

Roblox 1 : In-Game Players

Table of Contents

- [Players](#)
 - [Players' Tools](#)

Players

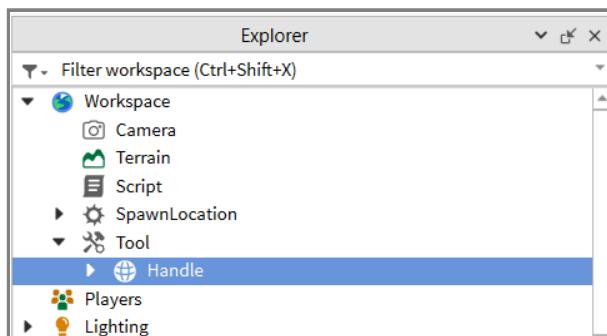
In Roblox, when a user enters an experience, they are represented by a **Player** object in the data model, which holds essential information like their username, friend list, and membership type across different experiences. The **Players** service manages all **Player** instances and provides access to key containers, allowing developers to customize the user's experience within the game environment.

Players' Tools

Within Roblox experiences, **Tools** offer users interactive items such as swords, rocket launchers, and magic wands for engaging gameplay sessions. Developers have the creative freedom to craft personalized tools, seamlessly integrate them into their experience, and script unique functionalities. To incorporate tools into your experience, follow these steps :

1. Go to the **Explorer** window to click the **+** button next to **Workspace** and add **Tool**.
2. Under **Tool** add a **MeshPart**. To do this, click the **+** button next **Tool**.
3. Rename the **MeshPart** to **Handle**

 It is very important to rename the **Meshpart** to **Handle** with capital **H** or else it would not work.



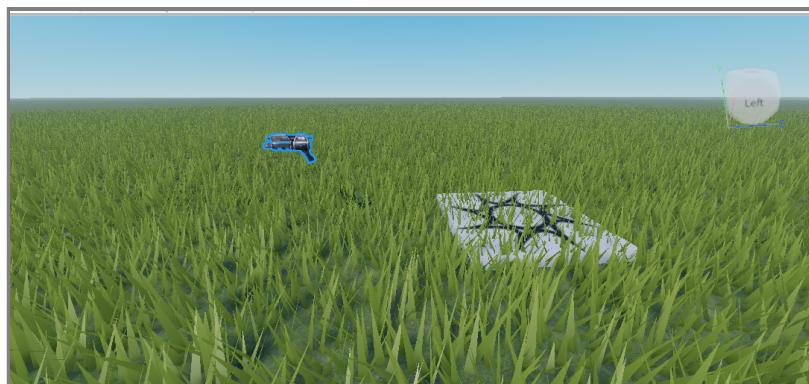
4. Open the **Properties** window of the **Handle** and paste these values in **MeshID** and **TextureID** in the **Appearance** section.

```
meshID : rbxassetid://92656610
```

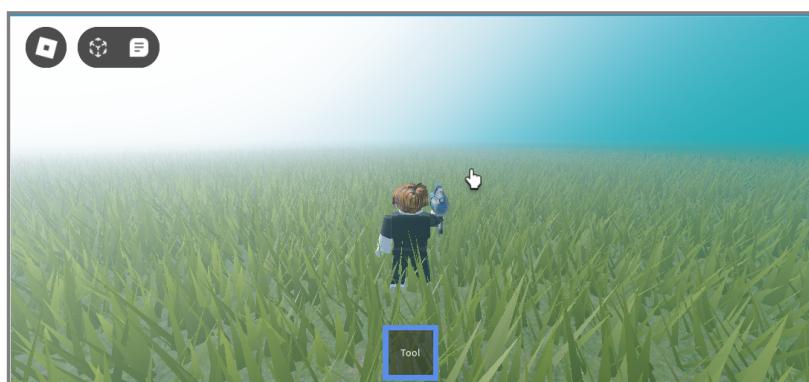
TextureID : rbxassetid://92658105

| Appearance | |
|-----------------|-------------------------------------|
| BrickColor | Medium stone grey |
| CastShadow | <input checked="" type="checkbox"/> |
| Color | [163, 162, 165] (Medium ston...) |
| DoubleSided | <input type="checkbox"/> |
| Material | Plastic |
| MaterialVariant | |
| MeshId | rbxassetid://92656610 |
| Reflectance | 0 |
| RenderFidelity | Precise |
| TextureID | rbxassetid://92658105 |
| Transparency | 0 |

