19CS34L-Augmented Reality for Developers

Project Title:

PLANE FINDER USING UNITY

Problem Analysis (5)	Coding & Implementation (10)	Report/Documentation Submission (10)	Presentation/Demo (5)

Done by,

MITHUNKUMAR C 2112100

ALAGU NAGASH G 2112050

KAVIRAJ N 2112064

AIM:

Plane Finder using Unity with Augmented Reality (AR) is an innovative application that combines the power of Unity, a popular game development engine, with AR technology to locate and visualize real-world surfaces such as floors, tables, and walls. By leveraging the camera feed from a mobile device, this application allows users to detect and interact with these surfaces in a virtual environment.

ABSTRACT:

The purpose of a plane finder is to enable users to place virtual objects accurately in the real world. It serves as a fundamental building block for various AR experiences, including virtual furniture placement, interior design, gaming, and more. With the Unity engine and AR capabilities, developers can create compelling and immersive applications that seamlessly blend the physical and virtual worlds.

PROCEDURE:

1.**Set up Unity:** Install Unity on your computer and create a new project.
2.**Import AR packages**: Import AR-related packages into your Unity project. For AR development, you can use Unity's built-in AR Foundation package or third-party AR frameworks like Vuforia or AR Core.
3.**Create a scene:** Set up a scene in Unity where you can display the AR camera feed and overlay virtual objects on top of the real world.
4.**Set up AR camera:** Add an AR camera to your scene, which will capture the device's camera feed and enable AR tracking.
5.**Add plane detection**: Enable plane detection in your AR session. This will allow the AR system to detect horizontal or vertical surfaces like floors, tables, or walls.
6.**Create plane visualization**: When a plane is detected, you can visualize it in the scene by creating a corresponding game object (e.g., a flat plane or a mesh) and rendering it in the AR world.
7.**Attach AR objects**: Create virtual objects that represent planes or markers for your plane finder. These objects can be icons, labels, or 3D models that will be placed in the AR world once the planes are detected.
8.**Handle user input:** Implement user interaction to allow users to trigger the plane finder functionality. For example, you can add a button that initiates the plane detection process.
9.**Overlay virtual objects:** Once a plane is detected, instantiate the virtual objects onto the detected planes. You can use the AR camera's position and rotation data to align the virtual objects with the real-world surfaces.
10.**Enhance user experience:** Consider adding additional features to enhance the user experience. For example, you can implement gesture controls, distance measurements, or information tooltips for the virtual objects.
11.**Build and test:** Build the project for your target platform (e.g., Android, iOS) and test it on an AR-compatible device. Ensure that the plane finder functionality works as expected and the virtual objects are properly aligned with the detected planes

Output:

