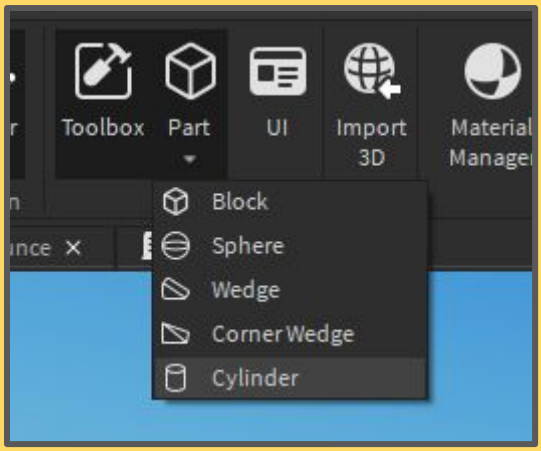


5 Once you have done this you can press the **Play** button to enter your roblox world. You will spawn on the first block. This is a special block called the **Spawn Block**.

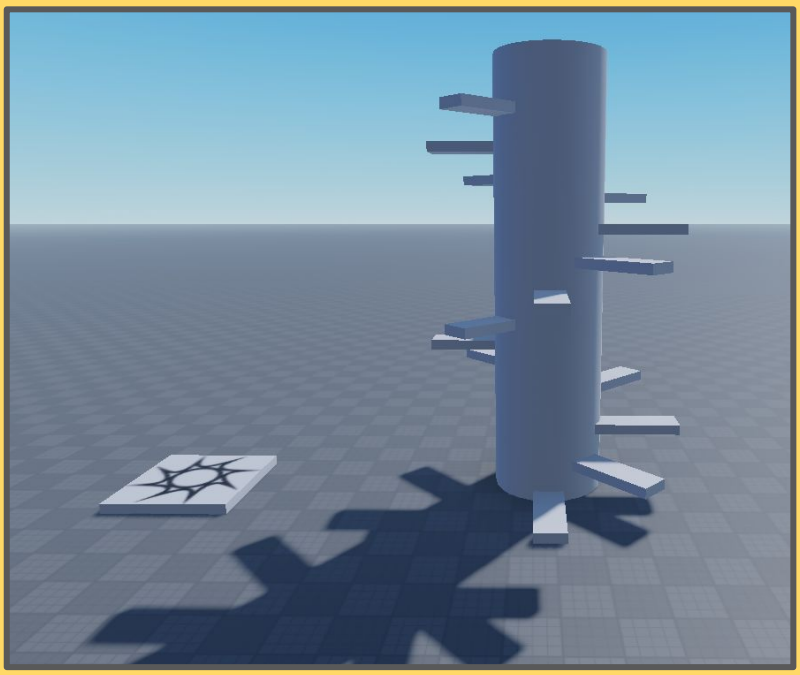


Additional Tasks!

Now that you can make blocks work on creating a spiraling staircase. You will see a new block here called a cylinder. You can make get these additional block types by pressing the small arrow below the **Part** button.



Using **Parts** create the spiral staircase below. We will be using this as part of the next lesson so make sure to complete it fully.

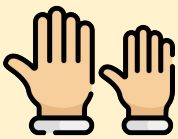


DON'T FORGET TO SAVE YOUR PROJECT

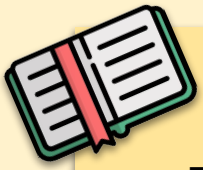
Lesson 2 - Fake floors

Learning Outcomes:

- Learn how to work with properties
- Create fake floors



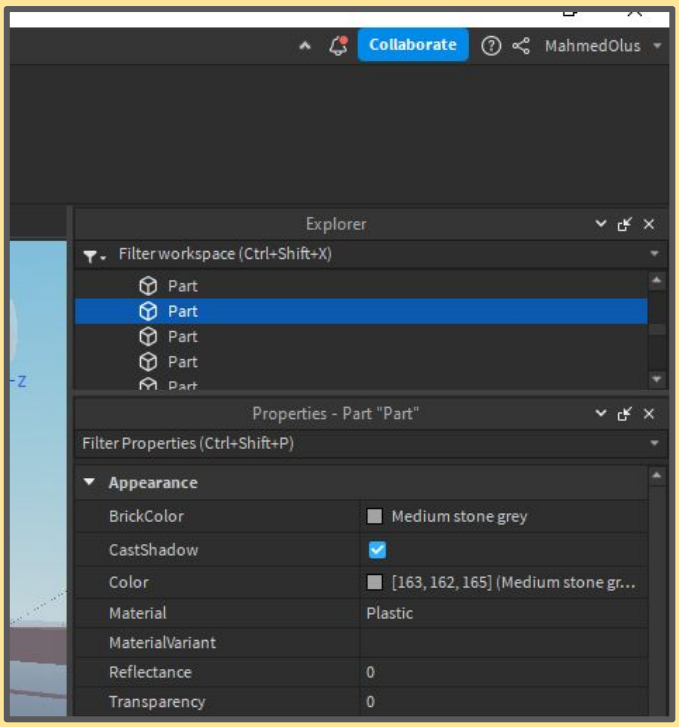
REMEMBER: Raise your hand. We love to help!



Block Properties

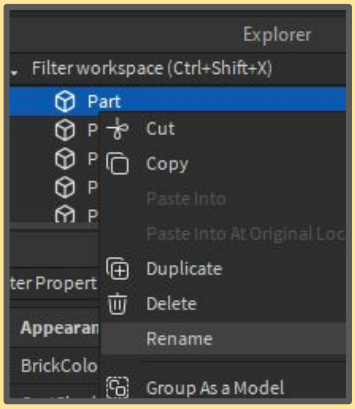
Things in roblox have “**Properties**”. Properties include stuff like the color and size of the block. They also include more complicated stuff like whether on not the block is “**Anchored”** which you would have seen in the last lesson.

When you click on a block it will highlight the block in the explorer (right side of the screen) and below the explorer you can see the properties below it

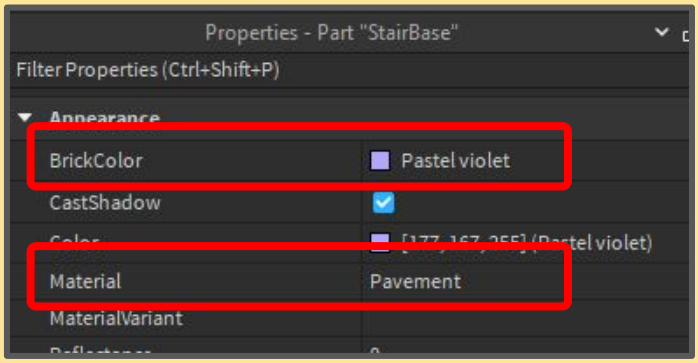


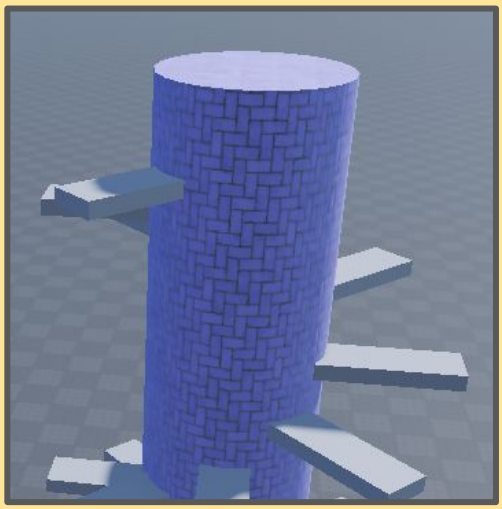
Let's play with Properties

1 We are going to start off with naming our parts. Click on the cylinder of the stairs and rename it to "StairBase". To rename it, right click the part and press rename. Once you do this, rename all the step to "step".