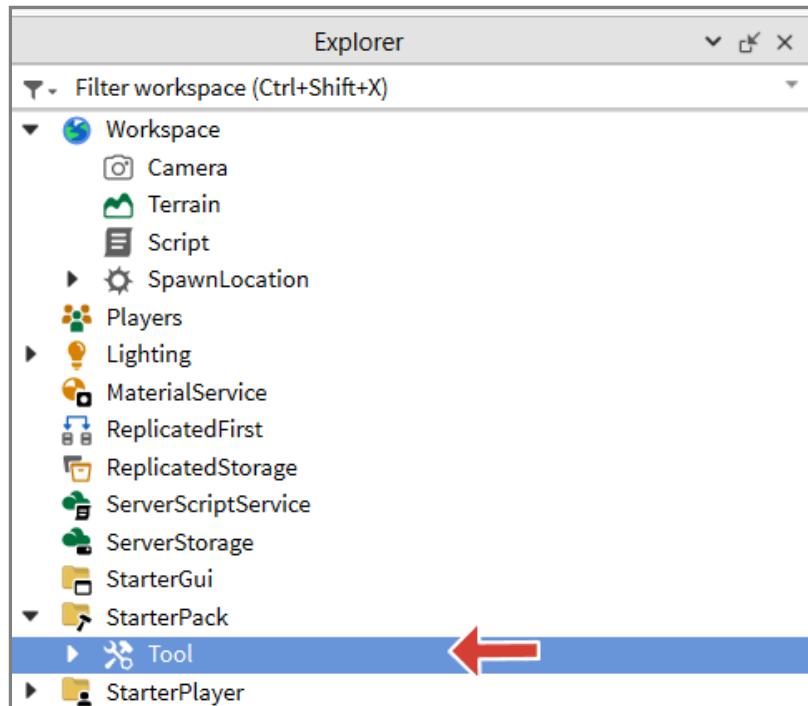
5. Look into your main map, make sure that the **Tools** is located near your **Spawn Location**. This is so that we can pickup the Tools easily without needing to walk for a long time.



6. Run the game and walk into the **Tools** to pick it up. You can also see now that the **Tools** have become a part of your **Inventory**.



7. If you want the tools to be already in your inventory when you start the game, you can drag the **Tools** from the **Workspace** to **StarterPack** in the **Explorer** windows.



8. Run the game again and now you can equip the tools by pressing on the box at the bottom of the game or by pressing **1**.

Leaderboard

In Roblox, leaderboards serve as an essential component of gameplay, providing players with a visual representation of their progress and achievements within a game. Leaderboards typically display various statistics, such as player scores, currency, or other custom metrics, allowing players to compare their performance with others.

1. To add a leaderboard to your game, add a **Script** to the **ServerScriptService**.
2. Open the script and type the below:

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

-- Function to set up leaderboard for a player
local function leaderboardSetup(player)
    local leaderstats = Instance.new("Folder")
    leaderstats.Name = "leaderstats"
    leaderstats.Parent = player
end

-- Connect the leaderboardSetup function to the PlayerAdded event
Players.PlayerAdded:Connect(leaderboardSetup)
```

 It is important to assign the value of `leaderstats.name` as "`leaderstats`", with all lowercase letters, with that spelling or else it would not work.

3. Within the `leaderboardSetup` function, add code to create leaderboard stats:

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

local function leaderboardSetup(player)
    local leaderstats = Instance.new("Folder")
    leaderstats.Name = "leaderstats"
    leaderstats.Parent = player

    local score = Instance.new("IntValue")
    score.Name = "Score"
    score.Value = 0
    score.Parent = leaderstats
end

Players.PlayerAdded:Connect(leaderboardSetup)
```

4. You can update the player's score every time they hit a part in the game. Here's an example of how to do this:

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

local function setupLeaderboard(player)
    local leaderstats = Instance.new("Folder")
    leaderstats.Name = "leaderstats"
    leaderstats.Parent = player

    local score = Instance.new("IntValue")
    score.Name = "Score"
    score.Value = 0
    score.Parent = leaderstats

    local function onPartHit(part)
        -- Check if the part was hit by the player's character
        local character = player.Character
        if character then
            -- Increment the player's score
            score.Value = score.Value + 1
        end
    end

    -- Connect the onPartHit function to the Touched event of all parts in the
    -- game
    for _, part in ipairs(game:GetService("Workspace"):GetDescendants()) do
        if part:IsA("Part") then
            part.Touched:Connect(function(hit)
                onPartHit(part)
            end)
        end
    end
end
```

```
end  
end
```

```
Players.PlayerAdded:Connect(setupLeaderboard)
```