Skip to content

ACOKing/ACTIONS\_DOWN.lua

Created May 23, 2017 22:04 • Report abuse

Code

Revisions 1

Weapon System (USFBP II)

ACTIONS\_DOWN.lua

local Actions = {

["Reload"] = function(S)

if (not S.Reloading) and (not S.isCrawling) then

S.Reload()

end

end;

["Sprint"] = function(S)

S.setRunKeyPressed(true)

if S.runReady then

if S.CanRun() then

S.monitorStamina()

end

end

end;

["ADS"] = function(S)

if S.holdToADS then

if S.hasNotAimedYet() then

S.holdAndAim()

end

else

S.toggleAim()

end

end;

["raiseStance"] = function(S)

if S.canChangeStance then

if not S.Running then

if S.isProne() then

if S.canCrouch then

S.changeStance("Crouch")

else

S.changeStance("Stand")

end

elseif S.isCrouching() then

S.changeStance("Stand")

end

end

end

end;

["lowerStance"] = function(S)

if S.canChangeStance then

if (not S.Running) then

if S.isStanding() then

if S.canCrouch then

S.changeStance("Crouch")

elseif S.canGoProne then

S.changeStance("Prone")

end

elseif S.isCrouching() then

if S.canGoProne then

S.changeStance("Prone")

end

end

elseif S.dolphinDive then

-- S.Humanoid.Jump = true

wait()

if S.canDive() then

local tempConnection = S.Humanoid.Changed:connect(function()

S.Humanoid.Jump = false

end)

S.runReady = false

S.changeStance("Dive")

S.Running = false

wait(S.rechargeTime)

tempConnection:disconnect()

S.runReady = true

end

end

end

end;

["selectFire"] = function(S)

if S.selectFire then

S.canSelectFire = false

S.changeMode()

S.setFireFunction()

local speedAlpha = S.selectFireAnimSpeed / 0.6

if S.hasSelectFireGUI then

S.runAsync(function()

S.toggleFireSelect(true)

local prevRawFireMode = S.getRawFireMode()

S.updateGUI("ModeLabels",((S.getRawFireMode() - 1) % S.numModes))

S.fireModes.FireModeController.Stopped:wait()

S.toggleFireSelect(false)

end)

end

if S.ableToSelectFire() then

S.runAsync(function()

local sequenceTable = {

function()

S.tween("Joint",S.RWeld, nil, S.CFANG(0, S.RAD(5), 0), S.Sine, speedAlpha \* 0.15)

S.tween("Joint",S.LWeld, S.LC0, S.CF(0.1, 1, -0.3) \* S.CFANG(S.RAD(-7), 0, S.RAD(-65)), S.Linear, speedAlpha \* 0.15)

wait(speedAlpha \* 0.2)

end;

function()

S.tween("Joint",S.LWeld, S.LC0, S.CF(0.1, 1, -0.3) \* S.CFANG(S.RAD(-10), 0, S.RAD(-65)), S.Linear, speedAlpha \* 0.1)

wait(speedAlpha \* 0.2)

end;

function()

S.tween("Joint",S.RWeld, nil, S.CF(), S.Sine, speedAlpha \* 0.2)

S.tween("Joint",S.LWeld, S.LC0, S.unAimedLeftArmC1, S.Sine, speedAlpha \* 0.2)

wait(speedAlpha \* 0.2)

end;

}

for \_, F in S.PAIRS(sequenceTable) do

if S.unAbleToSelectFire() then

break

end

F()

end

end)

end

if S.hasSelectFireMedia() then

wait(S.selectFireMediaSpeed)

end

S.canSelectFire = true

end

end;

["scopeSteady"] = function(S)

S.setupSteady()

if S.readyToAim() then

S.steadyCamera()

end

end;

}

return Actions

ACTIONS\_UP.lua

local Actions = {

["ADS"] = function(S)

if S.holdToADS then

if S.hasNotAimedOutYet() then

S.unAim()

end

end

end;

["Sprint"] = function(S)

S.stopRunning()

end;

["scopeSteady"] = function(S)

S.unSteadyCamera()

end;

}

return Actions

ANIMATIONS.lua

local Animations = {

Reload = function(S)

local W1 = nil

local W2 = nil

local animSpeed = S.isMagEmpty() and S.reloadTimeEmpty / 1.3 or S.reloadTimeLoaded / 0.9

return {

function()

if (not S.isNewMag()) then

if S.isMagVisible() then

local Mag1, magTable1 = S.createMag("Mag1")

Mag1.Parent = S.gunIgnore

W1 = Instance.new("Motor6D")

W1.Part0 = magTable1[1].magClone

W1.Part1 = S.Handle

W1.C0 = magTable1[1].Original.CFrame:toObjectSpace(S.Handle.CFrame)

W1.Parent = magTable1[1].magClone

S.tweenJoint(S.LWeld, nil, S.CF(-0.9, 2, -1.1) \* S.CFANG(S.RAD(-15), 0, S.RAD(-25)), S.Sine, 0.2 \* animSpeed)

S.tweenJoint(S.RWeld, nil, S.CF(0.3, 0.2, -0.31) \* S.CFANG(S.RAD(-12), 0, S.RAD(25)), S.Sine, 0.2 \* animSpeed)

S.tweenJoint(S.Grip, nil, S.CFANG(0, S.RAD(20), S.RAD(10)), S.Sine, 0.1 \* animSpeed)

wait(0.2 \* animSpeed)

end

end

end;

function()

if (not S.isNewMag()) then

if S.isMagVisible() then

S.makeMagInvisible()

W1:Destroy()

local Mag1, magTable1 = S.getMag("Mag1")

magTable1[1].magClone.Velocity = S.Handle.Velocity + S.Handle.CFrame:vectorToWorldSpace(S.V3(0, -1, 0)) \* 20

S.tweenJoint(S.RWeld, nil, S.CF(0.3, 0.2, -0.5) \* S.CFANG(S.RAD(-20), S.RAD(10), S.RAD(25)), S.Sine, 0.25 \* animSpeed)

S.tweenJoint(S.Grip, nil, S.CFANG(0, S.RAD(20), S.RAD(10)), S.Sine, 0.2 \* animSpeed)

else

S.tweenJoint(S.RWeld, nil, S.CF(0.6, 0.2, -0.61) \* S.CFANG(S.RAD(-12), S.RAD(20), S.RAD(25)), S.Sine, 0.2 \* animSpeed)

S.tweenJoint(S.Grip, nil, S.CFANG(S.RAD(-10), S.RAD(20), S.RAD(10)), S.Sine, 0.2 \* animSpeed)

end

S.tweenJoint(S.LWeld, nil, S.CF(0, 0.5, 0) \* S.CFANG(S.RAD(95), 0, S.RAD(-25)), S.Sine, 0.2 \* animSpeed)

wait(0.25 \* animSpeed)

end

end;

function()

if (not S.isNewMag()) then

local Mag1, magTable1 = S.getMag("Mag1")

if Mag1 then Mag1:Destroy() end

local Mag2, magTable2 = S.createMag("Mag2")

Mag2.Parent = S.gunIgnore

local LArmCF = S.LWeld.Part0.CFrame \* S.LWeld.C0 \* (S.CF(0.58, 1.63, -1.4) \* S.CFANG(S.RAD(-22), S.RAD(20), S.RAD(-60))):inverse()

local RArmCF = S.RWeld.Part0.CFrame \* S.RWeld.C0 \* (S.CF(0.6, 0.2, -0.61) \* S.CFANG(S.RAD(-15), S.RAD(20), S.RAD(25))):inverse()

local handleOffsetCF = S.RArm.CFrame:toObjectSpace(S.RArm.CFrame \* S.Grip.C0 \* (S.CFANG(S.RAD(-10), S.RAD(20), S.RAD(10))):inverse())

local originalMagOffsetCF = S.Handle.CFrame:toObjectSpace(magTable2[1].Original.CFrame)

local newMagC0 = LArmCF:toObjectSpace(RArmCF \* handleOffsetCF \* originalMagOffsetCF)

W2 = Instance.new("Motor6D")

W2.Part0 = S.LArm

W2.Part1 = magTable2[1].magClone

W2.C0 = newMagC0

W2.Parent = magTable2[1].magClone

S.tweenJoint(S.LWeld, nil, S.CF(0.55, 1, -2.4) \* S.CFANG(S.RAD(-20), S.RAD(20), S.RAD(-60)), S.Sine, 0.2 \* animSpeed)--0.25

S.tweenJoint(S.RWeld, nil, S.CF(0.6, 0.2, -0.61) \* S.CFANG(S.RAD(-12), S.RAD(20), S.RAD(25)), S.Sine, 0.2 \* animSpeed)

S.tweenJoint(S.Grip, nil, S.CFANG(S.RAD(-10), S.RAD(20), S.RAD(10)), S.Sine, 0.2 \* animSpeed)

wait(0.2 \* animSpeed)

end

end;

function()

if (not S.isNewMag()) then

S.tweenJoint(S.Grip, nil, S.CFANG(S.RAD(-10), S.RAD(20), S.RAD(10)), S.Sine, 0.15 \* animSpeed)

S.tweenJoint(S.LWeld, nil, S.CF(0.58, 1.63, -1.4) \* S.CFANG(S.RAD(-22), S.RAD(20), S.RAD(-60)), S.Sine, 0.15 \* animSpeed)--0.25

S.tweenJoint(S.RWeld, nil, S.CF(0.6, 0.2, -0.61) \* S.CFANG(S.RAD(-15), S.RAD(20), S.RAD(25)), S.Sine, 0.2 \* animSpeed)

wait(0.2 \* animSpeed)

end

end;

function()

if (not S.isNewMag()) then

local Mag1, \_ = S.getMag("Mag1")

local Mag2, \_ = S.getMag("Mag2")

S.makeMagVisible()

S.setNewMag()

if Mag1 then Mag1:Destroy() end

Mag2:Destroy()

end

end;

function()

if S.isMagEmpty() then

if S.isNewMag() then

S.tweenJoint(S.Grip, nil, S.CFANG(S.RAD(-10), S.RAD(20), S.RAD(10)), S.Sine, 0.15 \* animSpeed)

S.tweenJoint(S.LWeld, nil, S.CF(0.58, 1.63, -1.4) \* S.CFANG(S.RAD(-22), S.RAD(20), S.RAD(-60)), S.Sine, 0.15 \* animSpeed)--0.25

S.tweenJoint(S.RWeld, nil, S.CF(0.6, 0.2, -0.61) \* S.CFANG(S.RAD(-15), S.RAD(20), S.RAD(25)), S.Sine, 0.2 \* animSpeed)

end

S.tweenJoint(S.LWeld, nil, S.CF(0, 1.3, -0.55) \* S.CFANG(S.RAD(-26), 0, 0), S.Sine, 0.2 \* animSpeed)

wait(0.2 \* animSpeed)

end

end;

function()

if S.isMagEmpty() then

S.tweenJoint(S.LWeld, nil, S.CF(0.4, 1.6, -0.55) \* S.CFANG(S.RAD(-23), 0, S.RAD(-60)), S.Sine, 0.1 \* animSpeed)

wait(0.05 \* animSpeed)

end

end;

function()

if S.isMagEmpty() then

S.tweenJoint(S.RWeld, nil, S.CF(0.6, 0.2, -0.61) \* S.CFANG(S.RAD(-15), S.RAD(20), S.RAD(20)), S.Sine, 0.05 \* animSpeed)

wait(0.15 \* animSpeed)

end

end;

}

end;

Cocking = function(S)

end;

Crawling = function(X, moveDirection, moveSpeed)

return {

leftArm = CFrame.Angles(

0,

math.rad(90),

math.rad(-10)

) \* CFrame.new(

math.sin(moveDirection) \* (math.sin(X \* 6) / 4) - 0.2,

math.cos(moveDirection) \* (math.sin(X \* 6) / 2) - 0.1,

math.max(math.cos(X \* 6) / 4, 0) - 0.1

) \* CFrame.Angles(

-math.max(math.cos(X \* 6) / 4, 0),

0,

0

);

leftLeg = CFrame.new(

math.sin(moveDirection) \* (-math.sin(X \* 6) / 4) - 0.2,

math.cos(moveDirection) \* (math.sin(X \* 6) / 2) + 0.3,

math.max(math.cos(X \* 6) / 4, 0) - 0.1

):inverse() \* CFrame.Angles(

0,

0,

-math.rad(15) - math.cos(moveDirection) \* (math.rad(15) \* math.sin(X \* 6))

);

rightArm = CFrame.Angles(

0,

math.rad(-5),

math.rad(10)

) \* CFrame.new(

math.sin(moveDirection) \* (-math.sin(X \* 6) / 4) + 0.2,

math.cos(moveDirection) \* (-math.sin(X \* 6) / 5) - 0.2,

math.max(math.cos((X + math.rad(30)) \* 6) / 10, 0) - 0.1

) \* CFrame.Angles(

-math.max(math.cos((X + math.rad(30)) \* 6) / 10, 0),

0,

0

);

rightLeg = CFrame.new(

math.sin(moveDirection) \* (math.sin(X \* 6) / 4) + 0.2,

math.cos(moveDirection) \* (-math.sin(X \* 6) / 2) + 0.3,

math.max(math.cos((X + math.rad(30)) \* 6) / 4, 0) - 0.1

):inverse() \* CFrame.Angles(

0,

0,

math.rad(15) - math.cos(moveDirection) \* (math.rad(15) \* math.sin(X \* 6))

);

Grip = CFrame.Angles(

math.max(math.cos((X + math.rad(30)) \* 6) / 7, 0),

math.rad(5),

0

);

Camera = 1.5 \* math.rad(math.cos((X + math.rad(30)) \* 6)) + math.rad(0.5); --This is what the roll of the camera will be when you're crawling

}

end;

Idling = {

unAimed = function(X)

return {

Pos = Vector3.new(

math.sin(X / 2) / 70,

math.sin(X \* 5 / 4) / 70,

math.sin(X \* 3 / 4) / 70

);

Rot = Vector3.new(

0,

0,

0

);

}

end;

Aimed = function(X)

return {

Pos = Vector3.new(

math.sin(X \* 3 / 8) / 140,

math.sin(X \* 15 / 16) / 140,

0

);

Rot = Vector3.new(

0,

0,

0

);

}

end;

};

Walking = {

unAimed = function(X)

return {

Pos = Vector3.new(

4 \* math.sin(X \* 4.5) / 50,

1.5 \* math.sin(X \* 9) / 50,

0

);

Rot = Vector3.new(

0,

0,

math.rad(math.sin(X \* 4.5)) \* 2

);

}

end;

Aimed = function(X)

return {

Pos = Vector3.new(

2 \* math.sin(X \* 3) / 150,

0.75 \* math.sin(X \* 6) / 150,

0

);

Rot = Vector3.new(

0,

0,

math.rad(math.sin(X \* 3)) / 3

);

}

end;

};

Running = function(X)

return {

Pos = Vector3.new(

4 \* math.sin(X \* 4.5 \* 1.5) / 30,

1.5 \* math.sin(X \* 9 \* 1.5) / 40 + 0.2,

0

);

Rot = Vector3.new(

0,

-math.rad(math.sin(X \* 4.5 \* 1.5)) \* 5 + math.rad(3),

math.rad(math.sin(X \* 4.5 \* 1.5)) \* 5

);

}

end;

}

return Animations

clientMain.lua

--[[

Kit By TurboFusion

Remake By MuYhEt & Xander521

--]]

--------------------------------------------------------------------------------------

--------------------[ CHARACTER LOADING ]---------------------------------------------

--------------------------------------------------------------------------------------

local FFC = game.FindFirstChild

local WFC = game.WaitForChild

local Services = {}

Services.DS = game:GetService("Debris")

Services.CP = game:GetService("ContentProvider")

Services.RS = game:GetService("RunService")

Services.UIS = game:GetService("UserInputService")

Services.CAS = game.ContextActionService

Services.WS = workspace

Services.RepStorage = game.ReplicatedStorage

Services.KIAPI = require(Services.RepStorage.KeyInputAPI)

Services.Plrs = game.Players

repeat wait() until Services.Plrs.LocalPlayer.Character

repeat wait() until Services.Plrs.LocalPlayer.Character:IsDescendantOf(Services.WS)

wait(1 \* 0.05)

--------------------------------------------------------------------------------------

--------------------[ IGNORE MODEL ]--------------------------------------------------

--------------------------------------------------------------------------------------

local WalkingGamepad =false

local ignoreCode = WFC(script,"ignoreCode")

local TICK,UDIM2 = tick,UDim2.new

repeat wait() until ignoreCode.Value ~= 0

local ignoreModel = WFC(Services.WS,"ignoreModel\_"..ignoreCode.Value)

local RPGM = require(Services.RepStorage.RPGMathProvider)

local gbl = require(Services.RepStorage.Global)

local AttributeEffect =require(Services.WS.Settings.AttributeEffect)

local PerLevel = require(Services.WS.Settings.PerLevel)

local grenadeFolder = WFC(ignoreModel,"grenadeFolder")

--local KI = require(game.ReplicatedStorage.KeyInputAPI)

local GunMath = {}

--------------------------------------------------------------------------------------

--------------------[ CONSTANTS ]-----------------------------------------------------

--------------------------------------------------------------------------------------

local shortWait = Services.RS.RenderStepped.wait

local Gun = script.Parent

local serverMain = WFC(Gun,"serverMain",200)

local Handle = WFC(Gun,"Handle")

local AimPart = WFC(Gun,"AimPart")

local Main = WFC(Gun,"Main")

local Ammo = WFC(Gun,"Ammo")

local ClipSize = WFC(Gun,"ClipSize")

local StoredAmmo = WFC(Gun,"StoredAmmo")

local Libraries = {}

Libraries.Network = require(WFC(Services.RepStorage.RemoteService,"Plugin\_Gun",200))(game.Players.LocalPlayer,script.Parent.Name,true,Gun)

Libraries.GLFolder = Services.RepStorage:WaitForChild("GunLibraries",200);

Libraries.Particle = require(WFC(Libraries.GLFolder,"Particle",200));

Libraries.Spring = require(WFC(Libraries.GLFolder,"Spring",200));

Libraries.Anims = require(WFC(Gun,"ANIMATIONS",200));

Libraries.Plugins = require(WFC(Gun,"PLUGINS",200));

Libraries.Keybinds = require(WFC(Gun,"KEYBINDS",200));

Libraries.DownActions = require(WFC(Gun,"ACTIONS\_DOWN",200));

Libraries.UpActions = require(WFC(Gun,"ACTIONS\_UP",200));

S = require(WFC(Gun,"SETTINGS",200));

Libraries.PrimitiveAnim = require(WFC(Libraries.GLFolder,"PrimitiveAnim",600))

local Player = game.Players.LocalPlayer

local Char = Player.Character

Humanoid = Char:WaitForChild("Humanoid",200)

Torso = Char:WaitForChild("Torso",200)

Head = Char:WaitForChild("Head",200);

HRP = Char:WaitForChild("HumanoidRootPart",200)

Root = HRP:WaitForChild("RootJoint",200)

Neck = Torso:WaitForChild("Neck",200)

LArm = Char:WaitForChild("Left Arm",200)

RArm = Char:WaitForChild("Right Arm",200)

LLeg = Char:WaitForChild("Left Leg",200)

RLeg = Char:WaitForChild("Right Leg",200)

local fakeLArm,fakeRArm

local mainGUI = Player.PlayerGui:WaitForChild("blasterHUD",200)

local runAsync = function(threadFunc)

coroutine.resume(coroutine.create(threadFunc))

end

local PenetrationTries = S.penetrationSettings.maxTries

local ArmWelds = {

LWeld = nil;

RWeld = nil;

}

local KeyInputConnections = {}

local Activators = {}

local Interface = {

Cam = game.Workspace.CurrentCamera;

crossHair = mainGUI:WaitForChild("crossHair");

HUD = ( not Services.UIS.TouchEnabled and mainGUI:WaitForChild("HUD") or mainGUI:WaitForChild("MobileHUD"));

Scope = mainGUI:WaitForChild("Scope");

fireSelect = mainGUI:WaitForChild("fireSelect");

hitMarker = mainGUI:WaitForChild("hitMarker");

Sens = mainGUI:WaitForChild("Sens");

M2 = Player:GetMouse()

}

local FE = Services.WS.FilteringEnabled

local scopeMain = Interface.Scope:WaitForChild("Main")

local scopeSteady = Interface.Scope:WaitForChild("Steady")

local fireModes = Interface.fireSelect:WaitForChild("Modes")

local modeGUI = (not Services.UIS.TouchEnabled and FFC(Interface.HUD,"Mode"):FindFirstChild("Main") or Interface.HUD:WaitForChild("Mode"))

local clipAmmoGUI =(not Services.UIS.TouchEnabled and FFC(Interface.HUD,"Ammo"):FindFirstChild("Clip") or Interface.HUD:WaitForChild("Ammo"))

local AmmoBar =(not Services.UIS.TouchEnabled and Interface.HUD:WaitForChild("Ammo"):WaitForChild("AmmoBar2",20) or nil)

local crossA = Interface.crossHair:WaitForChild("A"):WaitForChild("Line")

local crossB = Interface.crossHair:WaitForChild("B"):WaitForChild("Line")

local crossC = Interface.crossHair:WaitForChild("C"):WaitForChild("Line")

local crossD = Interface.crossHair:WaitForChild("D"):WaitForChild("Line")

local VantagePoint = WFC(scopeMain,"VantagePoint",200)

local MIN,MAX = math.min,math.max

local ABS, HUGE, FLOOR, CEIL = math.abs, math.huge, math.floor, math.ceil

local RAD, SIN, COS, TAN = math.rad, math.sin, math.cos, math.tan

local V3 = {

RAW = Vector3.new;

ID = Vector3.new();

}

V3.lerp = V3.ID.lerp

local VEC2 = Vector2.new

local raycast = workspace.FindPartOnRayWithIgnoreList

local CF = {

RAW = CFrame.new,

ANG = CFrame.Angles,

ID = CFrame.new();

Inverse = CFrame.new().inverse,

TOS = CFrame.new().toObjectSpace

}

CF.FAxA = function(x,y,z)

if not y then

x,y,z=x.x,x.y,x.z

end

local m=(x\*x+y\*y+z\*z)^0.5

if m>1e-5 then

local si=SIN(m/2)/m

return CF.RAW(0,0,0,si\*x,si\*y,si\*z,COS(m/2))

else

return CF.ID

end

end

local RSEED = math.randomseed

ATAN,RANDOM = math.atan,math.random

local ATAN2 = function(x, y)

local r = ATAN(y / x)

if x < 0 then

r = r + math.pi

end

return r

end

local MeshTypes = Enum.MeshType

local OBJ = {

RAW = Instance.new;

Clone = game.Clone;

GetKids = game.GetChildren;

Destroy = game.Destroy

}

local SoundLib = {

Play = OBJ.RAW("Sound").Play;

}

local BC,Fonts,FontSizes = BrickColor.new,Enum.Font,Enum.FontSize

local C3,CameraTypes = Color3.new,Enum.CameraType

local Materials,InputTypes = Enum.Material,Enum.UserInputType

local INSERT = function(tableObj,item)

tableObj[#tableObj+1] = item

end

local REMOVE = function(tableObj,i)

tableObj[i] = nil

end

local PAIRS = pairs

local function tagHumanoid(humanoid, player, damage, previousHealth)

local creator\_tag = OBJ.RAW("ObjectValue")

creator\_tag.Value = player

creator\_tag.Name = "creator"

creator\_tag.Parent = humanoid

local weapon\_tag = OBJ.RAW("StringValue")

weapon\_tag.Name = "weapon"

weapon\_tag.Value = script.Parent.Name

weapon\_tag.Parent = creator\_tag

local damage\_tag = OBJ.RAW("IntValue")

if humanoid.Health < 1 then

damage\_tag.Value = CEIL(previousHealth)

else

damage\_tag.Value = CEIL(damage)

end

damage\_tag.Name = "damage"

damage\_tag.Parent = creator\_tag

end

local maxStamina = Player.Stats.MaxStamina.Value

local maxSteadyTime = S.scopeSettings.steadyTime \* 60

local armC0 = {

CF.RAW(-1.5, 0, 0) \* CF.ANG(RAD(90), 0, 0);

CF.RAW(1.5, 0, 0) \* CF.ANG(RAD(90), 0, 0);

}

local legC0 = {

Stand = {

CF.RAW(-0.5, -2, 0);

CF.RAW(0.5, -2, 0);

};

Crouch = {

CF.RAW(-0.5, -1.5, 0.5) \* CF.ANG(-RAD(90), 0, 0);

CF.RAW(0.5, -1, -0.75);

};

Prone = {

CF.RAW(-0.5, -2, 0);

CF.RAW(0.5, -2, 0);

};

}

local Ignore = {

Char;

ignoreModel;

}

local Shoulders = {

Right = Torso:WaitForChild("Right Shoulder");

Left = Torso:WaitForChild("Left Shoulder");

}

local Sine = function(X)

return SIN(RAD(X))

end

local Linear = function(X)

return (X \* 0.0111111111111)

end

local Quad = function(X)

return (X \* 0.0111111111111)^2

end

local RAY = Ray.new

--------------------------------------------------------------------------------------

--------------------[ VARIABLES ]-----------------------------------------------------

--------------------------------------------------------------------------------------

local playerMass = 0

local Forward = false

local Backward = false

local Idling = false

local Walking = false

local Running = false

local crawlCamRot = 0

local crawlAlpha = 0

local idleAlpha = 1

local walkAlpha = 0

local isCrawling = false

local isIdling = false

local isWalking = false

local isRunning = false

local Aimed = false

local Aiming = false

local aimAlpha = 0

local headOffset = VEC2(COS(RAD(90) - S.aimSettings.headTilt) \* 0.5, 1 + SIN(RAD(90) - S.aimSettings.headTilt) \* 0.5)

local Reloading = false

local breakReload = false

local magVisible = true

local newMag = false

local Knifing = false

local MB1Down = false

local Firing = false

local canFire = true

local fireFunction = nil

local firstShot = false

local shotCount = 0

local lastSideRecoil = {0, 0}

local recoilAnim = {

Pos = V3.RAW();

Rot = V3.RAW();

Code = nil;

}

local numModes = 0

local rawFireMode = 1

local canSelectFire = true

local guiAngOffset = 0

local Modes = {}

local onGround = true

local startFallHeight = 0

local jumpAnim = {

Pos = 0;

Rot = 0;

Code = 0;

}

local runReady = true

local runKeyPressed = false

local chargingStamina = false

local AimingIn = false

local AimingOut = false

local Stamina = Player.Stats.Stamina

local currentSteadyTime = S.scopeSettings.steadyTime \* 60

local camSteady = false

local takingBreath = false

local steadyKeyPressed = false

local Grip = nil

local aimedGripCF = nil

local spreadZoom = "unAimed"

local spreadStance = "Stand"

local spreadMotion = "Idling"

local baseSpread = S.spreadSettings.unAimed.Stand.Idling

local currentSpread = 0

local loweringSpread = false

Gun.LookSensitivity.Value = S.sensitivitySettings.Default

Gun.Sensitivity.Value = S.sensitivitySettings.Aim

local Sensitivity = {

mouse = Gun.LookSensitivity.Value;

aim = Gun.Sensitivity.Value;

}

local lastSensUpdate = 0

local ammoInClip = 0

local Stance = 0

local stanceSway = 1

local camSway = 1

local camAng = VEC2()

local armTilt = 0

local moveAng = 0

local animCode = 0

local desiredXOffset = 0

local desiredYOffset = 0

local currentXOffset = 0

local currentYOffset = 0

local aimHeadOffset = 0

local recoilAnimMultiplier = 1

local jumpAnimMultiplier = 1

local translationDivisor = 7

local rotationMultiplier = S.momentumSettings.Amplitude.unAimed

local armTiltMultiplier = 1

local JointC0,JointC1

local equipAnimPlaying = false

local armModel

local crossOffset = 0

local camOffsets = {

guiScope = Libraries.PrimitiveAnim.new(V3.RAW());

Reload = Libraries.PrimitiveAnim.new(V3.RAW(),nil);

Recoil = Libraries.PrimitiveAnim.new(V3.RAW(),nil);

}

local Anim = Libraries.PrimitiveAnim.new(V3.RAW(),0,V3.RAW(),0);

local lastBeat = 0

local gunParts = {}

local Connections = {}

local Keys = {}

local WalkingTouch = false

local function markHit()

runAsync(function()

if mainGUI:IsDescendantOf(game) then

Interface.hitMarker.Visible = true

local startMark = TICK()

Interface.hitMarker.lastMark.Value = startMark

wait(0.5)

if Interface.hitMarker.lastMark.Value <= startMark then

Interface.hitMarker.Visible = false

end

end

end)

end

local function DisplayDamage(damage,humanoid)

runAsync(function()

local part2 = OBJ.RAW("TextLabel")

local dt = OBJ.Clone(Services.RepStorage.DynamicText)

part2.Font = "SciFi"

part2.FontSize = "Size24"

part2.TextStrokeTransparency = 0

part2.Size = UDIM2(1,0,1,0)

part2.Position = UDIM2(0,0,0,0)

part2.BackgroundTransparency = 1

part2.Parent = humanoid.Parent.Head.DamageGUI

dt.Parent = part2

dt.Disabled = false

if (damage == 0) then

part2.TextColor3 = C3(0,0.5,1)

part2.Text = "Miss!"

else

part2.TextColor3 = C3(1,1,1)

part2.Text = damage

end

end)

end

local function loadAsset(assetType,assetEntity)

Services.CP:Preload(S[assetEntity.."Settings"][assetType])

end

--------------------------------------------------------------------------------------

--------------------[ PRE-LOADING ]---------------------------------------------------

--------------------------------------------------------------------------------------

S.bulletSettings.Velocity = RPGM.RangedCombatMath.MaxBoltVelocity(PerLevel,Player,AttributeEffect,S.bulletSettings.Velocity)

loadAsset("soundId","explosion")

loadAsset("Texture","hole")

loadAsset("Texture","spark")

loadAsset("Texture","smoke")

loadAsset("Texture","blood")

Services.CP:Preload("http://www.roblox.com/asset/?id=126877530") --The dark green arrow in the select fire gui

Services.CP:Preload("http://www.roblox.com/asset/?id=55754953") --The circle in the select fire gui

--------------------------------------------------------------------------------------

--------------------[ GUN SETUP ]-----------------------------------------------------

--------------------------------------------------------------------------------------

