Develop and deploy a simple marker-based AR game in which you have to write, say, a C# program to play video or tracking a particular marker. in unity and tell each and every step from the basics for it.

Tools You'll Need:

- Unity Hub + Unity Editor (preferably Unity 2021.3.x LTS)
- Vuforia Engine SDK
- Vuforia Developer Account (for license key and image targets)
- Any video file (e.g., .mp4)
- A webcam or phone camera for testing

Step 1: Install Unity and Set It Up

- Download and install Unity Hub from https://unity.com/.
- Inside Unity Hub:
    - Click "Install Editor" and choose Unity 2021.3.x LTS.
    - While installing, check:
      • Windows Build Support
      • Android Build Support (optional for mobile deployment)

Step 2: Create a New Unity Project

- Open Unity Hub, click "New Project", and choose the 3D Template.
- Name your project (e.g., "AR Video Marker Game").
- Click "Create".

Step 3: Add Vuforia to Your Project

- Go to "Window > Package Manager".
- Click the "+" button and select "Add package from git URL…".
- Paste: https://github.com/Vuforia/vuforia-unity-package.git
- Click "Add". Vuforia will now be imported.

Step 4: Set Up Vuforia

- Go to “File > Build Settings” and switch platform to Android (optional).
- Go to “Edit > Project Settings > Player”.
    - Under XR Settings, check “Vuforia Augmented Reality Supported”.
- Sign in at https://developer.vuforia.com and create an account.
- Navigate to License Manager and create a new license key.
- Back in Unity:
    - Delete the “Main Camera”.
    - Add a “Vuforia Engine > AR Camera”.
    - Paste the license key in AR Camera’s “App License Key” field.

Step 5: Add an Image Target

- In the Vuforia Developer Portal:
    - Go to “Target Manager > Add Database”.
    - Add Target: Type - Single Image, upload marker image, set width to 1.
- Download the database (Unity Editor format) and import into Unity.
- In Unity: “GameObject > Vuforia Engine > Image Target”.
- Set the ImageTarget to use your database and target image.

Step 6: Add a Video Player

- Right-click on the ImageTarget > “3D Object > Quad” → rename it “VideoScreen”.
- Set Scale to (1,1,1) and adjust its Position.
- Add Component: Video Player to the Quad.
- Set:
    - Source: Video Clip (drag and drop .mp4 into Assets and assign it).
    - Render Mode: Material Override.
    - Target Material: Quad’s Material.
    - Uncheck “Play on Awake”.
- Create a C# script named PlayOnDetect.cs and paste the following:

```
using UnityEngine;

using UnityEngine.Video;

using Vuforia;
```

```
public class PlayOnDetect : MonoBehaviour, ITrackableEventHandler
{
    private TrackableBehaviour mTrackableBehaviour;
    private VideoPlayer videoPlayer;

    void Start()
    {
        mTrackableBehaviour = GetComponent<TrackableBehaviour>();
        videoPlayer = GetComponentInChildren<VideoPlayer>();

        if (mTrackableBehaviour)
        {
            mTrackableBehaviour.RegisterTrackableEventHandler(this);
        }
    }

    public void OnTrackableStateChanged(
        TrackableBehaviour.Status previousStatus,
        TrackableBehaviour.Status newStatus)
    {
        if (newStatus == TrackableBehaviour.Status.DETECTED ||
            newStatus == TrackableBehaviour.Status.TRACKED)
```

```
        {
            videoPlayer.Play();
        }
        else
        {
            videoPlayer.Pause();
        }
    }
}
```

Step 7: Test the AR Game

1. Print or display the image marker on screen.
2. Connect a webcam.
3. Press “Play” in Unity.
4. Point the camera at the marker — your video should play.

Step 8: (Optional) Build for Android

1. Go to “File > Build Settings”.
2. Select Android and click “Switch Platform”.
3. Click “Add Open Scenes”.
4. Click “Build and Run” (enable USB debugging on your phone).