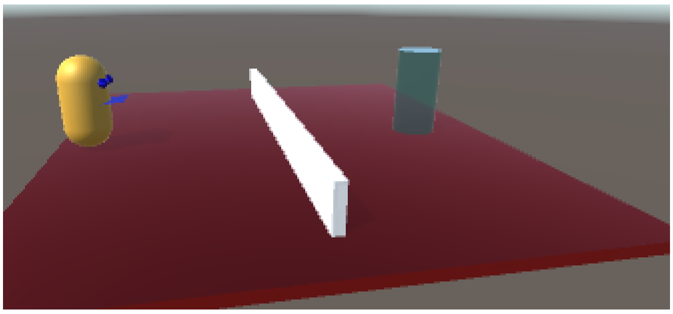
LAB 3: Create a player character as a 3D model and enable transformative actions such as moving left/right and forward/backward. (ADD AI)

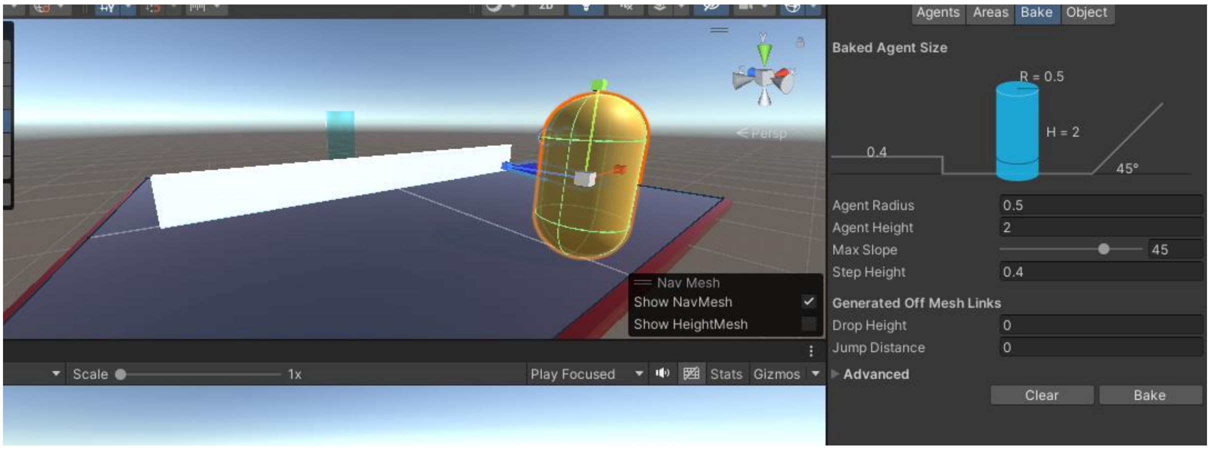
Sol:

Step 1: create environment for character, ground, Obstacle and AI



Step 2: Select Ground and click static

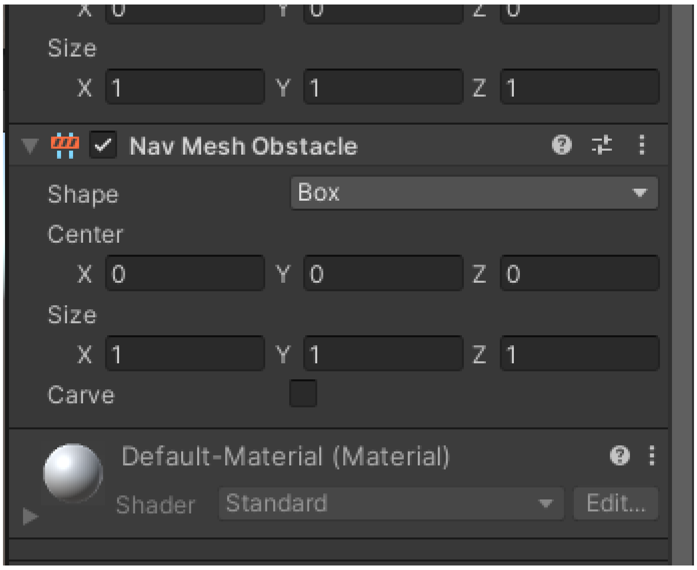
- Add Navigation path here it uses built-in A*
- Click bake it makes ground blue



