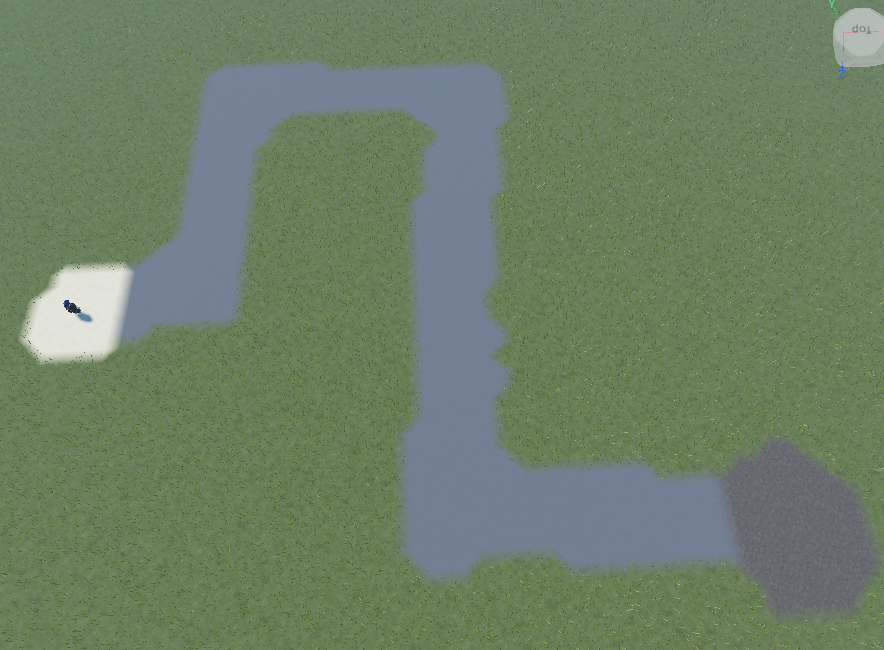
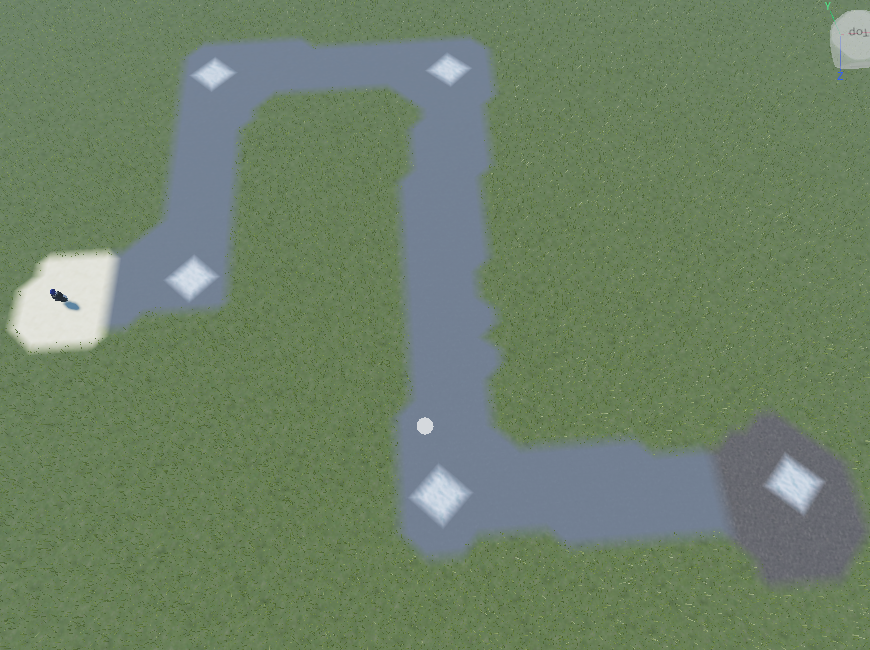
Як NPC повинен пройти шлях?