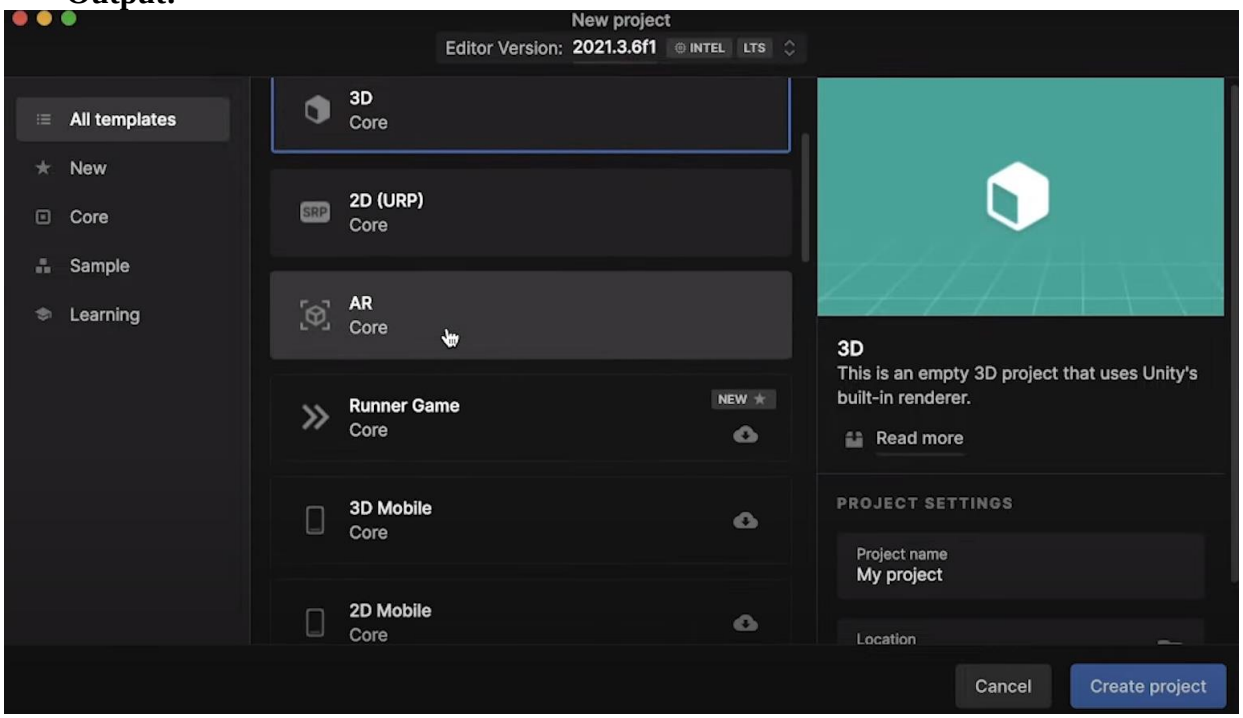

New project
Editor Version: 2021.3.6f1
INTEL
LTS
All templates
New
Core
Sample
Learning
3D
Core
2D (URP)
Core
AR
Core
Runner Game
Core
NEW
3D Mobile
Core
2D Mobile
Core
3D
This is an empty 3D project that uses Unity's built-in renderer.
Read more
PROJECT SETTINGS
Project name
My project
Location
Cancel
Create project

Fig.1

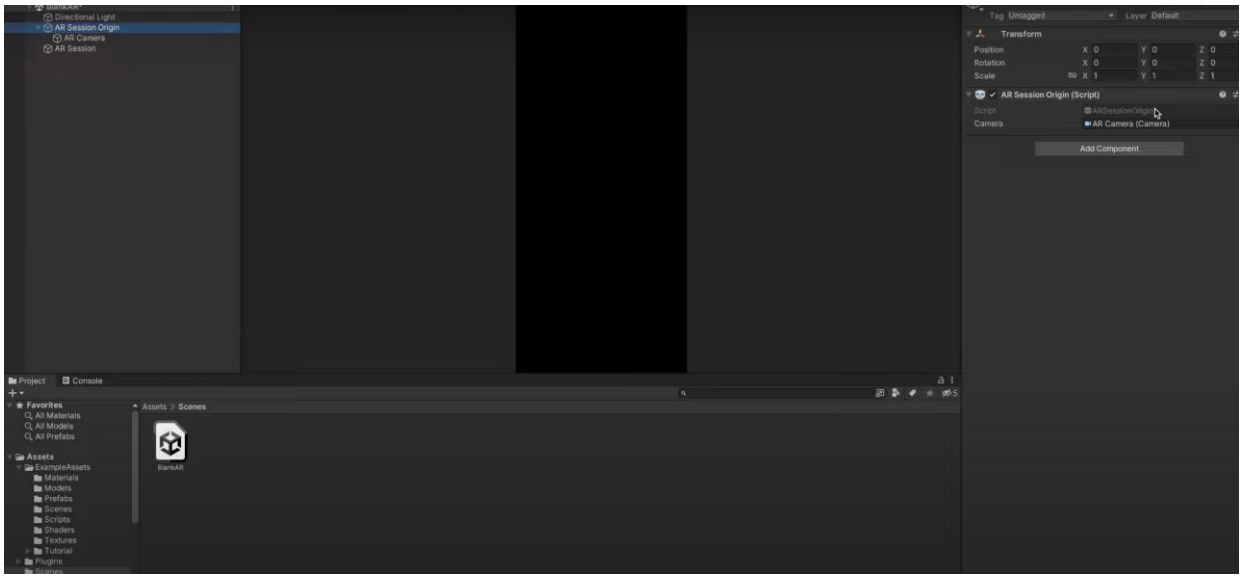

Directional Light
AR Session Origin
AR Camera
AR Session
Tag Untagged
Layer Default
Transform
Position
Rotation
Scale
AR Session Origin (Script)
Script
Camera
AR Camera (Camera)
Add Component
Project
Console
Favorites
All Materials
All Models
All Prefabs
Assets
Scenes
ExampleAssets
Materials
Models
Prefabs
Scenes
Scripts
Shaders
Textures
Tutorial
Plugins