Show NavMesh ☑
Show HeightMesh ☐

Agent Radius 0.5
Agent Height 2
Max Slope 45
Step Height 0.4
Drop Height 0
Jump Distance 0

Step 3: select white cube and add component Navmesh obstacle it will stop AI



Size
X 1 Y 1 Z 1

☑ Nav Mesh Obstacle

Shape: Box

Center
X 0 Y 0 Z 0

Size
X 1 Y 1 Z 1

Carve ☐

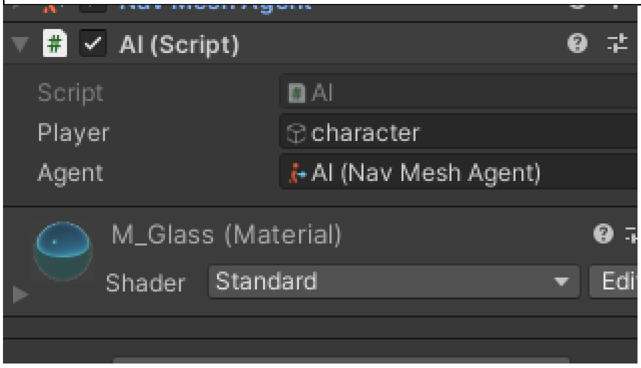
Default-Material (Material)

Shader: Standard

Step 4: Select AI ,cylinder and Add navmesh agent as component

- Create c# script AI and attach to Cylinder i.e our AI

```
using UnityEngine;
using UnityEngine.AI;

public class AI : MonoBehaviour
{
     public GameObject player;
    public NavMeshAgent agent;

    void Update () {
        agent.SetDestination(player.transform.position);
    }

}
```

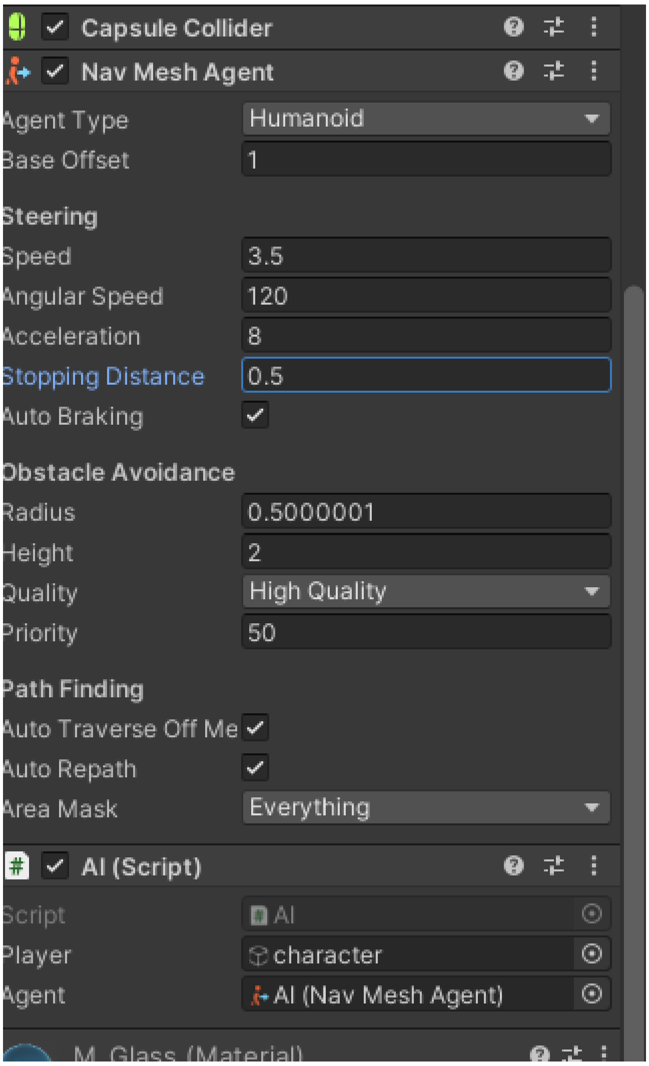


☑ AI (Script)