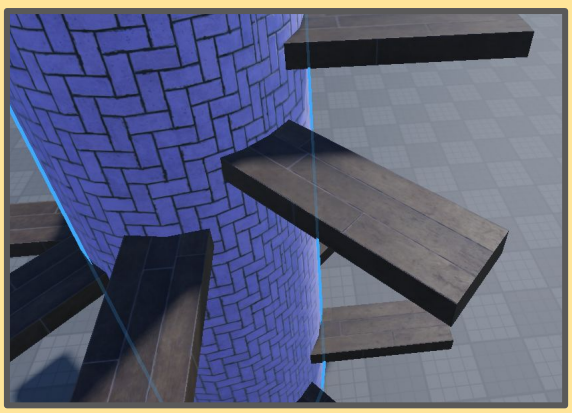


2 We are next going to mess around with the "Appearance" properties. Pick the "StairBase" block and find the appearance section in properties. Once you find it change the **color** and **material** of the block.



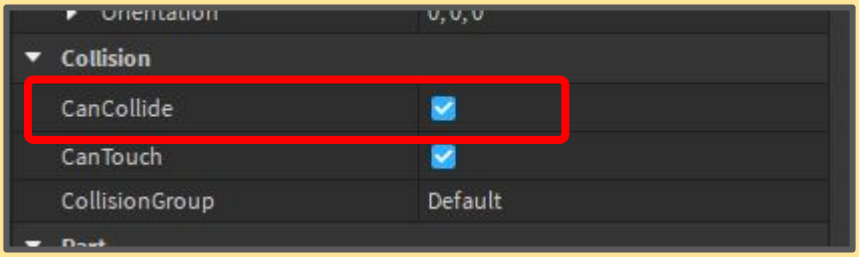


3 Finish off the rest staircase by giving colors and material to the steps. In this image here we used **wood plank** as the material.

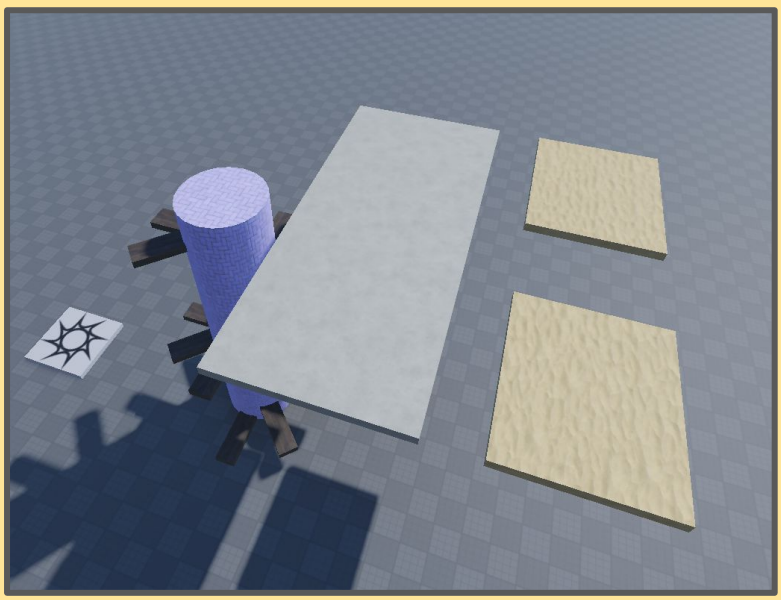


Invisible Blocks

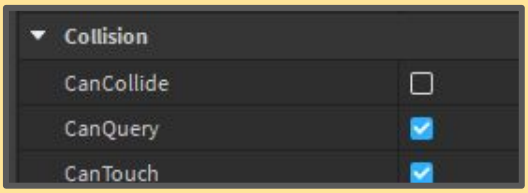
As part of **Properties**, there is a property called “CanCollide”. This means can the player (or anything) interact with the block. If this is not ticked, then you will fall through the block if you try to get on it.



1. Continuing on with our previous project. Add a platform and 2 mini platforms in front of it (choose a color and material). The idea is that one of these will be a trap.



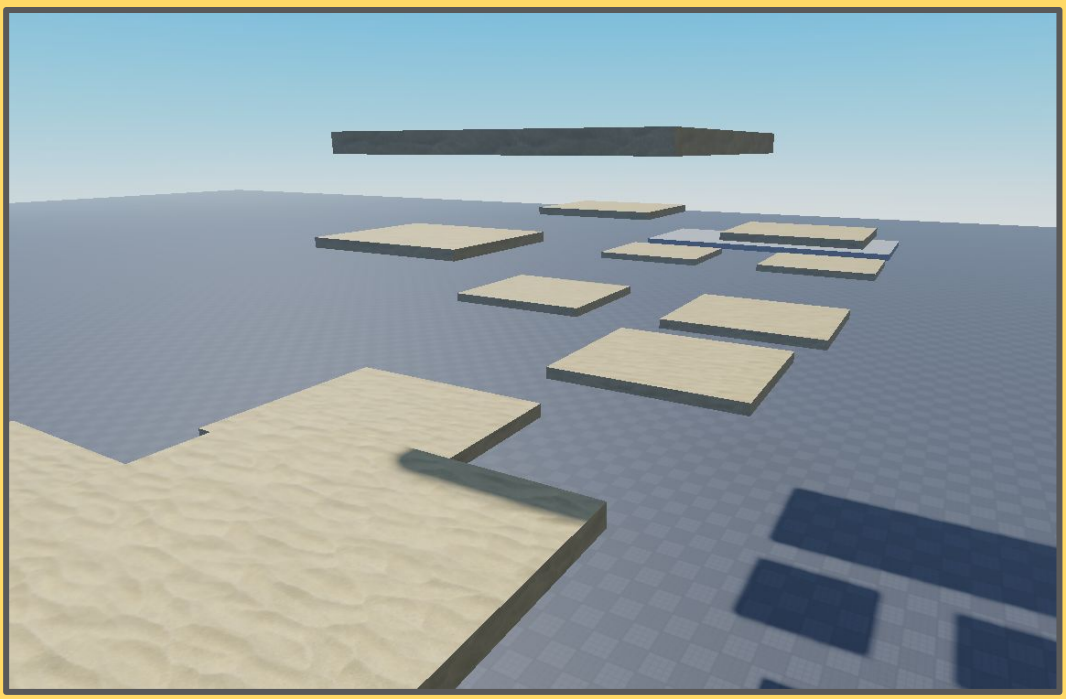
2 Pick one of the mini platforms and change the “canCollide” property to false. Play the world to test it out and you should now fall through one of the blocks



Challenge Tasks!

Now that you can make fake platforms. Work on creating your own platform challenge. Play with different heights to make it more challenging.

(for example you climb up and the next block is actually fake one and you have to jump down and land on the real block which is lower.)

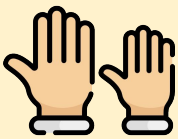


DON’T FORGET TO SAVE YOUR PROJECT

Lesson 3 - Attachments

Learning Outcomes:

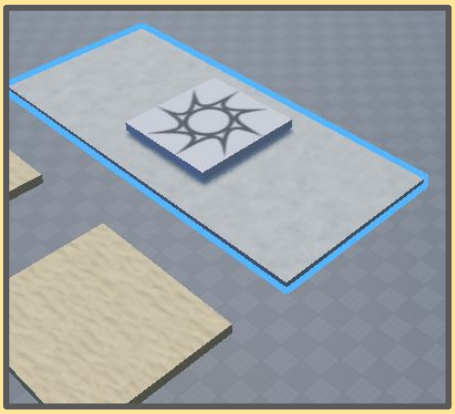
- Learn how to work with properties in Roblox



REMEMBER: Raise your hand. We love to help!

