Mini game recipe

Contents

Change properties of spawn location and base plate

1. In toolbox->decals search for ‘Rainbow’. Drag drop it on top of spawn point. Duplicate the decal 5 times and set it to ‘bottom’, ‘front’, ‘back’, ‘left’, ‘right’.
2. Set spawn location, properties, CFrame->Position to (0, 50, 0). Size to (12,1,12).
3. Set baseplate property->transparency to 1. Set baseplate color to ‘Really Red’ Add script to baseplate, using ‘+’ button next to it. And in script add code below.

```
function onTouch(part)
    if script.Parent.BrickColor == BrickColor.new("Really red") then
        local humanoid = part.Parent:FindFirstChild("Humanoid")
        if humanoid ~= nil then
            humanoid.Health =0
        else
            wait(2)
        end
    end
end

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

4. Play and see. As soon as character touches baseplate, he is killed.

Create check points

1. Use '+' next to workspace and create a folder, rename to 'Checkpoints'.
2. In 'Checkpoints' folder, create a part, rename it to 1. Set material to 'Diamond' color to 'Yellow'. Set size to (6,1,8) and position to (9,50,0). Ensure it is anchored.
3. Duplicate 1, rename to 2 and drag it along x-axis (red arrow). Set the color to green. Set position to (70,50,0).
4. Duplicate 2, rename to 3 and drag it along x-axis (red arrow). Set the color to blue. Set position to (118,50,0).
5. Duplicate 3, rename to 4 and drag it along x-axis (red arrow). Set color to red. Set position to (170, 60, 0).
6. Save to file. Play and see. We must see all the checkpoints in front of us.
7. Add a script to 'ServerScriptService' and add the below code

```
local checkpoints = workspace:WaitForChild("Checkpoints")


game.Players.PlayerAdded:Connect(function(player)
    local leaderstats = Instance.new("Folder")
    leaderstats.Name = "leaderstats"
    leaderstats.Parent = player

    local stage = Instance.new("IntValue")
    stage.Name = "Stage"
    stage.Value = 1
    stage.Parent = leaderstats

    player.CharacterAdded:Connect(function(char)
        local hum = char:WaitForChild("Humanoid")
        wait()
        char:MoveTo(checkpoints[stage.Value].Position)
        hum.Touched:Connect(function(hit)
            if hit.Parent == checkpoints then
```

```
                    if tonumber(hit.Name) == stage.Value+1 then
                        stage.Value = stage.Value + 1
                    end
                end
            end)
        end)

end)
```

8. Basically, we are getting a reference to folder checkpoints and storing in a variable. We are then creating folder called ‘leaderstats’ under player and int variable ‘Stage’ in this folder with initial value of 1. When the player gets added to game, we ‘move’ him to checkpoints[stage.Value].Position, which is checkpoints[1].Position when game starts. When the gamer touches another next checkpoint, for example checkpoint 2, we increment stage.Value so that next time he spawns, he will spawn in checkpoint he had reached before.

Create move up and down platforms

1. Create a part and anchor it. Rename to updown1. Give it brick material and whitish-blue color. Set size to (6,1,10) and position to (21,50,0). Using ‘+’ button add a script to it and add below code.

```
local tweenService = game:GetService("TweenService")

local part = script.Parent

local tweenInfo = TweenInfo.new(2,Enum.EasingStyle.Cubic, Enum.EasingDirection.InOut,-1,true)

local tween = tweenService:Create(part, tweenInfo,{

    CFrame = part.CFrame*CFrame.new(0,-10,0)

})

