Getting Up and Running with Mobile AR

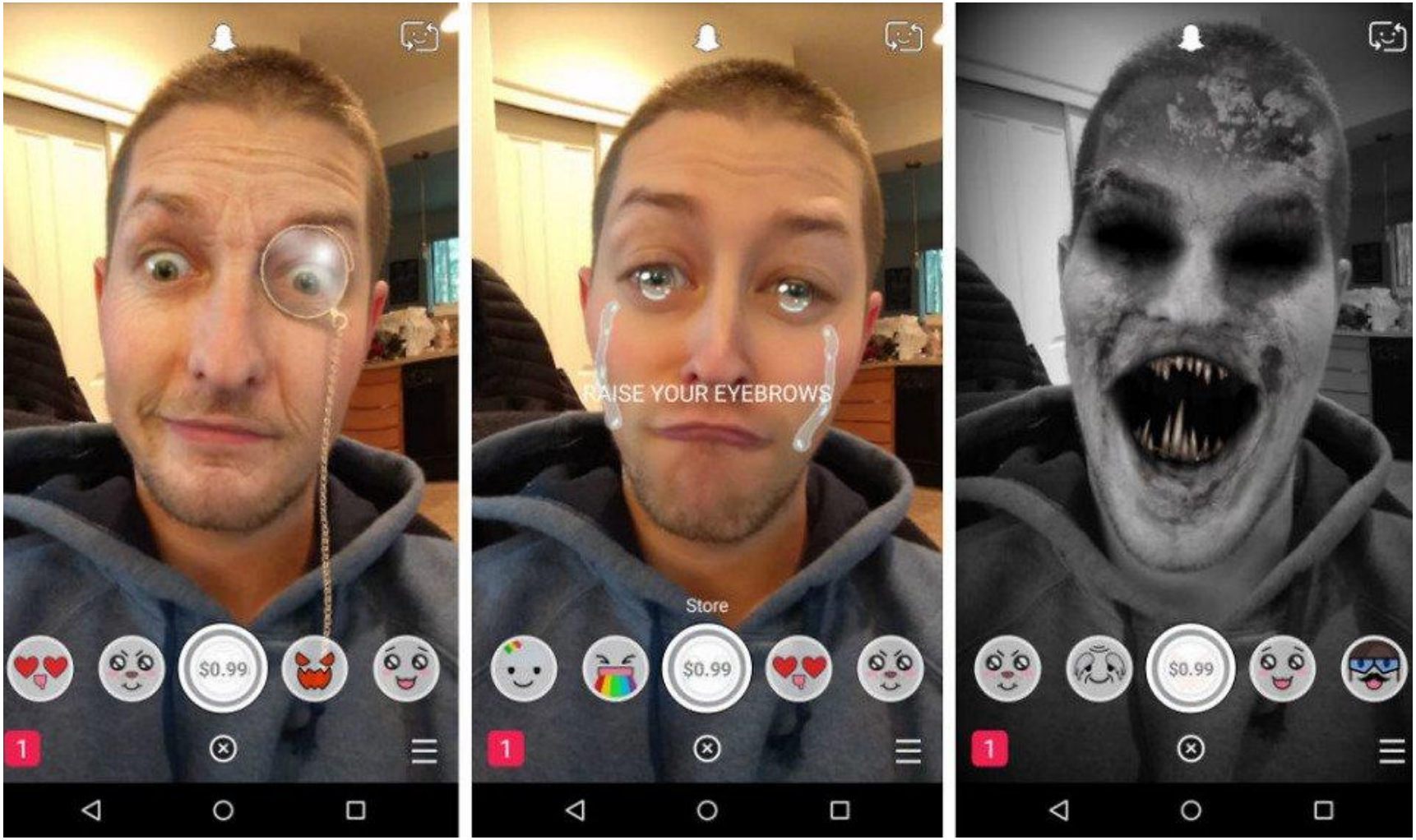
Brandon Johnson
Max Thorson

New York City
Minneapolis
Venice

Agenda

Augmented Reality

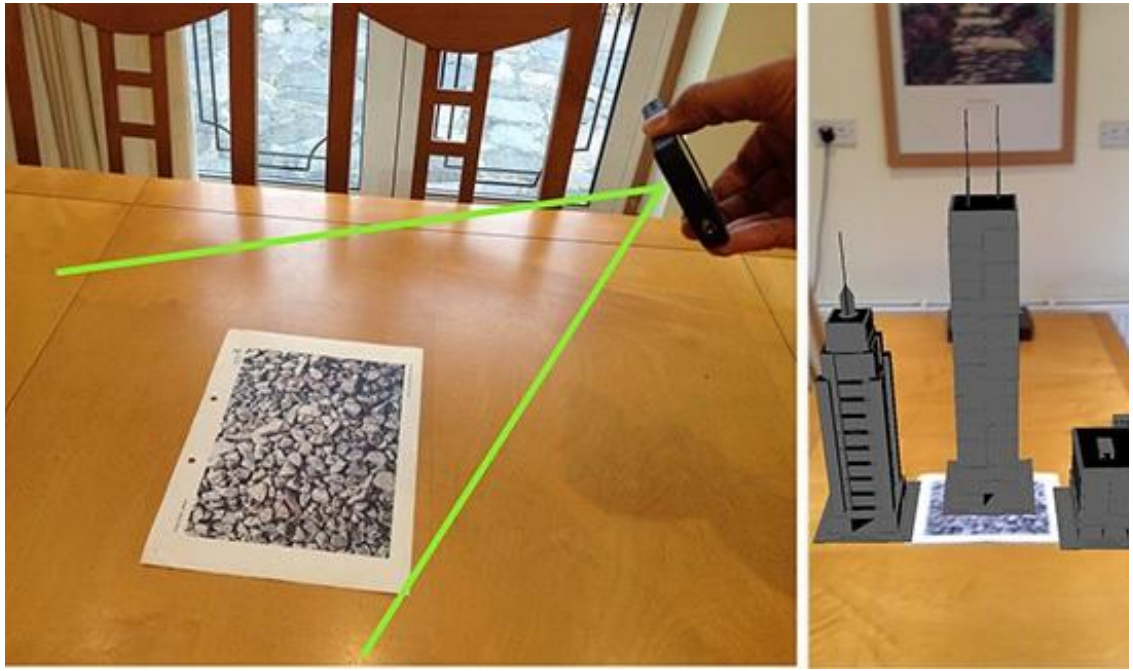

2:53
LTE
AR+
Pikachu / CP 31
64
3:23
AR+
Lapras / CP 1424
EXPERT HANDLER
63
3:16
AR+