Libraries.Network.send("Server","SetCam\_"..Player.UserId,Interface.Cam)

local gunMomentum = Libraries.Spring.new(V3.RAW())

gunMomentum.s = S.momentumSettings.Speed

gunMomentum.d = S.momentumSettings.Damper

local gunRecoilSpring = Libraries.Spring.new(V3.RAW())

gunRecoilSpring.s = S.recoilSettings.springSpeed

gunRecoilSpring.d = S.recoilSettings.springDamper

local camRecoilSpring = Libraries.Spring.new(V3.RAW())

camRecoilSpring.s = 35

camRecoilSpring.d = 0.5

local crossSpring = Libraries.Spring.new(V3.RAW(crossOffset + (baseSpread + currentSpread) \* 50, 0, 0))

crossSpring.s = 20

crossSpring.d = 0.75

--[[local function getModelMass(P)

for \_, v in PAIRS(P:GetChildren()) do

if v:IsA("BasePart") then

playerMass = playerMass + v:GetMass()

end

getModelMass(v)

end

end

getModelMass(Char)

Char.DescendantAdded:connect(function(Descendant)

if Descendant:IsA("BasePart") then

playerMass = playerMass + Descendant:GetMass()

end

end)

Char.DescendantRemoving:connect(function(Descendant)

if Descendant:IsA("BasePart") then

playerMass = playerMass - Descendant:GetMass()

end

end)]]

--------------------------------------------------------------------------------------

--------------------[ WELD CFRAMES ]--------------------------------------------------

--------------------------------------------------------------------------------------

runAsync(function()

--[[for \_, v in (Gun:GetChildren()) do

if v:IsA("BasePart") and v ~= Handle then

if v:FindFirstChild("mainWeld") then v.mainWeld:Destroy() end

if (not v:FindFirstChild("weldCF")) then

local weldCF = OBJ("CFrameValue")

weldCF.Name = "weldCF"

weldCF.Value = Handle.CFrame:toObjectSpace(v.CFrame)

weldCF.Parent = v

INSERT(gunParts, {Obj = v, Weld = nil})

end

if string.sub(v.Name, 1, 3) == "Mag" then

if (not v:FindFirstChild("magTrans")) then

local magTrans = OBJ("NumberValue")

magTrans.Name = "magTrans"

magTrans.Value = v.Transparency

magTrans.Parent = v

end

end

v.Anchored = false

end

end

Handle.Anchored = false]]

for i = 1,#OBJ.GetKids(Gun) do

if FFC(OBJ.GetKids(Gun)[i],"weldCF") then

INSERT(gunParts, {Obj = OBJ.GetKids(Gun)[i], Weld = nil})

OBJ.GetKids(Gun)[i].Anchored = false

end

end

end)

--------------------------------------------------------------------------------------

--------------------[ MAIN PROGRAM ]--------------------------------------------------

--------------------------------------------------------------------------------------

--------------------[ ARM CREATION FUNCTION ]-----------------------------------------

local creationFunctions = {

["Arms"] = function(args)

local Arms = {}

for i = 0, 1 do

local armModel = OBJ.RAW("Model")

armModel.Name = "armModel"

local Arm = OBJ.RAW("Part")

Arm.BrickColor = (S.fakeArmSettings.realBodyColor and (i == 0 and LArm.BrickColor or RArm.BrickColor) or S.fakeArmSettings.Color)

Arm.Transparency = S.fakeArmSettings.Transparency

Arm.Name = "Arm"

Arm.CanCollide = false

Arm.Size = V3.RAW(0.598, 2, 0.598)

Arm.Parent = armModel

local armMesh = OBJ.RAW("SpecialMesh")

armMesh.MeshId = "rbxasset://fonts//leftarm.mesh"

armMesh.MeshType = MeshTypes.FileMesh

armMesh.Scale = V3.RAW(0.598, 1, 0.598)

armMesh.Parent = Arm

local Glove1 = OBJ.RAW("Part")

Glove1.BrickColor = BC("Black")

Glove1.Name = "Glove1"

Glove1.CanCollide = false

Glove1.Size = V3.RAW(0.598, 2, 0.598)

Glove1.Parent = armModel

local glove1Mesh = OBJ.RAW("SpecialMesh")

glove1Mesh.MeshId = "rbxasset://fonts//leftarm.mesh"

glove1Mesh.Offset = V3.RAW(0, -0.5, 0)

glove1Mesh.Scale = V3.RAW(0.658, 0.205, 0.658)

glove1Mesh.Parent = Glove1

local glove1Weld = OBJ.RAW("Weld")

glove1Weld.Part0 = Arm

glove1Weld.Part1 = Glove1

glove1Weld.Parent = Arm

local Glove2 = OBJ.RAW("Part")

Glove2.BrickColor = BC("Black")

Glove2.Name = "Glove2"

Glove2.CanCollide = false

Glove2.Size = V3.RAW(0.598, 2, 0.598)

Glove2.Parent = armModel

local glove2Mesh = OBJ.RAW("SpecialMesh")

glove2Mesh.MeshId = "rbxasset://fonts//leftarm.mesh"

glove2Mesh.Offset = V3.RAW(0, -0.435, 0)

glove2Mesh.Scale = V3.RAW(0.69, 0.105, 0.69)

glove2Mesh.Parent = Glove2

local glove2Weld = OBJ.RAW("Weld")

glove2Weld.Part0 = Arm

glove2Weld.Part1 = Glove2

glove2Weld.Parent = Arm

local Glove3 = OBJ.RAW("Part")

Glove3.BrickColor = BC("Black")

Glove3.Name = "Glove3"

Glove3.CanCollide = false

Glove3.Size = V3.RAW(0.598, 2, 0.598)

Glove3.Parent = armModel

local glove3Mesh = OBJ.RAW("SpecialMesh")

glove3Mesh.MeshId = "rbxasset://fonts//leftarm.mesh"

glove3Mesh.Offset = V3.RAW(0.18 \* ((i \* 2) - 1), -0.7, 0)

glove3Mesh.Scale = V3.RAW(0.299, 0.305, 0.657)

glove3Mesh.Parent = Glove3

local glove3Weld = OBJ.RAW("Weld")

glove3Weld.Part0 = Arm

glove3Weld.Part1 = Glove3

glove3Weld.Parent = Arm

local Sleeve1 = OBJ.RAW("Part")

Sleeve1.BrickColor = BC("Sand green")

Sleeve1.Name = "Sleeve1"

Sleeve1.CanCollide = false

Sleeve1.Size = V3.RAW(0.598, 2, 0.598)

Sleeve1.Parent = armModel

local sleeve1Mesh = OBJ.RAW("SpecialMesh")

sleeve1Mesh.MeshId = "rbxasset://fonts//leftarm.mesh"

sleeve1Mesh.Offset = V3.RAW(0, 0.75, 0)

sleeve1Mesh.Scale = V3.RAW(0.656, 0.3, 0.656)

sleeve1Mesh.Parent = Sleeve1

local sleeve1Weld = OBJ.RAW("Weld")

sleeve1Weld.Part0 = Arm

sleeve1Weld.Part1 = Sleeve1

sleeve1Weld.Parent = Arm

local Sleeve2 = OBJ.RAW("Part")

Sleeve2.BrickColor = BC("Sand green")

Sleeve2.Name = "Sleeve2"

Sleeve2.CanCollide = false

Sleeve2.Size = V3.RAW(0.598, 2, 0.598)

Sleeve2.Parent = armModel

local sleeve2Mesh = OBJ.RAW("SpecialMesh")

sleeve2Mesh.MeshId = "rbxasset://fonts//leftarm.mesh"

sleeve2Mesh.Offset = V3.RAW(0, 0.55, 0)

sleeve2Mesh.Scale = V3.RAW(0.75, 0.1, 0.75)

sleeve2Mesh.Parent = Sleeve2

local sleeve2Weld = OBJ.RAW("Weld")

sleeve2Weld.Part0 = Arm

sleeve2Weld.Part1 = Sleeve2

sleeve2Weld.Parent = Arm

INSERT(Arms, {Model = armModel, armPart = Arm})

end

return Arms

end;

["Modes"] = function(args)

numModes = 0

for i, v in PAIRS(S.selectFireSettings.Modes) do

if v then

numModes = numModes + 1

end

end

local currentMode = 0

for i, v in PAIRS(S.selectFireSettings.Modes) do

if v then

local Frame = OBJ.RAW("Frame")

Frame.BackgroundTransparency = 1

Frame.Name = currentMode

Frame.Position = UDIM2(currentMode,0,0)

Frame.Size = UDIM2(1,0,1,0)

Frame.Parent = fireModes

Frame.ZIndex = 2

local modeLabel = OBJ.RAW("TextLabel")

modeLabel.BackgroundTransparency = 1

modeLabel.Name = "Label"

modeLabel.Position = UDIM2(0, 0, 0, 0)

modeLabel.Size = UDIM2(1,0,1,0)

modeLabel.Font = Fonts.Highway

modeLabel.FontSize = FontSizes.Size18

modeLabel.Text =i:upper()

modeLabel.TextColor3 = C3(1, 1, 1)

modeLabel.TextScaled = false

modeLabel.TextTransparency = 0

modeLabel.TextWrapped = true

modeLabel.Parent = Frame

INSERT(Modes, i:upper())

currentMode = currentMode + 1

end

end

-- guiAngOffset = -15 \* (numModes ^ 3) + 150 \* (numModes ^ 2) - 525 \* numModes + 660

return true

end;

["Bullet"] = function(args)

return (createBullet(args[3]))

end

}

function create(...)

local args = {...}

return creationFunctions[args[1]](args)

end

--------------------[ MATH FUNCTIONS ]------------------------------------------------

GunMath.Map = function(Val, fromLow, fromHigh, toLow, toHigh)

return (Val - fromLow) \* (toHigh - toLow) / (fromHigh - fromLow) + toLow

end

GunMath.numLerp = function(A, B, Alpha)

return A + (B - A) \* Alpha

end

GunMath.RAND = function(Min, Max, Accuracy)

return GunMath.numLerp(Min, Max, RANDOM())

--[[local Inverse = 1 / (Accuracy or 1)

return (math.random(Min \* Inverse, Max \* Inverse) / Inverse)]]

end

GunMath.Round = function(Num, toNearest)

return FLOOR(Num / toNearest + 0.5) \* toNearest

end

GunMath.getNearestPoint = function(A, B, Origin)

local A2 = (A - Origin).magnitude

local B2 = (B - Origin).magnitude

return (MIN(A2, B2) == A2 and A or B)

end

local function GetHitPos()

local H,P,N = AdvRayCast(Main.CFrame.p, Handle.CFrame.lookVector, S.bulletSettings.Range, {})

return P

end

--------------------[ TWEEN FUNCTIONS ]-----------------------------------------------

function tween(...)

local args = {...}

if args then

if args[1] == "Joint" then

local Joint = args[2]

local newC0 = args[3]

local newC1 = args[4]

local Alpha = args[5]

local Duration = args[6]

runAsync(function()

local newCode = RANDOM(-1e9, 1e9) --This creates a random code between -1000000000 and 1000000000

local tweenIndicator = nil

if (not FFC(Joint,"tweenCode")) then --If the joint isn't being tweened, then

tweenIndicator = Libraries.Network.fetch("Server","TweenIndicator\_"..Player.UserId,Joint, newCode)

else

tweenIndicator = Joint.tweenCode

tweenIndicator.Value = newCode --If the joint is already being tweened, this will change the code, and the tween loop will stop

if FE then

Libraries.Network.send("Server","setTween\_"..Player.UserId,tweenIndicator,newCode) --If the joint is already being tweened, this will change the code, and the tween loop will stop

end

end

if Duration <= 0 then --If the duration is less than or equal to 0 then there's no need for a tweening loop

if newC0 then Joint.C0 = newC0 end

if newC1 then Joint.C1 = newC1 end

else

local startC0 = Joint.C0

local startC1 = Joint.C1

local t0 = TICK()

while true do

shortWait(Services.RS.Heartbeat)

local X = MIN((TICK() - t0) / Duration, 1) \* 90

if tweenIndicator.Value ~= newCode then break end --This makes sure that another tween wasn't called on the same joint

if (not Selected) then break end --This stops the tween if the tool is deselected

if newC0 then Joint.C0 = startC0:lerp(newC0, Alpha(X)) if FE then Libraries.Network.send("Server","SetJointC0\_"..Player.UserId,Joint,Joint.C0) end end

if newC1 then Joint.C1 = startC1:lerp(newC1, Alpha(X)) if FE then Libraries.Network.send("Server","SetJointC1\_"..Player.UserId,Joint,Joint.C1) end end

if X == 90 then break end

end

if tweenIndicator.Value == newCode then --If this tween functions was the last one called on a joint then it will remove the code

Libraries.Network.send("Server","deleteTweenIndicator\_"..Player.UserId,tweenIndicator, newCode)

end

end

end)

end

end

if args[1] == "Cam" then

local Key = args[2]

local newRot = args[3]

local Alpha = args[4]

local Duration = args[5]

runAsync(function()

local newCode = RANDOM(-1e9, 1e9)

camOffsets[Key].Code = newCode

local Increment = 1.5 / Duration

local prevRot = camOffsets[Key].Rot

local X = 0

while true do

shortWait(Services.RS.Heartbeat)

local newX = X + Increment

X = (newX > 90 and 90 or newX)

if camOffsets[Key].Code ~= newCode then break end

if (not Selected) then break end

camOffsets[Key].Rot = V3.lerp(prevRot,newRot, Alpha(X))

if X == 90 then break end

end

if camOffsets[Key].Code == newCode then

camOffsets[Key].Code = nil

end

end)

end

if args[1] == "Recoil" then

local newPos = args[2]

local newRot = args[3]

local Alpha = args[4]

local Duration = args[5]

runAsync(function()

local newCode = RANDOM(-1e9, 1e9)

recoilAnim.Code = newCode

local Increment = 1.5 / Duration

local prevPos = recoilAnim.Pos

local prevRot = recoilAnim.Rot

local X = 0

while true do

shortWait(Services.RS.Heartbeat)

local newX = X + Increment

X = (newX > 90 and 90 or newX)

if recoilAnim.Code ~= newCode then break end

if (not Selected) then break end

recoilAnim.Pos = V3.lerp(prevPos,newPos, Alpha(X))

recoilAnim.Rot = V3.lerp(prevRot,newRot, Alpha(X))

if X == 90 then break end

end

if recoilAnim.Code == newCode then

recoilAnim.Code = nil

end

end)

elseif args[1] == "Aim" then

local currentFOV = Interface.Cam.FieldOfView

local targetFOV = args[2]

local frames = args[3]

local Alpha = args[4]

local currentTrans = (Interface.Scope.BackgroundTransparency == 1 and (S.guiScope and 0 or 1) or Interface.Scope.BackgroundTransparency)

local t0 = TICK()

while true do

shortWait(Services.RS.Heartbeat)

local X = MIN((TICK() - t0) / frames, 1) \* 90

if currentFOV > targetFOV then

aimAlpha = Alpha(X)

else

aimAlpha = 1 - Alpha(X)

end

aimHeadOffset = headOffset.X \* aimAlpha

jumpAnimMultiplier = GunMath.numLerp(1, S.fallSettings.aimEffect, aimAlpha)

translationDivisor = GunMath.numLerp(7, 20, aimAlpha)

rotationMultiplier = GunMath.numLerp(S.momentumSettings.Amplitude.unAimed, S.momentumSettings.Amplitude.Aimed, aimAlpha)

armTiltMultiplier = GunMath.numLerp(1, 0.2, aimAlpha)

local lerpedTrans = (currentFOV > targetFOV and GunMath.numLerp(currentTrans, 0, aimAlpha) or GunMath.numLerp(1, currentTrans, aimAlpha))

Interface.Cam.FieldOfView = (currentFOV > targetFOV and GunMath.numLerp(currentFOV,targetFOV,aimAlpha) or GunMath.numLerp(targetFOV,currentFOV,aimAlpha) )

if currentFOV > targetFOV then

if not Aimed then break end

else if Aimed then break end

end

if S.guiScope then

Interface.Scope.BackgroundTransparency = lerpedTrans

end

if X == 90 then break end

if (not Selected) then break end

end

end

end

--------------------[ GUI UPDATE FUNCTIONS ]------------------------------------------

local function updateGUI(...)

local args = {...}

if args[1] == "ClipAmmo" then

clipAmmoGUI.Text = Ammo.Value

clipAmmoGUI.TextColor3 = (Ammo.Value <= (ClipSize.Value / 3) and C3(1, 0, 0) or C3(1, 1, 1))

if AmmoBar then

AmmoBar.Size = UDIM2((Ammo.Value/ClipSize.Value),0,1,0)

end

elseif args[1] == "ModeLabels" then

fireModes.FireModeController:JumpTo(fireModes[args[2]])

end

end

--------------------[ GUI SETUP FUNCTION ]--------------------------------------------

function setUpGUI()

local currentNum = 1

if S.guiScope then

scopeSteady.Text = "Hold Left-Shift to Steady"

end

runAsync(function()

for i = clipAmmoGUI.TextTransparency,0,-0.05 do

clipAmmoGUI.TextTransparency = i;

shortWait(Services.RS.RenderStepped)

end

end)

INSERT(Connections, Ammo.Changed:connect(function(ammo)

updateGUI("ClipAmmo")

end))

if not Services.UIS.TouchEnabled then

clipAmmoGUI:TweenPosition(UDIM2(0.5,0,0,0),"Out", "Bounce", 1)

end

updateGUI("ClipAmmo")

create("Modes")

updateGUI("ModeLabels",numModes - 1, 0, 90)

if S.selectFire then

modeGUI.Text = Modes[((rawFireMode - 1) % numModes) + 1]

else

modeGUI.Text = (

S.gunType.Semi and "SEMI" or

S.gunType.Auto and "AUTO" or

S.gunType.Burst and "BURST" or

S.gunType.Stun and "STUN" or

"SAFETY"

)

end

end

--------------------[ CAMERA RENDERING FUNCTIONS ]-----------------------------------

local function changePlayerTrans(P, Trans)

for i = 1,#OBJ.GetKids(P) do

if OBJ.GetKids(P)[i]:IsA("BasePart") and (not OBJ.GetKids(P)[i]:IsDescendantOf(Gun)) then

runAsync(function()

OBJ.GetKids(P)[i].LocalTransparencyModifier = Trans

end)

end

runAsync(function()

changePlayerTrans(OBJ.GetKids(P)[i], Trans)

end)

end

end

GunMath.getYawPitch = function(Cf)

local LV = Cf.lookVector

local Yaw = ATAN2(LV.x, -LV.z)

local Pitch = ATAN(LV.y / ((LV.x ^ 2) + (LV.z ^ 2)^0.5))

return Yaw, Pitch

end

GunMath.getTotalCamOffset = function()

return camOffsets.guiScope.Rot + camOffsets.Reload.Rot + camRecoilSpring.p

end

function renderCamera()

local finalCamOffset = GunMath.getTotalCamOffset()

Interface.Cam.CameraType = CameraTypes.Scriptable

Interface.Cam.CFrame = CF.RAW(Head.Position) \* CF.ANG(0, camAng.X + finalCamOffset.X, 0) \* CF.ANG(camAng.Y + finalCamOffset.Y, 0, 0) \* CF.RAW(0, 0, 0.5)

Interface.Cam:SetRoll(crawlCamRot + finalCamOffset.Z)

end

--------------------[ ANIMATION FUNCTIONS ]-------------------------------------------

--local function gunbob(a,r)

-- local a,r=a or 1,r or 1

-- local d,s,v=char.distance\*6.28318\*3/4,char.speed,-char.velocity

-- local w=v3(r\*sin(d/4-1)/256+r\*(sin(d/64)-r\*v.z/4)/512,r\*cos(d/128)/128-r\*cos(d/8)/256,r\*sin(d/8)/128+r\*v.x/1024)\*s/20\*6.28318

-- return cf(r\*cos(d/8-1)\*s/196,1.25\*a\*sin(d/4)\*s/512,0)\*cframe.fromaxisangle(w)

-- end

function Animate()

local Increment = 90 \* 2.5 --1.5 / 0.4

local runAlpha = 0

local currentlyCrawling = false

local crawlTween = false

INSERT(Connections, Services.RS.Heartbeat:connect(function(dt)

--Movement Variable updating

isCrawling = (Stance == 2 and onGround and S.stanceSettings.crawlAnimation) and ((not Idling) and Walking) or false

isIdling = (((not onGround) and S.stopAnimsOnFall) and true or (Idling and (not Walking))) and (not Knifing) and (not isCrawling)

isWalking = (not Idling) and Walking and (not Running) and (not Knifing) and ((not S.stopAnimsOnFall) and true or onGround) and (not isCrawling)

isRunning = (not Idling) and Walking and Running and (not Knifing) and ((not S.stopAnimsOnFall) and true or onGround) and (not isCrawling)

crawlAlpha = MIN(MAX(crawlAlpha + (isCrawling and Increment or -Increment) \* dt, 0), 90)

idleAlpha = MIN(MAX(idleAlpha + (isIdling and Increment or -Increment) \* dt, 0), 90)

walkAlpha = MIN(MAX(walkAlpha + (isWalking and Increment or -Increment) \* dt, 0), 90)

runAlpha = MIN(MAX(runAlpha + (isRunning and Increment or -Increment) \* dt, 0), 90)

local posHip = (

Sine(idleAlpha) \* (Libraries.Anims.Idling["unAimed"](Anim.Ang)).Pos

) + (

Sine(walkAlpha) \* (Libraries.Anims.Walking["unAimed"](Anim.Ang)).Pos

) + (

Sine(runAlpha) \* (Libraries.Anims.Running(Anim.Ang)).Pos

)

local rotHip = (

Sine(idleAlpha) \* (Libraries.Anims.Idling["unAimed"](Anim.Ang)).Rot

) + (

Sine(walkAlpha) \* (Libraries.Anims.Walking["unAimed"](Anim.Ang)).Rot

) + (

Sine(runAlpha) \* (Libraries.Anims.Running(Anim.Ang)).Rot

)

local posAim = (

Sine(idleAlpha) \* (Libraries.Anims.Idling["Aimed"](Anim.Ang)).Pos

) + (

Sine(walkAlpha) \* (Libraries.Anims.Walking["Aimed"](Anim.Ang)).Pos

) + (

Sine(runAlpha) \* (Libraries.Anims.Running(Anim.Ang)).Pos

)

local rotAim = (

Sine(idleAlpha) \* (Libraries.Anims.Idling["Aimed"](Anim.Ang)).Rot

) + (

Sine(walkAlpha) \* (Libraries.Anims.Walking["Aimed"](Anim.Ang)).Rot

) + (

Sine(runAlpha) \* (Libraries.Anims.Running(Anim.Ang)).Rot

)

Anim.Pos = (1 - aimAlpha) \* posHip + aimAlpha \* posAim

Anim.Rot = (1 - aimAlpha) \* rotHip + aimAlpha \* rotAim

Anim.Ang = Anim.Ang + RAD(105 \* dt) \* stanceSway

--Gun Momentum updating

gunMomentum.t = V3.RAW(desiredXOffset, desiredYOffset, 0)

local newGunMomentum = gunMomentum.p

currentXOffset = newGunMomentum.X / S.momentumSettings.maxInput

currentYOffset = newGunMomentum.Y / S.momentumSettings.maxInput

--Recoil spring updating

gunRecoilSpring.t = recoilAnim.Rot

camRecoilSpring.t = camOffsets.Recoil.Rot

--Cross spring updating

crossSpring.t = (Aimed and V3.RAW(-2, 0, 0) or V3.RAW(crossOffset + (baseSpread + currentSpread) \* 50, 0, 0))

local newS = crossSpring.p.X

crossA.Position = UDIM2(0.5, -1, 1, -newS \* 0.5)

crossB.Position = UDIM2(0, newS \* 0.5 - 15, 0.5, -1)

crossC.Position = UDIM2(0.5, -1, 0, newS \* 0.5 - 15)

crossD.Position = UDIM2(1, -newS \* 0.5, 0.5, -1)

--Orientation updating

local finalCamOffset = GunMath.getTotalCamOffset()

headWeld.C1 = CF.ANG(-camAng.y - finalCamOffset.Y, 0, 0)

if FE then

Libraries.Network.send("Server","changeHeadC1\_"..Player.UserId,headWeld,headWeld.C1)

end

if (not Humanoid.Sit) then

HRP.CFrame = CF.RAW(HRP.Position) \* CF.ANG(0, camAng.x + finalCamOffset.X, 0)

headWeld.C1 = CF.ANG(-camAng.y - finalCamOffset.Y, 0, 0)

else

local HRPDir = CF.RAW(HRP.Position) \* CF.RAW(HRP.Position,Humanoid.TargetPoint).lookVector

HRP.CFrame = CF.RAW(HRP.Position,HRP.Position + HRPDir)

HRP.CFrame = CF.RAW(HRP.Position) \* CF.ANG(0, camAng.x + finalCamOffset.X, 0)

headWeld.C1 = CF.ANG(0 + (-camAng.y - finalCamOffset.Y) ,0,0)

end

--Walkspeed updating

if Running then

Humanoid.WalkSpeed = S.walkSpeeds.Sprinting

else

local SpeedRatio = S.walkSpeeds.Aimed / S.walkSpeeds.Base

if Stance == 0 then

Humanoid.WalkSpeed = (Aimed and S.walkSpeeds.Aimed or S.walkSpeeds.Base)

elseif Stance == 1 then

Humanoid.WalkSpeed = (Aimed and S.walkSpeeds.Crouched \* SpeedRatio or S.walkSpeeds.Crouched)

elseif Stance == 2 then

Humanoid.WalkSpeed = (Aimed and S.walkSpeeds.Prone \* SpeedRatio or S.walkSpeeds.Prone)

end

end

end))

local crawlAng = 0

while Selected do

if isCrawling then

breakReload = (Reloading and true or breakReload)

if Aimed then unAimGun(true) end

local tempCrawlAnim = Libraries.Anims.Crawling(crawlAng, moveAng)

runAsync(function()

local startCamRot = crawlCamRot

local startLLegCF = LLegWeld.C1

local startRLegCF = RLegWeld.C1

local t0 = TICK()

while true do

shortWait(Services.RS.Heartbeat)

local Alpha = MIN((TICK() - t0) \* 3.33333, 1) \* 90

if (not isCrawling) then break end

if (not Selected) then break end

crawlCamRot = GunMath.numLerp(startCamRot, tempCrawlAnim.Camera, Sine(Alpha))

if not FE then

LLegWeld.C1 = startLLegCF:lerp(tempCrawlAnim.leftLeg, Linear(Alpha))

RLegWeld.C1 = startRLegCF:lerp(tempCrawlAnim.rightLeg, Linear(Alpha))

else

runAsync(function()

Libraries.Network.send("Server","changeLLegWeldC1\_"..Player.UserId,LLegWeld,startLLegCF:lerp(tempCrawlAnim.leftLeg, Linear(Alpha)))

Libraries.Network.send("Server","changeRLegWeldC1\_"..Player.UserId,RLegWeld,startRLegCF:lerp(tempCrawlAnim.rightLeg, Linear(Alpha)))

end)

end

if Alpha == 90 then break end

end

end)

tween("Joint",ArmWelds.LWeld, nil, tempCrawlAnim.leftArm, Linear, 0.3)

tween("Joint",ArmWelds.RWeld, nil, tempCrawlAnim.rightArm, Linear, 0.3)

tween("Joint",Grip, nil, tempCrawlAnim.Grip, Linear, 0.3)

lowerSpread()

local t0 = TICK()

while true do

local dt = Services.RS.Heartbeat:wait()

if (not Selected) then break end

if (not isCrawling) then break end

if (TICK() - t0) >= 0.3 then

local crawlAnim = Libraries.Anims.Crawling(crawlAng, moveAng)

ArmWelds.RWeld.C1 = crawlAnim.rightArm

ArmWelds.RWeld.C1 = crawlAnim.rightArm

ArmWelds.LWeld.C1 = crawlAnim.leftArm

LLegWeld.C1 = crawlAnim.leftLeg

RLegWeld.C1 = crawlAnim.rightLeg

Grip.C1 = crawlAnim.Grip

if FE then

Libraries.Network.send("Server","changeLLegWeldC1\_"..Player.UserId,LLegWeld,LLegWeld.C1)

Libraries.Network.send("Server","changeRLegWeldC1\_"..Player.UserId,RLegWeld,RLegWeld.C1)

Libraries.Network.send("Server","changeGripC1\_"..Player.UserId,Grip,Grip.C1)

Libraries.Network.send("Server","changeLWeldC1\_"..Player.UserId,ArmWelds.LWeld,ArmWelds.LWeld.C1)

Libraries.Network.send("Server","changeRWeldC1\_"..Player.UserId,ArmWelds.RWeld,ArmWelds.RWeld.C1)

end

crawlCamRot = crawlAnim.Camera

crawlAng = crawlAng + 0.5 \* RAD(105 \* dt) \* (HRP.Velocity \* V3.RAW(1, 0, 1)).magnitude \* 0.333333333

end

end

else

crawlAng = 0

if (not equipAnimPlaying) then

runAsync(function()

local startCamRot = crawlCamRot

local startLLegCF = LLegWeld.C1

local startRLegCF = RLegWeld.C1

local t0 = TICK()

while true do

shortWait(Services.RS.Heartbeat)

local Alpha = MIN((TICK() - t0) \* 3.333333333, 1) \* 90

if isCrawling then break end

if (not Selected) then break end

crawlCamRot = GunMath.numLerp(startCamRot, 0, Sine(Alpha))

LLegWeld.C1 = startLLegCF:lerp(CF.RAW(), Linear(Alpha))

RLegWeld.C1 = startRLegCF:lerp(CF.RAW(), Linear(Alpha))

if Alpha == 90 then break end

end

end)

if (not isRunning) then

tween("Joint",ArmWelds.LWeld, nil, S.unAimedC1.leftArm, Sine, 0.3)

tween("Joint",ArmWelds.RWeld, nil, S.unAimedC1.rightArm, Sine, 0.3)

tween("Joint",Grip, nil, S.unAimedC1.Grip, Sine, 0.3)

end

end

while true do

if (not Selected) then break end

if isCrawling then break end

shortWait(Services.RS.Heartbeat)

end

end

wait()

end

end

function getAnimCF()

return CF.RAW(aimHeadOffset, 0, 0) \* CF.ANG(

jumpAnim.Rot \* COS(camAng.Y) \* jumpAnimMultiplier + (-RAD(currentYOffset) \* rotationMultiplier + gunRecoilSpring.p.X + Anim.Rot.X) \* stanceSway,