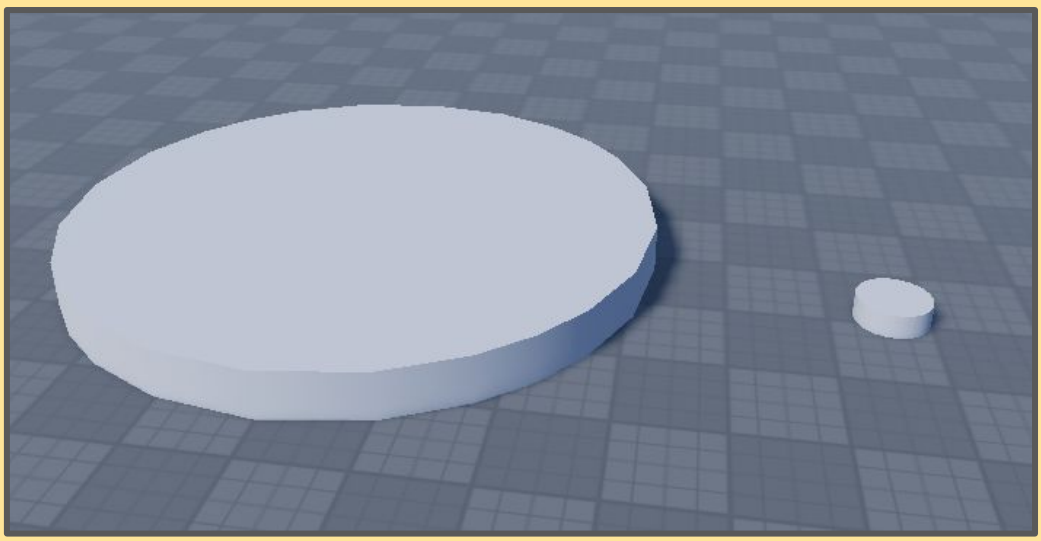
Let's Create!

1. To start off move the spawn plate to the end of the plates you've made. We will be continuing on with this and you want to test the new portion and not have to run through the whole course each time

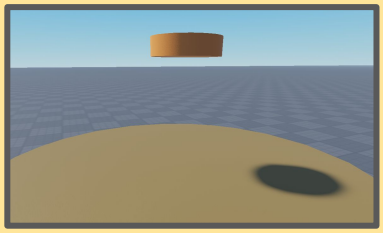


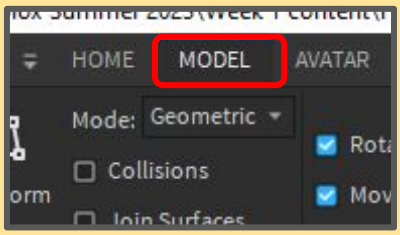
2. We want to first create 2 cylinder block. We want 1 to be a large circle and the other one to be a small circle.

   You want to Anchor the small circle but NOT the big circle



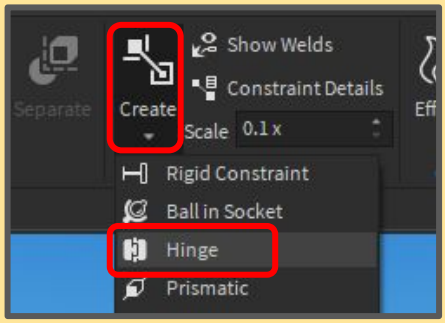
3 You want to aim to get the smaller circle in the middle of the bigger one and slightly up in the air.

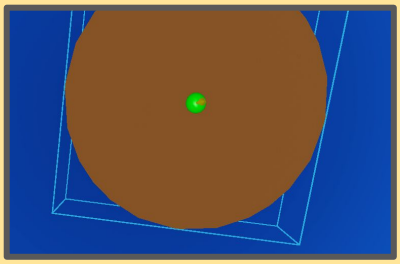




Next you want to go into the “**Model**” section (Top left of your screen)

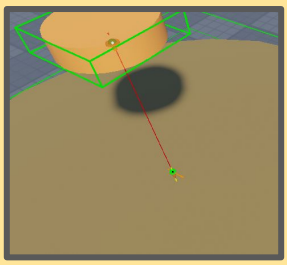
In this Model view, look to see if you can find the “**Create**” button near the top right of your screen. Press the drop down and you should see the **Hinge** option

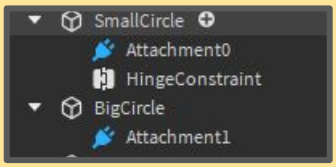




Once you find the **Hinge**. Press it and now you want to click the middle bottom side of the small circle. Once you do it you should see a green dot on it

And finally you will click the middle top side of the bigger circle. This makes a connection between these 2 blocks.



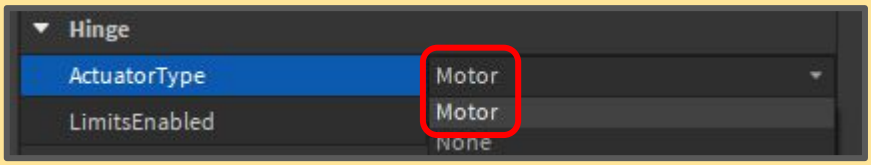


To check that you did it correctly, look for the parts in the explorer and you should see it look like this

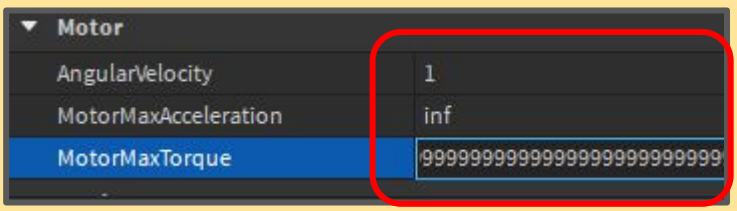
Get out the way!

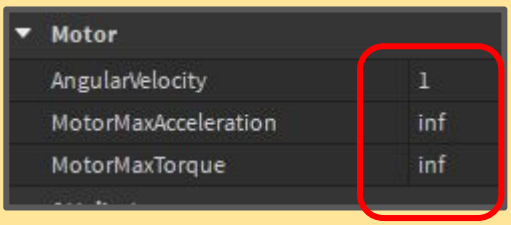
1. The hinge we just made, allows us to make the blocks spin, And we will be using this to make our next part of our obstacle course. Find the hinge in your explorer and go to the "**properties**" section (remember where that is)

2. In the **properties** find the "Hinge" section and in there look for the "**ActuatorType**" option and set this to "**Motor**"

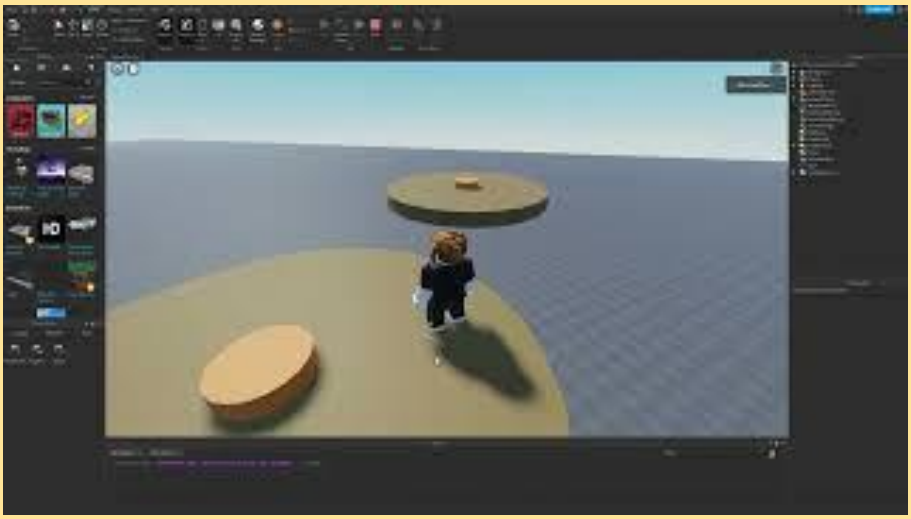


3. A section called "**Motor**" should now appear and see the values to this. (to get **infinity** to appear you need to hold 9 till the maximum number and when click on something else this will now be **inf**)





4 Once you have completed that you should now have a spinning block like the one shown below



Challenge Tasks!