Fig 2

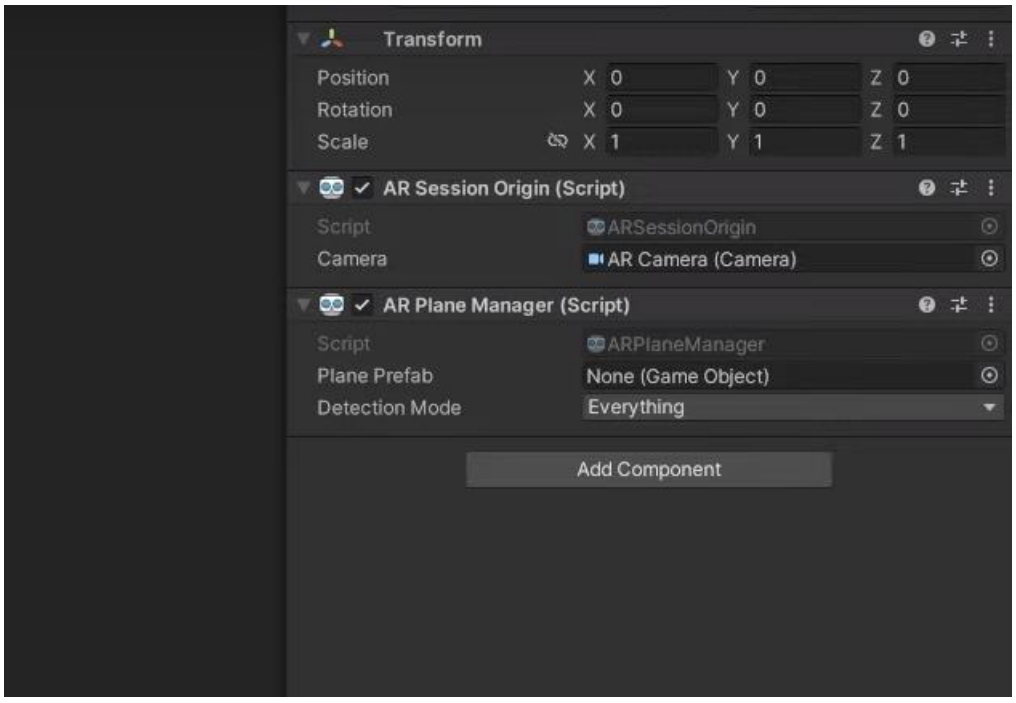

Transform
Position
Rotation
Scale
X 0
Y 0
Z 0
X 0
Y 0
Z 0
X 1
Y 1
Z 1
AR Session Origin (Script)
Script
ARSessionOrigin
Camera
AR Camera (Camera)
AR Plane Manager (Script)
Script
ARPlaneManager
Plane Prefab
None (Game Object)
Detection Mode
Everything
Add Component