(-RAD(currentXOffset) \* rotationMultiplier + gunRecoilSpring.p.Y + Anim.Rot.Y) \* stanceSway,

(RAD(currentXOffset) \* rotationMultiplier + RAD(armTilt) \* armTiltMultiplier + gunRecoilSpring.p.Z + Anim.Rot.Z) \* stanceSway

) \* CF.RAW(

(Anim.Pos.X + recoilAnim.Pos.X) \* stanceSway,

jumpAnim.Pos \* COS(camAng.Y) \* jumpAnimMultiplier + (Anim.Pos.Y + recoilAnim.Pos.Y) \* stanceSway,

-jumpAnim.Pos \* SIN(camAng.Y) \* jumpAnimMultiplier + (Anim.Pos.Z + recoilAnim.Pos.Z) \* stanceSway

), CF.ANG(-camAng.Y \* crawlAlpha \* 0.01111111, 0, 0) \* CF.RAW(aimHeadOffset, -1, 0)

end

--------------------[ FIRING FUNCTIONS ]----------------------------------------------

function lowerSpread()

if (not loweringSpread) then

loweringSpread = true

local Connection = nil

Connection = Services.RS.Heartbeat:connect(function(dt)

if MB1Down and Firing then

Connection:disconnect()

end

local newSpread = currentSpread - (S.spreadSettings.Decrease \* dt)

currentSpread = (newSpread < 0 and 0 or newSpread)

if currentSpread == 0 then

Connection:disconnect()

end

end)

loweringSpread = false

end

end

function getFiringFunction(canFireBullets)

local fireGun = function()

local fireSound = FFC(Handle,"FireSound")

Gun.Bolt.Transparency = 1

Gun.BoltBack.Transparency = 0

if fireSound then SoundLib.Play(fireSound) end

----------------------------------------------------------------------------------

for \_ = 1, (S.gunType.Shot and S.shotAmount or 1) do

local randSpread1 = RAD(GunMath.RAND(0, 365))

local randSpread2 = RAD(GunMath.RAND(-(baseSpread + currentSpread), baseSpread + currentSpread, 0.01))

local spreadDir = CF.FAxA(V3.RAW(0, 0, 1) \* randSpread1) \* CF.ANG(randSpread2, 0, 0)

local originCF = ((Aimed and S.guiScope) and Head.CFrame or Handle.CFrame) \* spreadDir

local bulletDirection = CF.RAW(originCF.p, originCF.p + originCF.lookVector).lookVector

if S.bulletSettings.instantHit then

local newRay

if S.guiScope then

newRay = Interface.Cam:ScreenPointToRay(VantagePoint.AbsolutePosition.X,VantagePoint.AbsolutePosition.Y,0)

newRay = RAY(newRay.Origin,newRay.Direction \* S.bulletSettings.Range)

else

newRay = RAY(Main.CFrame.p, bulletDirection \* S.bulletSettings.Range)

end

local H, P, N = raycast(Services.WS,newRay, Ignore)

local finalP = P

if H then

if S.gunType.Explosive then

if S.explosionSettings.soundId ~= "" then

local soundPart = OBJ.RAW("Part")

soundPart.Transparency = 1

soundPart.Anchored = true

soundPart.CanCollide = false

soundPart.Size = V3.RAW(1, 1, 1)

soundPart.CFrame = CF.RAW(P)

soundPart.Parent = gunIgnore

local Sound = OBJ.RAW("Sound")

Sound.Pitch = S.explosionSettings.Pitch

Sound.SoundId = S.explosionSettings.soundId

Sound.Volume = S.explosionSettings.Volume

Sound.Parent = soundPart

SoundLib.Play(Sound)

Services.DS:AddItem(soundPart, Sound.TimeLength)

end

Libraries.Network.send("bulletImpact\_"..Player.UserId,H, P, N, bulletDirection, false, gunIgnore, S)

Libraries.Network.send("Shockwave\_"..Player.UserId,P, S.explosionSettings.Radius, gunIgnore, S)

local E = OBJ.RAW("Explosion")

E.BlastPressure = S.explosionSettings.Pressure

E.BlastRadius = S.explosionSettings.Radius

E.DestroyJointRadiusPercent = (S.explosionSettings.rangeBasedDamage and 0 or 1)

E.ExplosionType = S.explosionSettings.Type

E.Position = P

E.Hit:connect(function(Obj, Dist)

if Obj.Name == "Torso" and (not Obj:IsDescendantOf(Char)) then

if S.explosionSettings.rangeBasedDamage then

local Dir = (Obj.Position - P).unit

local expH, \_ = raycast(Services.WS,

RAY(P - Dir \* 0.1, Dir \* 999),

Ignore

)

local rayHitHuman = expH:IsDescendantOf(Obj.Parent)

if (S.explosionSettings.rayCastExplosions and rayHitHuman) or (not S.explosionSettings.rayCastExplosions) then

local hitHumanoid = gbl:FindFirstClass(Obj.Parent, "Humanoid")

if hitHumanoid and hitHumanoid.Health > 0 and isEnemy(hitHumanoid) then

local distFactor = Dist / S.explosionSettings.Radius

local distInvert = MAX(1 - distFactor,0)

local newDamage = distInvert \* getBaseDamage((P - Main.CFrame.p).magnitude)

local Tag = OBJ.RAW("ObjectValue")

Tag.Value = Player

Tag.Name = "creator"

Tag.Parent = hitHumanoid

Services.DS:AddItem(Tag, 0.3)

hitHumanoid:TakeDamage(newDamage)

markHit()

end

end

else

local hitHumanoid = gbl:FindFirstClass(Obj.Parent, "Humanoid")

if hitHumanoid and hitHumanoid.Health > 0 and isEnemy(hitHumanoid) then

local Tag = OBJ.RAW("ObjectValue")

Tag.Value = Player

Tag.Name = "creator"

Tag.Parent = hitHumanoid

Services.DS:AddItem(Tag, 0.3)

markHit()

end

end

end

end)

E.Parent = Services.WS

else

\_, finalP = penetrateWall(H, P, bulletDirection, N, {Char, ignoreModel}, 0, (P - Main.CFrame.p).magnitude, nil)

end

end

if S.bulletTrail and S.trailSettings.Transparency ~= 1 then

Libraries.Network.fetch("Server","Trail\_"..Player.UserId,Main.CFrame.p, finalP, gunIgnore, S)

end

else

local bullet = create("Bullet",modeGUI.Text,bulletDirection,Main.CFrame.p)

--Gun.BulletObj.Value = bullet

local debounce = false

bullet.Touched:connect(function(hit)

local newRay = RAY(Main.CFrame.p, bulletDirection \* S.bulletSettings.Range)

local H, P, N = Services.WS:FindPartOnRayWithIgnoreList(newRay, {Char, ignoreModel})

local finalP = P

if (not hit:isDescendantOf(Char) and not hit:isDescendantOf(ignoreModel)) then

-- if hit.Parent:IsA("Tool") then

-- Libraries.Network.send("Server","Disarm\_"..Player.UserId,hit)

-- return

-- end

Libraries.Network.send("Server","bulletImpact\_"..Player.UserId,hit, P, N, bulletDirection, false, gunIgnore, S)

penetrateWall(hit, P, bulletDirection, N, {Char, ignoreModel}, 0, (P - Main.CFrame.p).magnitude, nil)

penetrateWall(H, P, bulletDirection, N, {Char, ignoreModel}, 0, (P - Main.CFrame.p).magnitude, nil)

wait(3)

OBJ.Destroy(bullet)

end

end)

end

end

----------------------------------------------------------------------------------

currentSpread = currentSpread + S.spreadSettings.Increase

for i = 1,#Libraries.Plugins.Firing do

runAsync(function()

Libraries.Plugins.Firing[i](Player)

end)

end

local backRecoil = GunMath.RAND(S.recoilSettings.Recoil.Back.Min, S.recoilSettings.Recoil.Back.Max, 0.01) --Get the kickback recoil

local upRecoil = GunMath.RAND(S.recoilSettings.Recoil.Up.Min, S.recoilSettings.Recoil.Up.Max, 0.01) --Get the up recoil

local sideRecoilAlpha = 0

if lastSideRecoil[1] < 0 and lastSideRecoil[2] < 0 then --This conditional basically makes sure the gun tilt isn't in the same direction for more than 2 shots

sideRecoilAlpha = GunMath.RAND(0, 1, 0.1)

elseif lastSideRecoil[1] > 0 and lastSideRecoil[2] > 0 then

sideRecoilAlpha = GunMath.RAND(-1, 0, 0.1)

else

sideRecoilAlpha = GunMath.RAND(-1, 1, 0.1)

end

local sideRecoil = GunMath.numLerp(S.recoilSettings.Recoil.Side.Left, S.recoilSettings.Recoil.Side.Right, sideRecoilAlpha / 2 + 0.5) --Get the side recoil

local tiltRecoil = GunMath.numLerp(S.recoilSettings.Recoil.Tilt.Left, S.recoilSettings.Recoil.Tilt.Right, sideRecoilAlpha / 2 + 0.5) --Get the tilt recoil

local recoilPos = V3.RAW(

0,---sideRecoil,

0,

-backRecoil

) \* (Aimed and S.recoilSettings.aimedMultiplier or 1)

local recoilRot = V3.RAW(

(Aimed and 0 or (-RAD(upRecoil \* 10) \* (firstShot and S.recoilSettings.firstShotMultiplier or 1))),

RAD(sideRecoil \* 10),

RAD(tiltRecoil \* 10)

) \* (Aimed and S.recoilSettings.aimedMultiplier or 1)

local camRecoilRot = V3.RAW(

-RAD(sideRecoil \* 10),

RAD(upRecoil \* 10) \* (firstShot and S.recoilSettings.firstShotMultiplier or 1) \* S.recoilSettings.camMultiplier,

0

) \* (Aimed and S.recoilSettings.aimedMultiplier or 1) \* stanceSway

tween("Recoil",recoilPos, recoilRot, Sine, 0.2)

tween("Cam","Recoil", camRecoilRot, Sine, 0.15 \* (firstShot and S.recoilSettings.firstShotMultiplier or 1))

for i = 1,#Main:GetChildren() do

if Main:GetChildren()[i].Name:sub(1, 7) == "FlashFX" then

Gun.Bolt.Transparency = 0

Gun.BoltBack.Transparency = 1

Main:GetChildren()[i].Enabled = true

end

end

delay(1 \* 0.05, function()

tween("Recoil",V3.ID, V3.ID, Sine, 0.2)

tween("Cam","Recoil", V3.ID, Sine, 0.2)

for i = 1,#Main:GetChildren() do

if Main:GetChildren()[i].Name:sub(1, 7) == "FlashFX" then

Gun.Bolt.Transparency = 1

Gun.BoltBack.Transparency = 0

Main:GetChildren()[i].Enabled = false

end

end

end)

firstShot = false

shotCount = shotCount + 1

lastSideRecoil[(shotCount % 2) + 1] = sideRecoilAlpha

if S.cockingAnim then

repeat shortWait(Services.RS.Heartbeat) until not Aimed

animateCock()

end

end

return (canFireBullets and fireGun or nil)

end

local function getFire(fireType)

if fireType == "AUTO" then

return function()

if (not canFire) then return end

canFire = false

if (not Knifing) then

Firing = true

while MB1Down and (not Reloading) and (not isCrawling) and (not Knifing) do

if Modes[((rawFireMode - 1) % numModes) + 1] ~= "AUTO" then break end

if Humanoid.Health == 0 then break end

if Ammo.Value > 0 then

Ammo.Value = Ammo.Value - 1

if Aimed and steadyKeyPressed and S.scopeSettings.unSteadyOnFire then

steadyKeyPressed = false

Stamina.Value = 0

end

newMag = false

if Player.UIStats.GamepadOn.Value then

game.HapticService:SetMotor(InputTypes.Gamepad1,Enum.VibrationMotor.Large,1)

end

getFiringFunction(true)()

end

if S.reloadSettings.magIsBullet then

for i = 1,#Gun:GetChildren() do

if Gun:GetChildren()[i].Name:sub(1, 3) == "Mag" then

Gun:GetChildren()[i].Transparency = 1

end

end

end

if Ammo.Value == 0 and S.reloadSettings.autoReload then

wait(0.2)

Reload()

end

wait(60 / S.roundsPerMin)

if Player.UIStats.GamepadOn.Value then

game.HapticService:SetMotor(InputTypes.Gamepad1,Enum.VibrationMotor.Large,0)

end

end

end

Firing = false

canFire = true

end

elseif fireType == "SEMI" then

return function()

if (not canFire) then return end

canFire = false

if (not Knifing) and (not isCrawling) and Humanoid.Health ~= 0 then

Firing = true

if Ammo.Value > 0 then

Ammo.Value = Ammo.Value - 1

if Aimed and steadyKeyPressed and S.scopeSettings.unSteadyOnFire then

steadyKeyPressed = false

Stamina.Value = 0

end

newMag = false

if Player.UIStats.GamepadOn.Value then

game.HapticService:SetMotor(InputTypes.Gamepad1,Enum.VibrationMotor.Large,1)

end

getFiringFunction(true)()

end

if S.reloadSettings.magIsBullet then

for i = 1,#Gun:GetChildren() do

if Gun:GetChildren()[i].Name:sub(1, 3) == "Mag" then

Gun:GetChildren()[i].Transparency = 1

end

end

end

if Ammo.Value == 0 and S.reloadSettings.autoReload then

wait(0.2)

Reload()

end

wait(60 / S.roundsPerMin)

if Player.UIStats.GamepadOn.Value then

game.HapticService:SetMotor(InputTypes.Gamepad1,Enum.VibrationMotor.Large,0)

end

end

if Player.UIStats.GamepadOn.Value then

game.HapticService:SetMotor(InputTypes.Gamepad1,Enum.VibrationMotor.Large,0)

end

Firing = false

canFire = true

end

elseif fireType == "STUN" then

return function()

if (not canFire) then return end

canFire = false

if (not Knifing) and (not isCrawling) and Humanoid.Health ~= 0 then

Firing = true

if Ammo.Value > 0 then

Ammo.Value = Ammo.Value - 1

if Aimed and steadyKeyPressed and S.scopeSettings.unSteadyOnFire then

steadyKeyPressed = false

Stamina.Value = 0

end

if Player.UIStats.GamepadOn.Value then

game.HapticService:SetMotor(InputTypes.Gamepad1,Enum.VibrationMotor.Large,1)

end

newMag = false

getFiringFunction(true)()

end

if S.reloadSettings.magIsBullet then

for i = 1,#Gun:GetChildren() do

if Gun:GetChildren()[i].Name:sub(1, 3) == "Mag" then

Gun:GetChildren()[i].Transparency = 1

end

end

end

if Ammo.Value == 0 and S.reloadSettings.autoReload then

wait(0.2)

Reload()

end

wait(60 / S.roundsPerMin)

if Player.UIStats.GamepadOn.Value then

game.HapticService:SetMotor(InputTypes.Gamepad1,Enum.VibrationMotor.Large,0)

end

end

Firing = false

canFire = true

end

elseif fireType == "HOOK" then

return function()

if (not canFire) then return end

canFire = false

if (not Knifing) and (not isCrawling) and Humanoid.Health ~= 0 then

Firing = true

if Ammo.Value > 0 then

Ammo.Value = Ammo.Value - 1

if Aimed and steadyKeyPressed and S.scopeSettings.unSteadyOnFire then

steadyKeyPressed = false

Stamina.Value = 0

end

if Player.UIStats.GamepadOn.Value then

game.HapticService:SetMotor(InputTypes.Gamepad1,Enum.VibrationMotor.Large,1)

end

newMag = false

getFiringFunction(false)()

end

if S.reloadSettings.magIsBullet then

for i = 1,#Gun:GetChildren() do

if Gun:GetChildren()[i].Name:sub(1, 3) == "Mag" then

Gun:GetChildren()[i].Transparency = 1

end

end

end

if Ammo.Value == 0 and S.reloadSettings.autoReload then

wait(0.2)

Reload()

end

wait(60 / S.roundsPerMin)

if Player.UIStats.GamepadOn.Value then

game.HapticService:SetMotor(InputTypes.Gamepad1,Enum.VibrationMotor.Large,0)

end

end

Firing = false

canFire = true

end

elseif fireType == "BURST" then

return function()

if (not canFire) then return end

canFire = false

local burstTime = 60 / S.roundsPerMin

if (not Knifing) and (not isCrawling) then

Firing = true

if Player.UIStats.GamepadOn.Value then

game.HapticService:SetMotor(InputTypes.Gamepad1,Enum.VibrationMotor.Large,1)

end

for i = 1, S.burstSettings.Amount do

if Ammo.Value > 0 then

Ammo.Value = Ammo.Value - 1

if Humanoid.Health ~= 0 then

if Aimed and steadyKeyPressed and S.scopeSettings.unSteadyOnFire then

steadyKeyPressed = false

currentSteadyTime = 0

end

newMag = false

getFiringFunction(true)()

end

end

if Ammo.Value == 0 and S.reloadSettings.autoReload then

wait(0.2)

Reload()

break

end

wait(S.burstSettings.fireRateBurst and burstTime or S.burstSettings.Time / S.burstSettings.Amount)

end

if Player.UIStats.GamepadOn.Value then

game.HapticService:SetMotor(InputTypes.Gamepad1,Enum.VibrationMotor.Large,0)

end

end

if S.reloadSettings.magIsBullet then

for i = 1,#Gun:GetChildren() do

if Gun:GetChildren()[i].Name:sub(1, 3) == "Mag" then

Gun:GetChildren()[i].Transparency = 1

end

end

end

Firing = false

wait(S.burstSettings.fireRateBurst and burstTime or S.burstSettings.Wait)

canFire = true

end

elseif fireType == "SAFETY" then

return function()

if canFire then

canFire = false

if Player.UIStats.GamepadOn.Value then

game.HapticService:SetMotor(InputTypes.Gamepad1,Enum.VibrationMotor.Small,1)

end

wait(5)

if Player.UIStats.GamepadOn.Value then

game.HapticService:SetMotor(InputTypes.Gamepad1,Enum.VibrationMotor.Small,0)

end

canFire = true

end

end

end

end

--------------------[ ADS FUNCTIONS ]-------------------------------------------------

function aimGun()

if Reloading or Knifing or isCrawling or (not S.canADS) then return end

mouseSensitivity = aimSensitivity

for i = 1,#Libraries.Plugins.Aimed do

runAsync(function()

Libraries.Plugins.Aimed[i](Player)

end)

end

Aimed = true

Aiming = true

Running = false

spreadZoom = "Aimed"

baseSpread = S.spreadSettings[spreadZoom][spreadStance][spreadMotion]

if S.aimSettings.Anim then

tween("Joint",ArmWelds.LWeld, armC0[1], S.aimedC1.leftArm, Sine, S.aimSettings.Speed)

tween("Joint",ArmWelds.RWeld, armC0[2], S.aimedC1.rightArm, Sine, S.aimSettings.Speed)

tween("Joint",LWeld2, nil, CF.RAW(), Sine, S.aimSettings.Speed)

tween("Joint",RWeld2, nil, CF.RAW(), Sine, S.aimSettings.Speed)

tween("Joint",Grip, nil, aimedGripCF, Sine, S.aimSettings.Speed)

tween("Joint",headWeld2, nil, CF.RAW(0, -0.5, 0) \* CF.ANG(0, 0, S.aimSettings.headTilt) \* CF.RAW(0, 0.5, 0), Sine, S.aimSettings.Speed)

tween("Aim",S.aimSettings.InFOV,S.aimSettings.Speed,Sine)

else

ArmWelds.LWeld.C0, ArmWelds.LWeld.C1 = armC0[1], S.aimedC1.leftArm

ArmWelds.RWeld.C0, ArmWelds.RWeld.C1 = armC0[2], S.aimedC1.rightArm

LWeld2.C1, RWeld2.C1 = CF.RAW(), CF.RAW()

animWeld.C0 = CF.RAW(0, 1, 0)

Grip.C1 = aimedGripCF

headWeld2.C1 = CF.RAW(0, -0.5, 0) \* CF.ANG(0, 0, S.aimSettings.headTilt) \* CF.RAW(0, 0.5, 0)

aimAlpha = 1

aimHeadOffset = headOffset.X

jumpAnimMultiplier = S.fallSettings.aimEffect

translationDivisor = 20

rotationMultiplier = S.momentumSettings.Amplitude.Aimed

armTiltMultiplier = 0.2

Interface.Cam.FieldOfView = S.aimSettings.InFOV

end

Aiming = (not Aimed)

if (not Aiming) and S.guiScope then

runAsync(function()

scopeSteady.Visible = true

Interface.HUD.Visible = false

Interface.Scope.BackgroundTransparency = 1

scopeMain.Visible = true

if armTable then

for \_, Obj in PAIRS(armTable[1].Model:GetChildren()) do

if Obj:IsA("BasePart") then

Obj.LocalTransparencyModifier = 1

end

end

for \_, Obj in PAIRS(armTable[2].Model:GetChildren()) do

if Obj:IsA("BasePart") then

Obj.LocalTransparencyModifier = 1

end

end

elseif armModel then

for \_, Obj in PAIRS(armModel:GetChildren()) do

if Obj:IsA("BasePart") then

Obj.LocalTransparencyModifier = 1

end

end

end

for \_, Obj in PAIRS(playerFolder:GetChildren()) do

if Obj:IsA("BasePart") then

Obj.LocalTransparencyModifier = 1

end

end

for \_, Obj in PAIRS(Gun:GetChildren()) do

if Obj:IsA("BasePart") then

Obj.LocalTransparencyModifier = 1

end

end

end)

runAsync(function()

local camAng = 0

local idleCam = function()

return V3.RAW(

RAD(SIN(camAng \* S.scopeSettings.Frequency.Idling)) \* stanceSway \* camSway \* S.scopeSettings.Amplitude.Idling,

RAD(SIN(camAng \* 2.5 \* S.scopeSettings.Frequency.Idling)) \* stanceSway \* camSway \* S.scopeSettings.Amplitude.Idling \* 0.75,

0

)

end

local walkCam = function()

return V3.RAW(

RAD(SIN(camAng \* S.scopeSettings.Frequency.Walking)) \* camSway \* stanceSway \* S.scopeSettings.Amplitude.Walking,

RAD(SIN(camAng \* 2.5 \* S.scopeSettings.Frequency.Walking)) \* camSway \* stanceSway \* S.scopeSettings.Amplitude.Walking \* 0.75,

0

)

end

while Aimed do

local dt = shortWait(Services.RS.RenderStepped)

camOffsets.guiScope.Rot = (Sine(idleAlpha) \* idleCam()) + (Sine(walkAlpha) \* walkCam())

camAng = camAng + RAD(105 \* dt) \* stanceSway \* camSway

end

end)

end

end

function unAimGun(Exception)

if (not S.canADS) then return end

mouseSensitivity = S.sensitivitySettings.Default

for \_, Plugin in PAIRS(Libraries.Plugins.UnAimed) do

runAsync(function()

Plugin()

end)

end

if S.guiScope then

runAsync(function()

if armTable then

for \_, Obj in PAIRS(armTable[1].Model:GetChildren()) do

if Obj:IsA("BasePart") then

Obj.LocalTransparencyModifier = 0

end

end

for \_, Obj in PAIRS(armTable[2].Model:GetChildren()) do

if Obj:IsA("BasePart") then

Obj.LocalTransparencyModifier = 0

end

end

elseif armModel then

for \_, Obj in PAIRS(armModel:GetChildren()) do

if Obj:IsA("BasePart") then

Obj.LocalTransparencyModifier = 0

end

end

end

for \_, Obj in PAIRS(playerFolder:GetChildren()) do

if Obj:IsA("BasePart") then

Obj.LocalTransparencyModifier = 0

end

end

for \_, Obj in PAIRS(Gun:GetChildren()) do

if Obj:IsA("BasePart") then

Obj.LocalTransparencyModifier = 0

end

end

end)

end

fakeLArm.LocalTransparencyModifier = 0

fakeRArm.LocalTransparencyModifier = 0

if (not Exception) then

if (not Aimed) then return end

if (Reloading and Exception) or Knifing then return end

spreadZoom = "unAimed"

baseSpread = S.spreadSettings[spreadZoom][spreadStance][spreadMotion]

Aimed = false

Aiming = true

if S.aimSettings.Anim then

local currentFOV = Interface.Cam.FieldOfView

scopeMain.Visible = false

scopeSteady.Visible = false

tween("Joint",ArmWelds.LWeld, armC0[1], S.unAimedC1.leftArm, Sine, S.aimSettings.Speed)

tween("Joint",ArmWelds.RWeld, armC0[2], S.unAimedC1.rightArm, Sine, S.aimSettings.Speed)

tween("Joint",headWeld2, nil, CF.RAW(), Sine, S.aimSettings.Speed)

tween("Joint",Grip, nil, S.unAimedC1.Grip, Sine, S.aimSettings.Speed)

tween("Aim",S.aimSettings.OutFOV,S.aimSettings.Speed,Sine)

Interface.HUD.Visible = true

else

scopeMain.Visible = false

scopeSteady.Visible = false

ArmWelds.LWeld.C0, ArmWelds.LWeld.C1 = armC0[1], S.unAimedC1.leftArm

ArmWelds.RWeld.C0, ArmWelds.RWeld.C1 = armC0[2], S.unAimedC1.rightArm

headWeld2.C0 = CF.RAW()

Grip.C1 = S.unAimedC1.Grip

aimAlpha = 0

aimHeadOffset = 0

jumpAnimMultiplier = 1

translationDivisor = 7

rotationMultiplier = S.momentumSettings.Amplitude.unAimed

armTiltMultiplier = 1

Interface.Cam.FieldOfView = 80

Interface.Scope.BackgroundTransparency = 1

Interface.HUD.Visible = true

end

Aiming = Aimed

else

runAsync(function()

Aimed = false

Aiming = false

spreadZoom = "unAimed"

baseSpread = S.spreadSettings[spreadZoom][spreadStance][spreadMotion]

local currentFOV = Interface.Cam.FieldOfView

scopeMain.Visible = false

scopeSteady.Visible = false

tween("Joint",headWeld2, nil, CF.RAW(), Sine, S.aimSettings.Speed)

if S.aimSettings.Anim then

tween("Aim",S.aimSettings.OutFOV,S.aimSettings.Speed,Quad)

else

scopeMain.Visible = false

scopeSteady.Visible = false

aimAlpha = 0

aimHeadOffset = 0

jumpAnimMultiplier = 1

translationDivisor = 7

rotationMultiplier = S.momentumSettings.Amplitude.unAimed

armTiltMultiplier = 1

Interface.Cam.FieldOfView = S.aimSettings.OutFOV

Interface.Scope.BackgroundTransparency = 1

end

end)

end

end

--------------------[ TEXTURE CREATION FUNCTIONS ]------------------------------------

function createBullet(Direction)

local Origin = Main.CFrame.p

local bulletCF = CF.RAW(Origin, Origin + Direction)

local Bullet

local Mesh

if modeGUI.Text ~= "STUN" then

Bullet = OBJ.RAW("Part")

Bullet.BrickColor = S.bulletSettings.Color

Bullet.Material = Enum.Material.Neon

Bullet.Name = "Bullet"

Bullet.CanCollide = false

Bullet.FormFactor = "Custom"

Bullet.Size = S.bulletSettings.Size

Bullet.BottomSurface = "Smooth"

Bullet.TopSurface = "Smooth"

if MIN(S.bulletSettings.Size.X, S.bulletSettings.Size.Y, S.bulletSettings.Size.Z) < 0.2 then

Mesh = OBJ.RAW("BlockMesh")

Mesh.Scale = S.bulletSettings.Size / V3.RAW(

MAX(S.bulletSettings.Size.X, 0.2),

MAX(S.bulletSettings.Size.Y, 0.2),

MAX(S.bulletSettings.Size.Z, 0.2)

)

Mesh.Parent = Bullet

end

else

Bullet = OBJ.Clone(Services.RepStorage.StunBullet)

end

local BF = OBJ.RAW("BodyForce")

BF.force = V3.RAW(0, Bullet:GetMass() \* (196.2 - S.bulletSettings.Acceleration), 0)

local BV = OBJ.RAW("BodyVelocity")

BV.MaxForce = V3.RAW(1000000,10000000,10000000)

BV.Velocity = Direction \* S.bulletSettings.Velocity

BF.Parent = Bullet

BV.Parent = Bullet

Bullet.CFrame = bulletCF + Direction \* S.bulletSettings.Size.Z / 2

Libraries.Network.send("Server","ReplicateBullet\_"..Player.UserId,Bullet.ClassName,Direction,gunIgnore,Bullet.CFrame)

Bullet.Parent = gunIgnore

return Bullet

end

--------------------[ HIT HANDLING FUNCTIONS ]----------------------------------------

function getBaseDamage(Dist)

local startDmg = S.damageSettings.Start.Damage

local startDist = S.damageSettings.Start.Dist

local endDmg = S.damageSettings.End.Damage

local endDist = S.damageSettings.End.Dist

return (

(

Dist < startDist \* S.bulletSettings.Range

) and startDmg or

(

Dist >= startDist \* S.bulletSettings.Range and

Dist < endDist \* S.bulletSettings.Range

) and GunMath.numLerp(startDmg, endDmg, GunMath.Map(Dist / S.bulletSettings.Range, startDist, endDist, 0, 1)) or

(

Dist >= endDist \* S.bulletSettings.Range

) and endDmg

)

end

local function DamageVehicle(part,Dist)

local h = nil

local t = nil

if part.Parent.ClassName == "Model" then

for i,v in pairs(part.Parent:GetChildren()) do

if v.Name == "Health" and v.ClassName == "IntValue" then

h = v

t = part.Parent:FindFirstChild("Team")

end

end

end

if part.Parent.Parent.ClassName == "Model" then

for i,v in pairs(part.Parent.Parent:GetChildren()) do

if v.Name == "Health" and v.ClassName == "IntValue" then

h = v

t = part.Parent.Parent:FindFirstChild("Team")

end

end

end

if h and t then

local curh = h.Value

if curh - getBaseDamage(Dist) > 0 then

h.Value = h.Value - getBaseDamage(Dist)

elseif curh - getBaseDamage(Dist) <=0 then

h.Value = 0

end

end

end

function Damage(H, P, N, D, Dist, customIgnore)