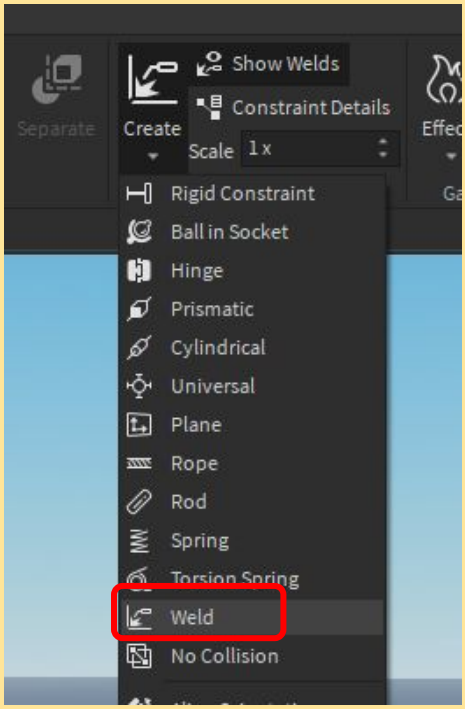
Now that you can make fake platforms. Work on creating your own platform challenge. Play with different heights to make it more challenging.

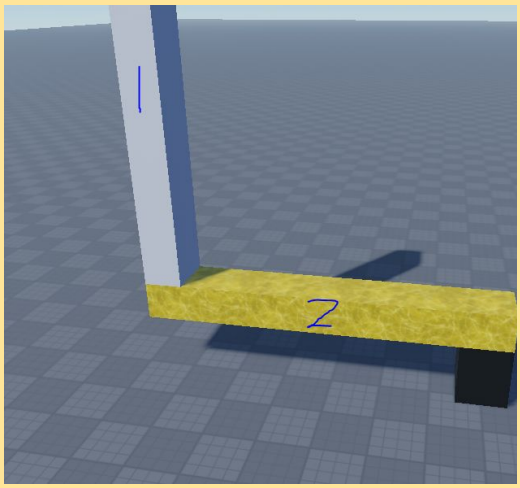
(for example you climb up and the next block is actually fake one and you have to jump down and land on the real block which is lower.)

What else can you create!

1

There are lots of different parts of the create tool other than **hinge**. One of them that is quite easy to use is the **weld** option. If you use the weld tool and press the 2 blocks below, it will stick them together so that they spin together





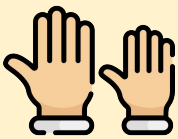
Advanced Challenge

Using **Weld**, create a more advanced version of spinning blocks to create something similar to what's shown below

Lesson 4 - More More MORE!!!

Learning Outcomes:

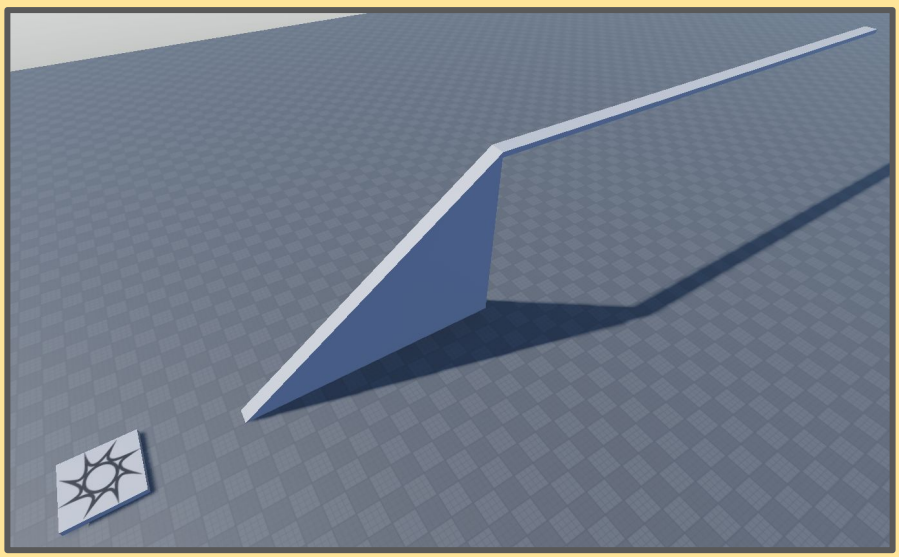
- Improve your understanding of Roblox
- Become more familiar with create attachments



REMEMBER: Raise your hand. We love to help!

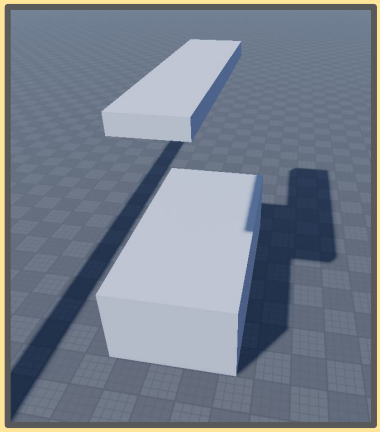
Let's Create!

1. We are going to be making a sort of wrecking ball, to start off with you want to make a bridge with a platform (remember to anchor all this)