Fig 3

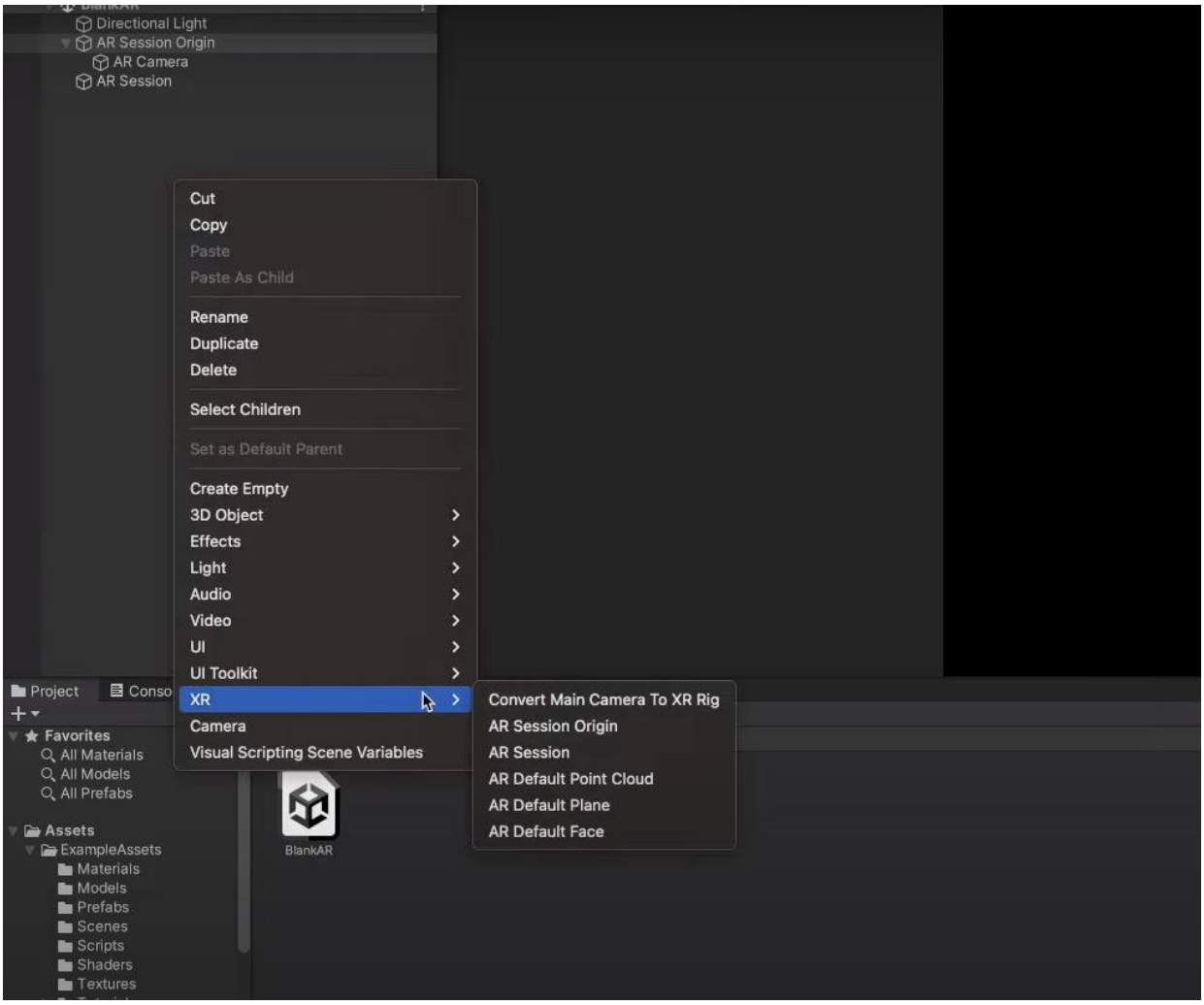

Directional Light
AR Session Origin
AR Camera
AR Session
Cut
Copy
Paste
Paste As Child
Rename
Duplicate
Delete
Select Children
Set as Default Parent
Create Empty
3D Object
Effects
Light
Audio
Video
UI
UI Toolkit
XR
Camera
Visual Scripting Scene Variables
Convert Main Camera To XR Rig
AR Session Origin
AR Session
AR Default Point Cloud
AR Default Plane
AR Default Face
Project
Favorites
All Materials
All Models
All Prefabs
Assets
ExampleAssets
Materials
Models
Prefabs
Scenes
Scripts
Shaders
Textures
BlankAR