Libraries.Network.doDamage(Humanoid, H, P, N, D, Dist, customIgnore, modeGUI.Text)

end

function isWallIgnored(Wall)

return (

Wall.Transparency >= S.penetrationSettings.transparencyThreshold or

(S.penetrationSettings.ignoreNonCanCollide and (not Wall.CanCollide)) or

isIgnored(Wall, S.penetrationSettings.ignoreCustom)

)

end

function penetrateWall(Wall, hitPos, Direction, Normal, Ignore, totalPDist, totalBDist, lastDamagedHumanoid)

local wallIgnore = isWallIgnored(Wall)

local hitHumanoid = Damage(Wall, hitPos, Normal, Direction, totalBDist, {Char, ignoreModel})

local damagedHumanoid = nil

if hitHumanoid and hitHumanoid ~= lastDamagedHumanoid then

lastDamagedHumanoid = hitHumanoid

damagedHumanoid = Damage(Wall, hitPos, Normal, Direction, totalBDist, {Char, ignoreModel})

else

lastDamagedHumanoid = nil

-- DamageVehicle(Wall,totalBDist)

if Wall.Name == "Window" then

--Libraries.Network.send("Server","BreakWindow",Wall, workspace)

end

end

local ignoreObject = hitHumanoid and (Wall.Parent:IsA("Hat") and Wall.Parent.Parent or Wall.Parent) or Wall

INSERT(Ignore, ignoreObject)

local rayLength = S.bulletSettings.Range - totalBDist

local testRay = RAY(hitPos, Direction \* (S.bulletSettings.Range - totalBDist))

local H1, P1, N1 = raycast(Services.WS,testRay, Ignore)

local newIgnore = removeElement(Ignore, ignoreObject)

local wallRay = RAY(P1 + Direction \* 0.1, -Direction \* (rayLength + 1))

local H2, P2, N2 = raycast(Services.WS,wallRay, Ignore)

local newPDist = totalPDist + (wallIgnore and 0 or (GunMath.getNearestPoint(P1, P2, hitPos) - hitPos).magnitude)

local newBDist = totalBDist + (P1 - hitPos).magnitude

local outOfRange = GunMath.Round(newPDist, 0.001) > S.penetrationSettings.Dist or GunMath.Round(newBDist, 0.001) > S.bulletSettings.Range

if (not wallIgnore) then

Libraries.Network.send("Server","bulletImpact\_"..Player.UserId,Wall, P2, N2, Direction, hitHumanoid, gunIgnore, S)

if (not hitHumanoid) then

Libraries.Network.send("Server","Shockwave\_"..Player.UserId,hitPos, S.shockwaveSettings.Radius, gunIgnore, S)

end

end

if hitHumanoid and modeGUI.Text ~= "STUN" and hitHumanoid.Health > 0 and isEnemy(hitHumanoid) and hitHumanoid == damagedHumanoid then

Libraries.Network.fetch("Server","Blood\_"..Player.UserId,Wall, P2, Direction, gunIgnore, S)

end

if outOfRange or (not H1) then

if (not outOfRange) and (not wallIgnore) and modeGUI.Text ~= "STUN" then

Libraries.Network.send("Server","bulletImpact\_"..Player.UserId,Wall, P2, N2, Direction, hitHumanoid, gunIgnore, S)

if (not hitHumanoid) then

Libraries.Network.send("Server","Shockwave\_"..Player.UserId,P2, S.shockwaveSettings.Radius, gunIgnore, S)

end

end

return Wall, hitPos

else

if Wall == H2 and (not wallIgnore) and modeGUI.Text ~= "STUN" then

Libraries.Network.send("Server","bulletImpact\_"..Player.UserId,Wall, P2, N2, Direction, hitHumanoid, gunIgnore, S)

if (not hitHumanoid) then

Libraries.Network.send("Server","Shockwave\_"..Player.UserId,P2, S.shockwaveSettings.Radius, gunIgnore, S)

end

end

PenetrationTries = PenetrationTries - 1

if PenetrationTries ~= 0 then

return penetrateWall(H1, P1, Direction, N1, Ignore, newPDist, newBDist, lastDamagedHumanoid)

else

PenetrationTries = S.penetrationSettings.maxTries

return nil

end

end

end

function isEnemy(Human)

local Plyr = game.Players:GetPlayerFromCharacter(Human.Parent)

if (not Plyr) then if S.CanDamageNPCs then

if FFC(Human.Parent,"BOT") then

return require(Human.Parent.BOT).Allegiance ~= Player.Allegiance.Value

end

end

end

return S.AllowFriendlyFire or (Plyr ~= nil and (Plyr.Allegiance.Value ~= Player.Allegiance.Value or Plyr.Neutral))

end

--------------------[ RELOAD FUNCTIONS ]----------------------------------------------

function animateCock()

tween("Joint",LWeld2, CF.RAW(), CF.RAW(), Sine, 0.15)

tween("Joint",RWeld2, CF.RAW(), CF.RAW(), Sine, 0.15)

local magParts = {}

local magTable = {}

for i = 1,#Gun:GetChildren() do

if Gun:GetChildren()[i].Name:sub( 1, 3) == "Mag" and Gun:GetChildren()[i]:IsA("BasePart") then

INSERT(magParts, Gun:GetChildren()[i])

end

end

local animVars = {

--FUNCTIONS--

tweenJoint = function(Joint, newC0, newC1, Alpha, Duration)

tween("Joint",Joint,newC0,newC1,Alpha,Duration)

end;

makeMagInvisible = function()

for \_, v in pairs(magParts) do

v.Transparency = 1

end

magVisible = false

end;

makeMagVisible = function()

for \_, v in pairs(magParts) do

v.Transparency = v:WaitForChild("magTrans").Value

end

magVisible = true

end;

isMagVisible = function()

return magVisible

end;

isMagEmpty = function()

return ammoInClip == 0

end;

setNewMag = function()

newMag = true

end;

isNewMag = function()

return newMag

end;

getMag = function(Key)

if magTable[Key] then

return magTable[Key][1], magTable[Key][2]

else

return nil, nil

end

end;

attachGripToHead = function()

local handleCF = RArm.CFrame \* Grip.C0

Grip.C0 = CF.TOS(Head.CFrame,handleCF)

Grip.Part0 = Head

end;

attachGripToArm = function()

local handleCF = Head.CFrame \* Grip.C0

Grip.C0 = CF.TOS(RArm.CFrame,handleCF)

Grip.Part0 = RArm

end;

Sine = Sine;

Linear = Linear;

--VARIABLES--

Handle = Handle;

LArm = LArm;

RArm = RArm;

LWeld = ArmWelds.LWeld;

RWeld = ArmWelds.RWeld;

LC0 = armC0[1];

RC0 = armC0[2];

Grip = Grip;

gunIgnore = gunIgnore;

Cam = Interface.Cam;

CF = CF.RAW;

CFANG = CF.ANG;

V3 = V3.RAW;

RAD = RAD;

reloadTimeLoaded = S.reloadSettings.Times.Loaded;

reloadTimeEmpty = S.reloadSettings.Times.Empty

}

local sequenceTable = Libraries.Anims.Cocking(animVars)

print(sequenceTable)

--local T = TICK()

for \_, reloadFunction in PAIRS(sequenceTable) do

if breakReload then

break

end

reloadFunction()

if (not magVisible) then

Ammo.Value = 0

end

updateGUI("ClipAmmo")

end

--print(TICK() - T) --I divide the reloadTime by this number to get the animation speed

if (not isCrawling) then

if Running and (not S.canFireWhileRunning) then

tween("Joint",ArmWelds.LWeld, armC0[1], S.runningC1.leftArm, Sine, 0.4)

tween("Joint",ArmWelds.RWeld, armC0[2], S.runningC1.rightArm, Sine, 0.4)

tween("Joint",Grip, nil, S.runningC1.Grip, Sine, 0.4)

else

tween("Joint",ArmWelds.LWeld, armC0[1], S.unAimedC1.leftArm, Sine, 0.4)

tween("Joint",ArmWelds.RWeld, armC0[2], S.unAimedC1.rightArm, Sine, 0.4)

tween("Joint",Grip, nil, S.unAimedC1.Grip, Sine, 0.4)

end

end

for \_, v in pairs(magTable) do --In case the reload animations was stopped mid way and there were still fake mags that weren't deleted

OBJ.Destroy(v[1])

end

end

function animateReload()

tween("Joint",LWeld2, CF.RAW(), CF.RAW(), Sine, 0.15)

tween("Joint",RWeld2, CF.RAW(), CF.RAW(), Sine, 0.15)

local magParts = {}

local magTable = {}

for \_, Obj in pairs(Gun:GetChildren()) do

if Obj.Name:sub( 1, 3) == "Mag" and Obj:IsA("BasePart") then

INSERT(magParts, Obj)

end

end

local animVars = {

--FUNCTIONS--

tweenJoint = function(Joint, newC0, newC1, Alpha, Duration)

tween("Joint",Joint,newC0,newC1,Alpha,Duration)

end;

makeMagInvisible = function()

for \_, v in PAIRS(magParts) do

v.Transparency = 1

end

magVisible = false

end;

makeMagVisible = function()

for \_, v in PAIRS(magParts) do

v.Transparency = v:WaitForChild("magTrans").Value

end

magVisible = true

end;

isMagVisible = function()

return magVisible

end;

isMagEmpty = function()

return ammoInClip == 0

end;

setNewMag = function()

newMag = true

end;

isNewMag = function()

return newMag

end;

createMag = function(Key)

local magModel = OBJ.RAW("Model")

local magClones = {}

for i, v in PAIRS(magParts) do

local vClone = OBJ.Clone(v)

vClone.Transparency = v:WaitForChild("magTrans").Value

vClone.CanCollide = false

vClone.Parent = magModel

INSERT(magClones, {Original = v, magClone = vClone})

if i ~= 1 then

local W = OBJ.RAW("Motor6D")

W.Part0 = magClones[1].magClone

W.Part1 = vClone

W.C0 = CF.TOS(magClones[1].magClone.CFrame,vClone.CFrame)

W.Parent = magClones[1].magClone

end

end

magTable[Key] = {magModel, magClones}

return magModel, magClones

end;

getMag = function(Key)

if magTable[Key] then

return magTable[Key][1], magTable[Key][2]

else

return nil, nil

end

end;

attachGripToHead = function()

local handleCF = RArm.CFrame \* Grip.C0

Grip.C0 = CF.TOS(Head.CFrame,handleCF)

Grip.Part0 = Head

end;

attachGripToArm = function()

local handleCF = Head.CFrame \* Grip.C0

Grip.C0 = CF.TOS(RArm.CFrame,handleCF)

Grip.Part0 = RArm

end;

Sine = Sine;

Linear = Linear;

--VARIABLES--

Handle = Handle;

LArm = LArm;

RArm = RArm;

LWeld = ArmWelds.LWeld;

RWeld = ArmWelds.RWeld;

LC0 = armC0[1];

RC0 = armC0[2];

Grip = Grip;

gunIgnore = gunIgnore;

Cam = Interface.Cam;

CF = CF.RAW;

CFANG = CF.ANG;

V3 = V3.RAW;

RAD = RAD;

reloadTimeLoaded = S.reloadSettings.Times.Loaded;

reloadTimeEmpty = S.reloadSettings.Times.Empty

}

local sequenceTable = Libraries.Anims.Reload(animVars)

--local T = TICK()

if Player.UIStats.GamepadOn.Value then

game.HapticService:SetMotor(InputTypes.Gamepad1,Enum.VibrationMotor.Small,0.5)

end

for \_, reloadFunction in PAIRS(sequenceTable) do

if breakReload then

break

end

reloadFunction()

if (not magVisible) then

Ammo.Value = 0

end

end

--print(TICK() - T) --I divide the reloadTime by this number to get the animation speed

if (not isCrawling) then

if Running and (not S.canFireWhileRunning) then

tween("Joint",ArmWelds.LWeld, armC0[1], S.runningC1.leftArm, Sine, 0.4)

tween("Joint",ArmWelds.RWeld, armC0[2], S.runningC1.rightArm, Sine, 0.4)

tween("Joint",Grip, nil, S.runningC1.Grip, Sine, 0.4)

else

tween("Joint",ArmWelds.LWeld, armC0[1], S.unAimedC1.leftArm, Sine, 0.4)

tween("Joint",ArmWelds.RWeld, armC0[2], S.unAimedC1.rightArm, Sine, 0.4)

tween("Joint",Grip, nil, S.unAimedC1.Grip, Sine, 0.4)

end

end

for \_, v in PAIRS(magTable) do --In case the reload animations was stopped mid way and there were still fake mags that weren't deleted

OBJ.Destroy(v[1])

end

if Player.UIStats.GamepadOn.Value then

game.HapticService:SetMotor(InputTypes.Gamepad1,Enum.VibrationMotor.Small,0)

end

end

function Reload()

local reloadCursorThread = coroutine.create(function()

for j = 1, 4, 1 do

for i = 1, 3,1 do

Interface.crossHair.Reload.Text = "Reloading" .. string.rep(".", i)

wait(0.4)

end

end

end)

if Ammo.Value < (ClipSize.Value + 1) and (not Reloading) and StoredAmmo.Value > 0 then

Firing = false

ammoInClip = (ammoInClip == 0 and Ammo.Value or ammoInClip)

Reloading = true

lowerSpread()

if Aimed then unAimGun(S.reloadSettings.Anim) end

Interface.crossHair.Reload.Visible = true

if FFC(Handle,"ReloadSound") then Handle.ReloadSound:Play() end

coroutine.resume(reloadCursorThread)

if S.reloadSettings.Anim then

wait()

animateReload()

else

local startReload = TICK()

local initialReloadTime = Ammo.Value == 0 and S.reloadSettings.Times.Empty or S.reloadSettings.Times.Loaded

while true do

if breakReload then break end

if (TICK() - startReload) >= initialReloadTime then break end

shortWait(Services.RS.Heartbeat)

end

end

if (not breakReload) then

newMag = false

if StoredAmmo.Value >= ClipSize.Value then

if ammoInClip > 0 then

StoredAmmo.Value = StoredAmmo.Value - ((ClipSize.Value + 1) - ammoInClip)

Ammo.Value = ClipSize.Value + 1

else

StoredAmmo.Value = StoredAmmo.Value - ClipSize.Value

Ammo.Value = ClipSize.Value

end

elseif StoredAmmo.Value < ClipSize.Value and StoredAmmo.Value > 0 then

Ammo.Value = StoredAmmo.Value

StoredAmmo.Value = 0

end

end

Reloading = false

if Selected then

ammoInClip = (breakReload and ammoInClip or 0)

Interface.crossHair.Reload.Visible = false

end

breakReload = false

end

end

--------------------[ EXTERNAL DATA LOCATING FUNCTIONS ]-----------------------------

function removeElement(Table, Element) --removes the first instance of Element from Table

for i, v in PAIRS(Table) do

if v == Element then

REMOVE(Table, i)

break

end

end

return Table

end

function isIgnored(Obj, Table)

for \_,v in PAIRS(Table) do

if Obj == v or Obj:IsDescendantOf(v) then

return true

end

end

return false

end

function GetHitSurfaceCFrame(HitPos,Obj)

local SurfaceCF = {

{"Back",Obj.CFrame \* CF.RAW(0,0,Obj.Size.z)};

{"Bottom",Obj.CFrame \* CF.RAW(0,-Obj.Size.y,0)};

{"Front",Obj.CFrame \* CF.RAW(0,0,-Obj.Size.z)};

{"Left",Obj.CFrame \* CF.RAW(-Obj.Size.x,0,0)};

{"Right",Obj.CFrame \* CF.RAW(Obj.Size.x,0,0)};

{"Top",Obj.CFrame \* CF.RAW(0,Obj.Size.y,0)}

}

local ClosestDist = HUGE

local ClosestSurface = nil

for \_,v in PAIRS(SurfaceCF) do

local SurfaceDist = (HitPos - v[2].p).magnitude

if SurfaceDist < ClosestDist then

ClosestDist = SurfaceDist

ClosestSurface = v

end

end

return ClosestSurface[2]

end

function AdvRayCast(Origin, Direction, Dist, CustomIgnore)

local NewIgnore = (CustomIgnore and CustomIgnore or Ignore)

local NewRay = RAY(Origin, Direction \* (Dist > 999 and 999 or Dist))

local HitObj, HitPos, HitNorm = raycast(Services.WS,NewRay, NewIgnore)

local LastPos = HitPos

local FinalHitObj, FinalHitPos = nil, nil

local RepTimes = FLOOR(Dist \* 0.001001001)

if (not HitObj) and (Dist > 999) then

for i = 0, RepTimes do

local NewDist = (i == RepTimes and (Dist - (LastPos - Origin).magnitude) or 999)

local Ray2 = RAY(LastPos, Direction \* NewDist)

local HitObj2, HitPos2 = raycast(Services.WS,Ray2, NewIgnore)

if i ~= RepTimes then

if HitObj2 then

FinalHitObj, FinalHitPos = HitObj2, HitPos2

break

end

elseif i == RepTimes then

FinalHitObj, FinalHitPos = HitObj2, HitPos2

end

LastPos = HitPos2

end

return FinalHitObj, FinalHitPos

elseif HitObj or (Dist <= 999) then

return HitObj, HitPos, HitNorm

end

end

--------------------[ JUMPING ANIMATION ]---------------------------------------------

function onFall(initialVelocity)

runAsync(function()

local velocityAlpha = MAX(MIN(initialVelocity / Humanoid.JumpPower, 1), 0)

local startJumpPos = jumpAnim.Pos

local startJumpRot = jumpAnim.Rot

local endJumpPos = 0.04 \* S.fallSettings.fallMultiplier \* velocityAlpha

local endJumpRot = RAD(4) \* S.fallSettings.fallMultiplier \* velocityAlpha

local t0 = TICK()

while true do

Services.RS.Heartbeat:wait()

local Alpha = MIN((TICK() - t0) \* 6.66666667, 1) \* 90

if onGround then break end

jumpAnim.Pos = GunMath.numLerp(startJumpPos, endJumpPos, Sine(Alpha))

jumpAnim.Rot = GunMath.numLerp(startJumpRot, endJumpRot, Sine(Alpha))

if Alpha == 90 then break end

end

startJumpPos = endJumpPos

startJumpRot = endJumpRot

endJumpPos = -0.08 \* S.fallSettings.fallMultiplier

endJumpRot = -RAD(8) \* S.fallSettings.fallMultiplier

local X = 1

while true do

local dt = Services.RS.Heartbeat:wait()

X = X + (dt \* 60) / X

local Alpha = (X - 1) \* 0.0666667

if onGround then break end

jumpAnim.Pos = GunMath.numLerp(startJumpPos, endJumpPos, Alpha)

jumpAnim.Rot = GunMath.numLerp(startJumpRot, endJumpRot, Alpha)

end

end)

end

function onLand(fallDist)

runAsync(function()

local animAlpha = MIN(fallDist, S.fallSettings.maxDist) \* (0.6666667)

local startJumpPos = jumpAnim.Pos

local startJumpRot = jumpAnim.Rot

local endJumpPos = animAlpha \* 0.01 \* S.fallSettings.landMultiplier \* (runReady and 1 or 2)

local endJumpRot = RAD(animAlpha) \* S.fallSettings.landMultiplier \* (runReady and 1 or 2)

local t0 = TICK()

while true do

Services.RS.Heartbeat:wait()

local Alpha = MIN((TICK() - t0) \* 5, 1)

if (not onGround) then break end

jumpAnim.Pos = GunMath.numLerp(startJumpPos, endJumpPos, Alpha)

jumpAnim.Rot = GunMath.numLerp(startJumpRot, endJumpRot, Alpha)

if Alpha == 1 then break end

end

t0 = TICK()

while true do

Services.RS.Heartbeat:wait()

local Alpha = MIN((TICK() - t0) \* 30, 1) \* 90

if (not onGround) then break end

jumpAnim.Pos = GunMath.numLerp(endJumpPos, 0, Sine(Alpha))

jumpAnim.Rot = GunMath.numLerp(endJumpRot, 0, Sine(Alpha))

if Alpha == 90 then break end

end

end)

end

function onHumanoidStateChanged(oldState, newState)

if newState == Enum.HumanoidStateType.Freefall then

onGround = false

if S.fallAnimation then

onFall(HRP.Velocity.Y)

while HRP.Velocity.Y > 0 do shortWait(Services.RS.RenderStepped) end

startFallHeight = HRP.Position.Y

end

elseif oldState == Enum.HumanoidStateType.Freefall then

onGround = true

if S.fallAnimation then

local fallDist = startFallHeight - HRP.Position.Y

onLand(fallDist)

end

end

end

--------------------[ CAMERA STEADYING FUNCTIONS ]------------------------------------

function steadyCamera()

scopeSteady.Text = "Steadying..."

scopeSteady.TextColor3 = C3(1, 1, 0)

camSteady = true

local originalSway = camSway

local Increment = 1.5 / 0.6

local X = 0

while true do

shortWait(Services.RS.Heartbeat)

local newX = X + Increment

X = (newX > 90 and 90 or newX)

if (not steadyKeyPressed) then break end

camSway = GunMath.numLerp(originalSway, 0, Sine(X))

if X == 90 then break end

end

while steadyKeyPressed and Aimed do

if Stamina.Value > 0 then

local NewSteadyTime = Stamina.Value - 1

Stamina.Value = (NewSteadyTime < 0 and 0 or NewSteadyTime)

camSway = 0

elseif Stamina.Value == 0 then

break

end

shortWait(Services.RS.Heartbeat)

end

camSteady = false

runAsync(function()

local Increment = 1.5 \* 4

local X = 0

while true do

shortWait(Services.RS.Heartbeat)

local newX = X + Increment

X = (newX > 90 and 90 or newX)

if camSteady then break end

camSway = GunMath.numLerp(0, S.scopeSettings.camSwayOnBreath, 1 - COS(RAD(X)))

if X == 90 then break end

end

Increment = 1.5 / S.scopeSettings.breathTime

X = 0

while true do

shortWait(Services.RS.Heartbeat)

local newX = X + Increment

X = (newX > 90 and 90 or newX)

if camSteady then break end

camSway = GunMath.numLerp(S.scopeSettings.camSwayOnBreath, 1, Sine(X))

if X == 90 then break end

end

--[[for X = 0, 90, 1.5 / 0.2 do

local Alpha = 1 - COS(RAD(X))--math.log10(X) / math.log10(90)

camSway = GunMath.numLerp(0, 3, Alpha)

RS.RenderStepped:wait()

end]]

--[[for X = 0, 90, 1.5 / S.scopeSettings.steadyTime do

if camSteady then break end

local Alpha = SIN(RAD(X))

camSway = numLerp(3, 1, Alpha)

RS.RenderStepped:wait()

end]]

end)

retakeBreath()

end

function retakeBreath()

scopeSteady.Text = "Re-taking Breath"

scopeSteady.TextColor3 = C3(1, 0, 0)

takingBreath = true

local Increment = S.scopeSettings.steadyTime / S.scopeSettings.breathTime

while takingBreath do

if Stamina.Value < maxStamina then

local newSteadyTime = Stamina.Value + Increment

Stamina.Value = (newSteadyTime > maxStamina and maxStamina or newSteadyTime)

elseif Stamina.Value >= maxStamina then

break

end

wait(0.25)

end

if takingBreath then

scopeSteady.Text = "Hold Left-Shift to Steady"

scopeSteady.TextColor3 = C3(1, 1, 0)

takingBreath = false

end

end

--------------------[ SPRINTING FUNCTIONS ]-------------------------------------------

function can(...)

local args = {...}

if args[1] == "Run" then

return ((Forward and (not Backward)) and

Walking and (Stamina.Value > 0) and Running and

Selected and (args[2] and true or onGround) and

runReady and (S.canFireWhileRunning and true or (not Firing))

)

end

end

function monitorStamina()

Running = true

if (not can("Run",false)) then

Running = false

return

end

if Aimed then unAimGun(true) end

if Stance == 1 or Stance == 2 then

changeStance("Stand")

end

if (not (Reloading and S.reloadSettings.Anim)) then

if S.canFireWhileRunning then

tween("Joint",ArmWelds.LWeld, armC0[1], S.unAimedC1.leftArm, Sine, 0.4)

tween("Joint",ArmWelds.RWeld, armC0[2], S.unAimedC1.rightArm, Sine, 0.4)

tween("Joint",Grip, nil, S.unAimedC1.Grip, Sine, 0.4)

else

tween("Joint",ArmWelds.LWeld, armC0[1], S.runningC1.leftArm, Sine, 0.4)

tween("Joint",ArmWelds.RWeld, armC0[2], S.runningC1.rightArm, Sine, 0.4)

tween("Joint",Grip, nil, S.runningC1.Grip, Sine, 0.4)

end

end

crossOffset = 50

while runKeyPressed do

if can("Run",true) then

if onGround and WalkingTouch then

local newStamina = Stamina.Value - 1

Stamina.Value = (newStamina < 0 and 0 or newStamina)

end

else

break

end

wait(0.25)

end

Running = false

if (not Aimed) and (not (Reloading and S.reloadSettings.Anim)) and (not S.canFireWhileRunning) then

crossOffset = 0

tween("Joint",ArmWelds.LWeld, armC0[1], S.unAimedC1.leftArm, Sine, 0.4)

tween("Joint",ArmWelds.RWeld, armC0[2], S.unAimedC1.rightArm, Sine, 0.4)

tween("Joint",Grip, nil, S.unAimedC1.Grip, Sine, 0.4)

end

end

--------------------[ STANCE FUNCTIONS ]----------------------------------------------

function changeStance(...)

local args = {...}

if args[1] == "Stand" then

local LHip = Torso["Left Hip"]

local RHip = Torso["Right Hip"]

LLegWeld.Part1 = nil

LHip.Part1 = LLeg

RLegWeld.Part1 = nil

RHip.Part1 = RLeg

Stance = 0

spreadStance = "Stand"

baseSpread = S.spreadSettings[spreadZoom][spreadStance][spreadMotion]

if S.stanceSettings.Anim and (not args[2]) then

runAsync(function()

local prevStanceSway = stanceSway

local X = 0

local Increment = 1.5 / S.stanceSettings.Speed

while true do

shortWait(Services.RS.Heartbeat)

local newX = X + Increment

X = (newX > 90 and 90 or newX)

if Stance ~= 0 then break end

stanceSway = GunMath.numLerp(prevStanceSway, 1, Sine(X))

if X == 90 then break end

end

end)

tween("Joint",ABWeld, CF.RAW(), nil, Sine, S.stanceSettings.Speed)

tween("Joint",LLegWeld, legC0.Stand[1], nil, Sine, S.stanceSettings.Speed)

tween("Joint",RLegWeld, legC0.Stand[2], nil, Sine, S.stanceSettings.Speed)

tween("Joint",LHip, CF.RAW(-1, -1, 0) \* CF.ANG(0, RAD(-90), 0), CF.RAW(-0.5, 1, 0) \* CF.ANG(0, RAD(-90), 0), Sine, S.stanceSettings.Speed)

tween("Joint",RHip, CF.RAW(1, -1, 0) \* CF.ANG(RAD(-180), RAD(90), 0), CF.RAW(0.5, 1, 0) \* CF.ANG(RAD(-180), RAD(90), 0), Sine, S.stanceSettings.Speed)

tween("Joint",Root, CF.ANG(RAD(-90), 0, RAD(180)), nil, Sine, S.stanceSettings.Speed)

tween("Joint",headWeld, CF.RAW(0, 1.5, 0), nil, Sine, S.stanceSettings.Speed)

elseif args[2] or (not S.stanceSettings.Anim) then

ABWeld.C0 = CF.RAW()

LLegWeld.C0 = legC0.Stand[1]

RLegWeld.C0 = legC0.Stand[2]

LHip.C0, LHip.C1 = CF.RAW(-1, -1, 0) \* CF.ANG(0, RAD(-90), 0), CF.RAW(-0.5, 1, 0) \* CF.ANG(0, RAD(-90), 0)

RHip.C0, RHip.C1 = CF.RAW(1, -1, 0) \* CF.ANG(RAD(-180), RAD(90), 0), CF.RAW(0.5, 1, 0) \* CF.ANG(RAD(-180), RAD(90), 0)

Root.C0 = CF.ANG(RAD(-90), 0, RAD(180))

headWeld.C0 = CF.RAW(0, 1.5, 0)

end

elseif args[1] == "Crouch" then

local LHip = Torso["Left Hip"]

local RHip = Torso["Right Hip"]

LHip.Part1 = nil

LLegWeld.Part1 = LLeg

RHip.Part1 = nil

RLegWeld.Part1 = RLeg

Stance = 1

spreadStance = "Crouch"

baseSpread = S.spreadSettings[spreadZoom][spreadStance][spreadMotion]

if S.stanceSettings.Anim then

runAsync(function()

local prevStanceSway = stanceSway

local X = 0

local Increment = 1.5 / S.stanceSettings.Speed

while true do

shortWait(Services.RS.Heartbeat)

local newX = X + Increment

X = (newX > 90 and 90 or newX)

if Stance ~= 1 then break end

stanceSway = GunMath.numLerp(prevStanceSway, 0.75, Sine(X))

if X == 90 then break end

end

end)

tween("Joint",ABWeld, CF.RAW(0, 0, -0.05), nil, Sine, S.stanceSettings.Speed)

tween("Joint",LLegWeld, legC0.Crouch[1], nil, Sine, S.stanceSettings.Speed)

tween("Joint",RLegWeld, legC0.Crouch[2], nil, Sine, S.stanceSettings.Speed)

tween("Joint",LHip, CF.RAW(-1, -0.5, 0) \* CF.ANG(0, RAD(-90), 0), CF.RAW(-0.5, 0.5, 1) \* CF.ANG(0, RAD(-90), RAD(-90)), Sine, S.stanceSettings.Speed)

tween("Joint",RHip, CF.RAW(1, -0.5, 0.25) \* CF.ANG(RAD(-180), RAD(90), 0), CF.RAW(0.5, 0.5, 1) \* CF.ANG(RAD(-180), RAD(90), 0), Sine, S.stanceSettings.Speed)

tween("Joint",Root, CF.RAW(0, -1, 0) \* CF.ANG(RAD(-90), 0, RAD(180)), nil, Sine, S.stanceSettings.Speed)

tween("Joint",headWeld, CF.RAW(0, 1.5, 0), nil, Sine, S.stanceSettings.Speed)

else

ABWeld.C0 = CF.RAW(0, 0, -0.0625)