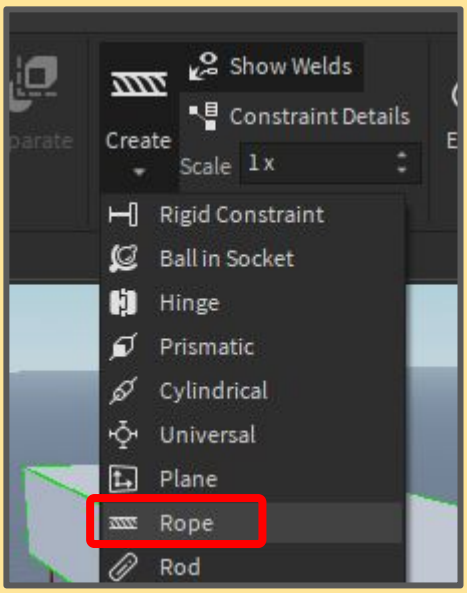


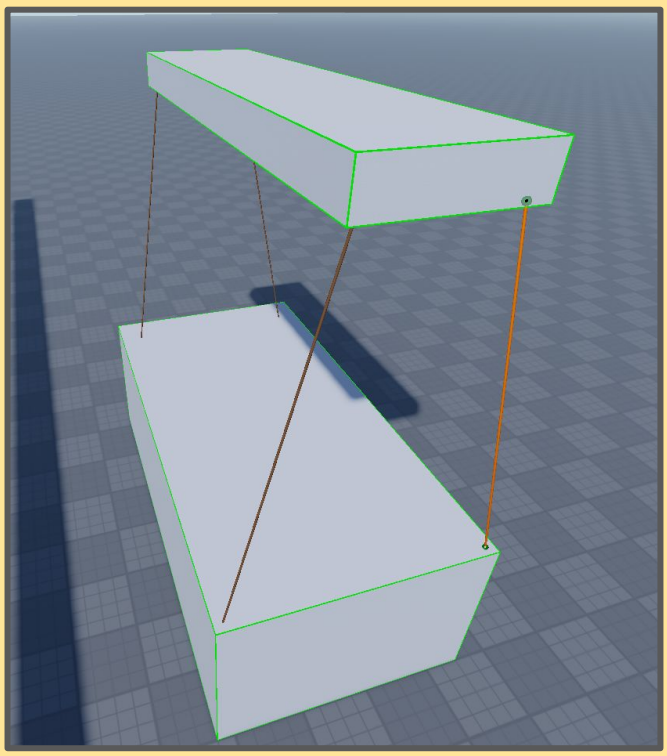
Rope Attachments

2 Make 2 blocks. 1 big block and another smaller one above it. The big block is going to be the wrecking block.

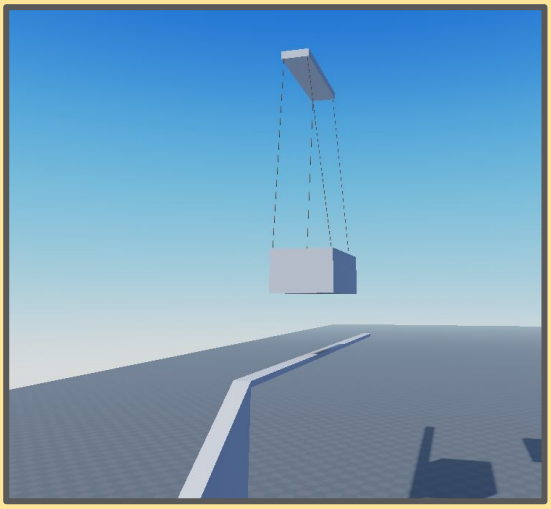


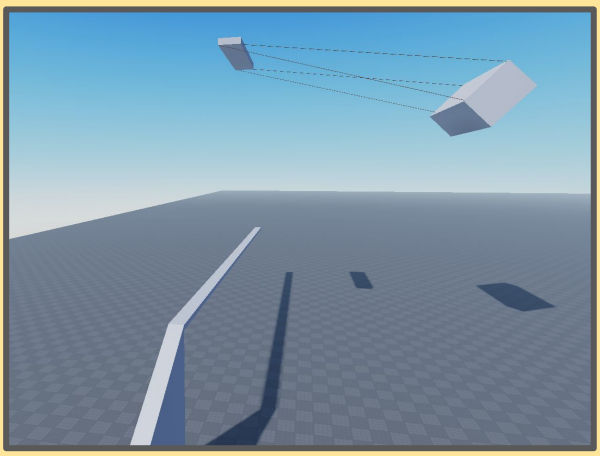
3 Going back to the “**create**” button (this is were we made the rotating blocks) look for the **rope** option. Use this rope option to make **4 rope attachments** from e**ach corner of one block** to the **same corner on the other block.** (put the block far away from each other to make LONG ROPES)





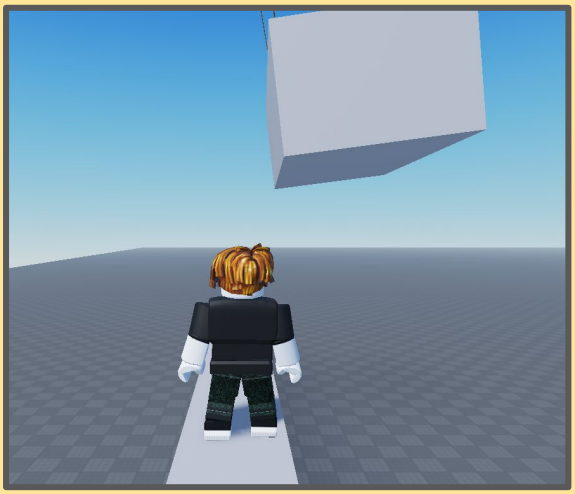
4 Place your blocks over the platform you built previously. The top block should be anchored, the bottom block should not be.





5 With the blocks on top of the platform. Move and rotate the big wrecking block to the side. We do this so that when we press play, the wrecking block (which is the one that is not anchored) will fall with momentum and the rope will swing it back and forth

6 When you press play now check how the block is swinging. The first time you press play it probably won't be perfect. In the image below you will see that it's swinging too far above the platform



7 Here are some tips to improve it.

- Change the height of the blocks and the length of the ropes (you can find this in properties)
- If the block is the right height but can't knock the player off, make the wrecking ball bigger (BIGGER BLOCK BIGGER HIT)

Work on the wrecking ball until you have it knocking you off the platform as such.

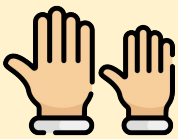
Challenge Tasks!

Now that you have the basics of an obstacle course learned, work on creating your own fun challenge. Use inspiration from obstacle course game shows/ games such as fall guys or total wipeout!!!!!

Lesson 5 - Scripts

Learning Outcomes:

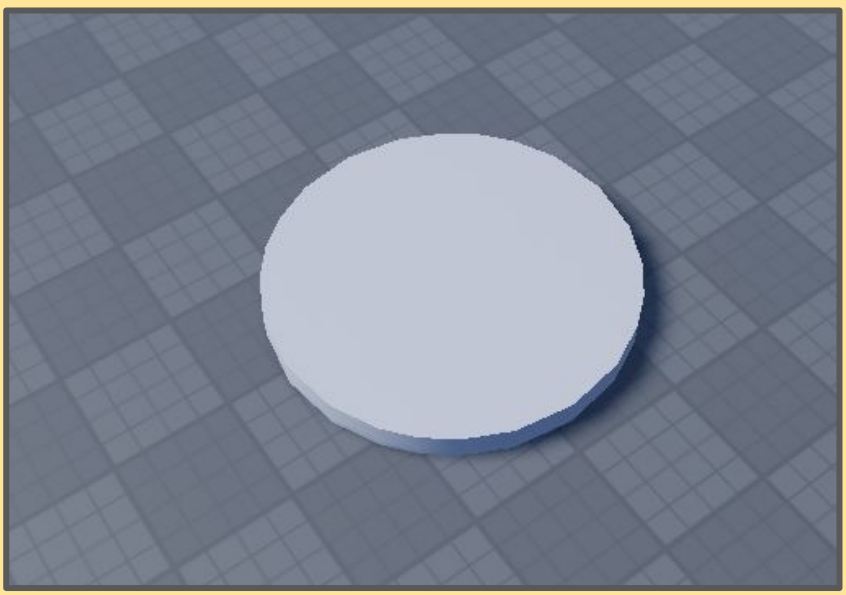
- Improve your understanding of Roblox
- Become familiar with Scripting



REMEMBER: Raise your hand. We love to help!

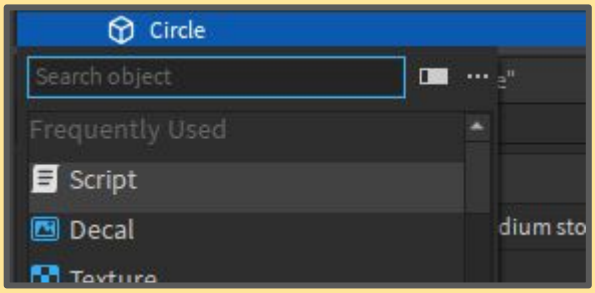
Let's Create!

1. To start off with make a circle block on the floor. We will be using this to make a scripted block

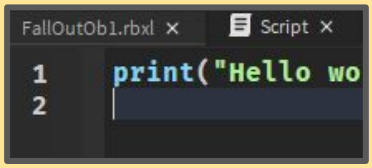


Adding Scripts

2 Find the block you just made in the explorer and press the plus button beside. When you press + find the script option and click it



3 This will create a script with a line of code in it already. Delete this line and instead add the following line of code




```
script.Parent.Touched:Connect(speedUp)
```

Writing code in roblox can be difficult to start off with but as you get more practice you will be confident in writing it on your on. The way the line above works is;

- **script**: the script you are working on
- **Parent**: this is saying whatever the thing the script is inside which in this case is the circle block we just made
- **Touched**: this is looking for the property which is asking “is something touching it”
- **Connect(speedUp)**: run the code called **speedUp** (we haven't made that yet but we will)
-

In combination what the line says is “if someone is touching the block that i am in, run the code called speedUp

4 Now we are going to add the code that we want to run. Make sure to be careful with brackets, brackets are the easiest way to break your code!

```
function speedUp(part)

end

script.Parent.Touched:Connect(speedUp)
```

What we are doing now is saying that once our block is touched, any code between these 2 lines will run

5 Now we are going to make a variable. Write the following code below

```
function speedUp(part)
    local humanoid
end

script.Parent.Touched:Connect(speedUp)
```

6 We want this variable to check if the thing that touched the block is a human character. To do that we make it equal the following line of code. Can you guess what the line is trying to say?

```
function speedUp(part)
    local humanoid = part.Parent:FindFirstChild("Humanoid")
end

script.Parent.Touched:Connect(speedUp)
```