Charizard / CP 818
63


RAISE YOUR EYEBROWS
Store
$0.99
$0.99
$0.99
1
1
1


0
9:28
1ST
3
1ST & 10
FOX SPORTS
1ST & 10
USA Football

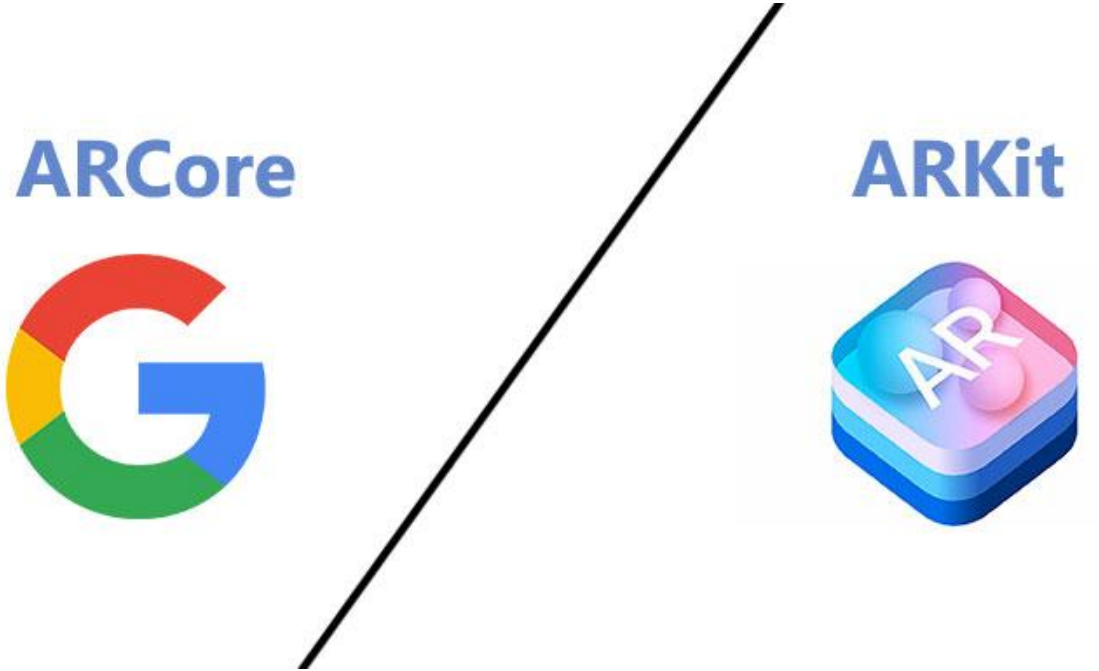
Implementations

Marker-based (Fiducial) AR



Head-mounted AR

Mobile AR



Apple ARKit

Google ARCore

How They Work(ish)