LLegWeld.C0 = legC0.Crouch[1]

RLegWeld.C0 = legC0.Crouch[2]

LHip.C0, LHip.C1 = CF.RAW(-1, -0.5, 0) \* CF.ANG(0, RAD(-90), 0), CF.RAW(-0.5, 0.5, 1) \* CF.ANG(0, RAD(-90), RAD(-90))

RHip.C0, RHip.C1 = CF.RAW(1, -0.5, 0.25) \* CF.ANG(RAD(-180), RAD(90), 0), CF.RAW(0.5, 0.5, 1) \* CF.ANG(RAD(-180), RAD(90), 0)

Root.C0 = CF.RAW(0, -1, 0) \* CF.ANG(RAD(-90), 0, RAD(180))

headWeld.C0 = CF.RAW(0, 1.5, 0)

end

elseif args[1] == "Prone" then

local LHip = Torso["Left Hip"]

local RHip = Torso["Right Hip"]

LHip.Part1 = nil

LLegWeld.Part1 = LLeg

RHip.Part1 = nil

RLegWeld.Part1 = RLeg

Stance = 2

spreadStance = "Prone"

baseSpread = S.spreadSettings[spreadZoom][spreadStance][spreadMotion]

if S.stanceSettings.Anim then

runAsync(function()

local prevStanceSway = stanceSway

local X = 0

local Increment = 1.5 / S.stanceSettings.Speed

while true do

shortWait(Services.RS.Heartbeat)

local newX = X + Increment

X = (newX > 90 and 90 or newX)

if Stance ~= 2 then break end

stanceSway = GunMath.numLerp(prevStanceSway, 0.5, Sine(X))

if X == 90 then break end

end

end)

tween("Joint",ABWeld, CF.RAW(0, 0, -0.1), nil, Sine, S.stanceSettings.Speed)

tween("Joint",LLegWeld, legC0.Prone[1], nil, Sine, S.stanceSettings.Speed)

tween("Joint",RLegWeld, legC0.Prone[2], nil, Sine, S.stanceSettings.Speed)

tween("Joint",Root, CF.RAW(0, -2.5, 1) \* CF.ANG(RAD(180), 0, RAD(180)), nil, Sine, S.stanceSettings.Speed)

tween("Joint",headWeld, CF.RAW(0, 1, 1) \* CF.ANG(RAD(90), 0, 0), nil, Sine, S.stanceSettings.Speed)

else

ABWeld.C0 = CF.RAW(0, 0, -0.125)

LLegWeld.C0 = legC0.Prone[1]

RLegWeld.C0 = legC0.Prone[2]

Root.C0 = CF.RAW(0, -2.5, 1) \* CF.ANG(RAD(180), 0, RAD(180))

headWeld.C0 = CF.RAW(0, 1, 1) \* CF.ANG(RAD(90), 0, 0)

end

elseif args[1] == "Dive" then

onGround = false

local diveDirection = (HRP.CFrame \* CF.ANG(S.diveSettings.Angle, 0, 0)).lookVector \* S.walkSpeeds.Sprinting \* S.diveSettings.Force

spawn(function()

HRP.Velocity=HRP.CFrame.lookVector\*60+V3.RAW(0,40,0)

wait(.1)

HRP.Velocity=HRP.CFrame.lookVector\*70+V3.RAW(0,30,0)

wait(.4)

HRP.Velocity=HRP.CFrame.lookVector\*30+V3.RAW(0,-10,0)

end)

runAsync(function()

while true do

local newRay = RAY(HRP.Position, V3.RAW(0, -3.1, 0))

local H, \_ = raycast(Services.WS,newRay, Ignore)

if H then

onGround = true

break

end

wait()

end

end)

changeStance("Prone")

wait(0.1)

end

end

--------------------[ MOUSE FUNCTIONS ]-----------------------------------------------

function onMB1Down()

MB1Down = true

firstShot = true

if fireFunction then

fireFunction()

end

end

function onMB1Up()

MB1Down = false

lowerSpread()

end

function onMB2Down()

if modeGUI.Text ~= "HOOK" then

if S.aimSettings.holdToADS then

if (not AimingIn) and (not Aimed) then

AimingIn = true

aimGun()

AimingIn = false

end

else

if Aimed then

unAimGun()

else

aimGun()

end

end

else

script.Parent.ReelValue.Value = (Services.UIS:IsKeyDown("S") and -1 or (Services.UIS:IsKeyDown("W") and 1 or 0))

end

end

function onMB2Up()

if S.aimSettings.holdToADS then

if (not AimingOut) and Aimed then

AimingOut = true

unAimGun()

AimingOut = false

end

end

end

Gun.Sensitivity.Changed:connect(function(sensitivity)

Sensitivity.aim = (

sensitivity < S.sensitivitySettings.Min and S.sensitivitySettings.Min or

sensitivity > S.sensitivitySettings.Max and S.sensitivitySettings.Max or

sensitivity

)

Gun.LookSensitivity.Value = (Aimed and Sensitivity.aim or Gun.LookSensitivity.Value)

Sensitivity.mouse = Gun.LookSensitivity.Value

end)

--local newAimSensitivity = aimSensitivity + S.sensitivitySettings.Increment

--local newAimSensitivity = aimSensitivity - S.sensitivitySettings.Increment

--------------------[ KEYBOARD FUNCTIONS ]--------------------------------------------

local function GamepadDown(input)

local ActionVars = {

Player = Player;

Reload = Reload;

Reloading = Reloading;

isCrawling = isCrawling;

setRunKeyPressed = function(bool)

runKeyPressed = bool

end;

CanRun = function()

return (not Idling) and Walking and (not Running) and (not Knifing) and (not (Aimed and S.guiScope and S.Keys.Sprint == S.Keys.scopeSteady))

end;

monitorStamina = monitorStamina;

runReady = runReady;

holdToADS = S.aimSettings.holdToADS;

hasNotAimedYet = function()

return (not AimingIn) and (not Aimed)

end;

holdAndAim = function()

AimingIn = true

aimGun()

AimingIn = false

end;

rechargeTime = S.diveSettings.rechargeTime;

toggleAim = function()

if Aimed then

unAimGun()

else

aimGun()

end

end;

canChangeStance = S.canChangeStance;

Running = Running;

isProne = function()

return Stance == 2

end;

canCrouch = S.stanceSettings.Stances.Crouch;

canGoProne = S.stanceSettings.Stances.Prone;

selectFire = S.selectFire;

canSelectFire = canSelectFire;

runAsync = runAsync;

toggleFireSelect = function(bool)

Interface.fireSelect.Visible = bool

end;

selectFireAnimSpeed = S.selectFireSettings.animSpeed;

waitRenderStepped = function()

shortWait(Services.RS.RenderStepped)

end;

getRawFireMode = function()

return rawFireMode

end;

updateGUI = updateGUI;

numModes = numModes;

fireModes = fireModes;

ableToSelectFire = function()

return S.selectFireSettings.Animation and (not Aimed) and (not isRunning) and (not isCrawling)

end;

CF = CF.RAW;

CFANG = CF.ANG;

RAD = RAD;

Sine = Sine;

tween = tween;

LWeld = ArmWelds.LWeld;

RWeld = RWeld2;

LC0 = armC0[1];

Linear = Linear;

unAimedLeftArmC1 = S.unAimedC1.leftArm;

PAIRS = PAIRS;

setupSteady = function()

steadyKeyPressed = true;

end;

unAbleToSelectFire = function()

return Aimed or isRunning or isCrawling or Reloading

end;

hasSelectFireMedia = function()

return S.selectFireSettings.Animation or S.selectFireSettings.GUI

end;

hasSelectFireGUI = S.selectFireSettings.GUI;

selectFireMediaSpeed = S.selectFireSettings.animSpeed;

changeMode = function()

rawFireMode = rawFireMode + 1

modeGUI.Text = Modes[((rawFireMode - 1) % numModes) + 1]

end;

setFireFunction = function()

fireFunction = getFire(modeGUI.Text)

end;

dolphinDive = S.dolphinDive;

canDive = function()

return Humanoid:GetState() ~= Enum.HumanoidStateType.Freefall and (not Services.UIS:IsKeyDown("Space")) and runKeyPressed

end;

Humanoid = Humanoid;

changeStance = function(stance)

changeStance(stance)

end;

isCrouching = function()

return Stance == 1;

end;

isStanding = function()

return Stance == 0;

end;

readyToAim = function()

return Aimed and (not Aiming);

end;

holdBreath = function()

takingBreath = false;

end;

steadyCamera = steadyCamera;

}

local Key =(input.KeyCode)

if S.Debug then

mainGUI.Debug.Key.Text = Key.Name

end

if Key ~= S.GamepadButtons.SpecialKey then

if Libraries.DownActions[Libraries.Keybinds.GamepadButtons[Key]] ~= nil then

Libraries.DownActions[Libraries.Keybinds.GamepadButtons[Key]](ActionVars)

elseif Key == S.GamepadButtons.Unequip then

Humanoid:UnequipTools()

end

else

if Aimed and S.guiScope then

Libraries.DownActions[Libraries.Keybinds.GamepadButtons[S.GamepadButtons.SpecialKey][2]](ActionVars)

else

Libraries.DownActions[Libraries.Keybinds.GamepadButtons[S.GamepadButtons.SpecialKey][1]](ActionVars)

end

end

end

local function GamepadUp(input)

local ActionVars = {

holdToADS = S.aimSettings.holdToADS;

hasNotAimedOutYet = function()

return (not AimingOut) and Aimed

end;

unAim = function()

AimingOut = true

unAimGun()

AimingOut = false

end;

stopRunning = function()

runKeyPressed = false

Running = false

end;

unSteadyCamera = function()

steadyKeyPressed = false

end

}

local Key = input.KeyCode

if S.Debug then

mainGUI.Debug.Key.Text = Key.Name

end

if Key ~= S.Keys.SpecialKey then

if Libraries.UpActions[Libraries.Keybinds.GamepadButtons[Key]] ~= nil then

Libraries.UpActions[Libraries.Keybinds.GamepadButtons[Key]](ActionVars)

end

else

if Aimed and S.guiScope then

Libraries.UpActions[Libraries.Keybinds.GamepadButtons[S.GamepadButtons.SpecialKey][2]](ActionVars)

else

Libraries.UpActions[Libraries.Keybinds.GamepadButtons[S.GamepadButtons.SpecialKey][1]](ActionVars)

end

end

end

function keyDown(inputObj)

local ActionVars = {

Player = Player;

Reload = Reload;

Reloading = Reloading;

isCrawling = isCrawling;

setRunKeyPressed = function(bool)

runKeyPressed = bool

end;

CanRun = function()

return (not Idling) and Walking and (not Running) and (not Knifing) and (not (Aimed and S.guiScope and S.Keys.Sprint == S.Keys.scopeSteady))

end;

monitorStamina = monitorStamina;

runReady = runReady;

holdToADS = S.aimSettings.holdToADS;

hasNotAimedYet = function()

return (not AimingIn) and (not Aimed)

end;

holdAndAim = function()

AimingIn = true

aimGun()

AimingIn = false

end;

rechargeTime = S.diveSettings.rechargeTime;

toggleAim = function()

if Aimed then

unAimGun()

else

aimGun()

end

end;

canChangeStance = S.canChangeStance;

Running = Running;

isProne = function()

return Stance == 2

end;

canCrouch = S.stanceSettings.Stances.Crouch;

canGoProne = S.stanceSettings.Stances.Prone;

selectFire = S.selectFire;

canSelectFire = canSelectFire;

runAsync = runAsync;

toggleFireSelect = function(bool)

Interface.fireSelect.Visible = bool

end;

selectFireAnimSpeed = S.selectFireSettings.animSpeed;

waitRenderStepped = function()

shortWait(Services.RS.RenderStepped)

end;

getRawFireMode = function()

return rawFireMode

end;

updateGUI = updateGUI;

numModes = numModes;

fireModes = fireModes;

ableToSelectFire = function()

return S.selectFireSettings.Animation and (not Aimed) and (not isRunning) and (not isCrawling)

end;

CF = CF.RAW;

CFANG = CF.ANG;

RAD = RAD;

Sine = Sine;

tween = tween;

LWeld = ArmWelds.LWeld;

RWeld = RWeld2;

LC0 = armC0[1];

Linear = Linear;

unAimedLeftArmC1 = S.unAimedC1.leftArm;

PAIRS = PAIRS;

setupSteady = function()

steadyKeyPressed = true;

end;

unAbleToSelectFire = function()

return Aimed or isRunning or isCrawling or Reloading

end;

hasSelectFireMedia = function()

return S.selectFireSettings.Animation or S.selectFireSettings.GUI

end;

hasSelectFireGUI = S.selectFireSettings.GUI;

selectFireMediaSpeed = S.selectFireSettings.animSpeed;

changeMode = function()

rawFireMode = rawFireMode + 1

modeGUI.Text = Modes[((rawFireMode - 1) % numModes) + 1]

end;

setFireFunction = function()

fireFunction = getFire(modeGUI.Text)

end;

dolphinDive = S.dolphinDive;

canDive = function()

return Humanoid:GetState() ~= Enum.HumanoidStateType.Jumping and (not Services.UIS:IsKeyDown("Space")) and runKeyPressed

end;

Humanoid = Humanoid;

changeStance = function(stance)

changeStance(stance)

end;

isCrouching = function()

return Stance == 1;

end;

isStanding = function()

return Stance == 0;

end;

readyToAim = function()

return Aimed and (not Aiming);

end;

holdBreath = function()

takingBreath = false;

end;

steadyCamera = steadyCamera;

}

local Key =(inputObj.KeyCode)

if S.Debug then

mainGUI.Debug.Key.Text = Key.Name

end

if Key ~= S.Keys.SpecialKey then

if Libraries.DownActions[Libraries.Keybinds.Keys[Key]] ~= nil then

Libraries.DownActions[Libraries.Keybinds.Keys[Key]](ActionVars)

end

else

if Aimed and S.guiScope then

Libraries.DownActions[Libraries.Keybinds.Keys[S.Keys.SpecialKey][2]](ActionVars)

else

Libraries.DownActions[Libraries.Keybinds.Keys[S.Keys.SpecialKey][1]](ActionVars)

end

end

for \_, PTable in PAIRS(Libraries.Plugins.KeyDown) do

if Key ==(PTable.Key) then

spawn(function()

PTable.Plugin()

end)

end

end

end

function keyUp(inputObj)

local ActionVars = {

holdToADS = S.aimSettings.holdToADS;

hasNotAimedOutYet = function()

return (not AimingOut) and Aimed

end;

unAim = function()

AimingOut = true

unAimGun()

AimingOut = false

end;

stopRunning = function()

runKeyPressed = false

Running = false

end;

unSteadyCamera = function()

steadyKeyPressed = false

end

}

local Key = inputObj.KeyCode

if S.Debug then

mainGUI.Debug.Key.Text = Key.Name

end

if Key ~= S.Keys.SpecialKey then

if Libraries.UpActions[Libraries.Keybinds.Keys[Key]] ~= nil then

Libraries.UpActions[Libraries.Keybinds.Keys[Key]](ActionVars)

end

else

if Aimed and S.guiScope then

Libraries.UpActions[Libraries.Keybinds.Keys[S.Keys.SpecialKey][2]](ActionVars)

else

Libraries.UpActions[Libraries.Keybinds.Keys[S.Keys.SpecialKey][1]](ActionVars)

end

end

for \_, PTable in PAIRS(Libraries.Plugins.KeyUp) do

if Key == (PTable.Key) then

runAsync(function()

PTable.Plugin()

end)

end

end

end

for \_, v in pairs(Gun.Keybinds:GetChildren()) do

v.Changed:connect(function(value)

Libraries.Keybinds:Recalibrate()

end)

end

--------------------[ END FUNCTIONS ]-------------------------------------------------

--------------------------------------------------------------------------------------

--------------------[ PRE-CONNECTIONS ]-----------------------------------------------

--------------------------------------------------------------------------------------

local function updateAnimVars()

wait()

Forward = (Services.UIS:IsKeyDown("W") or Services.UIS:IsKeyDown("Up") or WalkingGamepad or WalkingTouch)

Backward = (Services.UIS:IsKeyDown("S") or Services.UIS:IsKeyDown("Down"))

local Right = Services.UIS:IsKeyDown("D")

local Left = Services.UIS:IsKeyDown("A")

local walkingForward = (Forward and (not Backward))

local walkingBackward = ((not Forward) and Backward)

local walkingRight = (Right and (not Left))

local walkingLeft = ((not Right) and Left)

if (Forward or Backward or Right or Left) then

Walking, Idling = true, false

if (not Running) and (not Aimed) then

spreadMotion = "Moving"

baseSpread = S.spreadSettings[spreadZoom][spreadStance][spreadMotion]

end

elseif (not (Forward and Backward and Right and Left)) then

Walking, Idling = false, true

if (not Aimed) then

spreadMotion = "Idling"

baseSpread = S.spreadSettings[spreadZoom][spreadStance][spreadMotion]

end

end

local newArmTilt = (

((walkingForward or walkingBackward) and walkingRight) and 2.5 or

((walkingForward or walkingBackward) and walkingLeft) and -2.5 or

((not (walkingForward and walkingBackward)) and walkingRight) and 5 or

((not (walkingForward and walkingBackward)) and walkingLeft) and -5 or 0

)

local newMoveAng = (

(walkingForward and (not (walkingRight or walkingLeft))) and 0 or

(walkingForward and walkingRight) and RAD(-45) or

((not (walkingForward or walkingBackward)) and walkingRight) and RAD(-90) or

(walkingBackward and walkingRight) and RAD(-135) or

(walkingBackward and (not (walkingRight or walkingLeft))) and (moveAng < 0 and RAD(-180) or RAD(180)) or

(walkingBackward and walkingLeft) and RAD(135) or

((not (walkingForward or walkingBackward)) and walkingLeft) and RAD(90) or

(walkingForward and walkingLeft) and RAD(45) or 0

)

local newAnimCode = RANDOM(-1e9, 1e9)

animCode = newAnimCode

local startTilt = armTilt

local startAng = (ABS(moveAng) == RAD(180)) and (newMoveAng > 0 and RAD(180) or RAD(-180)) or moveAng

local Increment = (startTilt == newArmTilt and 1.5 / 0.7 or 1.5 / (0.35 \* ABS(startTilt - newArmTilt) / 5))

local X = 0

local armLerp, angLerp

while true do

shortWait(Services.RS.Heartbeat)

local newX = X + Increment

X = (newX > 90 and 90 or newX)

if animCode ~= newAnimCode then break end

armLerp = GunMath.numLerp(startTilt, newArmTilt, Sine(X))

armTilt = armLerp

angLerp = GunMath.numLerp(startAng, newMoveAng, Sine(X))

moveAng = angLerp

if X == 90 then break end

end

end

Services.UIS.InputBegan:connect(updateAnimVars)

Services.UIS.InputEnded:connect(updateAnimVars)

updateAnimVars()

function setUpTouchControls()

if S.aimSettings.holdToADS == false then

INSERT(Connections,Services.UIS.TouchPinch:connect(function()

local aimFunc = nil

aimFunc = (Aimed and unAimGun or aimGun)

aimFunc()

end))

end

if Services.UIS.TouchEnabled then

INSERT(Connections,Services.RS.RenderStepped:connect(function()

if runKeyPressed then

monitorStamina()

end

end))

INSERT(Connections, mainGUI.MobileCombat.ToggleSprint.Button.MouseButton1Click:connect(function()

runKeyPressed = not runKeyPressed

end))

INSERT(Connections, mainGUI.MobileCombat.Reload.Button.MouseButton1Click:connect(function()

if (not Reloading) and (not isCrawling) then

Reload()

end

end))

INSERT(Connections, mainGUI.MobileCombat.SelectFire.Button.MouseButton1Click:connect(function()

if canSelectFire then

canSelectFire = false

rawFireMode = rawFireMode + 1

modeGUI.Text = Modes[((rawFireMode - 1) % numModes) + 1]

fireFunction = getFire(modeGUI.Text)

local speedAlpha = S.selectFireSettings.animSpeed / 0.6

if S.selectFireSettings.GUI then

spawn(function()

Interface.crossHair.Visible = false

Interface.fireSelect.Visible = true

local prevRawFireMode = rawFireMode

updateGUI("ModeLabels",((rawFireMode - 1) % numModes))

fireModes.FireModeController.Stopped:wait()

Interface.fireSelect.Visible = false

Interface.crossHair.Visible = true

end)

end

if S.selectFireSettings.Animation and (not Aimed) and (not isRunning) and (not isCrawling) then

spawn(function()

local sequenceTable = {

function()

tween("Joint",RWeld2, nil, CF.ANG(0, RAD(5), 0), Sine, speedAlpha \* 0.15)

tween("Joint",ArmWelds.LWeld, armC0[1], CF.RAW(0.1, 1, -0.3) \* CF.ANG(RAD(-7), 0, RAD(-65)), Linear, speedAlpha \* 0.15)

wait(speedAlpha \* 0.2)

end;

function()

tween("Joint",ArmWelds.LWeld, armC0[1], CF.RAW(0.1, 1, -0.3) \* CF.ANG(RAD(-10), 0, RAD(-65)), Linear, speedAlpha \* 0.1)

wait(speedAlpha \* 0.2)

end;

function()

tween("Joint",RWeld2, nil, CF.RAW(), Sine, speedAlpha \* 0.2)

tween("Joint",ArmWelds.LWeld, armC0[1], S.unAimedC1.leftArm, Sine, speedAlpha \* 0.2)

wait(speedAlpha \* 0.2)

end;

}

for \_, F in pairs(sequenceTable) do

if Aimed or isRunning or isCrawling or Reloading then

break

end

F()

end

end)

end

if S.selectFireSettings.Animation or S.selectFireSettings.GUI then

wait(S.selectFireSettings.animSpeed)

end

canSelectFire = true

end

end))

INSERT(Connections, mainGUI.MobileCombat.ADS.Button.MouseButton1Click:connect(function()

if Aimed then

unAimGun()

else

aimGun()

end

end))

INSERT(Connections, mainGUI.MobileCombat.LowerStance.Button.MouseButton1Click:connect(function()

if (not Running) then

if Stance == 0 then

if S.stanceSettings.Stances.Crouch then

changeStance("Crouch")

elseif S.stanceSettings.Stances.Prone then

changeStance("Prone")

end

elseif Stance == 1 then

if S.stanceSettings.Stances.Prone then

changeStance("Prone")

end

end

elseif S.dolphinDive then

wait()

if Humanoid:GetState() ~= Enum.HumanoidStateType.Freefall and (not Services.UIS:IsKeyDown("Space")) and runReady then

local tempConnection = Humanoid.Changed:connect(function()

Humanoid.Jump = false

end)

runReady = false

changeStance("Dive")

Running = false

wait(S.diveSettings.rechargeTime)

tempConnection:disconnect()

runReady = true

end

end

end))

INSERT(Connections,mainGUI.MobileCombat.RaiseStance.Button.MouseButton1Click:connect(function()

if (not Running) then

if Stance == 2 then

if S.stanceSettings.Stances.Crouch then

changeStance("Crouch")

else

changeStance("Stand")

end

elseif Stance == 1 then

changeStance("Stand")

end

end

end))

end

INSERT(Connections,Player.Backpack.SetInput.Event:connect(function(t,direction,inputObject)

-- WalkingTouch = inputObject.Position.Y > 0.2

if t == "Camera" then

local rawCamAng = camAng - (VEC2(RAD(direction.x/2), RAD(direction.y/2)) \* Gun.Sensitivity.Value \* 0.25)

camAng = VEC2(rawCamAng.x, (rawCamAng.y > RAD(80) and RAD(80) or rawCamAng.y < RAD(-80) and RAD(-80) or rawCamAng.y))

desiredXOffset = math.min(math.max(inputObject.Position.X, -S.momentumSettings.maxInput), S.momentumSettings.maxInput)

desiredYOffset = math.min(math.max(inputObject.Position.Y, -S.momentumSettings.maxInput), S.momentumSettings.maxInput)

elseif t == "Movement" then

WalkingTouch = direction.Y > 0.2

updateAnimVars()

end

print("Touch Detected")

end))

end

--------------------------------------------------------------------------------------

--------------------[ TOOL SELECTION AND DESELECTION ]--------------------------------

--------------------------------------------------------------------------------------

local function InitArms()

if S.fakeArmSettings.showArmor then

if Char:FindFirstChild("Arm1") then

local fakeArm1 = OBJ.Clone(Char.Arm1)

fakeArm1.Parent = gunIgnore

local C = fakeArm1:GetChildren()

for i=1, #C do

if C[i]:IsA("BasePart") then

local W = OBJ.RAW("Weld")

W.Part0 = fakeArm1.Middle

W.Part1 = C[i]

local CJ = CF.RAW(fakeArm1.Middle.Position)

local C0 = CF.inverse(fakeArm1.Middle.CFrame)\*CJ

local C1 = CF.inverse(C[i].CFrame)\*CJ

W.C0 = C0

W.C1 = C1

W.Parent = fakeArm1.Middle

end

local Y = OBJ.RAW("Weld")

Y.Part0 = fakeLArm

Y.Part1 = fakeArm1.Middle

Y.C0 = CF.RAW(0, 0, 0)

Y.Parent = Y.Part0

end

local h = fakeArm1:GetChildren()

for i = 1, # h do

if h[i]:IsA("BasePart") then

h[i].Anchored = false

h[i].CanCollide = false

end

end

fakeLArm.Transparency = 1

end

if FFC(Char,"Arm2") then

local fakeArm2 = OBJ.Clone(Char.Arm2)

fakeArm2.Parent = gunIgnore

local C = OBJ.GetKids(fakeArm2)

for i=1, #C do

if C[i]:IsA("BasePart") then

local W = OBJ.RAW("Weld")

W.Part0 = fakeArm2.Middle

W.Part1 = C[i]

local CJ = CF.RAW(fakeArm2.Middle.Position)

local C0 = CF.inverse(fakeArm2.Middle.CFrame)\*CJ

local C1 = CF.inverse(C[i].CFrame)\*CJ

W.C0 = C0

W.C1 = C1

W.Parent = fakeArm2.Middle

end

local Y = OBJ.RAW("Weld")

Y.Part0 = fakeRArm

Y.Part1 = fakeArm2.Middle

Y.C0 = CF.RAW(0, 0, 0)

Y.Parent = Y.Part0

end

fakeRArm.Transparency = 1

local h = OBJ.GetKids(fakeArm2)

for i = 1, # h do

if h[i]:IsA("BasePart") then

h[i].Anchored = false

h[i].CanCollide = false

end

end

end

end

end

local function enableModing()

Player.UIStats.ModingEnabled.Value = false

Player.UIStats.GamepadMode.Value = 12

end

local function calculateMovement(inputObj)

if inputObj.UserInputType == InputTypes.MouseMovement then

local rawCamAng = camAng - (VEC2(RAD(inputObj.Delta.x), RAD(inputObj.Delta.y)) \* Gun.LookSensitivity.Value \* 0.25)

camAng = VEC2(rawCamAng.x, (rawCamAng.y > RAD(80) and RAD(80) or rawCamAng.y < RAD(-80) and RAD(-80) or rawCamAng.y))

desiredXOffset = MIN(MAX(inputObj.Delta.x, -S.momentumSettings.maxInput), S.momentumSettings.maxInput)

desiredYOffset = MIN(MAX(inputObj.Delta.y, -S.momentumSettings.maxInput), S.momentumSettings.maxInput)

elseif inputObj.UserInputType == InputTypes.Gamepad1 then

if inputObj.KeyCode == Enum.KeyCode.Thumbstick2 then

while VEC2(inputObj.Position.X,-inputObj.Position.Y).magnitude > 0.2 do

local rawCamAng = camAng - (VEC2(RAD(inputObj.Position.X)\*2, RAD(-inputObj.Position.Y)\*2) \* Gun.LookSensitivity.Value \* 0.25)

camAng = VEC2(rawCamAng.x, (rawCamAng.y > RAD(80) and RAD(80) or rawCamAng.y < RAD(-80) and RAD(-80) or rawCamAng.y))

desiredXOffset = MIN(MAX(inputObj.Position.X, -S.momentumSettings.maxInput), S.momentumSettings.maxInput)

desiredYOffset = MIN(MAX(-inputObj.Position.Y, -S.momentumSettings.maxInput), S.momentumSettings.maxInput)

shortWait(Services.RS.Heartbeat)

end

elseif inputObj.KeyCode == Enum.KeyCode.Thumbstick1 then

WalkingGamepad = (VEC2(inputObj.Position.X,-inputObj.Position.Y).magnitude > 0.2)

updateAnimVars()

end

end

end

Libraries.Network.listen("Client","Send","markHit\_"..Player.UserId,function()

markHit()

end)

Libraries.Network.listen("Client","Fetch","getAnimCF\_"..Player.UserId,function()

return getAnimCF()

end)

Libraries.Network.listen("Client","Fetch","penetrateWall\_"..Player.UserId,function(Wall, hitPos, Direction, Normal, Ignore, totalPDist, totalBDist, lastDamagedHumanoid)

return penetrateWall(Wall, hitPos, Direction, Normal, Ignore, totalPDist, totalBDist, lastDamagedHumanoid)

end)

local function onEquipped()

wait()

if Humanoid.Health ~= 0 and (not Selected) and Gun.Parent == Char then

Selected = true

breakReload = false

equipAnimPlaying = true

RSEED(TICK()) --This sets a new seed for the random function each time you select the gun

--------------------[ FAILSAFE RESETING ]-------------------------------------

for \_, GM in PAIRS(OBJ.GetKids(ignoreModel)) do

if GM.Name == "gunIgnore\_"..Player.Name then

OBJ.Destroy(GM)

end

end

for \_, c in PAIRS(Connections) do

c:disconnect()

end

Connections = {}

enableModing()

--------------------[ REMOTE GUN SETUP ]--------------------------------------

Libraries.Network.startClient();

local fakeLWeld,fakeRWeld,gIgnoreHum

if FE then

local Vars = {

ignoreModel = ignoreModel;

Humanoid = Humanoid;

Shoulders = Shoulders;

Torso = Torso;