Script: AI
Player: character
Agent: AI (Nav Mesh Agent)

M_Glass (Material)

Shader: Standard



Capsule Collider

Nav Mesh Agent

Agent Type: Humanoid
Base Offset: 1

Steering
Speed: 3.5
Angular Speed: 120
Acceleration: 8
Stopping Distance: 0.5
Auto Braking: ✓

Obstacle Avoidance
Radius: 0.5000001
Height: 2
Quality: High Quality
Priority: 50

Path Finding
Auto Traverse Off Me: ✓
Auto Repath: ✓
Area Mask: Everything

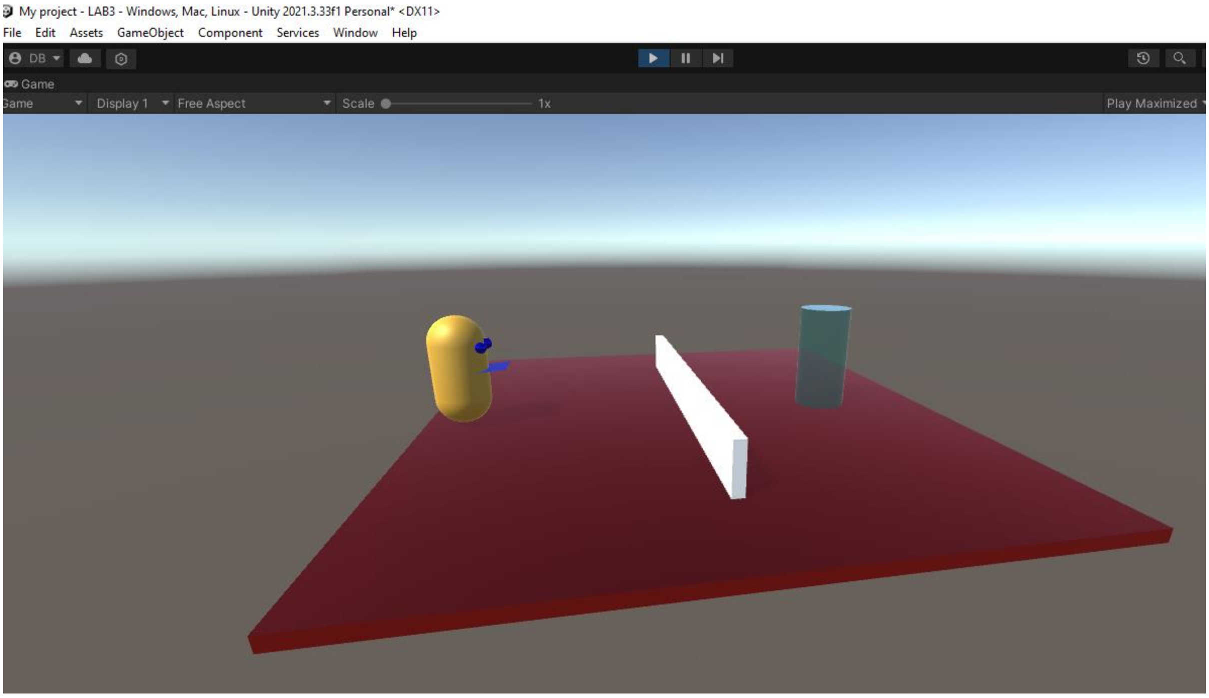
AI (Script)
Script: AI
Player: character
Agent: AI (Nav Mesh Agent)