Fig 4

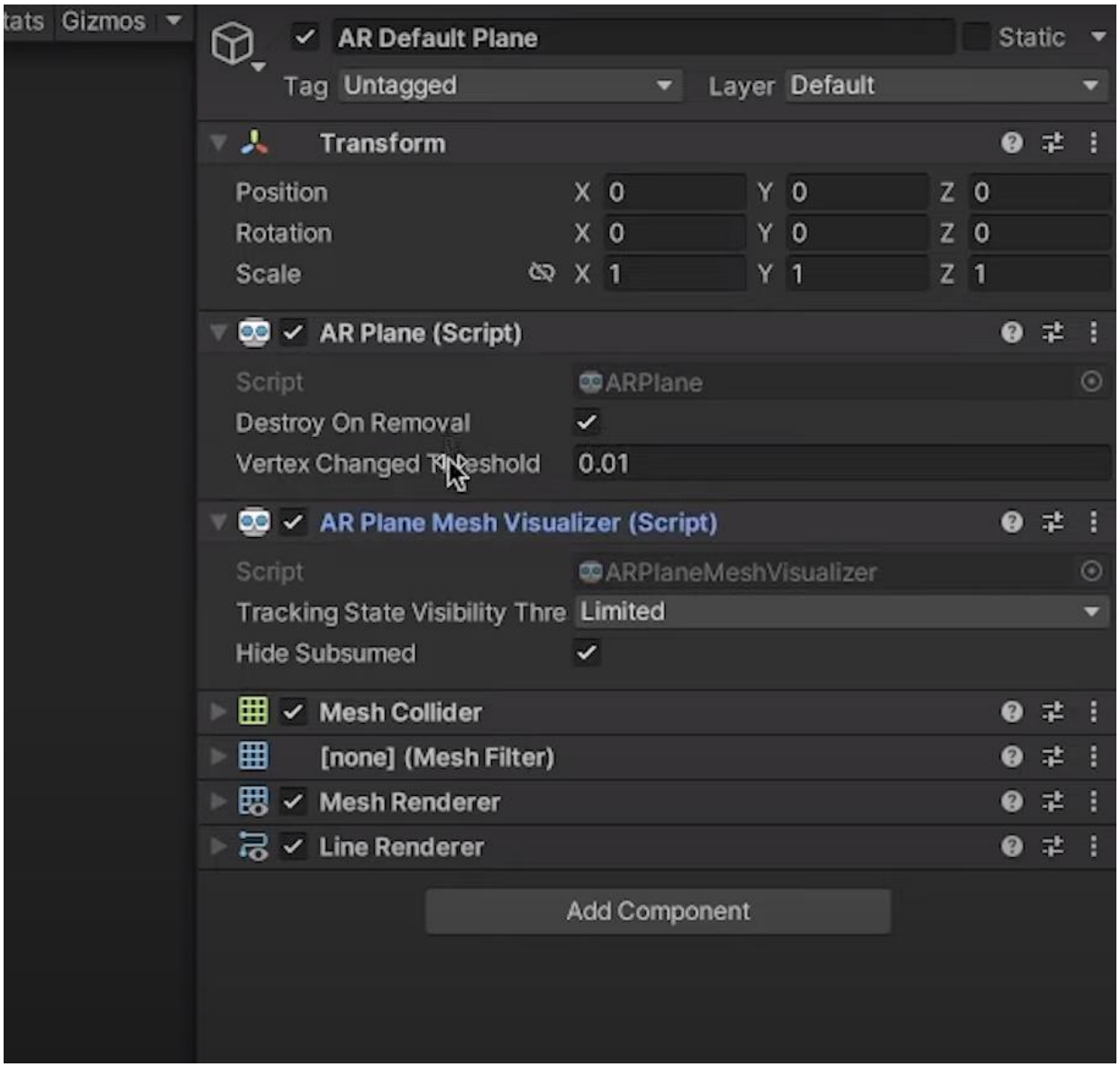

tats
Gizmos
AR Default Plane
Static
Tag
Untagged
Layer
Default
Transform
Position
X 0
Y 0
Z 0
Rotation
X 0
Y 0
Z 0
Scale
X 1
Y 1
Z 1
AR Plane (Script)
Script
ARPlane
Destroy On Removal
Vertex Changed Threshold
0.01
AR Plane Mesh Visualizer (Script)
Script
ARPlaneMeshVisualizer
Tracking State Visibility Thre
Limited
Hide Subsumed
Mesh Collider
[none] (Mesh Filter)
Mesh Renderer
Line Renderer
Add Component

Fig 5

```
    /// </summary>
    public float vertexChangedThreshold
    {
        get => m_VertexChangedThreshold;
        set => m_VertexChangedThreshold = Mathf.Max(0f, value);
    }

    /// <summary>
    /// Invoked when any vertex in the plane's boundary changes by more
        than <see cref="vertexChangedThreshold"/>.
    /// </summary>
    public event Action<ARPlaneBoundaryChangedEventArgs> boundaryChanged;

    /// <summary>
    /// Gets the normal to this plane in world space.
    /// </summary>
    public Vector3 normal => transform.up;

    /// <summary>
    /// The <see cref="ARPlane"/> which has subsumed this plane, or
        <c>null</c>
    /// if this plane has not been subsumed.
    /// </summary>
    public ARPlane subsumedBy { get; internal set; }

    /// <summary>
    /// The alignment of this plane.
    /// </summary>
    public PlaneAlignment alignment => sessionRelativeData.alignment;
```