Playing with the Properties

1. Now that we have the code checking if a player has touched the block, we now want to do something to the player. We are going to be changing the players “**properties**” you should have already started changing **properties** of different parts and objects in **Lesson 2**. Write the following code that.

```
function speedUp(part)
    local humanoid = part.Parent:FindFirstChild("Humanoid")
    if humanoid  then


    end
end

script.Parent.Touched:Connect(speedUp)
```

2. Now add this following line of code to turn this into a speed up block which makes you run and jump further!

```
function speedUp(part)
    local humanoid = part.Parent:FindFirstChild("Humanoid")
    if humanoid  then
        humanoid.WalkSpeed = 40
        humanoid.JumpHeight = 20
    end
end

script.Parent.Touched:Connect(speedUp)
```

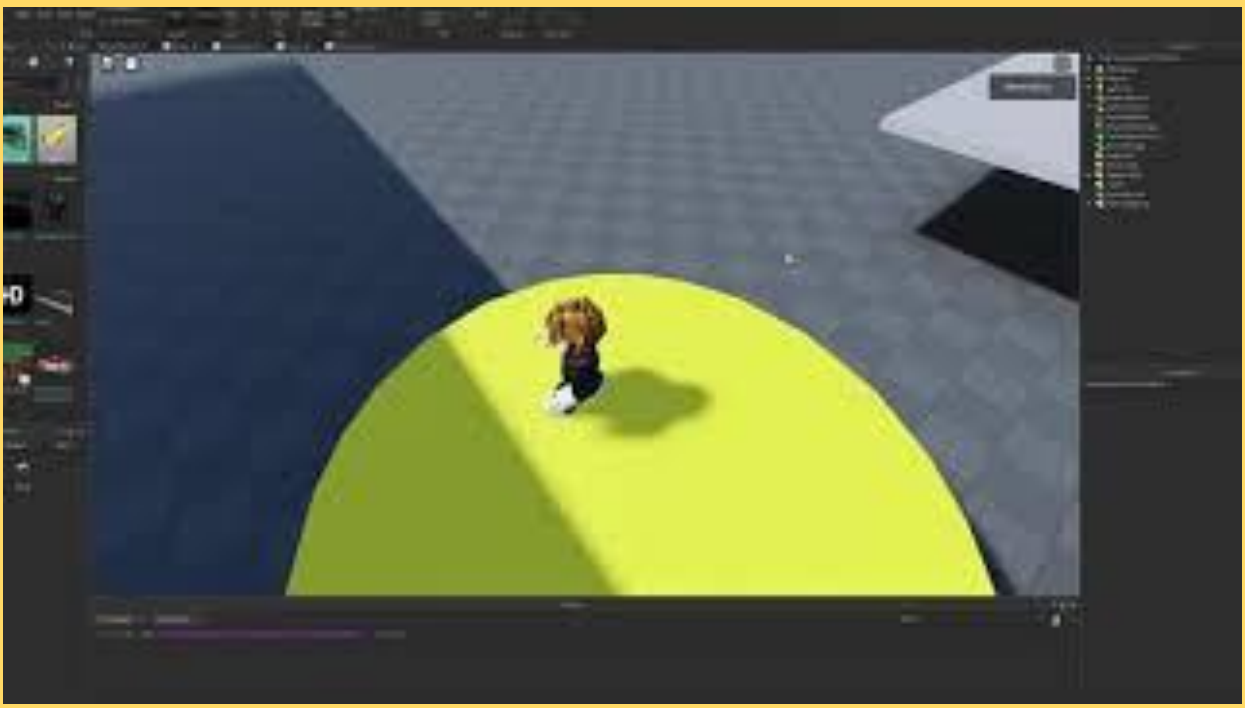
Challenge Tasks!

With the speed block working your task now is to create a **jump pass**. The idea is that the player will have to **choose between 2 paths**. 1 is the block that will **speed you up**, and the other **slows you down**.

The player needs to **choose correctly** to be able to make the jump to the next platform or they will **fall down**.

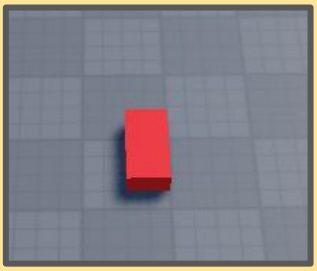
Add a block that returns you back to **normal speed** when you try the obstacle again. Normal speed is

- **humanoid.WalkSpeed = 20**
- **humanoid.JumpHeight = 8**



Making Kill blocks

1. Making a kill block is very similar to how we did the speed blocks. Start by adding a block and colour it red.



2. Similar as before we are going to write some code that will say when the parent (the kill block) is touched by something, run the code called kill.

```
local function kill(part)

end

script.Parent.Touched:Connect(kill)
```

3. Inside this function we are going to do the same code structure which is
   - Check if it is a human
   - If its a human do something (in this case we are setting the human health to 0)

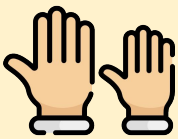
```
local function kill(part)
    local humanoid = part.Parent:FindFirstChild("Humanoid")

    if humanoid then
        humanoid.Health = 0
    end
end


script.Parent.Touched:Connect(kill)
```

4 Once you have completed the kill block, add this to the bottom of the previous jump so that if you fail the jump it sends you back to the start

Lesson 6 - Moving Walls

Learning Outcomes:

- Improve your understanding of Roblox
- Become familiar with Scripting

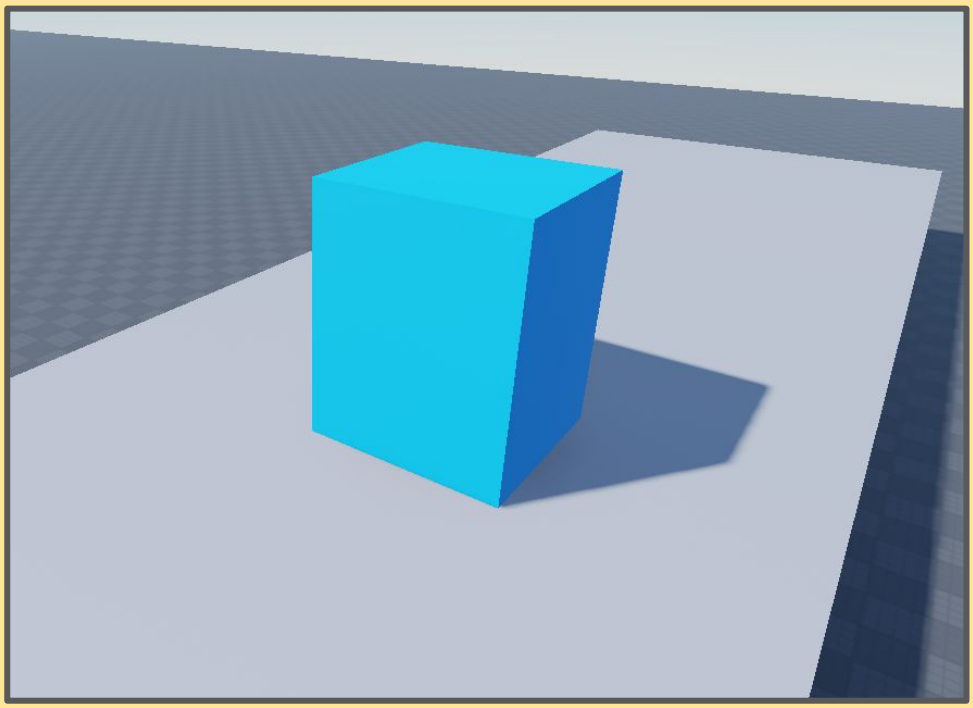


REMEMBER: Raise your hand. We love to help!

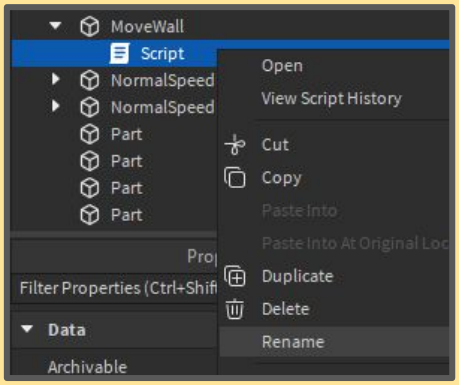
Let's Create!