tween:Play()
```

2. Play and see. You must be able to see a platform move up and down. You must be able to time yourself correctly and jump on it and move with it.
3. Duplicate updown1, drag along x axis, rename to updown2. Change color to whitish-green. Position to (34,50,0) In it’s script change TweenInfo duration to 3 and offset to (0,10,0).
4. Similarly create updown3, position = (46,50,0) , duration =1 , offset = (0,-5,0). Any color.
5. Similarly create updown4, position = (58,50,0) , duration =0.5 , offset = (0,5,0). Any color.

6. Play and see. Player must be able to jump and navigate this ‘obby’ and reach check point 2. Once he reaches checkpoint 2, even if he jumps off and gets killed, he/she must respawn in check point 2.

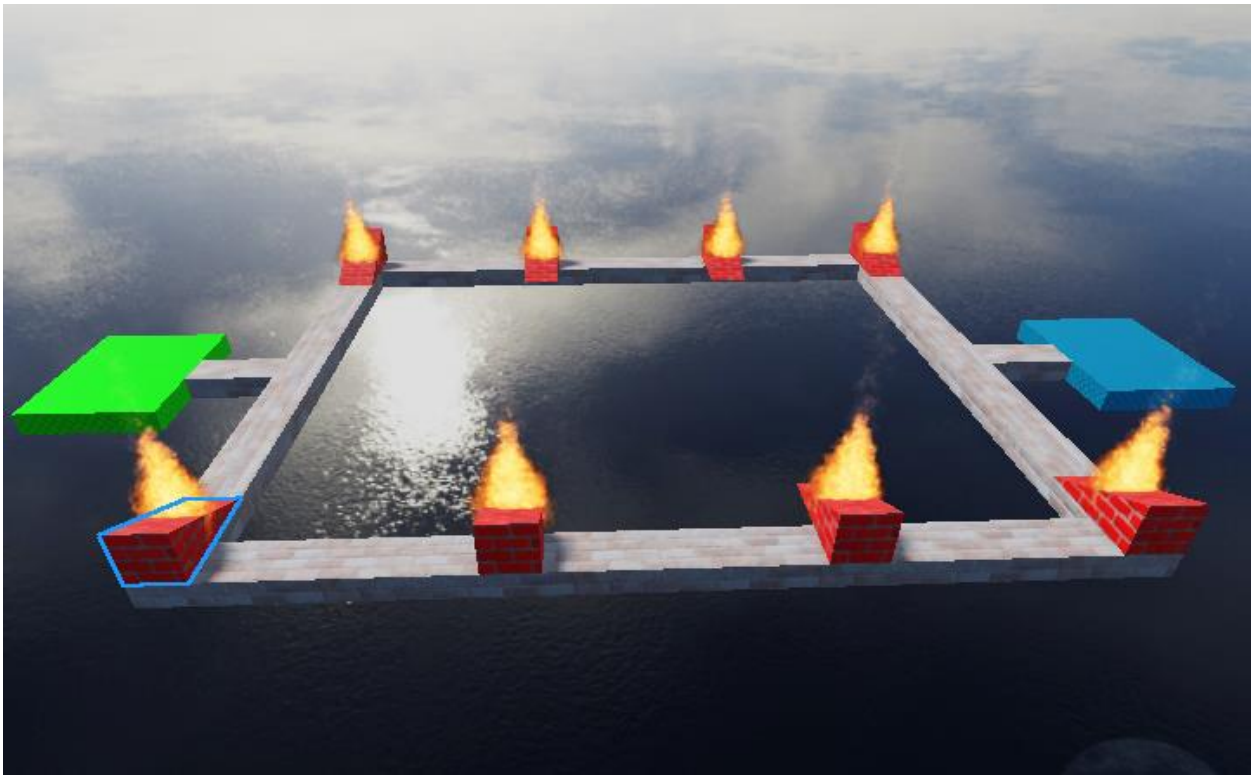
Create the fire path

1. Make below 6 parts with brick material, size and position as given below, name as FP1 etc.
2. Fp1 – size = (4,1,2), position = (75,50,0) and make sure to anchor all Fp1 – Fp6
3. Fp2 – size = (30,1,2), position = (78,50,0) , orientation= (0,90,0)
4. Fp3 – size = (30,1,2), position = (94,50,-14),
5. Fp4 – size = (30,1,2), position = (94,50,14),
6. Fp5 – size = (30,1,2), position = (110,50,0), orientation= (0,90,0)
7. Fp6 – size = (4,1,2), position = (113,50,0)
8. Play and see. Basically, must be able to walk or jump along this and reach the next check point, which is check point 3. And even if player gets killed, he should respawn here.
9. Now, we can design some ‘wedges with fire’ obstacles to place along this path.
10. Create a part->Wedge. Set the Material to brick and color to ‘really red’. Rename to F1. Set Size to (2,2,4.5), and position to (78, 51.5, 12.75).
11. Using ‘+’ button, add ‘Fire’ to it. Using ‘+’ button add script to it and to script add code below.

```
function onTouch(part)
    if script.Parent.BrickColor == BrickColor.new("Really red") then
        local humanoid = part.Parent:FindFirstChild("Humanoid")
        if humanoid ~= nil then
            humanoid.Health =0
        else
            wait(2)
        end
    end
end

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

12. Use duplicate, rename, rotate and move to place other ‘fire obbys’ along the path as shown in the picture below.

13. 
14. Play and see.
15. If gamer touches the wedge with fire, he will be killed. Jump above them and cross the ‘fire obby’.

Create the trampoline

1. Create part->cylinder. Rename to T1 (trampoline 1). Make the material rubber color black. Set position to (128,50,0) and size(0.6,3,3). Anchor it. Set AssemblyLinearVelocity to (0,200,0).
2. Duplicate it, rename to T2. Set position to (142,58,0) and size to (0.6,7,7).
3. Create a wedge, size(15,15,15). Give it red color and diamond material. Anchor it. Position to (159,52,0)
4. Play and see, you must be able to hit spacebar (jump) exactly when you touch trampoline, jump very high and then jump onto next trampoline and finally to check point 4.

Win

1. Use the ‘+’ next the ‘4’ in checkpoint and add a script. In script add below code.

```
function onTouch(part)

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

    if humanoid ~= nil then
```

```
            print('You have won the game!')
            humanoid.WalkSpeed = 0
    end
end

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

2. Play and see, when you reached the check point 4, you must see the message in output!
3. Now create a ScreenGui under StarterGui.
4. Under ScreenGui create a frame. Set Frame's background color to black, background transparency to 0.3. Rename the frame to WinInfo.
5. Under WinInfo, add a text label that says 'You have won the game!'. Adjust the size, color, font.
6. Set the WinInfo, property Visible to unchecked. Click on workspace.
7. In the '4' script, instead of printing out a message, add below code to make the WinInfo visible.

```
8.          local player = game.Players:FindFirstChild(part.Parent.Name)
9.          if player then
10.             player.PlayerGui.ScreenGui.WinInfo.Visible = true
11.         end
```

12. Play and see the information – You have won the game! – as soon as you reach checkpoint 4.
13. Enjoy!