Examples

Cambria AR

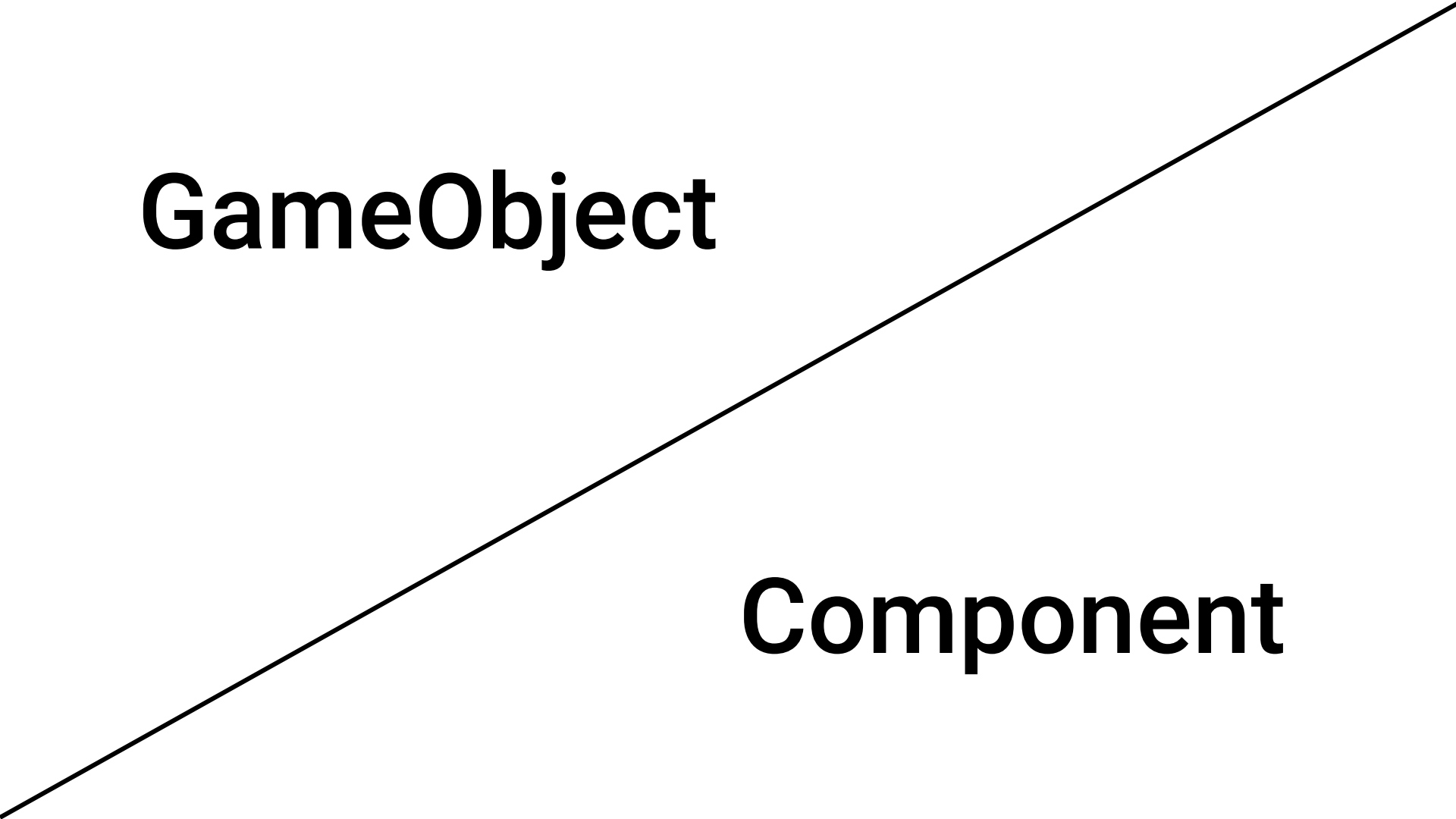
Primos SurroundView

Authoring AR Apps

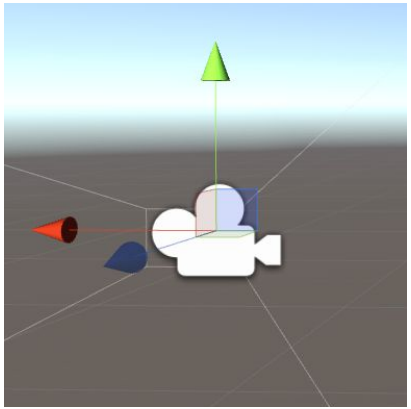
Unity

Quick Introduction

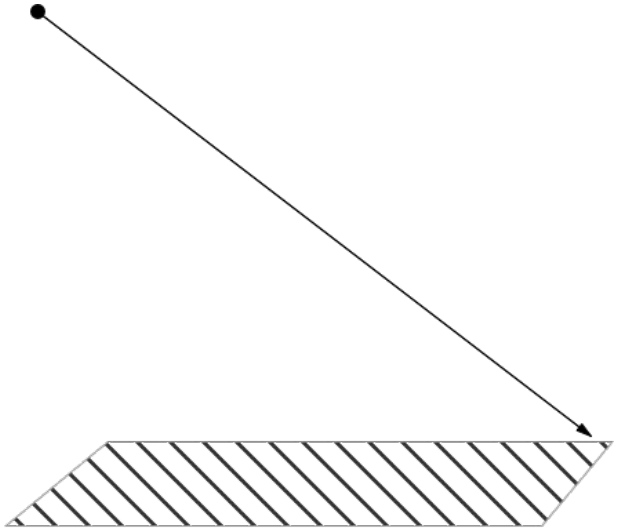
Terms


GameObject
Component

“Virtual” camera

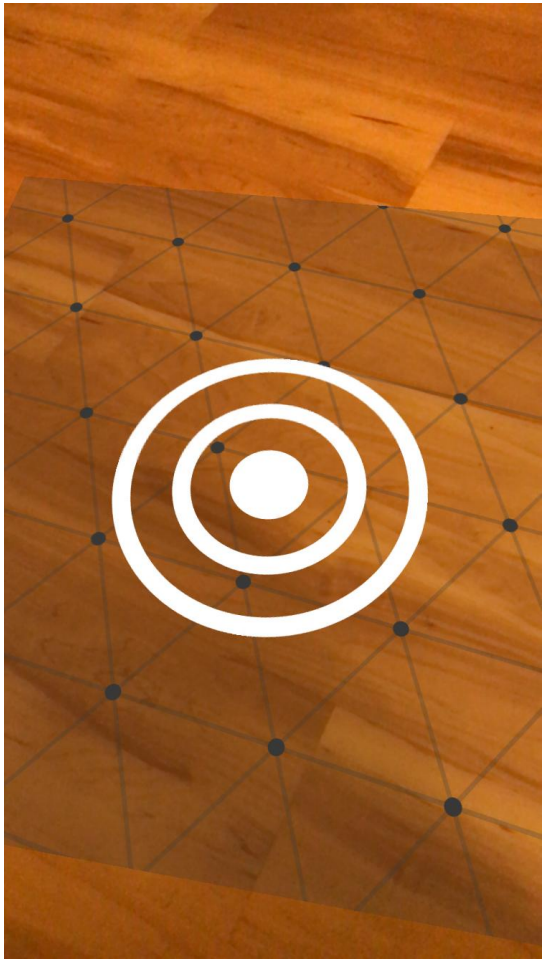


Raycasting



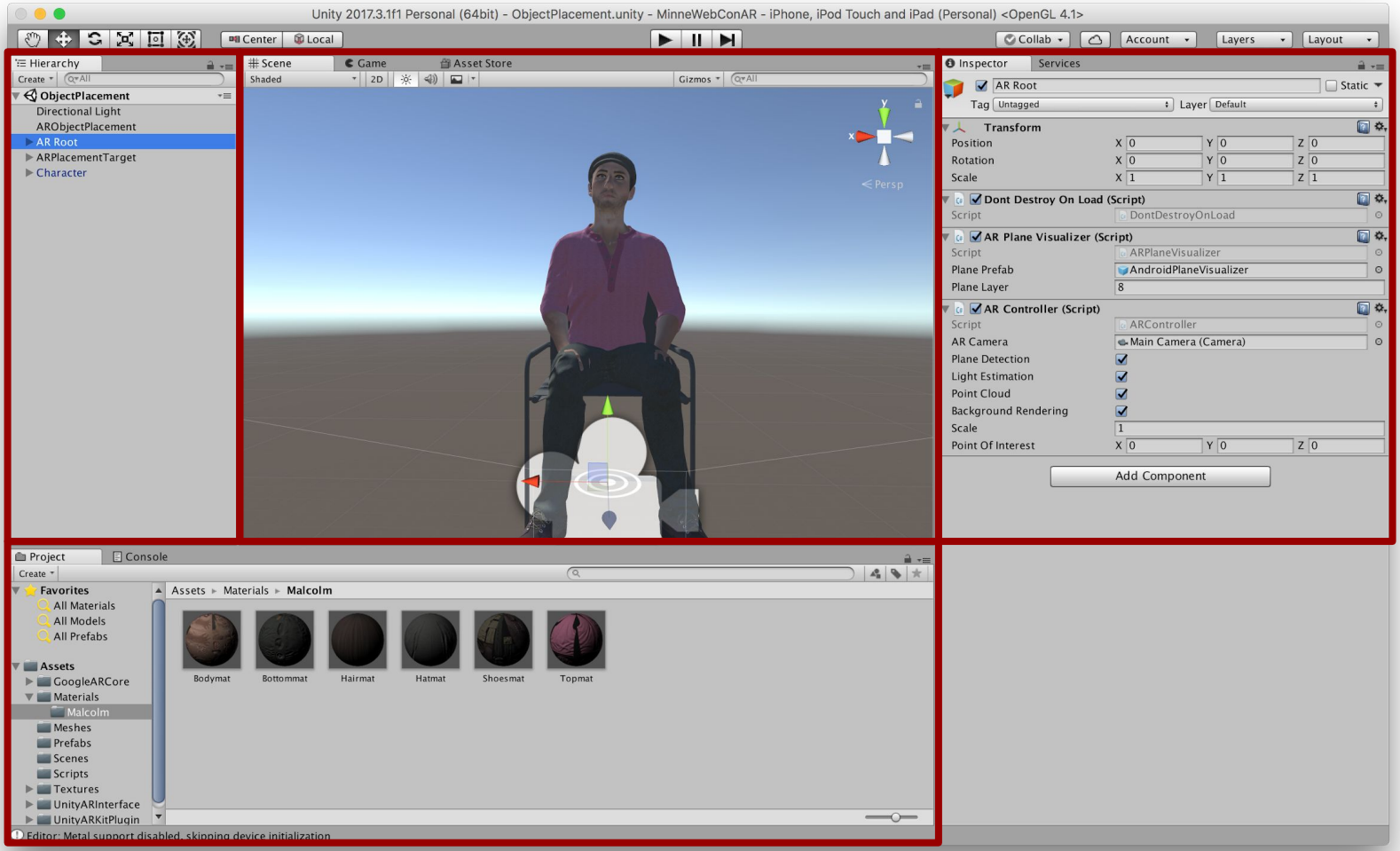
AR Feature Point

AR Plane



Unity ARInterface

Using Unity


Unity 2017.3.1f1 Personal (64bit) - ObjectPlacement.unity - MinneWebConAR - iPhone, iPod Touch and iPad (Personal) <OpenGL 4.1>
Center
Local
Collab
Account
Layers
Layout
Hierarchy
Create
ObjectPlacement
Directional Light
ARObjectPlacement
AR Root
ARPlacementTarget
Character
Scene
Game
Asset Store
Shaded
2D
Gizmos
Persp
Inspector
Services
AR Root
Static
Tag
Untagged
Layer
Default
Transform
Position
Rotation
Scale
Dont Destroy On Load (Script)
Script
DontDestroyOnLoad
AR Plane Visualizer (Script)
ARPlaneVisualizer
Plane Prefab
AndroidPlaneVisualizer
Plane Layer
AR Controller (Script)
ARController
AR Camera
Main Camera (Camera)
Plane Detection
Light Estimation
Point Cloud
Background Rendering
Point Of Interest
Add Component
Project
Console
Favorites
All Materials
All Models
All Prefabs
Assets
GoogleARCore
Materials
Malcolm
Meshes
Prefabs
Scenes
Scripts
Textures
UnityARInterface
UnityARKitPlugin
Assets ▸ Materials ▸ Malcolm
Bodymat
Bottommat
Hairmat
Hatmat
Shoesmat
Topmat
Editor: Metal support disabled, skipping device initialization