Fig 6

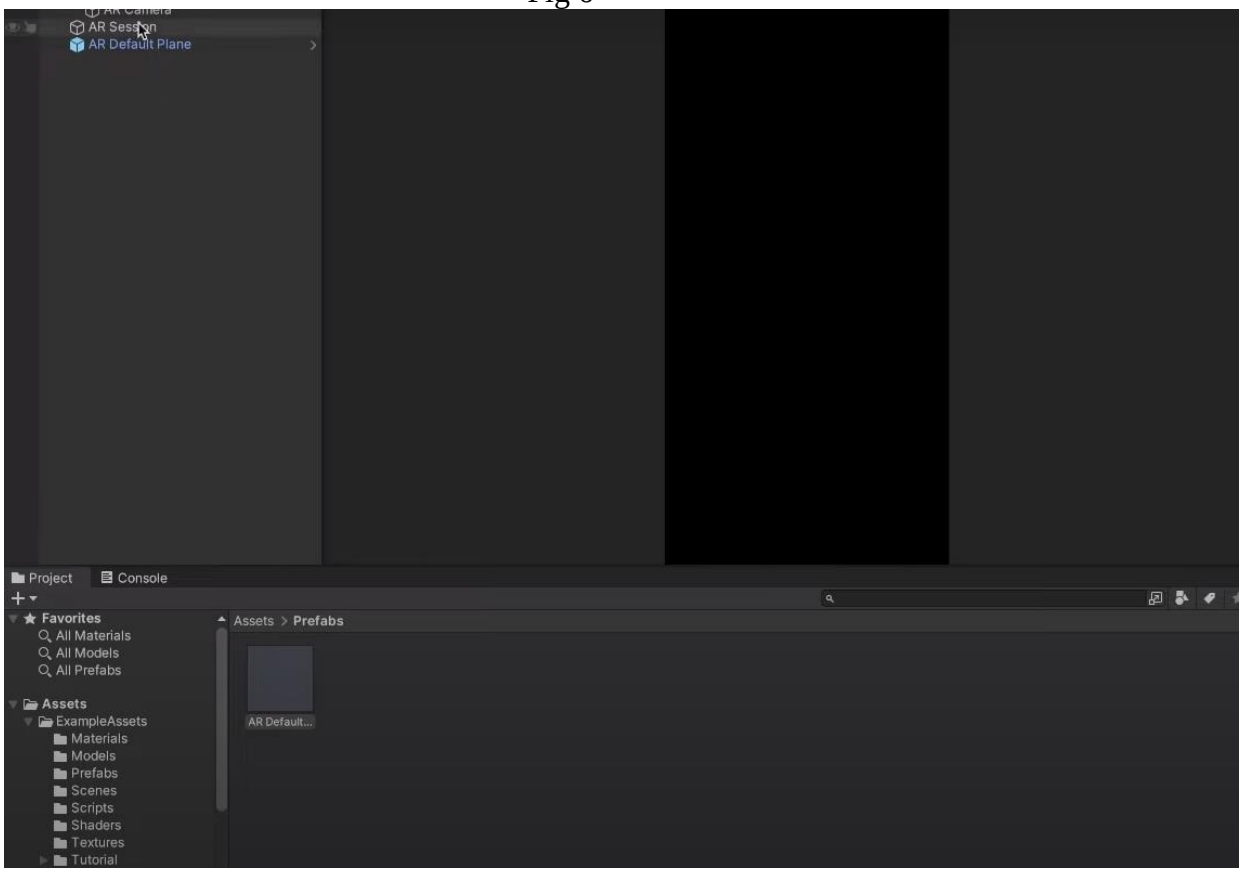


Fig 7

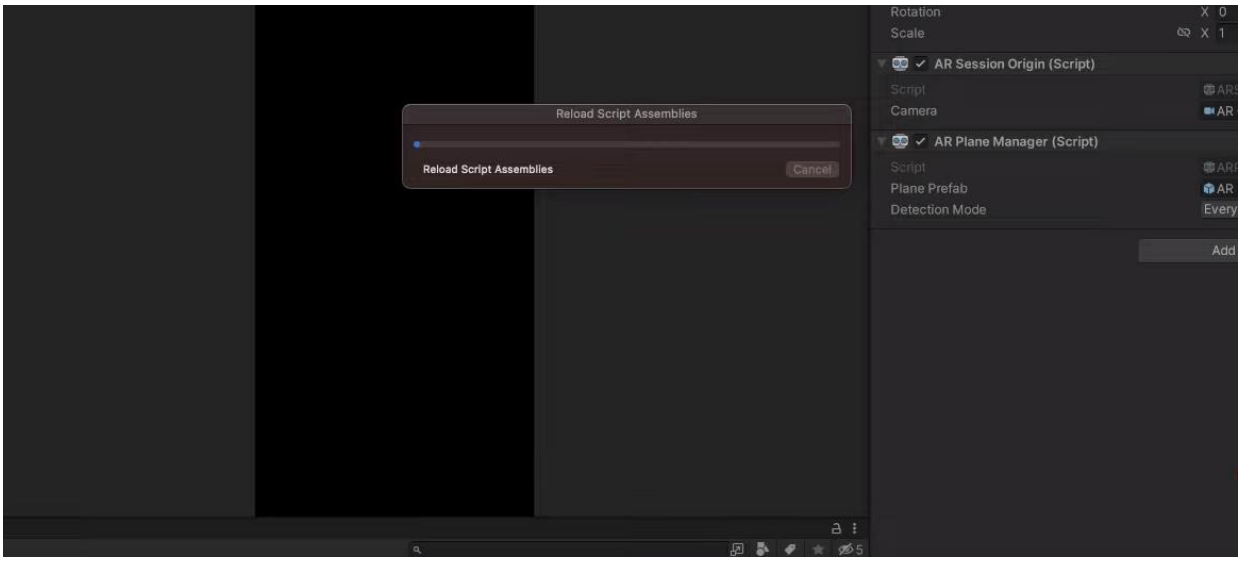

Reload Script Assemblies
Reload Script Assemblies
Cancel
Rotation
Scale
AR Session Origin (Script)
Script
Camera
AR Plane Manager (Script)
Script
Plane Prefab
Detection Mode
Add