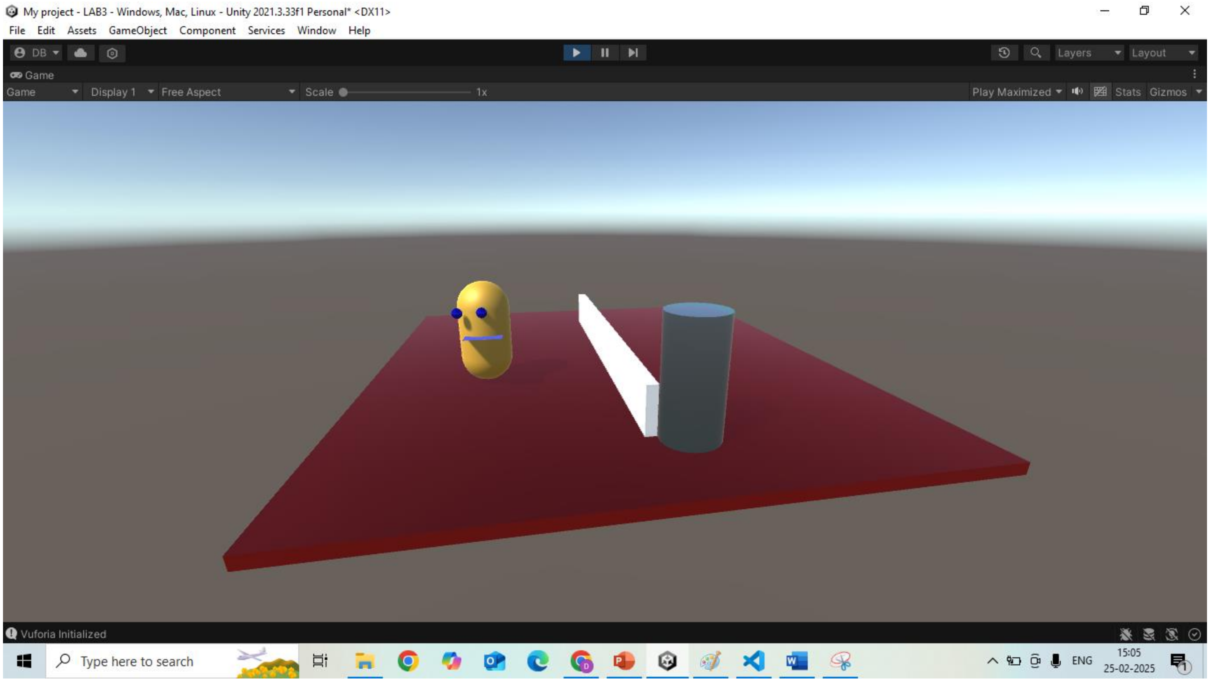
M_Glass (Material)

Step 5: Create C# Script ,Attach script to character and attach reference for character controller

```
using UnityEngine;

public class LAB4 : MonoBehaviour
{
    public float moveSpeed = 50f;
    public float rotationSpeed = 700f;
    public CharacterController controller;
    private Vector3 moveDirection;
    void Update()
    {

        float moveX = Input.GetAxis("Horizontal");
        float moveZ = Input.GetAxis("Vertical");
        // Calculate movement direction based on input
        moveDirection = new Vector3(moveX, 0f, moveZ);
        if (moveDirection.magnitude > 0)
        {
            Quaternion toRotation = Quaternion.LookRotation(moveDirection, Vector3.up);
            transform.rotation = Quaternion.RotateTowards(transform.rotation, toRotation,
rotationSpeed * Time.deltaTime);
        }
        // Apply the movement to the character
        controller.Move(moveDirection * moveSpeed * Time.deltaTime);
    }

}
```

Output:


My project - LAB3 - Windows, Mac, Linux - Unity 2021.3.33f1 Personal* <DX11>
File Edit Assets GameObject Component Services Window Help
DB
Game
Game
Display 1
Free Aspect
Scale
1x
Play Maximized



Vuforia Initialized