1. After the jump pass make a platform and add a big square block to it.

   Rename this block to MoveWall



2 Add a script to this block you just made, when you add the script you can also rename it by right clicking the script in the explorer. Rename it to **movingScript**

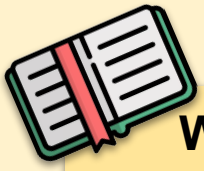


3 The first thing we are going to do in this script is delete the code inside it and right this line of code. We have been using “script.parent” a few times now and this connects to the block itself. This time we are going to make a variable that will represent the block we made

```
wall = script.Parent
```

4 Every time before we have made scripts where something happens when you touch it. This time we want this to be a script that happens by itself forever. We do this by adding the following lines

```
wall = script.Parent


while true do

end
```



What are Loops

You will only be having a glance at loops in this lesson but the idea of loops is repetition. Instead of doing the same thing 1 million times, you can tell your computer what you want to do once and then tell it to repeat itself for as long as you want

1. Add in the following lines of code. This code is saying to repeat the lines of code you about to receive 30 times.

```
while true do

    for i=1, 30 do

    end


end
```

2. What we want to happen is to make the block move. We do that by writing the following lines of code

```
for i=1, 30 do
    wall.CFrame = wall.CFrame * CFrame.new(0,1,0)
    wait(0)
end
```

CFrame is the roblox variable for the coordinates of something. What the line look complicated the only part you need to focus on is the last part. **The 3 numbers here control which direction, left-right, up-down, forward-backwards**

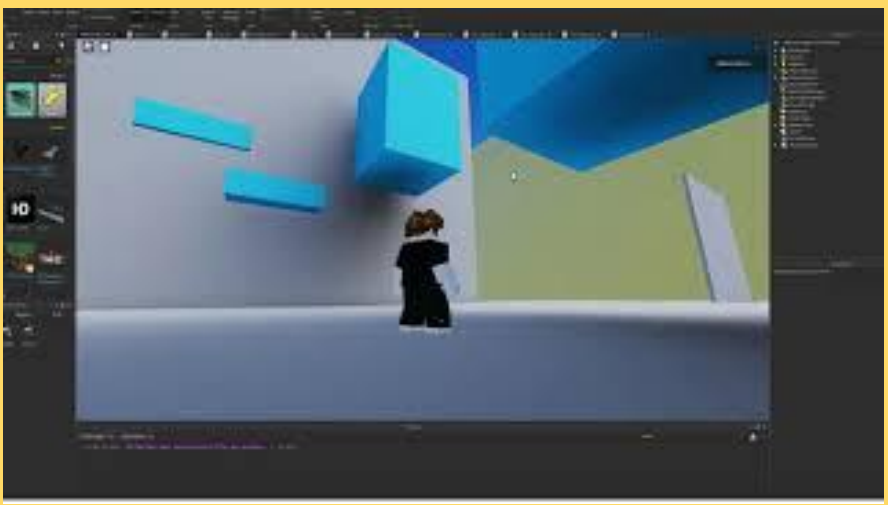
```
CFrame.new(0,1,0)
```

3 Now to finish off the block we want to tell it to go back to the start. This is the exact same as before but instead we say **-1** to go the other direction. Now you should have a block that moved up and down

```
wall = script.Parent

while true do


    for i=1, 30 do
        wall.CFrame = wall.CFrame * CFrame.new(0,1,0)
        wait(0)
    end

    for i=1, 30 do
        wall.CFrame = wall.CFrame * CFrame.new(0,-1,0)
        wait(0)
    end



end
```

Challenge Tasks!

By changing which one of the numbers in the CFrame.new() number you set to 1 will change which direction it will move in,



Advanced Challenge

Your goal now is to create 2 structures using what you have learned so far (moving blocks, speed blocks, kill blocks). Below will be some ideas to give you a creative jump start on what you want to do

Create a moving wall that will punch you off a platform and push you towards a kill block wall

Create a spinning platform (remember hinges) that has kill blocks attached to it, you have to run through this and avoid the kill blocks

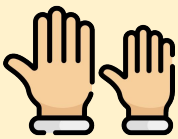
Make an area where pillars appear from the floor and make them kill blocks, players will have to navigate through the area, make people either super fast or super slow to make it much more difficult

DON'T FORGET TO SAVE YOUR PROJECT

Lesson 7 - Checkpoints

Learning Outcomes:

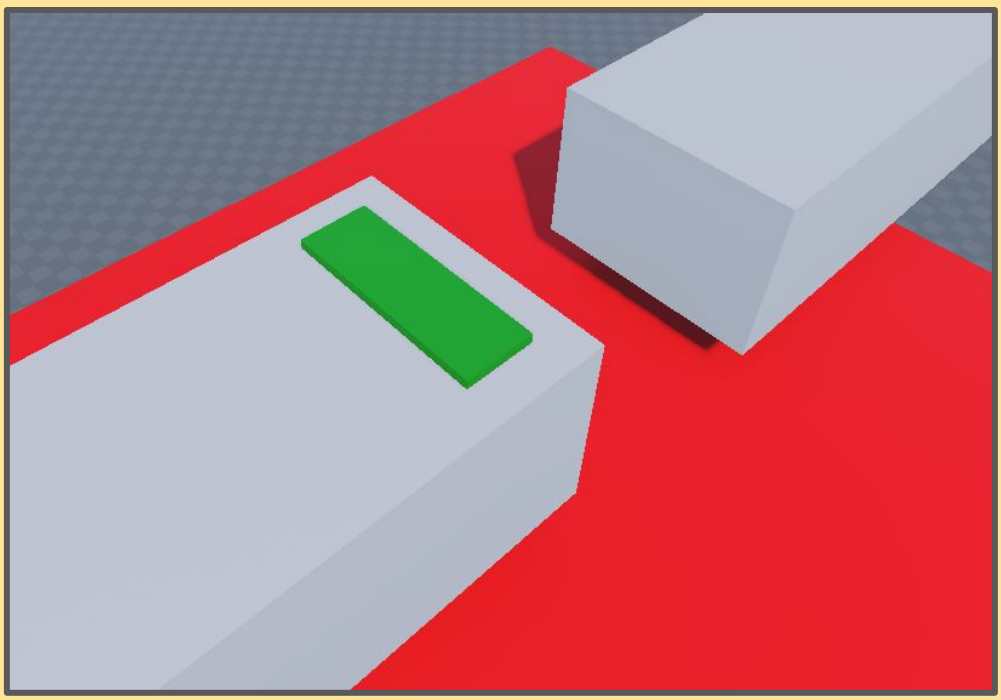
- Improve your understanding of Scripting
- Create checkpoints



REMEMBER: Raise your hand. We love to help!

Checkpoints

1 So far we have made different obstacles. A many feature of an obstacle course is the if you fail you can jump back to the last checkpoint. To start off make a block that will be our **first checkpoint**.



2 Add a **script** to this **block** you just made, when you add the **script** you can also add in the following code. We are sticking with the **same method** of getting the **block itself into a variable** and if you **touch the variable** call a **function**.

```
local checkpoint = script.Parent

function checkPointReached(part)

end

checkpoint.Touched:Connect(checkPointReached)
```

3 This is going to be a large piece of code so we are going to be writing comments to make everything clear. Comments are pieces of writing that your computer will ignore so we use it as explanations for things.

We write a comment by adding 2 dashed (--) then the rest of that line will be considered a comment.