Fig 8

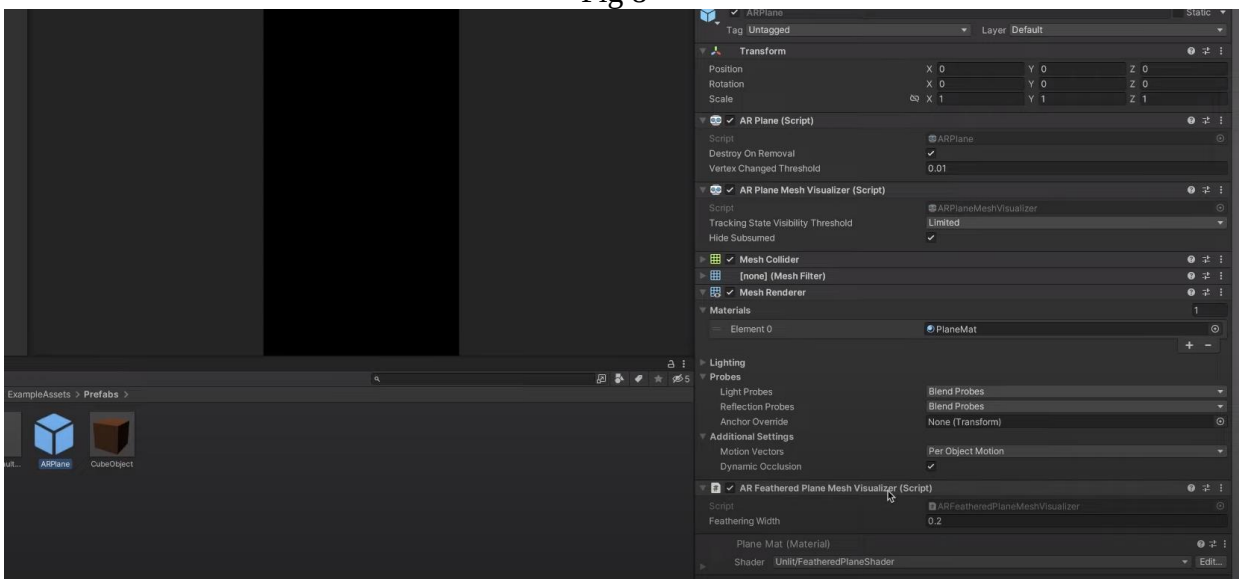
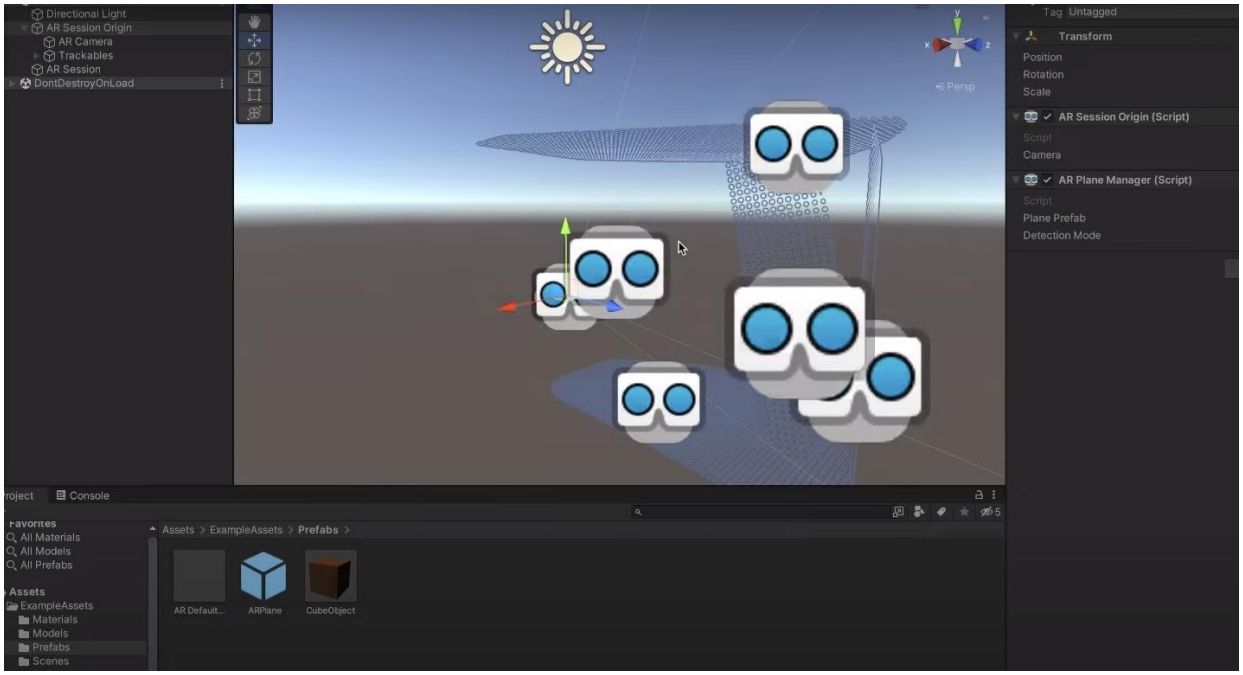
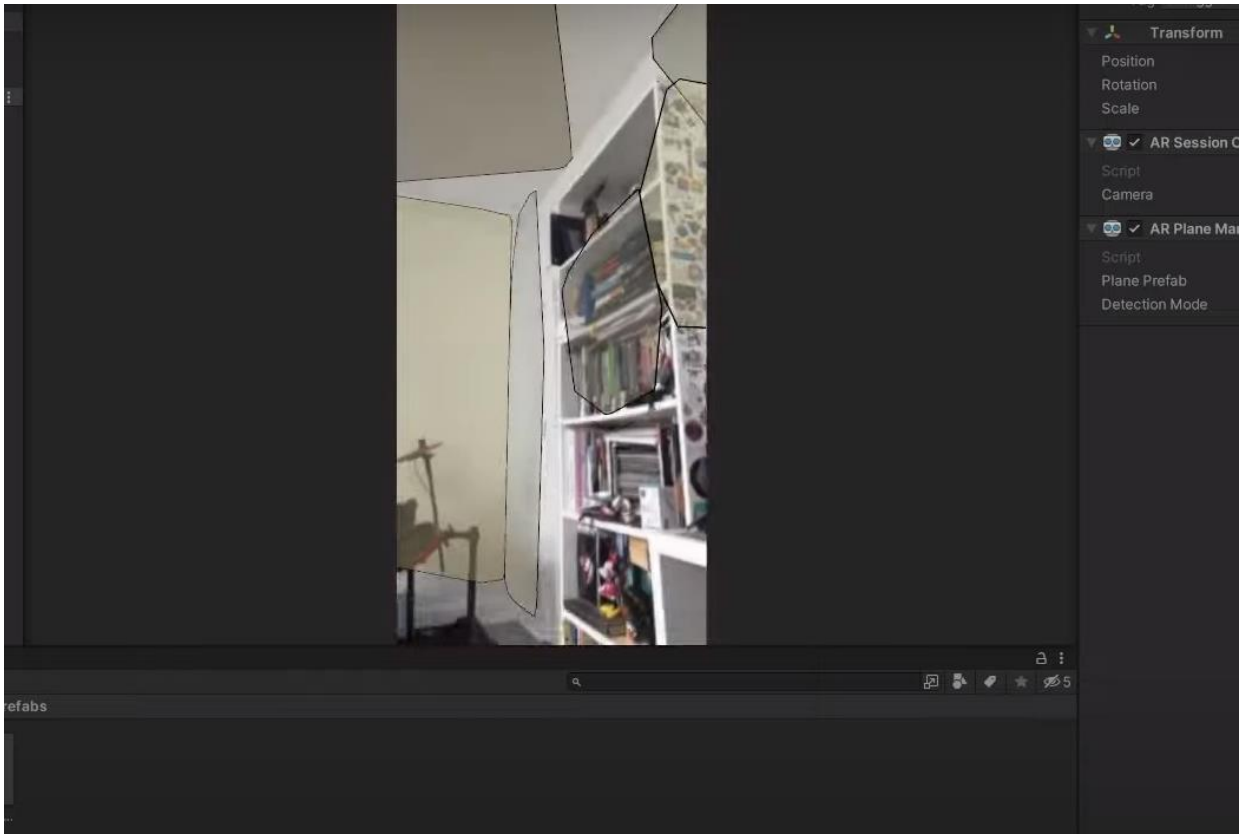