Head = Head;

armC0 = armC0;

leftArmC1 = S.equipSettings.leftArmC1;

rightArmC1 = S.equipSettings.rightArmC1;

LArm = LArm;

RArm = RArm;

gunParts = gunParts;

Handle = Handle;

LArm = LArm;

RArm = RArm;

S = S;

gunIgnore = gunIgnore;

}

gunIgnore, playerFolder, headWeld, headWeld2, animWeld, ABWeld, ArmWelds.LWeld, ArmWelds.RWeld, LWeld2, RWeld2, LLegWeld, RLegWeld, gunParts2 = Libraries.Network.fetch("Server","gunSetup\_"..Player.UserId,Vars)

local Vars2 = {

ignoreModel = ignoreModel;

Humanoid = Humanoid;

Shoulders = Shoulders;

Torso = Torso;

Head = Head;

armC0 = armC0;

leftArmC1 = S.equipSettings.leftArmC1;

rightArmC1 = S.equipSettings.rightArmC1;

LArm = LArm;

RArm = RArm;

gunParts = gunParts;

Handle = Handle;

LArm = LArm;

RArm = RArm;

S = S;

playerFolder = playerFolder;

gunIgnore = gunIgnore;

}

armModel = Libraries.Network.fetch("Server","Arms\_"..tostring(Player.UserId),Vars2)

fakeLArm = armModel["Left Arm"]

fakeRArm = armModel["Right Arm"]

fakeLWeld = armModel["Left Arm"]:FindFirstChild("FLWeld")

fakeRWeld = armModel["Right Arm"]:FindFirstChild("FRWeld")

gIgnoreHum = armModel.Humanoid

Grip = RArm:WaitForChild("RightGrip",200)

else

--------------------[ CREATING IGNORE MODELS ]--------------------------------

gunIgnore = OBJ.RAW("Model")

gunIgnore.Name = "gunIgnore\_"..Player.Name

gunIgnore.Parent = ignoreModel

print(gunIgnore)

--------------------[ MODIFYING THE PLAYER ]----------------------------------

Humanoid.AutoRotate = false

Shoulders.Right.Part1 = nil

Shoulders.Left.Part1 = nil

playerFolder = OBJ.RAW("Model")

playerFolder.Name = "playerFolder"

playerFolder.Parent = gunIgnore

local headBase = OBJ.RAW("Part")

headBase.Transparency = 1

headBase.Name = "headBase"

headBase.CanCollide = false

headBase.FormFactor = Enum.FormFactor.Custom

headBase.Size = V3.RAW(0.2, 0.2, 0.2)

headBase.BottomSurface = Enum.SurfaceType.Smooth

headBase.TopSurface = Enum.SurfaceType.Smooth

headBase.Parent = playerFolder

headWeld = OBJ.RAW("Motor6D")

headWeld.Part0 = Torso

headWeld.Part1 = headBase

headWeld.C0 = CF.RAW(0, 1.5, 0)

headWeld.Parent = Torso

headWeld2 = OBJ.RAW("Weld")

headWeld2.Part0 = headBase

headWeld2.Part1 = Head

headWeld2.Parent = headBase

neckClone = OBJ.Clone(Neck)

--[[local stanceBase = OBJ("Part")

stanceBase.Transparency = 1

stanceBase.Name = "stanceBase"

stanceBase.CanCollide = false

stanceBase.FormFactor = Enum.FormFactor.Custom

stanceBase.Size = V3(0.2, 0.2, 0.2)

stanceBase.BottomSurface = Enum.SurfaceType.Smooth

stanceBase.TopSurface = Enum.SurfaceType.Smooth

stanceBase.Parent = playerFolder

stanceWeld = OBJ("Weld")

stanceWeld.Part0 = stanceBase

stanceWeld.Part1 = Torso

stanceWeld.Parent = stanceBase]]

local animBase = OBJ.RAW("Part")

animBase.Transparency = 1

animBase.Name = "animBase"

animBase.CanCollide = false

animBase.FormFactor = Enum.FormFactor.Custom

animBase.Size = V3.RAW(0.2, 0.2, 0.2)

animBase.BottomSurface = Enum.SurfaceType.Smooth

animBase.TopSurface = Enum.SurfaceType.Smooth

animBase.Parent = playerFolder

animWeld = OBJ.RAW("Motor6D")

animWeld.Part0 = animBase

animWeld.Part1 = headBase

animWeld.Parent = animBase

local ArmBase = OBJ.RAW("Part")

ArmBase.Transparency = 1

ArmBase.Name = "ArmBase"

ArmBase.CanCollide = false

ArmBase.FormFactor = Enum.FormFactor.Custom

ArmBase.Size = V3.RAW(0.2, 0.2, 0.2)

ArmBase.BottomSurface = Enum.SurfaceType.Smooth

ArmBase.TopSurface = Enum.SurfaceType.Smooth

ArmBase.Parent = playerFolder

ABWeld = OBJ.RAW("Weld")

ABWeld.Part0 = ArmBase

ABWeld.Part1 = animBase

ABWeld.Parent = ArmBase

local LArmBase = OBJ.RAW("Part")

LArmBase.Transparency = 1

LArmBase.Name = "LArmBase"

LArmBase.CanCollide = false

LArmBase.FormFactor = Enum.FormFactor.Custom

LArmBase.Size = V3.RAW(0.2, 0.2, 0.2)

LArmBase.BottomSurface = Enum.SurfaceType.Smooth

LArmBase.TopSurface = Enum.SurfaceType.Smooth

LArmBase.Parent = playerFolder

local RArmBase = OBJ.RAW("Part")

RArmBase.Transparency = 1

RArmBase.Name = "RArmBase"

RArmBase.CanCollide = false

RArmBase.FormFactor = Enum.FormFactor.Custom

RArmBase.Size = V3.RAW(0.2, 0.2, 0.2)

RArmBase.BottomSurface = Enum.SurfaceType.Smooth

RArmBase.TopSurface = Enum.SurfaceType.Smooth

RArmBase.Parent = playerFolder

ArmWelds.LWeld = OBJ.RAW("Motor6D")

ArmWelds.LWeld.Name = "LWeld"

ArmWelds.LWeld.Part0 = ArmBase

ArmWelds.LWeld.Part1 = LArmBase

ArmWelds.LWeld.C0 = ArmWelds.LWeld.C0 \* armC0[1]

ArmWelds.LWeld.C1 = S.equipSettings.leftArmC1

ArmWelds.LWeld.Parent = ArmBase

ArmWelds.RWeld = OBJ.RAW("Motor6D")

ArmWelds.RWeld.Name = "RWeld"

ArmWelds.RWeld.Part0 = ArmBase

ArmWelds.RWeld.Part1 = RArmBase

ArmWelds.RWeld.C0 = ArmWelds.RWeld.C0 \* armC0[2]

ArmWelds.RWeld.C1 = S.equipSettings.rightArmC1

ArmWelds.RWeld.Parent = ArmBase

LWeld2 = OBJ.RAW("Weld")

LWeld2.Name = "LWeld"

LWeld2.Part0 = LArmBase

LWeld2.Part1 = LArm

LWeld2.Parent = LArmBase

RWeld2 = OBJ.RAW("Weld")

RWeld2.Name = "RWeld"

RWeld2.Part0 = RArmBase

RWeld2.Part1 = RArm

RWeld2.Parent = RArmBase

LLegWeld = OBJ.RAW("Motor6D")

LLegWeld.Name = "LLegWeld"

LLegWeld.Part0 = Torso

LLegWeld.Part1 = nil

LLegWeld.C0 = LLegWeld.C0 \* CF.RAW(-0.5, -2, 0)

LLegWeld.Parent = Torso

RLegWeld = OBJ.RAW("Motor6D")

RLegWeld.Name = "RLegWeld"

RLegWeld.Part0 = Torso

RLegWeld.Part1 = nil

RLegWeld.C0 = RLegWeld.C0 \* CF.RAW(0.5, -2, 0)

RLegWeld.Parent = Torso

if S.playerArms then

armModel = OBJ.RAW("Model")

armModel.Parent = (FE and playerFolder or Interface.Cam)

fakeLArm = OBJ.Clone(LArm)

fakeLArm.Parent = armModel

fakeLArm.Transparency = S.fakeArmSettings.Transparency

fakeLArm.CanCollide = false

fakeLArm.Size = S.fakeArmSettings.armSize

fakeLArm:BreakJoints()

--LArm.Transparency = 1

fakeLWeld = OBJ.RAW("Motor6D")

fakeLWeld.Part0 = fakeLArm

fakeLWeld.Part1 = LArm

fakeLWeld.Parent = fakeLArm

fakeRArm = OBJ.Clone(RArm)

fakeRArm.Parent = armModel

fakeRArm.Transparency = S.fakeArmSettings.Transparency

fakeRArm.CanCollide = false

fakeRArm.Size = S.fakeArmSettings.armSize

fakeRArm:BreakJoints()

InitArms()

--RArm.Transparency = 1

fakeRWeld = OBJ.RAW("Motor6D")

fakeRWeld.Part0 = fakeRArm

fakeRWeld.Part1 = RArm

fakeRWeld.Parent = fakeRArm

gIgnoreHum = OBJ.RAW("Humanoid")

gIgnoreHum.Parent = armModel

if S.fakeArmSettings.characterMeshes then

for \_,Obj in PAIRS(OBJ.GetKids(Char)) do

if Obj:IsA("CharacterMesh") then

OBJ.Clone(Obj).Parent = armModel

end

end

end

for \_,Obj in PAIRS(OBJ.GetKids(Char)) do

if Obj:IsA("Shirt") then

OBJ.Clone(Obj).Parent = armModel

end

end

else

armTable = create("Arms")

armTable[1].Model.Parent =(FE and playerFolder or Interface.Cam)--playerFolder

armTable[2].Model.Parent = (FE and playerFolder or Interface.Cam)--playerFolder

fakeLArm = armTable[1].armPart

--LArm.Transparency = 1

fakeLWeld = OBJ.RAW("Weld")

fakeLWeld.Part0 = fakeLArm

fakeLWeld.Part1 = LArm

fakeLWeld.Parent = fakeLArm

fakeRArm = armTable[2].armPart

--RArm.Transparency = 1

fakeRWeld = OBJ.RAW("Weld")

fakeRWeld.Part0 = fakeRArm

fakeRWeld.Part1 = RArm

fakeRWeld.Parent = fakeRArm

end

--------------------[ MODIFYING THE GUN ]-------------------------------------

for \_, Tab in PAIRS(gunParts) do

local Weld = OBJ.RAW("Weld")

Weld.Name = "MainWeld"

Weld.Part0 = Handle

Weld.Part1 = Tab.Obj

Weld.C0 = Tab.Obj.weldCF.Value

Weld.Parent = Handle

Tab.Weld = Weld

end

Grip = RArm:WaitForChild("RightGrip",200)

Grip.C1 = S.equipSettings.GripC1

end

local handleCF = Torso.CFrame \* CF.RAW(0, 0.5, 0) \* armC0[2] \* S.aimedC1.rightArm:inverse() \* Grip.C0

local handleOffset = CF.TOS(Gun.AimPart.CFrame,Handle.CFrame)

aimedGripCF = CF.TOS(((Torso.CFrame \* CF.RAW(headOffset.X, headOffset.Y, 0)) \* handleOffset),handleCF)

Player.CameraMode = Enum.CameraMode.LockFirstPerson

Interface.Cam.CameraType = CameraTypes.Scriptable

Interface.Cam.FieldOfView = 80

Services.UIS.MouseBehavior = (Services.UIS.TouchEnabled and Enum.MouseBehavior.Default or Enum.MouseBehavior.LockCenter)

Services.UIS.MouseIconEnabled = false

local initialX, initialY = GunMath.getYawPitch(Interface.Cam.CFrame)

camAng = -VEC2(initialX, initialY)

mainGUI.Enabled = true

setUpGUI()

runAsync(function()

local currentMode = Modes[((rawFireMode - 1) % numModes) + 1]

fireFunction =(S.selectFire and getFire(currentMode) or (S.GunType.Semi and getFire("SEMI") or (S.GunType.Auto and getFire("AUTO") or (S.GunType.Burst and getFire("BURST") or (S.GunType.Stun and getFire("STUN") and (S.GunType.Hook and getFire("HOOK") or getFire("SAFETY")))))))

end)

changePlayerTrans(Char, 1)

--------------------[ RUNNING PLUGINS ]---------------------------------------

for \_, Plugin in PAIRS(Libraries.Plugins.OnEquipped) do

runAsync(function()

Plugin()

end)

end

--------------------[ GETTING PLAYER MASS ]-----------------------------------

local connectedParts = HRP:GetConnectedParts(false)

runAsync(function()

for i = 1,#connectedParts do

if connectedParts[i]:IsA("BasePart") and connectedParts[i].Parent == Char then

runAsync(function()

playerMass = playerMass + connectedParts[i]

end)

end

end

end)

--------------------[ CONNECTIONS ]-------------------------------------------

setUpTouchControls()

INSERT(Connections, Humanoid.Died:connect(function()

onUnequipped(true)

end))

INSERT(Connections, Humanoid.Jumping:connect(function()

if Stance ~= 0 then

changeStance("Stand")

end

end))

INSERT(Connections, Humanoid.StateChanged:connect(onHumanoidStateChanged))

INSERT(Connections, Interface.M2.Button1Down:connect(onMB1Down))

INSERT(Connections, Interface.M2.Button1Up:connect(onMB1Up))

INSERT(Connections, Interface.M2.Button2Down:connect(onMB2Down))

INSERT(Connections, Interface.M2.Button2Up:connect(onMB2Up))

INSERT(Connections, Services.UIS.InputBegan:connect(keyDown))

INSERT(Connections, Services.UIS.InputBegan:connect(GamepadDown))

INSERT(Connections, Services.UIS.InputEnded:connect(GamepadUp))

INSERT(Connections, Services.UIS.InputEnded:connect(keyUp))

INSERT(Connections, Services.UIS.InputChanged:connect(function(inputObj,processed)

calculateMovement(inputObj)

end))

INSERT(Connections, Interface.M2.Idle:connect(function(inputObj)

desiredXOffset = 0

desiredYOffset = 0

end))

if Services.UIS.TouchEnabled then

mainGUI.MobileCombat.Visible = true

end

Interface.HUD.Visible = true

--[[ local SwitchLoadoutChanged,SwitchLConn = KI:CreateConnection({

"LeftShift","L"

})

INSERT(Connections,SwitchLoadoutChanged.Changed:connect(function(Activated)

if Activated then

Humanoid:UnequipTools()

Humanoid:EquipTool(script.Parent.Secondary.Value)

end

end))

]]--

INSERT(Connections,Services.RS.Heartbeat:connect(function()

--Main animation

local animC0, animC1 = getAnimCF()

animWeld.C0 = animC0

animWeld.C1 = animC1

if FE then

Libraries.Network.send("Server","changeAnimC1\_"..Player.UserId,animWeld,animWeld.C0,animWeld.C1)

end

--Camera updating

renderCamera()

end))

--------------------[ ANIMATE GUN ]-------------------------------------------

runAsync(function()

tween("Joint",ArmWelds.LWeld, nil, S.unAimedC1.leftArm, Sine, S.equipSettings.Time)

tween("Joint",ArmWelds.RWeld, nil, S.unAimedC1.rightArm, Sine, S.equipSettings.Time)

tween("Joint",Grip, nil, S.unAimedC1.Grip, Sine, S.equipSettings.Time)

end)

runAsync(function()

local T = TICK()

while true do

if TICK() - T > S.equipSettings.Time then break end

if (not Selected) then break end

Services.RS.Heartbeat:wait()

end

equipAnimPlaying = false

end)

--tween("Joint",ArmWelds.RWeld, nil, CFrame.new(-0.543740213, 0.246546745, -1.1437645) \* CF.ANG( 0.868195355, 2.16905711e-008, -0.496222436) , function(X) return math.sin(math.rad(X)) end, 0.3)

-- tween("Joint",ArmWelds.LWeld, nil, CFrame.new(0.870444059, 0.659437597, -1.52524424, 0.98510921, 0.171723664, -0.00840575993, -0.0107716024, 0.110439532, 0.993824482, 0.171591491, -0.978935182, 0.110644706), function(X) return math.sin(math.rad(X)) end, 0.3)

-- wait(0.2)

Animate()

end

end

function onUnequipped(deleteTool)

if Selected then

Selected = false

breakReload = true

--------------------[ RUNNING PLUGINS ]---------------------------------------

for \_, Plugin in PAIRS(Libraries.Plugins.OnUnEquipped) do

runAsync(function()

Plugin()

end)

end

Player.UIStats.ModingEnabled.Value = true

Player.UIStats.GamepadMode.Value = 1

Interface.HUD.Visible = false

--------------------[ MODIFYING THE PLAYER ]----------------------------------

if not deleteTool then

Interface.Cam.FieldOfView = 70

Interface.Cam.CameraType = Enum.CameraType.Custom

end

Services.UIS.MouseBehavior = Enum.MouseBehavior.Default

Services.UIS.MouseIconEnabled = true

mainGUI.Enabled = false

Player.CameraMode = Enum.CameraMode.Classic

if armTable then

OBJ.Destroy(armTable[1].Model)

OBJ.Destroy(armTable[2].Model)

elseif armModel ~= nil then

Libraries.Network.send("Server","RemoveArmModel",armModel)

end

LLegWeld:Destroy()

RLegWeld:Destroy()

changePlayerTrans(Char, 0)

Libraries.Network.send("Server","ResetShoulders\_"..Player.UserId,Shoulders,LArm,RArm)

if neckClone then

neckClone.Parent = Torso

end

Libraries.Network.send("Server","DestroyHeadWeld\_"..Player.UserId,headWeld)

Humanoid.WalkSpeed = 16

Humanoid.AutoRotate = true

Interface.Cam.CameraType = Enum.CameraType.Custom

--------------------[ RESETING THE TOOL ]-------------------------------------

mouseSensitivity = S.sensitivitySettings.Default

MB1Down = false

playerMass = 0

Aimed = false

camOffsets = {

guiScope = Libraries.PrimitiveAnim.new(V3.RAW());

Reload = Libraries.PrimitiveAnim.new(V3.RAW(),nil);

Recoil = Libraries.PrimitiveAnim.new(V3.RAW(),nil);

}

recoilAnim = Libraries.PrimitiveAnim.new(V3.RAW(),nil,V3.RAW());

--Setting the aim variables to unaimed

spreadZoom = "unAimed"

scopeMain.Visible = false

scopeSteady.Visible = false

aimAlpha = 0

aimHeadOffset = 0

jumpAnimMultiplier = 1

translationDivisor = 7

rotationMultiplier = S.momentumSettings.Amplitude.unAimed

armTiltMultiplier = 1

Interface.Scope.BackgroundTransparency = 1

if S.guiScope then

runAsync(function()

for \_, Obj in PAIRS(OBJ.GetKids(Gun)) do

if Obj:IsA("BasePart") then

Obj.LocalTransparencyModifier = 0

end

end

end)

end

onGround = true

for \_,c in PAIRS(Connections) do

c:disconnect()

end

Libraries.Network.send("Server","ShowRegularArms",LArm,RArm)

Connections = {}

if FE then

Libraries.Network.fetch("Server","unSetupGun\_"..Player.UserId,gunParts2)

else

for \_, Tab in PAIRS(gunParts) do

OBJ.Destroy(Tab.Weld)

Tab.Weld = nil

end

end

Libraries.Network.fetch("Server","destroyGunIgnore\_"..Player.UserId,gunIgnore)

if deleteTool then

Interface.Cam:ClearAllChildren()

OBJ.Destroy(Gun)

end

-- Services.CAS:UnbindAction("Walk")

wait() --This is here in case you dolphin dived and deselected the tool instantly

if S.stanceSettings.standOnDeselect and Stance ~= 0 then

crawlCamRot = 0

isCrawling = false

stanceSway = 1

spreadStance = "Stand"

changeStance("Stand",true)

end

baseSpread = S.spreadSettings[spreadZoom][spreadStance][spreadMotion]

Libraries.Network.reset()

end

end

Gun.Equipped:connect(onEquipped)

Gun.Unequipped:connect(function() onUnequipped(false) end)

--------------------------------------------------------------------------------------

--------------------[ END PROGRAM ]---------------------------------------------------

--------------------------------------------------------------------------------------

KEYBINDS.lua

local function getBindName(key,MapMode)

for \_, v in pairs(script.Parent:FindFirstChild(MapMode):GetChildren()) do

if v.Value == key.Name and v.Name ~= "scopeSteady" then

return v.Name

end

end

end

local Settings = require(script.Parent.SETTINGS)

local KeyBinds = {

Keys = {};

GamepadButtons = {};

}

local function insertKeyBind(key,MapMode)

if key.Name:find("Button") or key.Name:find("DPad") or key.Name:find("Thumbstick") then

KeyBinds.GamepadButtons[key] = getBindName(key,MapMode)

else

KeyBinds.Keys[key] = getBindName(key,MapMode)

end

end

insertKeyBind(Enum.KeyCode[script.Parent.Keybinds.Reload.Value],"Keybinds")

insertKeyBind(Enum.KeyCode[script.Parent.Keybinds.ADS.Value],"Keybinds")

insertKeyBind(Enum.KeyCode[script.Parent.Keybinds.lowerStance.Value],"Keybinds")

insertKeyBind(Enum.KeyCode[script.Parent.Keybinds.raiseStance.Value],"Keybinds")

KeyBinds.Keys[Settings.Keys.SpecialKey] = {"Sprint","scopeSteady"}

insertKeyBind(Enum.KeyCode[script.Parent.Keybinds.selectFire.Value],"Keybinds")

insertKeyBind(Enum.KeyCode[script.Parent.Keybinds.Hook.Value],"Keybinds")

insertKeyBind(Enum.KeyCode[script.Parent.GamepadBinds.Reload.Value],"GamepadBinds")

insertKeyBind(Enum.KeyCode[script.Parent.GamepadBinds.ADS.Value],"GamepadBinds")

insertKeyBind(Enum.KeyCode[script.Parent.GamepadBinds.lowerStance.Value],"GamepadBinds")

insertKeyBind(Enum.KeyCode[script.Parent.GamepadBinds.raiseStance.Value],"GamepadBinds")

KeyBinds.GamepadButtons[Settings.GamepadButtons.SpecialKey] = {"Sprint","scopeSteady"}

insertKeyBind(Enum.KeyCode[script.Parent.GamepadBinds.selectFire.Value],"GamepadBinds")

insertKeyBind(Enum.KeyCode[script.Parent.GamepadBinds.Hook.Value],"GamepadBinds")

function KeyBinds:Recalibrate()

KeyBinds.Keys = {}

KeyBinds.GamepadButtons = {};

insertKeyBind(Enum.KeyCode[script.Parent.Keybinds.Reload.Value],"Keybinds")

insertKeyBind(Enum.KeyCode[script.Parent.Keybinds.ADS.Value],"Keybinds")

insertKeyBind(Enum.KeyCode[script.Parent.Keybinds.lowerStance.Value],"Keybinds")

insertKeyBind(Enum.KeyCode[script.Parent.Keybinds.raiseStance.Value],"Keybinds")

KeyBinds.Keys[Settings.Keys.SpecialKey] = {"Sprint","scopeSteady"}

insertKeyBind(Enum.KeyCode[script.Parent.Keybinds.selectFire.Value],"Keybinds")

insertKeyBind(Enum.KeyCode[script.Parent.Keybinds.Hook.Value],"Keybinds")

insertKeyBind(Enum.KeyCode[script.Parent.GamepadBinds.Reload.Value],"GamepadBinds")

insertKeyBind(Enum.KeyCode[script.Parent.GamepadBinds.ADS.Value],"GamepadBinds")

insertKeyBind(Enum.KeyCode[script.Parent.GamepadBinds.lowerStance.Value],"GamepadBinds")

insertKeyBind(Enum.KeyCode[script.Parent.GamepadBinds.raiseStance.Value],"GamepadBinds")

KeyBinds.GamepadButtons[Settings.GamepadButtons.SpecialKey] = {"Sprint","scopeSteady"}

insertKeyBind(Enum.KeyCode[script.Parent.GamepadBinds.selectFire.Value],"GamepadBinds")

insertKeyBind(Enum.KeyCode[script.Parent.GamepadBinds.Hook.Value],"GamepadBinds")

end

return KeyBinds

PLUGINS.lua

--[[

Plugins for this gun kit are basically functions that will run at specific times, i.e. When a key is pressed, when the gun is

fired, when the gun is aimed, etc.

HOW TO USE IT:

KEYDOWN PLUGIN:

Let's say you wanted to toggle a laser whenever you press the "v" key:

You would create a table like the example below

The first element would be "Key = INSERT\_KEY\_HERE"

The second element would be "Description = INSERT\_DESCRIPTION\_HERE"

The third element would be "Plugin = INSERT\_FUNCTION\_HERE"

So whenever you press the "v" key, the plugin function will run

Pretty useful if you want to add extra code to the script without actually having to modify the script

NOTE: Only the keydown plugin requires this table. Every other plugin just needs a function

EVERY OTHER PLUGIN:

Let's say you wanted to make a shell eject whenever the gun was fired:

You would add function called "Plugin = INSERT\_FUNCTION\_HERE"

That's it; If you want other stuff to happen when the gun is fired, you would either put it all into 1 function, or

you would add more Plugins to the "Firing" table. Like so:

Firing = {

Plugin = function()

--Code

end;

Plugin = function()

--Code

end;

Plugin = function()

--Code

end;

};

That's really it, you just need to know some basic scripting to use it. If you have more questions, pm me.

NOTE: The plugins run seperate from the code in the Gun\_Main. For example, if you have a plugin that ejects a shell 1 second

after the gun is fired, the gun's firing speed won't be affected in any way.

--]]

local Gun = script.Parent

local S = require(Gun.SETTINGS)

local Plugins = {

KeyDown = {

{ --This is a plugin for a toggleable laser

Key = Enum.KeyCode.V; --This is the key you press to activate the plugin

Description = "Toggle Laser"; --This is what the description of the key will be in the controls

Plugin = function() --This is the actual plugin function

if Gun.Upgrades:WaitForChild("hasLaser",200).Value then

game.ReplicatedStorage.RemoteService.AdvanceTutorialRemote:FireServer("TutorialBountyHunter",11)

local Laser = Gun:WaitForChild("Laser") --These few lines wait for the necessary bricks/models

local Handle = Gun:WaitForChild("Handle")

local ignoreCode = Gun.clientMain:WaitForChild("ignoreCode")

local ignoreModel = game.Workspace:WaitForChild("ignoreModel\_"..ignoreCode.Value)

local PlyrName = game.Players:GetPlayerFromCharacter(Gun.Parent).Name

local playerFolder = ignoreModel:WaitForChild("gunIgnore\_"..PlyrName)

local RS = game:GetService("RunService")

local function createLaserDot() --This function creates the red laser dot

local laserDot = Instance.new("Part")

laserDot.Transparency = 1

laserDot.Name = "laserDot"

laserDot.Anchored = true

laserDot.CanCollide = false

laserDot.FormFactor = Enum.FormFactor.Custom

laserDot.Size = Vector3.new(0.25, 0.25, 1)

local laserGui = Instance.new("SurfaceGui")

laserGui.CanvasSize = Vector2.new(100, 100)

laserGui.Parent = laserDot

local laserImage = Instance.new("ImageLabel")

laserImage.BackgroundTransparency = 1

laserImage.Size = UDim2.new(1, 0, 1, 0)

laserImage.Image = "http://www.roblox.com/asset/?id=131394739"

laserImage.Parent = laserGui

local laserLight = Instance.new("SurfaceLight")

laserLight.Angle = 180

laserLight.Brightness = math.huge

laserLight.Color = Color3.new(1, 0, 0)

laserLight.Face = Enum.NormalId.Back

laserLight.Range = 5

laserLight.Shadows = true

laserLight.Parent = laserDot

return laserDot

end

local function getHitSurfaceCFrame(Pos, Obj) --This function returns the proper cframe based on the face that the position is on

local surfaceCF = {

{"Back", Obj.CFrame \* CFrame.new(0, 0, Obj.Size.z)};

{"Bottom", Obj.CFrame \* CFrame.new(0, -Obj.Size.y, 0)};

{"Front", Obj.CFrame \* CFrame.new(0, 0, -Obj.Size.z)};

{"Left", Obj.CFrame \* CFrame.new(-Obj.Size.x, 0, 0)};

{"Right", Obj.CFrame \* CFrame.new(Obj.Size.x, 0, 0)};

{"Top", Obj.CFrame \* CFrame.new(0, Obj.Size.y, 0)}

}

local closestDist = math.huge

local closestSurface = nil

for \_,v in pairs(surfaceCF) do

local surfaceDist = (Pos - v[2].p).magnitude

if surfaceDist < closestDist then

closestDist = surfaceDist

closestSurface = v

end

end

local surfaceDir = CFrame.new(Obj.CFrame.p, closestSurface[2].p)

local surfaceDist = surfaceDir.lookVector \* ((Obj.CFrame.p - closestSurface[2].p).magnitude / 2 - 0.25)

local surfaceOffset = Pos - closestSurface[2].p + surfaceDist

local surfaceCFrame = surfaceDir + surfaceDist + surfaceOffset

return surfaceCFrame

end

local laserDot = createLaserDot() --The code is cleaner when the laser creating code is in a function

Laser.Transparency = (Laser.Transparency == 1 and 0 or 1) --Toggles the laser on or off

while math.floor(Laser.Transparency) == 0 do --This loop will keep running as long as the laser is visible

if (not game.Players:GetPlayerFromCharacter(Gun.Parent)) then break end --This checks if the gun is a child of the character

local newRay = Ray.new(Laser.Position, Handle.CFrame.lookVector \* 999)

local H, P = game.Workspace:FindPartOnRay(newRay, ignoreModel)

local Distance = (P - Laser.Position).magnitude

Laser.Mesh.Offset = Vector3.new(0, Distance / 2, 0)

Laser.Mesh.Scale = Vector3.new(0.075, Distance / 0.2, 0.075)

Laser.BrickColor = BrickColor.new("Really red")

Laser.Material = Enum.Material.Neon

if H then

laserDot.CFrame = getHitSurfaceCFrame(P, H) --If the laser hits a part then position the dot on the part

laserDot.Parent = playerFolder

else

laserDot.Parent = nil --If the laser doesn't hit a part then temporarily remove the laser dor

end

RS.RenderStepped:wait()

end

laserDot:Destroy() --These lines reset the laser if the laser is transparent or the gun was deselected