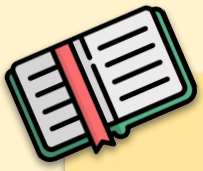
```
local checkpoint = script.Parent

function checkPointReached(part)
    -- check if it is a player
end

checkpoint.Touched:Connect(checkPointReached)
```

4 Next we want to check as usual if it is a player. Write the highlighted code in the correct space. Again this will be a large code file so be careful to do each step correctly.

```
local checkpoint = script.Parent

function checkPointReached(part)

    -- check if it is a player
    if part.Parent:FindFirstChild('Humanoid') then

    end
end

checkpoint.Touched:Connect(checkPointReached)
```

5 Now that we know a player has touched the checkpoint, We now want to get the specific player. Write the comment as well as the line of code to get the player themselves

```
-- check if it is a player
if part.Parent:FindFirstChild('Humanoid') then

    --get said player
    local player = game.Players:GetPlayerFromCharacter(part.Parent)
end
```



Recap break!

So far we have done 3 things;

- Create a function for when we touch the block
- Check if a player has touched the block
- Get the player who touched the block

The code piece so far should look like this

```lua
local checkpoint = script.Parent

function checkPointReached(part)

    -- check if it is a player
    if part.Parent:FindFirstChild('Humanoid') then

        --get said player
        local player = game.Players:GetPlayerFromCharacter(part.Parent)

    end

end

checkpoint.Touched:Connect(checkPointReached)
```

1 Underneath the code that connects to the player, write this code here. This code will connect to this checkpoint in the roblox server

```lua
--get said player
local player = game.Players:GetPlayerFromCharacter(part.Parent)

-- get checkpoint
local checkpointData = game.ServerStorage:FindFirstChild('checkpointData')
```

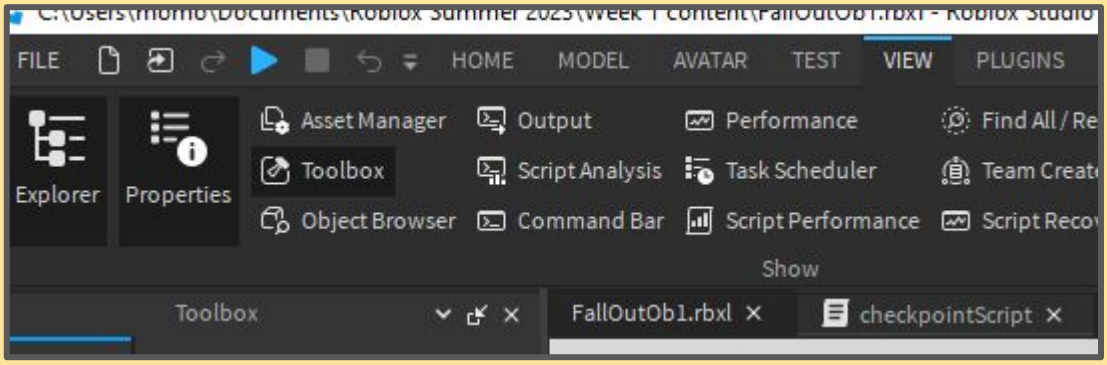
2 The next chunk of code is going directly after the previous line. It's a complicated chunk so make sure to write this EXACTLY the same. The first time you step on the checkpoint you need to add it to the server storage

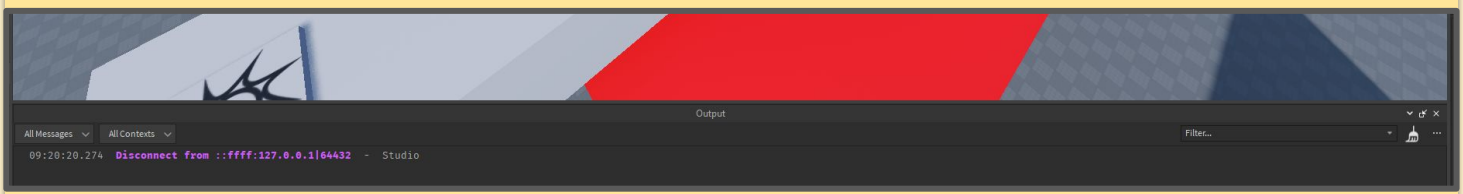
```lua
--make checkpoints if none exist
if not checkpointData then
    checkpointData = Instance.new('Model', game.ServerStorage)
    checkpointData.Name = 'checkpointData'

end
```

3 Similar to the checkpoint, we need to make a savedCheckpoint for the player so we know where they are supposed to revive. Below the checkpoint code write down the following code.

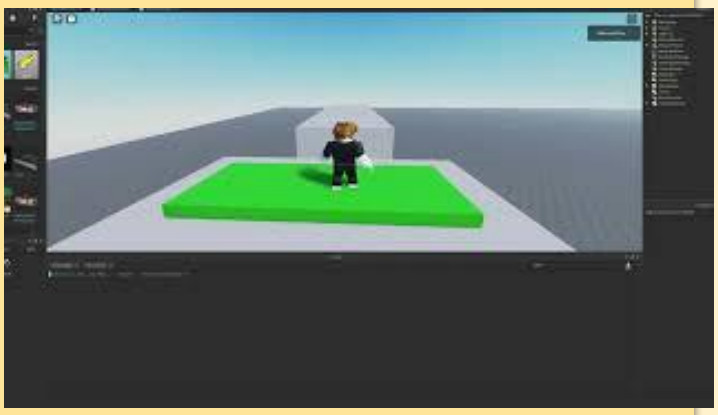
```
--get player and the checkpoint they currently are on
local savedCheckpoint = checkpointData:FindFirstChild(player.Name)

if not savedCheckpoint then

    savedCheckpoint = Instance.new('ObjectValue', checkpointData)
    savedCheckpoint.Name = player.Name
    print("cp new")
end
```

4 The next step is to prepare to test it. We want the output view to be shown. To do this, go to the view menu at the top and look for the output button. When you press it an output box should appear on the bottom of roblox.





5 Now if you press play, when you step on the checkpoint it should print out **cp new** for the first time. Below you will see the code we should have so far incase it's not working



```lua
-- get the checkpoint
local checkpoint = script.Parent

function checkPointReached(part)

    -- check if player
    if part.Parent and part.Parent:FindFirstChild('Humanoid') then

        --get said player
        local player = game.Players:GetPlayerFromCharacter(part.Parent)


        -- get checkpoint
        local checkpointData = game.ServerStorage:FindFirstChild('checkpointData')

        --make checkpoints if none exist
        if not checkpointData then
            checkpointData = Instance.new('Model', game.ServerStorage)
            checkpointData.Name = 'checkpointData'

        end



        --get player and the checkpoint they currently are on
        local savedCheckpoint = checkpointData:FindFirstChild(player.Name)

        if not savedCheckpoint then

            savedCheckpoint = Instance.new('ObjectValue', checkpointData)
            savedCheckpoint.Name = player.Name
            print("cp new")
        end




    end



end

checkpoint.Touched:Connect(checkPointReached)

```

Finishing off the checkpoint

1. You should not move on from here if you don't have the last part working!!!!!

2. To finish off the checkpoints we need to first save the new checkpoint to the player. Insert this line of code when you see it below

```
        --get player and the checkpoint they currently are on
        local savedCheckpoint = checkpointData:FindFirstChild(player.Name)

        if not savedCheckpoint then

            savedCheckpoint = Instance.new('ObjectValue', checkpointData)
            savedCheckpoint.Name = player.Name
            print("cp new")
        end

        --save next checkpoint to the player
        savedCheckpoint.Value = checkpoint


    end


end

checkpoint.Touched:Connect(checkPointReached)
```

The very last step is to add a function right below this line that will tell the player, when they die, revive back to this block

```
--save next checkpoint to the player
savedCheckpoint.Value = checkpoint

--Tell player to respawn at that checkpoint
function goToCheckpoint(character)
    wait()
    local location = savedCheckpoint.Value.CFrame
    --so we revive over rather than in the checkpoint
    character:WaitForChild('HumanoidRootPart').CFrame = location + Vector3.new(0,5,0)
end
player.CharacterAdded:Connect(goToCheckpoint)
```

3 Now you should have a checkpoint that works, place this check point over a kill block and check that you respawn at the correct checkpoint correctly

Challenge Tasks!

You can make as many checkpoints as you want just by duplicating this block now. You should have made an obstacle course by now, go back and add checkpoints throughout the course!