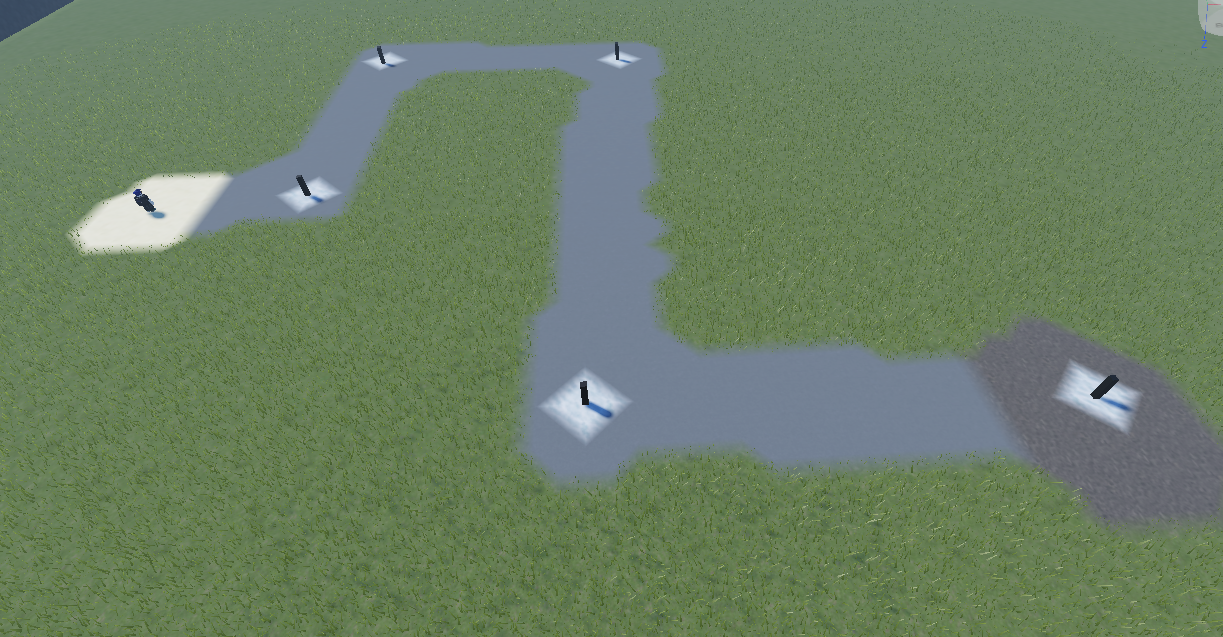
Я хочу, щоб персонаж пройшов від однієї позиції до іншої



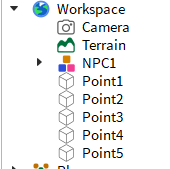
1. Для початку, потрібно обрати, скільки точок повинен пройти персонаж перед поворотом



1. Тепер, відносно цих точок, потрібно розробити деталі для проходу персонажа



1. Кожну з деталей потрібно переназвати на Point1, Point2, Point3 і так далі. Персонажа, який буде рухатись, потрібно теж переназвати на NPC1(не обов’язково, тоді потрібно буде змінювати код)



1. Тепер потрібно створити скрипт у ServerScriptService і переназвати його у Move



1. Потрібно внести наступний скрипт:

local npc = game.Workspace.NPC1 -- Ім'я персонажа можна змінити

local humanoid = npc.Humanoid

local pointA = game.Workspace.Point1 -- перша точка

local pointB = game.Workspace.Point2 -- друга точка

local pointC = game.Workspace.Point3 -- наступна точка

local pointD = game.Workspace.Point4 -- наступна точка

local pointE = game.Workspace.Point5 -- наступна точка

humanoid:MoveTo(pointA.Position) -- перейти до першої точки

humanoid.MoveToFinished:Wait() -- почекати

humanoid:MoveTo(pointB.Position) -- перейти до наступної точки

humanoid.MoveToFinished:Wait() -- почекати

humanoid:MoveTo(pointC.Position) -- перейти до наступної точки

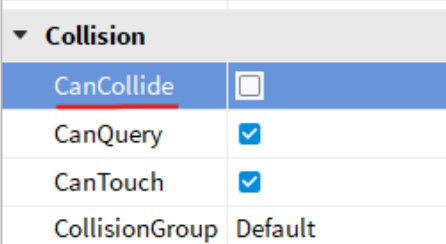
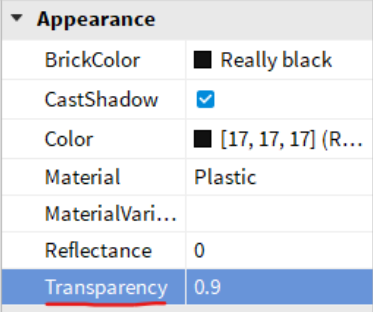
humanoid.MoveToFinished:Wait() -- почекати

humanoid:MoveTo(pointD.Position) -- перейти до наступної точки

humanoid.MoveToFinished:Wait() -- почекати

humanoid:MoveTo(pointE.Position) -- перейти до наступної точки

1. Обов’язково: перечитати зелений текст/коментар біля самого коду. Якщо потрібно буде менше або більше точок, тоді або скопіювати код, або видалити частину кода.
2. Тепер зробимо деталі прозорими для нашого персонажа. Змінимо властивості Transparency та CanCollide



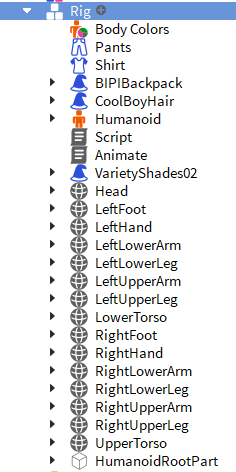
Ще один спосіб!

Якщо я хочу, щоб цей персонаж просто ходив навколо, мені потрібно:

1. додати персонажа



1. Додати до персонажа Script і перенести туди наступний код:



local npc = script.Parent

local humanoid = npc:WaitForChild("Humanoid")

while true do

wait(math.random(4))

humanoid:MoveTo(npc.PrimaryPart.Position + Vector3.new(math.random(-30,30),0,math.random(-30,30)))

humanoid.MoveToFinished:Wait()

end

1. Якщо ми хочемо, щоб персонаж обходив в різні перешкоди, тоді додамо в цей скрипт інший код:

local NPC = script.Parent

local Humanoid = NPC:FindFirstChildOfClass("Humanoid")

local Root = Humanoid.RootPart

local Pathfinding = game:GetService("PathfindingService")

local walkingAnimation = Instance.new("Animation")

walkingAnimation.AnimationId = "http://www.roblox.com/asset/?id=913376220" -- ID анімації

local walkingAnimTrack = Humanoid:LoadAnimation(walkingAnimation)

function PickARandomNearestLoc(range)

return Root.Position + Vector3.new(math.random(-range,range),0,math.random(-range,range))

end

function Path(From,To)

local Path = Pathfinding:CreatePath({

AgentCanJump = true,

AgentCanClimb = true

})

Path:ComputeAsync(From,To)

if Path.Status == Enum.PathStatus.Success then

local Waypoints = Path:GetWaypoints()

for i,v in pairs(Waypoints) do

Humanoid:MoveTo(v.Position)

if v.Action == Enum.PathWaypointAction.Jump then

Humanoid.Jump = true

end

Humanoid.MoveToFinished:Wait(2)

end

else

warn("NPC Stuck or no path was found.")

end

end

function Start()

while true do

wait(math.random(1,5))

local To = PickARandomNearestLoc(50)

walkingAnimTrack:Play()

Path(Root.Position,To)

walkingAnimTrack:Stop()

end

end

Start()