Laser.Transparency = 1

Laser.Mesh.Offset = Vector3.new()

Laser.Mesh.Scale = Vector3.new(0.075, 0, 0.075)

end

end;

};

{

Key = Enum.KeyCode.U,

Description = "Unequip Weapon",

Plugin = function()

game.Players.LocalPlayer.Character.Humanoid:UnequipTools()

end

};

{ --This is a plugin for a toggleable flashlight

Key = Enum.KeyCode.Z;

Description = "Toggle Flashlight";

Plugin = function()

if Gun.Upgrades:WaitForChild("hasLight",200).Value then

local Flashlight = Gun:WaitForChild("Flashlight")

if Flashlight then

for \_, Light in pairs(Flashlight:GetChildren()) do

if Light.Name == "Light" then

Light.Enabled = (not Light.Enabled)

end

end

end

end

end

};

};

KeyUp = {

};

Firing = {

Plugin = function(Player)

end;

};

Aimed = {

};

UnAimed = {

Plugin = function()

--Put code here

end;

};

OnEquipped = {

Plugin = function()

--Put code here

end

};

OnUnEquipped = {

Plugin = function()

--Put code here

end

};

}

return Plugins

SCOPES.lua

local AvailableScopes = {

"CompM4";

"ACOG";

"EoTech";

"Reflex";

"Kobra";

}

return AvailableScopes

serverMain.lua

local Plyr = script:WaitForChild("Plyr")

Plyr.Value = game.Players:GetPlayerFromCharacter(script.Parent.Parent) or script.Parent.Parent.Parent

local Network = require(game.ReplicatedStorage.RemoteService.Plugin\_Gun)(Plyr.Value,script.Parent.Name,true,script.Parent)

Network.listen("Server","Send","SetCam\_"..Plyr.Value.UserId,function(player, Cam)

script:WaitForChild("Cam").Value = Cam

end)

local RS = game:GetService("RunService")

local BC = BrickColor.new

local FFC = game.FindFirstChild

local PL = game.Players

local Cam = script.Cam.Value

local TOS = CFrame.new().toObjectSpace

local Gun = script.Parent

local Handle = Gun:WaitForChild("Handle")

local RAY = Ray.new

local RANDOM = math.random

local FE = workspace.FilteringEnabled

local V3 = Vector3.new

local CF, CFANG, CFTOS = CFrame.new, CFrame.Angles, CFrame.new().toObjectSpace

local PAIRS = pairs

local S = require(Gun.SETTINGS)

local MIN,MAX = math.min,math.max

local RAD,CEIL,FLOOR = math.rad,math.ceil,math.floor

local NSEQ,CSEQ,OBJ = NumberSequence.new,ColorSequence.new,Instance.new

local UD2 = UDim2.new

local NRANGE = NumberRange.new

local RAY = Ray.new

local JointC0,JointC1

local WS = workspace

local AttributeEffect =require(WS.Settings.AttributeEffect)

local PerLevel = require(WS.Settings.PerLevel)

local gunIgnore

local animWeldC1,animWeldC0

local animWeld

local AvailableScopes = require(script.Parent.SCOPES)

local Optics = require(script.Optics)

function ArcTan(x, y)

local r = math.atan(y / x)

if x < 0 then

r = r + math.pi

end

return r

end

local gbl = require(game.ReplicatedStorage.Global)

local RPGM = require(game.ReplicatedStorage.RPGMathProvider)

local numLerp = function(A, B, Alpha)

return A + (B - A) \* Alpha

end

local runAsync = function(threadFunc)

coroutine.resume(coroutine.create(threadFunc))

end

local raycast = WS.FindPartOnRayWithIgnoreList

local function isEnemy(Human)

local Plyr2 = game.Players:GetPlayerFromCharacter(Human.Parent)

if (not Plyr2) then if S.CanDamageNPCs then

if Human.Parent:FindFirstChild("BOT") then

return (require(Human.Parent.BOT).Allegiance ~= Plyr.Value.Allegiance.Value)

end

end

end

return S.AllowFriendlyFire or (Plyr2 ~= nil and (Plyr2.Allegiance.Value ~= Plyr.Value.Allegiance.Value or Plyr2.Neutral))end

local function Weld(p0,p1,c0,c1,par)

local w = OBJ("Weld")

w.Part0 = p0

w.Part1 = p1

w.C0 = c0 or CF()

w.C1 = c1 or CF()

w.Parent = p0 or par

return w

end

local function Motor(p0,p1,c0,c1,des,vel,par)

local w = OBJ("Motor6D")

w.Part0 = p0

w.Part1 = p1

w.C0 = c0 or CF()

w.C1 = c1 or CF()

w.MaxVelocity = tonumber(vel) or .05

w.DesiredAngle = tonumber(des) or 0

w.Parent = p0 or par

return w

end

Network.listen("Server","Send","SetPlayer\_"..Plyr.Value.UserId,function(player, Cam)

script.Plyr.Value = player

end)

local function DisplayDamage(damage,humanoid)

local part2 = OBJ("TextLabel")

part2.Font = "Highway"

part2.FontSize = "Size24"

part2.TextStrokeTransparency = 0

part2.Size = UD2(1,0,1,0)

part2.Position = UD2(0,0,0,0)

part2.BackgroundTransparency = 1

part2.Parent = humanoid.Parent.Head.DamageGUI

runAsync(function()

part2:TweenPosition(UDim2.new(0,0,0,-98))

wait(1)

for i = 1, 20 do

part2.TextTransparency = part2.TextTransparency + 0.05

wait(0.025)

end

end)

if (damage == 0) then

part2.TextColor3 = Color3.new(0,0.5,1)

part2.Text = "Miss!"

else

part2.TextColor3 = Color3.new(1,1,1)

part2.Text = damage

end

end

local function tagHumanoid(humanoid, player, damage, previousHealth)

local creator\_tag = OBJ("ObjectValue")

creator\_tag.Value = player

creator\_tag.Name = "creator"

creator\_tag.Parent = humanoid

local weapon\_tag = OBJ("StringValue")

weapon\_tag.Name = "weapon"

weapon\_tag.Value = script.Parent.Name

weapon\_tag.Parent = creator\_tag

local damage\_tag = OBJ("IntValue")

if humanoid.Health < 1 then

damage\_tag.Value = CEIL(previousHealth)

else

damage\_tag.Value = CEIL(damage)

end

damage\_tag.Name = "damage"

damage\_tag.Parent = creator\_tag

end

local inList = function(Element, List)

for \_, v in PAIRS(List) do

if v == Element then

return true

end

end

return false

end

local INSERT = function(tableObj,item)

tableObj[#tableObj+1] = item

end

local REMOVE = function(tableObj,i)

tableObj[i] = nil

end

local getObject = function(Model, Class, Name)

for \_, v in PAIRS(Model:GetChildren()) do

if v:IsA(Class) and v.Name == Name then

return v

end

end

return nil

end

----------------------------------------------------------------------

--------------------[ IGNORE MODEL HANDLING ]-------------------------

----------------------------------------------------------------------

wait(RANDOM(0, 20) \* 0.025) --This is to prevent more than one ignoreModel from being created

if \_G.ignoreCode then --If the ignoreCode already exists, then the script creates the ignoreModel

--[[

The purpose of this is so that every gun in a game that uses this gun kit will share one ignoreModel. That way,

bullet trails, bullet holes, and other fake arms will be ignored by the gun which makes the bullets more likely to

hit a character part

--]]

if (not FFC(WS,"ignoreModel\_"..\_G.ignoreCode)) then

local ignoreModel = OBJ("Model")

ignoreModel.Name = "ignoreModel\_"..\_G.ignoreCode

ignoreModel.Parent = WS

local grenadeFolder = OBJ("Model")

grenadeFolder.Name = "grenadeFolder"

grenadeFolder.Parent = ignoreModel

runAsync(function()

while true do

ignoreModel.Parent = WS

grenadeFolder.Parent = ignoreModel

wait(1 \* 0.05)

end

end)

end

script.Parent:WaitForChild("clientMain"):WaitForChild("ignoreCode").Value = \_G.ignoreCode

else

--[[

If there isn't already an ignoreCode, then this creates one. The purpose of it being random is so that if there is

an ignoreModel for something else in the game, the script won't end up placing the ignored objects in that ignoreModel

--]]

\_G.ignoreCode = RANDOM(1, 1e4)

if (not FFC(WS,"ignoreModel\_"..\_G.ignoreCode)) then

local ignoreModel = OBJ("Model")

ignoreModel.Name = "ignoreModel\_"..\_G.ignoreCode

ignoreModel.Parent = WS

local grenadeFolder = OBJ("Model")

grenadeFolder.Name = "grenadeFolder"

grenadeFolder.Parent = ignoreModel

runAsync(function()

while true do

ignoreModel.Parent = WS

grenadeFolder.Parent = ignoreModel

wait(1 \* 0.05)

end

end)

end

script.Parent:WaitForChild("clientMain"):WaitForChild("ignoreCode").Value = \_G.ignoreCode

end

runAsync(function()

--[[

This function deletes any Player Folders that were left in the ignoreModel because the player left the game without

deselecting the Gun first

--]]

repeat wait() until \_G.ignoreCode

local ignoreModel = WS:WaitForChild("ignoreModel\_"..\_G.ignoreCode)

while true do

for \_, gunIgnore in PAIRS(ignoreModel:GetChildren()) do

if gunIgnore.Name ~= "grenadeFolder" then

if (not FFC(PL,gunIgnore.Name:sub(11))) then

gunIgnore:Destroy()

end

end

end

RS.Heartbeat:wait()

end

end)

----------------------------------------------------------------------

--------------------[ GET WELD CFRAMES ]------------------------------

----------------------------------------------------------------------

for \_, v in PAIRS(Gun:GetChildren()) do

if v:IsA("BasePart") and v ~= Handle then

if v:FindFirstChild("mainWeld") then v.mainWeld:Destroy() end

if (not v:FindFirstChild("weldCF")) then

local weldCF = OBJ("CFrameValue")

weldCF.Name = "weldCF"

weldCF.Value = Handle.CFrame:toObjectSpace(v.CFrame)

weldCF.Parent = v

end

if string.sub(v.Name, 1, 3) == "Mag" then

if (not v:FindFirstChild("magTrans")) then

local magTrans = OBJ("NumberValue")

magTrans.Name = "magTrans"

magTrans.Value = v.Transparency

magTrans.Parent = v

end

end

v.Anchored = true

v.CanCollide = false

end

end

Handle.Anchored = false

Handle.CanCollide = true

Network.listen("Server","Fetch","gunSetup\_"..tostring(Plyr.Value.UserId),function(player,Vars)

gunIgnore = OBJ("Model")

gunIgnore.Name = "gunIgnore\_"..player.Name

gunIgnore.Parent = Vars.ignoreModel

local playerFolder = OBJ("Model")

playerFolder.Name = "playerFolder"

playerFolder.Parent = gunIgnore

local BulletStorage = OBJ("Folder")

BulletStorage.Name = "bulletStorage\_"..player.Name

BulletStorage.Parent = game.ReplicatedStorage

Gun.BulletStorage.Value = BulletStorage

Vars.Humanoid.AutoRotate = false

Vars.Shoulders.Right.Part1 = nil

Vars.Shoulders.Left.Part1 = nil

local headBase = Instance.new("Part")

headBase.Transparency = 1

headBase.Name = "headBase"

headBase.CanCollide = false

headBase.FormFactor = Enum.FormFactor.Custom

headBase.Size = V3(0.2, 0.2, 0.2)

headBase.BottomSurface = Enum.SurfaceType.Smooth

headBase.TopSurface = Enum.SurfaceType.Smooth

headBase.Parent = playerFolder

local headWeld = OBJ("Motor6D")

headWeld.Part0 = Vars.Torso

headWeld.Part1 = headBase

headWeld.C0 = CF(0, 1.5, 0)

headWeld.Parent = Vars.Torso

local headWeld2 = OBJ("Weld")

headWeld2.Part0 = headBase

headWeld2.Part1 = Vars.Head

headWeld2.Parent = headBase

local animBase = OBJ("Part")

animBase.Transparency = 1

animBase.Name = "animBase"

animBase.CanCollide = false

animBase.FormFactor = Enum.FormFactor.Custom

animBase.Size = V3(0.2, 0.2, 0.2)

animBase.BottomSurface = Enum.SurfaceType.Smooth

animBase.TopSurface = Enum.SurfaceType.Smooth

animBase.Parent = playerFolder

animWeld = Motor(animBase,headBase,nil,nil,nil,nil,nil)

animWeld.Name = "animWeld"

local armBase = OBJ("Part")

armBase.Transparency = 1

armBase.Name = "ArmBase"

armBase.CanCollide = false

armBase.FormFactor = Enum.FormFactor.Custom

armBase.Size = V3(0.2, 0.2, 0.2)

armBase.BottomSurface = Enum.SurfaceType.Smooth

armBase.TopSurface = Enum.SurfaceType.Smooth

armBase.Parent = playerFolder

local ABWeld = Motor(armBase,animBase,nil,nil,nil,nil,nil)

local LArmBase = OBJ("Part")

LArmBase.Transparency = 1

LArmBase.Name = "LArmBase"

LArmBase.CanCollide = false

LArmBase.FormFactor = Enum.FormFactor.Custom

LArmBase.Size = V3(0.2, 0.2, 0.2)

LArmBase.BottomSurface = Enum.SurfaceType.Smooth

LArmBase.TopSurface = Enum.SurfaceType.Smooth

LArmBase.Parent = playerFolder

local RArmBase = OBJ("Part")

RArmBase.Transparency = 1

RArmBase.Name = "RArmBase"

RArmBase.CanCollide = false

RArmBase.FormFactor = Enum.FormFactor.Custom

RArmBase.Size = V3(0.2, 0.2, 0.2)

RArmBase.BottomSurface = Enum.SurfaceType.Smooth

RArmBase.TopSurface = Enum.SurfaceType.Smooth

RArmBase.Parent = playerFolder

local LWeld = Motor(armBase,LArmBase,Vars.armC0[1],Vars.leftArmC1,nil,nil,nil)

LWeld.Name = "LWeld"

local RWeld = Motor(armBase,RArmBase,Vars.armC0[2],Vars.rightArmC1,nil,nil,nil)

RWeld.Name = "RWeld"

local LWeld2 = Motor(LArmBase,Vars.LArm,nil,nil,nil,nil,nil)

LWeld2.Name = "LWeld"

local RWeld2 = Motor(RArmBase,Vars.RArm,nil,nil,nil,nil,nil)

RWeld2.Name = "RWeld"

local LLegWeld = Motor(Vars.Torso,nil,CF(-0.5, -2, 0),nil,nil,nil,nil)

LLegWeld.Name = "LLegWeld"

local RLegWeld = Motor(Vars.Torso,nil,CF(0.5, -2, 0),nil,nil,nil,nil)

RLegWeld.Name = "RLegWeld"

for \_, Tab in PAIRS(Vars.gunParts) do

Tab.Obj.Anchored = false

local Weld = OBJ("Motor6D")

Weld.Name = "mainWeld"

Weld.Part0 = Vars.Handle

Weld.Part1 = Tab.Obj

Weld.C0 = Tab.Obj.weldCF.Value

Weld.Parent = Vars.Handle

Tab.Weld = Weld

end

return gunIgnore, playerFolder, headWeld, headWeld2, animWeld, ABWeld, LWeld, RWeld, LWeld2, RWeld2, LLegWeld, RLegWeld, Vars.gunParts

end)

--Network.listen("Server","Fetch","makeBullet\_"..Plyr.Value.UserId,function(player,Mode,bulletDirection,Origin,ignoreList,gunIgnore)

-- local Origin = Origin

-- local Char = Gun.Parent

-- local ignoreModel = ignoreList[2]

-- local bulletCF = CF(Origin, Origin + bulletDirection)

-- local Bullet

-- if Mode ~= "STUN" then

--

-- Bullet = OBJ("Part")

--

-- Bullet.BrickColor = S.bulletSettings.Color

-- Bullet.Material = Enum.Material.Neon

-- Bullet.Name = "Bullet"

-- Bullet.CanCollide = false

-- Bullet.Size = S.bulletSettings.Size

-- Bullet.BottomSurface = "Smooth"

-- Bullet.TopSurface = "Smooth"

-- if S.bulletSettings.trueBolt then

-- local Mesh2 = game.ReplicatedStorage.BlasterEffects.boltMesh:Clone()

-- Mesh2.Parent = Bullet

--

-- end

-- if not S.bulletSettings.trueBolt then

-- if MIN(S.bulletSettings.Size.X, S.bulletSettings.Size.Y, S.bulletSettings.Size.Z) < 0.2 then

-- local Mesh = OBJ("BlockMesh")

--

-- Mesh.Scale = S.bulletSettings.Size / V3(

-- MAX(S.bulletSettings.Size.X, 0.2),

-- MAX(S.bulletSettings.Size.Y, 0.2),

-- MAX(S.bulletSettings.Size.Z, 0.2)

-- )

-- Mesh.Parent = Bullet

--

-- end

--

-- end

-- else

-- Bullet = game.ReplicatedStorage.StunBullet:Clone()

-- end

--

-- local BF = OBJ("BodyForce")

-- BF.force = V3(0, Bullet:GetMass() \* (196.2 - S.bulletSettings.Acceleration), 0)

-- local BV = OBJ("BodyVelocity")

-- BV.MaxForce = V3(1000000,10000000,10000000)

-- BV.Velocity = bulletDirection \* S.bulletSettings.Velocity

-- BV.Parent = Bullet

-- BF.Parent = Bullet

-- Bullet.Parent = gunIgnore

-- Bullet.CFrame = bulletCF + bulletDirection \* S.bulletSettings.Size.Z / 2

-- return Bullet

--end)

Network.listen("Server","Fetch","unSetupGun\_"..Plyr.Value.UserId,function(player,gunParts)

for \_, Tab in PAIRS(gunParts) do

Tab.Weld:Destroy()

Tab.Weld = nil

end

return true

end)

Network.listen("Server","Fetch","lerpedRecoilRotation\_"..Plyr.Value.UserId,function(player,recoilAnim,prevRot,newRot,AlphaX)

recoilAnim.Rot = prevRot:lerp(newRot,AlphaX)

return true

end)

Network.listen("Server","Fetch","lerpedRecoilPosition\_"..Plyr.Value.UserId, function(player,recoilAnim,prevPos,newPos,AlphaX)

recoilAnim.Pos = prevPos:lerp(newPos,AlphaX)

return true

end)

Network.listen("Server","Fetch","TweenIndicator\_"..Plyr.Value.UserId,function(player,Joint, newCode)

local tweenIndicator = nil

if (not Joint:findFirstChild("tweenCode")) then --If the joint isn't being tweened, then

tweenIndicator = OBJ("IntValue")

tweenIndicator.Name = "tweenCode"

tweenIndicator.Value = newCode

tweenIndicator.Parent = Joint

else

tweenIndicator = Joint.tweenCode

tweenIndicator.Value = newCode --If the joint is already being tweened, this will change the code, and the tween loop will stop

end

return tweenIndicator

end)

Network.listen("Server","Fetch","Arms\_"..tostring(Plyr.Value.UserId),function(player,Vars)

local armModel = OBJ("Model")

armModel.Parent = (FE and Vars.gunIgnore.playerFolder or script.Cam.Value)

local fakeLArm = Vars.LArm:Clone()

fakeLArm.Parent = armModel

--fakeLArm.Transparency = Vars.S.fakeArmSettings.Transparency

fakeLArm.CanCollide = false

fakeLArm.Size = Vars.S.fakeArmSettings.armSize

fakeLArm:BreakJoints()

-- Vars.LArm.Transparency = 1

local fakeLWeld = OBJ("Motor6D")

fakeLWeld.Name = "FLWeld"

fakeLWeld.Part0 = fakeLArm

fakeLWeld.Part1 = Vars.LArm

fakeLWeld.Parent = fakeLArm

local fakeRArm = Vars.RArm:Clone()

fakeRArm.Parent = armModel

fakeRArm.Transparency = Vars.S.fakeArmSettings.Transparency

fakeRArm.CanCollide = false

fakeRArm.Size = Vars.S.fakeArmSettings.armSize

fakeRArm:BreakJoints()

--Vars.RArm.Transparency = 1

if Vars.S.fakeArmSettings.showArmor then

if Gun.Parent:FindFirstChild("Arm1") then

local fakeArm1 = Gun.Parent.Arm1:Clone()

fakeArm1.Parent = Vars.gunIgnore

local C = fakeArm1:GetChildren()

for i=1, #C do

if C[i].className == "Part" or C[i]:IsA('UnionOperation') then

local W = OBJ("Weld")

W.Part0 = fakeArm1.Middle

W.Part1 = C[i]

local CJ = CF(fakeArm1.Middle.Position)

local C0 = fakeArm1.Middle.CFrame:inverse()\*CJ

local C1 = C[i].CFrame:inverse()\*CJ

W.C0 = C0

W.C1 = C1

W.Parent = fakeArm1.Middle

end

local Y = OBJ("Weld")

Y.Part0 = fakeLArm

Y.Part1 = fakeArm1.Middle

Y.C0 = CF(0, 0, 0)

Y.Parent = Y.Part0

end

local h = fakeArm1:GetChildren()

for i = 1, # h do

if h[i]:IsA("BasePart") then

h[i].Anchored = false

h[i].CanCollide = false

end

end

fakeLArm.Transparency = 1

end

if Gun.Parent:FindFirstChild("Arm2") then

local fakeArm2 = Gun.Parent.Arm2:Clone()

fakeArm2.Parent = Vars.gunIgnore

local C = fakeArm2:GetChildren()

for i=1, #C do

if C[i].className == "Part" or C[i]:IsA('UnionOperation') then

local W = OBJ("Weld")

W.Part0 = fakeArm2.Middle

W.Part1 = C[i]

local CJ = CF(fakeArm2.Middle.Position)

local C0 = fakeArm2.Middle.CFrame:inverse()\*CJ

local C1 = C[i].CFrame:inverse()\*CJ

W.C0 = C0

W.C1 = C1

W.Parent = fakeArm2.Middle

end

local Y = OBJ("Weld")

Y.Part0 = fakeRArm

Y.Part1 = fakeArm2.Middle

Y.C0 = CF(0, 0, 0)

Y.Parent = Y.Part0

end

fakeRArm.Transparency = 1

local h = fakeArm2:GetChildren()

for i = 1, # h do

if h[i]:IsA("BasePart") then

h[i].Anchored = false

h[i].CanCollide = false

end

end

end

end

local fakeRWeld = OBJ("Motor6D")

fakeRWeld.Name = "FRWeld"

fakeRWeld.Part0 = fakeRArm

fakeRWeld.Part1 = Vars.RArm

fakeRWeld.Parent = fakeRArm

local gIgnoreHum = OBJ("Humanoid")

gIgnoreHum.Parent = armModel

if Vars.S.fakeArmSettings.characterMeshes then

for \_,Obj in PAIRS(Gun.Parent:GetChildren()) do

if Obj:IsA("CharacterMesh") then

Obj:Clone().Parent = armModel

end

end

end

for \_,Obj in PAIRS(Gun.Parent:GetChildren()) do

if Obj:IsA("Shirt") then

Obj:Clone().Parent = armModel

end

end

return armModel

end)

Network.listen("Server","Fetch","Blood\_"..Plyr.Value.UserId,function(player, H, P, D, gunIgnore, S)

local bloodCF = CF(P, P + D) \* CFANG(RAD(-90), 0, 0)

local Blood = OBJ("Part")

Blood.Transparency = 1

Blood.Anchored = true

Blood.CanCollide = false

Blood.FormFactor = "Custom"

Blood.Size = V3(0.2, 1, 0.2)

Blood.TopSurface = 0

Blood.BottomSurface = 0

local Particles = OBJ("ParticleEmitter")

Particles.Color = CSEQ(S.bloodSettings.Color)

Particles.LightEmission = 0

Particles.Size = NSEQ(S.bloodSettings.Size)

Particles.Texture = S.bloodSettings.Texture

Particles.Transparency = NSEQ(

{

NumberSequenceKeypoint.new(0, S.bloodSettings.startTransparency);

NumberSequenceKeypoint.new(1, 1);

}

)

Particles.EmissionDirection = Enum.NormalId.Top

Particles.Lifetime = NRANGE(S.bloodSettings.Lifetime - 0.05, S.bloodSettings.Lifetime + 0.05)

Particles.Rate = S.bloodSettings.Rate

Particles.Rotation = NRANGE(0, 90)

Particles.Speed = NRANGE(S.bloodSettings.Speed)

Particles.VelocitySpread = S.bloodSettings.Spread

Particles.Parent = Blood

Blood.Parent = gunIgnore

Blood.CFrame = bloodCF

if (not H.Anchored) then

local Weld = OBJ("Weld", Blood)

Weld.Part0 = H

Weld.Part1 = Blood

Weld.C0 = H.CFrame:toObjectSpace(bloodCF)

Blood.Anchored = false

end

delay(0.15, function()

Particles.Enabled = false

wait(S.bloodSettings.Lifetime + 0.05)

Blood:Destroy()

end)

return true

end)

--Network.listen("Server","Fetch","Trail\_"..Plyr.Value.UserId,function(player,Origin, P, gunIgnore, S)

-- local Trail = OBJ("Part")

-- Trail.BrickColor = S.trailSettings.Color

-- Trail.Transparency = S.trailSettings.Transparency

-- Trail.Anchored = true

-- Trail.CanCollide = false

-- Trail.Size = V3(1, 1, 1)

-- local Mesh = OBJ("CylinderMesh")

-- Mesh.Offset = V3(0, -(P - Origin).magnitude / 2, 0)

-- Mesh.Scale = V3(S.trailSettings.Thickness, (P - Origin).magnitude, S.trailSettings.Thickness)

-- Mesh.Parent = Trail

-- Trail.Parent = gunIgnore

-- Trail.CFrame = CF(Origin, P) \* CFANG(RAD(90), 0, 0)

-- delay(S.trailSettings.visibleTime, function()

-- if S.trailSettings.disappearTime > 0 then

-- local t0 = tick()

-- while true do

-- local Alpha = math.min((tick() - t0) / S.trailSettings.disappearTime, 1)

-- Trail.Transparency = numLerp(S.trailSettings.Transparency, 1, Alpha)

-- if Alpha == 1 then break end

-- RS.Heartbeat:wait()

-- end

-- Trail:Destroy()

-- else

-- Trail:Destroy()

-- end

-- end)

-- result = true

--

--end)

local function Damage(player,Humanoid,H, P, N, D, Dist, customIgnore, mode )

local hVal = S.damageSettings.Multipliers.Head

local cVal = S.damageSettings.Multipliers.Chest

local lVal = S.damageSettings.Multipliers.Limbs

if Humanoid.Health ~= 0 then

local hitHumanoid = nil

if H.Parent then

if H.Parent:IsA("Hat") then

INSERT(customIgnore, H)

local newRay = RAY(P - D \* 0.1, D \* (S.bulletSettings.Range - Dist + 0.1))

local newH, newP, newN = raycast(WS,newRay, customIgnore)

if newH then

hitHumanoid = Damage(player, Humanoid, newH, newP, newN, D, Dist + (newP - P).magnitude, customIgnore)

end

else

hitHumanoid = gbl:FindFirstClass(H.Parent, "Humanoid")

if hitHumanoid and hitHumanoid.Health > 0 and isEnemy(hitHumanoid) then

if mode ~= "STUN" then

local chosenDamage = 0

if H.Name == "Head" then

chosenDamage = RPGM.RangedCombatMath.HeadShotDamage(script.Parent.Damage.Value,script.Parent.HeadshotDamageMagnitude.Value)

elseif H.Name == "Torso" then

chosenDamage = (RPGM.RangedCombatMath.RegularDamage(script.Parent.Damage.Value,Plyr.Value,AttributeEffect,PerLevel))

else

chosenDamage = (RPGM.RangedCombatMath.RegularDamage(script.Parent.Damage.Value,Plyr.Value,AttributeEffect,PerLevel))

end

local sniperRand

if H.Name == "Head" and S.sniperDamage then

sniperRand = RANDOM(1,2)

chosenDamage = (sniperRand == 2 and hitHumanoid.Health or chosenDamage)

end

local damageHum = 0

if hitHumanoid.Parent:FindFirstChild("SwordScript",true) then

if hitHumanoid.Parent:FindFirstChild("SwordScript",true).Parent:FindFirstChild("Deflecting") then

if hitHumanoid.Parent:FindFirstChild("SwordScript",true).Parent:FindFirstChild("Deflecting").Value then

damageHum = RANDOM(1,2)

end

end

else

damageHum = 1

end

if damageHum ~= 0 and damageHum == 1 then

tagHumanoid(hitHumanoid,Plyr.Value,chosenDamage,hitHumanoid.Health)

hitHumanoid:TakeDamage(chosenDamage)

elseif damageHum ~= 0 and damageHum == 2 then

Humanoid:TakeDamage(chosenDamage\*2)

script.DeflectionSound:Play()

elseif damageHum == 0 then

tagHumanoid(hitHumanoid,Plyr.Value,chosenDamage,hitHumanoid.Health)

hitHumanoid:TakeDamage(chosenDamage)

end

Network.send("Client",Plyr.Value,"MarkHit\_"..Plyr.Value.UserId)

if hitHumanoid.Parent.Head:FindFirstChild("DamageGUI") then

DisplayDamage(chosenDamage,hitHumanoid)

end

if not game.Players:GetPlayerFromCharacter(hitHumanoid.Parent)and hitHumanoid.Name == "Human" then

end

else

hitHumanoid.WalkSpeed = 0

hitHumanoid.JumpPower = 0

hitHumanoid.AutoRotate = false

wait(5)

hitHumanoid.WalkSpeed = 16

hitHumanoid.JumpPower = 50

hitHumanoid.AutoRotate = true

end

end

end

return hitHumanoid

end

end

end

Network.setupDamageListener(Damage)

Network.listen("Server","Send","bulletImpact\_"..Plyr.Value.UserId,function(player, H, P, N, D, humanoidFound, gunIgnore)

local surfaceCF = CF(P, P + N)

----------------------------------------------------------------------------------

--Creating the bullet hole--------------------------------------------------------

----------------------------------------------------------------------------------

if S.bulletHoles and ((not humanoidFound)) then

local Hole = OBJ("Part")