Tag Untagged
Layer Default
Transform
Position
Rotation
Scale
AR Plane (Script)
Destroy On Removal
Vertex Changed Threshold
0.01
AR Plane Mesh Visualizer (Script)
Tracking State Visibility Threshold
Limited
Hide Subsumed
Mesh Collider
[none] (Mesh Filter)
Mesh Renderer
Materials
Element 0
PlaneMat
Lighting
Probes
Light Probes
Blend Probes
Reflection Probes
Blend Probes
Anchor Override
None (Transform)
Additional Settings
Motion Vectors
Per Object Motion
Dynamic Occlusion
AR Feathered Plane Mesh Visualizer (Script)
Feathering Width
0.2
Plane Mat (Material)
Shader
Unlit/FeatheredPlaneShader
ExampleAssets
Prefabs
ARPlane
CubeObject

Fig 9


Directional Light
AR Session Origin
AR Camera
Trackables
AR Session
DontDestroyOnLoad
Persp
Tag Untagged
Transform
Position
Rotation
Scale
AR Session Origin (Script)
Script
Camera
AR Plane Manager (Script)
Script
Plane Prefab
Detection Mode
Console
Favorites
All Materials
All Models
All Prefabs
Assets
ExampleAssets
Materials
Models
Prefabs
Scenes
Assets > ExampleAssets > Prefabs
AR Default...
ARPlane
CubeObject

OUTPUT:



CONCLUSION:

By implementing a plane finder using Unity with AR, users can easily explore their surroundings and identify suitable surfaces for virtual object placement. The application analyzes the camera feed in real-time, detecting flat surfaces and generating a representation of them within the virtual environment. This empowers users to visualize and interact with virtual objects as if they were present in the physical space.