Hole.Transparency = 1

Hole.Anchored = true

Hole.CanCollide = false

Hole.Size = V3(1, 1, 0.2)

Hole.TopSurface = 0

Hole.BottomSurface = 0

local Mesh = OBJ("BlockMesh")

Mesh.Offset = V3(0, 0, -0.05)

Mesh.Scale = V3(S.holeSettings.Size, S.holeSettings.Size, 0)

Mesh.Parent = Hole

local Decal = OBJ("Decal")

Decal.Face = Enum.NormalId.Front

Decal.Texture = S.holeSettings.Texture

Decal.Parent = Hole

Hole.Parent = gunIgnore

Hole.CFrame = surfaceCF

if (not H.Anchored) then

local Weld = OBJ("Weld")

Weld.Part0 = H

Weld.Part1 = Hole

Weld.Parent = Hole

Weld.C0 = H.CFrame:toObjectSpace(surfaceCF)

Hole.Anchored = false

end

delay(S.holeSettings.visibleTime, function()

if S.holeSettings.disappearTime > 0 then

local t0 = tick()

while true do

local Alpha = math.min((tick() - t0) / S.holeSettings.disappearTime, 1)

Decal.Transparency = numLerp(0, 1, Alpha)

if Alpha == 1 then break end

RS.Heartbeat:wait()

end

Hole:Destroy()

else

Hole:Destroy()

end

end)

end

----------------------------------------------------------------------------------

--Creating the spark effect-------------------------------------------------------

----------------------------------------------------------------------------------

if S.bulletSparks and (not humanoidFound) and inList(H.Material, S.sparkSettings.Materials) then

local Sparks = OBJ("Part")

Sparks.Transparency = 1

Sparks.Anchored = true

Sparks.CanCollide = false

Sparks.FormFactor = "Custom"

Sparks.Size = V3(1, 1, 1)

Sparks.TopSurface = 0

Sparks.BottomSurface = 0

local Particles = nil

if S.customSparks then

Particles = getObject(game.ServerStorage, "ParticleEmitter", "bulletSpark"):Clone()

else

Particles = OBJ("ParticleEmitter")

Particles.Color = CSEQ(S.sparkSettings.Color.Start, S.sparkSettings.Color.End)

Particles.LightEmission = 1

Particles.Size = NSEQ(

{

NumberSequenceKeypoint.new(0, S.sparkSettings.Size, 0.25);

NumberSequenceKeypoint.new(1, 0);

}

)

Particles.Texture = S.sparkSettings.Texture

Particles.Transparency = NSEQ(0)

Particles.Acceleration = V3(0, -196.2, 0)

Particles.EmissionDirection = Enum.NormalId.Front

Particles.Lifetime = NRANGE(S.sparkSettings.Lifetime - 0.05, S.sparkSettings.Lifetime + 0.05)

Particles.Rate = S.sparkSettings.Rate

Particles.Rotation = NRANGE(0, 360)

Particles.Speed = NRANGE(S.sparkSettings.Speed - 5, S.sparkSettings.Speed + 5)

Particles.VelocitySpread = S.sparkSettings.Spread

end

Particles.Enabled = false

Particles.Parent = Sparks

Sparks.Parent = gunIgnore

Sparks.CFrame = surfaceCF

if (not H.Anchored) then

local Weld = OBJ("Weld")

Weld.Part0 = H

Weld.Part1 = Sparks

Weld.Parent = Sparks

Weld.C0 = H.CFrame:toObjectSpace(surfaceCF)

Sparks.Anchored = false

end

Particles:Emit(8)

wait(Particles.Lifetime.Max)

Sparks:Destroy()

end

----------------------------------------------------------------------------------

--Creating the smoke effect-------------------------------------------------------

----------------------------------------------------------------------------------

if S.bulletSmoke and (not humanoidFound) then

local Smoke = OBJ("Part")

Smoke.Transparency = 1

Smoke.Anchored = true

Smoke.CanCollide = false

Smoke.FormFactor = "Custom"

Smoke.Size = V3(1, 1, 1)

Smoke.TopSurface = 0

Smoke.BottomSurface = 0

local Particles = OBJ("ParticleEmitter")

Particles.Color = CSEQ(S.smokeSettings.objColor and H.Color or S.smokeSettings.Color)

Particles.LightEmission = 0

Particles.Size = NSEQ(

{

NumberSequenceKeypoint.new(0, S.smokeSettings.Size.Start);

NumberSequenceKeypoint.new(1, S.smokeSettings.Size.End);

}

)

Particles.Texture = S.smokeSettings.Texture

Particles.Transparency = NSEQ(

{

NumberSequenceKeypoint.new(0, S.smokeSettings.startTransparency);

NumberSequenceKeypoint.new(0.5, 0.75 \* S.smokeSettings.startTransparency + 0.25);

NumberSequenceKeypoint.new(1, 1);

}

)

Particles.Acceleration = V3(0, -196.2, 0)

Particles.EmissionDirection = Enum.NormalId.Front

Particles.Lifetime = NRANGE(S.smokeSettings.Lifetime - 0.05, S.smokeSettings.Lifetime + 0.05)

Particles.Rate = S.smokeSettings.Rate

Particles.Rotation = NRANGE(0, 360)

Particles.RotSpeed = NRANGE(10)

Particles.Speed = NRANGE(S.smokeSettings.Speed - 5, S.smokeSettings.Speed + 5)

Particles.VelocitySpread = S.smokeSettings.Spread

Particles.Parent = Smoke

Smoke.Parent = gunIgnore

Smoke.CFrame = surfaceCF

if (not H.Anchored) then

local Weld = OBJ("Weld")

Weld.Part0 = H

Weld.Part1 = Smoke

Weld.Parent = Smoke

Weld.C0 = CFTOS(H.CFrame,surfaceCF)

Smoke.Anchored = false

end

Particles:Emit(7)

wait(S.smokeSettings.Lifetime + 0.05)

Smoke:Destroy()

end

end)

Network.listen("Server","Send","SetJointC0\_"..Plyr.Value.UserId,function(player,Joint, C0)

Joint.C0 = C0

end)

Network.listen("Server","Send","deleteTweenIndicator\_"..Plyr.Value.UserId,function(player,tweenIndicator,newCode)

if tweenIndicator then

if tweenIndicator.Value == newCode then --If this tween functions was the last one called on a joint then it will remove the code

tweenIndicator:Destroy()

end

end

end)

Network.listen("Server","Send","SetJointC1\_"..Plyr.Value.UserId,function(player,Joint,C1)

Joint.C1 = C1

end)

Network.listen("Server","Send","changeGripC1\_"..Plyr.Value.UserId,function(player,Grip,GripC1)

Grip.C1 = GripC1

end)

Network.listen("Server","Send","changeAnimC1\_"..Plyr.Value.UserId,function(player,animWeld,animC0,animC1)

animWeld.C0 = animC0

animWeld.C1 = animC1

end)

Network.listen("Server","Send","setTween\_"..Plyr.Value.UserId,function(player,tweenIndicator,newCode)

tweenIndicator.Value = newCode

end)

Network.listen("Server","Send","destroyGunIgnore\_"..Plyr.Value.UserId,function(player,gunIgnore)

gunIgnore:Destroy()

end)

Network.listen("Server","Send","Shockwave\_"..Plyr.Value.UserId,function(player,Center, Radius, gunIgnore, S)

local Shockwave = OBJ("Part")

Shockwave.BrickColor = S.shockwaveSettings.Color

Shockwave.Material = Enum.Material.SmoothPlastic

Shockwave.Name = "Shockwave"

Shockwave.Anchored = true

Shockwave.CanCollide = false

Shockwave.FormFactor = Enum.FormFactor.Symmetric

Shockwave.Size = V3(1, 1, 1)

Shockwave.BottomSurface = Enum.SurfaceType.Smooth

Shockwave.TopSurface = Enum.SurfaceType.Smooth

local Mesh = OBJ("SpecialMesh")

Mesh.MeshType = Enum.MeshType.Sphere

Mesh.Scale = V3()

Mesh.Parent = Shockwave

Shockwave.Parent = gunIgnore

Shockwave.CFrame = CF(Center)

runAsync(function()

local t0 = tick()

while true do

local Alpha = math.min((tick() - t0) / S.shockwaveSettings.Duration, 1)

local Scale = 2 \* Radius \* Alpha

Mesh.Scale = V3(Scale, Scale, Scale)

Shockwave.Transparency = Alpha

if Alpha == 1 then break end

RS.Heartbeat:wait()

end

Shockwave:Destroy()

end)

end)

Network.listen("Server","Send","ShowRegularArms",function(player,LArm,RArm)

LArm.Transparency = 0

RArm.Transparency = 0

end)

Network.listen("Server","Send","ChangeRLegWeldC1\_"..Plyr.Value.UserId,function(player,RLegWeld,legC1)

RLegWeld.C1 = legC1

end)

Network.listen("Server","Send","changeHeadC1\_"..Plyr.Value.UserId,function(player,headWeld, headC1)

headWeld.C1 = headC1

end)

Network.listen("Server","Send","ResetShoulders\_"..Plyr.Value.UserId,function(player,Shoulders,LArm,RArm)

Shoulders.Right.Part1 = RArm

Shoulders.Left.Part1 = LArm

end)

Network.listen("Server","Send","DestroyHeadWeld\_"..Plyr.Value.UserId,function(player,headWeld)

headWeld:Destroy()

end)

Network.listen("Server","Send","changeLLegWeldC1\_"..Plyr.Value.UserId,function(player,LLegWeld, legC1)

LLegWeld.C1 = legC1

end)

Network.listen("Server","Send","RemoveArmModel",function(player,armModel)

armModel:Destroy()

end)

Network.listen("Server","Send","Disarm\_"..Plyr.Value.UserId,function(player,part)

local tools = part.Parent

if tools:IsA("Tool") and isEnemy(tools.Parent:FindFirstChildOfClass("Humanoid")) then

tools.Parent = game.Workspace

local BAV = Instance.new("BodyAngularVelocity")

BAV.Name = "Flip"

BAV.Parent = tools.Handle

local BV = Instance.new("BodyVelocity")

BV.MaxForce = V3(4e6,4e6,4e6)

BV.Velocity = Handle.CFrame.upVector \* 20

BV.Parent = tools.Handle

BAV.AngularVelocity = V3(1,1,1) \* RAD(RANDOM(1,360))

delay(4,function()

BAV:Destroy()

BV:Destroy()

end)

end

end)

Network.listen("Server","Send","changeRWeldC1\_"..Plyr.Value.UserId,function(player, RWeld,rC1)

RWeld.C1 = rC1

end)

Network.listen("Server","Send","changeLWeldC1\_"..Plyr.Value.UserId,function(player, LWeld,lC1)

LWeld.C1 = lC1

end)

Network.listen("Server","Send","ReplicateBullet\_"..Plyr.Value.UserId,function(player,className,Direction,gunIgnore,bCFrame)

local Bullet

if className == "Part" then

Bullet = OBJ("Part")

Bullet.BrickColor = S.bulletSettings.Color

Bullet.Material = Enum.Material.Neon

Bullet.Name = "Bullet"

Bullet.CanCollide = false

Bullet.FormFactor = "Custom"

Bullet.Size = S.bulletSettings.Size

Bullet.BottomSurface = "Smooth"

Bullet.TopSurface = "Smooth"

if MIN(S.bulletSettings.Size.X, S.bulletSettings.Size.Y, S.bulletSettings.Size.Z) < 0.2 then

local Mesh = OBJ("BlockMesh")

Mesh.Scale = S.bulletSettings.Size / V3(

MAX(S.bulletSettings.Size.X, 0.2),

MAX(S.bulletSettings.Size.Y, 0.2),

MAX(S.bulletSettings.Size.Z, 0.2)

)

Mesh.Parent = Bullet

end

elseif className == "MeshPart" then

Bullet = game.ReplicatedStorage.StunBullet:Clone()

end

local BF = OBJ("BodyForce")

BF.force = V3(0, Bullet:GetMass() \* (196.2 - S.bulletSettings.Acceleration), 0)

local BV = OBJ("BodyVelocity")

BV.MaxForce = V3(1000000,10000000,10000000)

BV.Velocity = Direction \* S.bulletSettings.Velocity

BF.Parent = Bullet

BV.Parent = Bullet

Bullet.Parent = gunIgnore

Bullet.CFrame = bCFrame

Bullet:SetNetworkOwner(nil)

Gun.BulletObj.Value = Bullet

end)

Network.listen("Server","Fetch","GetCurrentBullet\_"..Plyr.Value.UserId,function(player)

return Gun.BulletObj.Value

end)

--Gun.BulletObj.Changed:connect(function(bullet)

-- if bullet then

-- local Bullet = bullet:Clone()

--

-- end

--end)

local Humanoid = Plyr.Value.Character.Humanoid

local Torso = Plyr.Value.Character.Torso

Gun.Equipped:connect(function()

Network.startServer();

Optics:SetupScope(Gun.CurrentScope.Value)

end)

Gun.Unequipped:connect(function()

local ignoreCode = Gun:WaitForChild("clientMain"):WaitForChild("ignoreCode").Value

local resetCam = script:WaitForChild("resetCam"):Clone()

resetCam.Disabled = false

resetCam.GunServer.Value = script

resetCam:WaitForChild("ignoreCode").Value = ignoreCode

resetCam.Parent = Plyr.Value.PlayerGui

Network.reset()

end)

SETTINGS.lua

local CF = CFrame.new

local RAD = math.rad

local CFANG = CFrame.Angles

local BC = BrickColor.new

local Settings = {

gunType = {

Semi = false;

Auto = true;

Burst = false;

Shot = false;

Stun = false;

Hook = false;

Explosive = false;

};

selectFire = true;

selectFireSettings = {

Animation = true;

GUI = true;

Modes = {

Safety = true;

Semi = true;

Stun = true;

Burst = true;

Auto = true;

Hook = false;

};

animSpeed = 0.5;

};

burstSettings = {

fireRateBurst = true;

Amount = 3;

Time = 0.2;

Wait = 0.1;

};

shotAmount = 5;

explosionSettings = {

Radius = 20;

Pressure = 5e5;

Type = Enum.ExplosionType.NoCraters;

soundId = "rbxassetid://138499093";

soundPitch = 1;

soundVolume = 1;

rayCastExplosions = false;

rangeBasedDamage = true;

};

Debug = true;

playerArms = true;

fakeArmSettings = {

Transparency = 0;

armSize = Vector3.new(0.6, 2, 0.6);

characterMeshes = true;

realBodyColor = true;

Color = BC("Pastel brown");

};

canFireWhileGrappling = true;

unAimedC1 = {

leftArm = CF(-0.7, 2, -.8) \* CFANG(RAD(-10), 0, RAD(-30));

rightArm = CF(0.4, 0.25, -0.3) \* CFANG(0, 0, RAD(25));

Grip = CFANG(0, RAD(25), 0);

};

aimedC1 = {

leftArm = CF(-0.1, 1, -0.3) \* CFANG(RAD(-10), 0, 0) \* CFANG(0, 0, RAD(-40));

rightArm = CF(0.5, 0.3, 0.1) \* CFANG(0, 0, RAD(45));

};

runningC1 = {

leftArm = CF(-0.65, 0.85, -1.1) \* CFANG(RAD(1), RAD(-8.5), RAD(16));

rightArm = CF(0.16, 1, -0.14) \* CFANG(RAD(15), RAD(2), RAD(50));

Grip = CFANG(0, RAD(-5), 0);

};

equipAnimation = true;

equipSettings = {

Time = 0.25;

leftArmC1 = CF(0.2, 1.2, 0) \* CFANG(RAD(105), RAD(-30), RAD(90));

rightArmC1 = CF(-0.5, 0.75, 0) \* CFANG(RAD(45), 0, RAD(75));

GripC1 = CF();

};

stopAnimsOnFall = true;

fallAnimation = true;

fallSettings = {

maxDist = 35;

landMultiplier = 1;

fallMultiplier = 1;

aimEffect = 0.25;

};

gunMomentum = true;

momentumSettings = {

maxInput = 18;

Speed = 20;

Damper = 0.5;

Amplitude = {

unAimed = 5;

Aimed = 1;

}

};

cockingAnim = false;

movementAnims = true;

canADS = true;

aimSettings = {

Anim = true;

Speed = 0.3;

OutFOV = 80;

InFOV = 25;

holdToADS = true;

headTilt = math.rad(25);

};

sensitivitySettings = {

Default = 1;

Aim = 0.3;

Increment = 0.05;

};

guiScope = false;

scopeSettings = {

Frequency = {

Idling = 0.7;

Walking = 2;

};

Amplitude = {

Idling = 0.75;

Walking = 0.75;

};

steadyTime = 8;

breathTime = 5;

camSwayOnBreath = 2.5;

unSteadyOnFire = true;

};

roundsPerMin = 400;

bulletSettings = {

instantHit = false;

Range = 2400;

Velocity = 1000;

trueBolt = true;

Acceleration = 0.2;

Color = BC("Lime green");

Transparency = 0;

Size = Vector3.new(0.1, 0.1, 5);

};

damageSettings = {

Start = {

Damage = 24;

Dist = 0.08;

};

End = {

Damage = 12;

Dist = 0.5;

};

Multipliers = {

Chest = 1;

Head = 1.5;

Limbs = 1;

};

};

AllowFriendlyFire = false;

CanDamageNPCs = true;

bulletTrail = false;

trailSettings = {

Color = BC("White");

Transparency = 0;

Thickness = 0;

visibleTime = 0;

disappearTime = 0;

};

bulletHoles = true;

holeSettings = {

Texture = "http://www.roblox.com/asset/?id=315656769";

Size = 1;

visibleTime = 3;

disappearTime = 2;

};

bulletSparks = true;

customSparks = false;

sparkSettings = {

Color = {

Start = Color3.new(1, 158 / 255, 24 / 255);

End = Color3.new(212 / 255, 242 / 255, 8 / 255);

};

Size = 0.25;

Texture = "http://www.roblox.com/asset/?id=311395391";

Lifetime = 0.2;

Rate = 75;

Speed = 25;

Spread = 45;

Materials = {

Enum.Material.Plastic;

Enum.Material.Slate;

Enum.Material.Concrete;

Enum.Material.CorrodedMetal;

Enum.Material.DiamondPlate;

Enum.Material.Foil;

Enum.Material.Marble;

Enum.Material.Granite;

Enum.Material.Brick;

Enum.Material.Grass;

Enum.Material.Pebble;

Enum.Material.SmoothPlastic;

Enum.Material.Metal;

Enum.Material.Cobblestone;

};

};

bulletSmoke = false;

smokeSettings = {

objColor = false;

Color = Color3.new(0.5, 0.5, 0.5);

Size = {

Start = 0.25;

End = 0.5;

};

Texture = "http://www.roblox.com/asset/?id=244514423";

startTransparency = 0;

Lifetime = 0.2;

Rate = 100;

Speed = 35;

Spread = 15;

};

bloodEffect = false;

bloodSettings = {

Color = Color3.new(1, 0, 0);

Size = 0.1;

Texture = "http://www.roblox.com/asset/?id=3419963";

startTransparency = 0.125;

Lifetime = 0.1;

Rate = 200;

Speed = 50;

Spread = 15;

};

bulletShockwave = false;

shockwaveSettings = {

Radius = 0.3;

Color = BC("Light stone grey");

Duration = 0.2;

};

penetrationSettings = {

Dist = 0;

transparencyThreshold = 1;

ignoreNonCanCollide = true;

ignoreCustom = {};

maxTries = 10;

};

recoilSettings = {

firstShotMultiplier = 2.4;

aimedMultiplier = 0.5;

camMultiplier = 2;

springSpeed = 15;

springDamper = 0.5;

Recoil = {

Side = {

Left = -0.07;

Right = 0.14;

};

Up = {

Min = 0.13;

Max = 0.14;

};

Back = {

Min = 0.1;

Max = 0.15;

};

Tilt = {

Left = -0.25;

Right = 0.25;

};

}

};

spreadSettings = {

Increase = 0.1;

Decrease = 15;

Aimed = {

Stand = {

Idling = 0.2;

Moving = 0.4;

};

Crouch = {

Idling = 0.15;

Moving = 0.3;

};

Prone = {

Idling = 0.1;

Moving = 0.3;

};

};

unAimed = {

Stand = {

Idling = 2;

Moving = 2.5;

};

Crouch = {

Idling = 1.5;

Moving = 2;

};

Prone = {

Idling = 1;

Moving = 2;

};

};

};

reloadSettings = {

Anim = true;

Times = {

Loaded = 2.3;

Empty = 2.85;

};

autoReload = true;

magIsBullet = false;

reloadWhileRunning = true;

};

sprintTime = 15;

staminaCoolTime = 4;

canFireWhileRunning = false;

dolphinDive = true;

diveSettings = {

rechargeTime = 1;

Force = 350;

Angle = math.rad(30);

};

canChangeStance = true;

stanceSettings = {

Anim = true;

Speed = 0.3;

Stances = {

Crouch = true;

Prone = true;

};

standOnDeselect = true;

crawlAnimation = true;

};

walkSpeeds = {

Base = 16;

Sprinting = 25;

Aimed = 10;

Crouched = 7;

Prone = 4;

};

sheatheSettings = {

gunProfile = 1;

weldProfile = 1;

CFrameSettings = {

position = 0.75;

rotation = 45;

y = -.2;

x = -.7;

};

};

--NOTE: For extra keys, go here: http://wiki.roblox.com/index.php?title=Taking\_keyboard\_input

Keys = {

SpecialKey = Enum.KeyCode.LeftShift;

};

GamepadButtons = {

SpecialKey = Enum.KeyCode.ButtonR1;

Unequip = Enum.KeyCode.ButtonY;

};

}

return Settings

SheatheController.lua

local Gun = script.Parent

local S = require(Gun:WaitForChild("SETTINGS"))

local Character = Gun.Parent

local OBJ = Instance.new

local sheatheModel = nil

local guntype = S.sheatheSettings.gunProfile

local weldmode = S.sheatheSettings.weldProfile

local distance = S.sheatheSettings.CFrameSettings.position

local rotation = S.sheatheSettings.CFrameSettings.rotation

local y = S.sheatheSettings.CFrameSettings.y

local x = S.sheatheSettings.CFrameSettings.x

-----------------------------

parts = {}

local n = 1

--can i have my

function on(mouse)

if sheatheModel == nil then

n = 1

local m = OBJ("Model")

local all = Gun:GetChildren()

for i = 1, #all do

if all[i]:IsA("BasePart") then

parts[n] = all[i].Transparency

local brick = all[i]:Clone()

brick.Parent = m

n = n +1

end

end

wait()

if sheatheModel == nil then

local weld = script:FindFirstChild("SheatheWeld")

if weld ~= nil then

local new = weld:clone()

new.Disabled = false

new.Parent = m

m.Name = Gun.Name

m.Parent = Character

sheatheModel = m

local handle = sheatheModel:FindFirstChild("Handle")

if handle ~= nil then

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

if S.sheatheSettings.gunProfile == 1 then

local torso = sheatheModel.Parent:FindFirstChild("Torso")

if torso ~= nil then

if weldmode == 1 then--barrel pointing upper-right

local w = Instance.new("Motor6D")

w.Part0 = torso

w.Parent = w.Part0

w.Part1 = handle

w.C1 = CFrame.new(distance \*-1, 0.25 +y, -0.75 +x \*-1) \* CFrame.fromEulerAnglesXYZ(math.rad(rotation \*-1), (math.pi / 2), 0)

elseif weldmode == 2 then--barrel pointing upper-left

local w = Instance.new("Weld")

w.Part0 = torso

w.Parent = w.Part0

w.Part1 = handle

w.C1 = CFrame.new(distance, 0.25 +y, -0.75 +x \*-1) \* CFrame.fromEulerAnglesXYZ(math.rad(rotation \*-1), (math.pi / 2 ) \*-1, 0)

elseif weldmode == 3 then--barrel pointing upside-right

local w = Instance.new("Weld")

w.Part0 = torso

w.Parent = w.Part0

w.Part1 = handle

w.C1 = CFrame.new(distance \*-1, -0.1+y, 0.2 +x \*-1) \* CFrame.fromEulerAnglesXYZ(math.rad(rotation), (math.pi / 2), -1.5)

elseif weldmode == 4 then--barrel pointing upside-left

local w = Instance.new("Weld")

w.Part0 = torso

w.Parent = w.Part0

w.Part1 = handle

w.C1 = CFrame.new(distance, 0.25+y, -0.75 +x \*-1) \* CFrame.fromEulerAnglesXYZ(math.rad(rotation), (math.pi / 2 +rotation) \*-1.1, 1)

end

end

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

elseif guntype == 2 then--BullPup

local torso = Character:FindFirstChild("Torso")

if torso ~= nil then

if weldmode == 1 then--barrel pointing upper-right

local w = Instance.new("Weld")

w.Part0 = torso

w.Parent = w.Part0

w.Part1 = handle

w.C1 = CFrame.new(distance \*-1, 0.25+y, -0.5 +x \*-1) \* CFrame.fromEulerAnglesXYZ(math.rad(rotation \*-1), math.pi / 2, 0)

elseif weldmode == 2 then--barrel pointing upper-left

local w = Instance.new("Weld")

w.Part0 = torso

w.Parent = w.Part0

w.Part1 = handle

w.C1 = CFrame.new(distance, 0.25 +y, -0.5 +x \*-1) \* CFrame.fromEulerAnglesXYZ(math.rad(rotation \*-1), math.pi / 2 \*-1, 0)

elseif weldmode == 3 then--barrel pointing upside-right

local w = Instance.new("Weld")

w.Part0 = torso

w.Parent = w.Part0

w.Part1 = handle

w.C1 = CFrame.new(distance \*-1, 0.25 +y, -0.5 +x \*-1) \* CFrame.fromEulerAnglesXYZ(math.rad(rotation), math.pi / 2, 0)

elseif weldmode == 4 then--barrel pointing upside-left

local w = Instance.new("Weld")

w.Part0 = torso

w.Parent = w.Part0

w.Part1 = handle

w.C1 = CFrame.new(distance, 0.25 +y, -0.5 +x \*-1) \* CFrame.fromEulerAnglesXYZ(math.rad(rotation), math.pi / 2 \*-1, 0)

end

end

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

elseif guntype == 3 then--pistol

local lleg = Character:FindFirstChild("Left Leg")

local rleg = Character:FindFirstChild("Right Leg")

local torso = Character:FindFirstChild("Torso")

if lleg ~= nil and rleg ~= nil then

if weldmode == 1 then--pistol on right leg

local w = Instance.new("Weld")

w.Part0 = rleg

w.Parent = w.Part0

w.Part1 = handle

w.C1 = CFrame.new(distance \*-1, 0 +y, -0.25 +x \*-1) \* CFrame.fromEulerAnglesXYZ(math.pi / 2, 0, 0)

elseif weldmode == 2 then--pistol on left leg

local w = Instance.new("Weld")

w.Part0 = lleg

w.Parent = w.Part0

w.Part1 = handle

w.C1 = CFrame.new(distance, 0 +y, -0.25 +x \*-1) \* CFrame.fromEulerAnglesXYZ(math.pi / 2, 0, 0)

elseif weldmode == 3 then--knife in pant's back, gangsta like

local w = Instance.new("Weld")

w.Part0 = torso

w.Parent = w.Part0

w.Part1 = handle

w.C1 = CFrame.new(distance \*-1, 0 +y, 0.25 +x) \* CFrame.fromEulerAnglesXYZ(math.pi / 2 , math.pi / 2, 0)

elseif weldmode == 4 then--knife in pant's front, gangsta like

local w = Instance.new("Weld")

w.Part0 = torso

w.Parent = w.Part0

w.Part1 = handle

w.C1 = CFrame.new(distance, 0 +y, 0.25 +x) \* CFrame.fromEulerAnglesXYZ(math.pi / 2 , math.pi / 2 \*-1, 0)

end

end

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

elseif guntype == 4 then--knife

local lleg = Character:FindFirstChild("Left Leg")

local rleg = Character:FindFirstChild("Right Leg")

local torso = Character:FindFirstChild("Torso")

if lleg ~= nil and rleg ~= nil and torso ~= nil then

if weldmode == 1 then--pistol on right leg

local w = Instance.new("Weld")

w.Part0 = rleg

w.Parent = w.Part0

w.Part1 = handle

w.C1 = CFrame.new(distance \*-1, 0.15 +y, -0.25 +x \*-1) \* CFrame.fromEulerAnglesXYZ(math.pi, 0, 0)

elseif weldmode == 2 then--pistol on left leg

local w = Instance.new("Weld")

w.Part0 = lleg

w.Parent = w.Part0

w.Part1 = handle

w.C1 = CFrame.new(distance, 0.15 +y, -0.25 +x \*-1) \* CFrame.fromEulerAnglesXYZ(math.pi, 0, 0)

elseif weldmode == 3 then--knife in pant's back, gangsta like

local w = Instance.new("Weld")

w.Part0 = torso

w.Parent = w.Part0

w.Part1 = handle

w.C1 = CFrame.new(distance \*-1, 0 +y, 0.25 +x) \* CFrame.fromEulerAnglesXYZ(math.pi , math.pi / 2, 0)

elseif weldmode == 4 then--knife in pant's front, gangsta like

local w = Instance.new("Weld")

w.Part0 = torso

w.Parent = w.Part0

w.Part1 = handle

w.C1 = CFrame.new(distance, 0 +y, 0.25 +x) \* CFrame.fromEulerAnglesXYZ(math.pi , math.pi / 2 \*-1, 0)

end

end

end

end

end

end

end

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

--------------------------------------------------------------------------------

if sheatheModel ~= nil then

n = 1

local all = sheatheModel:GetChildren()

for i = 1, #all do

if all[i]:IsA("BasePart") then

all[i].Transparency = 1

end

end

end

end

--check

function off(mouse)

if sheatheModel ~= nil then

n = 1

local all = sheatheModel:GetChildren()

for i = 1, #all do

if all[i]:IsA("BasePart") then

all[i].Transparency = parts[n]

local Do = true

if Do then

Do = false--dude!

n = n +1

end

end

end

end

end

--please?

script.Parent.Equipped:connect(on)

script.Parent.Unequipped:connect(off)

Comment

Leave a comment

Footer

© 2024 GitHub, Inc.

Footer navigation

Terms

Privacy

Security

Status

Docs

Contact