PointsAndPlanes

ARController.cs

```
0 references
123     private void OnEnable()
124     {
125         // Set global application settings
126         Application.targetFrameRate = 60;
127         Screen.sleepTimeout = SleepTimeout.NeverSleep;
128         Input.simulateMouseWithTouches = true;
129
130         if (m_ARInterface == null)
131             SetupARInterface();
132
133         // See if we are on a camera
134         if (m_ARCamera == null)
135             m_ARCamera = GetComponent<Camera>();
136
137         // Fallback to main camera
138         if (m_ARCamera == null)
139             m_ARCamera = Camera.main;
140
141         StopAllCoroutines();
142         StartCoroutine(StartServiceRoutine());
143
144     }
```

ARController.cs

```
0 references
172     void Update()
173     {
174         m_ARInterface.Update();
175     }
176
```

ARController.cs

```
102     // Before the next frame is rendered, we need to pose the camera.
        2 references
103     void OnBeforeRender()
104     {
105         m_ARInterface.UpdateCamera(m_ARCamera);
106
107         Pose pose = new Pose();
108         if (m_ARInterface.TryGetPose(ref pose))
109         {
110             m_ARCamera.transform.localPosition = pose.position;
111             m_ARCamera.transform.localRotation = pose.rotation;
112             var parent = m_ARCamera.transform.parent;
113             if (parent != null)
114                 parent.localScale = Vector3.one * scale;
115         }
116     }
117
```

ARPointCloudVisualizer.cs

```
26      // Use this for initialization
        0 references
27      void Start()
28      {
29          m_ParticleSystem = Instantiate(m_PointCloudParticlePrefab, GetRoot());
30          m_NoParticles = new ParticleSystem.Particle[1];
31          m_NoParticles[0].startSize = 0f;
32      }
```

ARPointCloudVisualizer.cs

```
34          // Update is called once per frame
            0 references
35          void Update()
36          {
37              if (ARInterface.GetInterface().TryGetPointCloud(ref m_PointCloud))
38              {
39                  var scale = GetScale();
40
41                  var numParticles = Mathf.Min(m_PointCloud.points.Count, m_MaxPointsToShow);
42                  if (m_Particles == null || m_Particles.Length != numParticles)
43                      m_Particles = new ParticleSystem.Particle[numParticles];
44
45                  // Iterate through all particles, adding color, scale, and position.
46                  for (int i = 0; i < numParticles; ++i)
47                  {
48                      m_Particles[i].position = m_PointCloud.points[i] * scale;
49                      m_Particles[i].startColor = new Color(1.0f, 1.0f, 1.0f);
50                      m_Particles[i].startSize = m_ParticleSize * scale;
51                  }
52
53                  m_ParticleSystem.SetParticles(m_Particles, numParticles);
54              }
55              else
56              {
57                  m_ParticleSystem.SetParticles(m_NoParticles, 1);
58              }
59          }
60      }
61  }
```

ARPlaneVisualizer.cs

```
        4 references
17      private Dictionary<string, GameObject> m_Planes = new Dictionary<string, GameObject>();
18
        0 references
19      void OnEnable()
20      {
21          m_PlaneLayer = LayerMask.NameToLayer ("ARGameObject");
22
23          // We want ARInterface to tell us when it detects a plane, so we use the following event handlers:
24          ARInterface.planeAdded += PlaneAddedHandler;
25          ARInterface.planeUpdated += PlaneUpdatedHandler;
26          ARInterface.planeRemoved += PlaneRemovedHandler;
27      }
28
        0 references
29      void OnDisable()
30      {
31          ARInterface.planeAdded -= PlaneAddedHandler;
32          ARInterface.planeUpdated -= PlaneUpdatedHandler;
33          ARInterface.planeRemoved -= PlaneRemovedHandler;
34      }
35
```

PlacementTarget

ARPlacementTarget.cs

```
13      // Use this for initialization
        0 references
14      void Start () {
15          // Make our layermask that we'll use in a raycast to only check for collisions against a certain layer
16          // Don't forget to add this layer to Unity layer list in editor!
17          // https://answers.unity.com/questions/8715/how-do-i-use-layermasks.html
18          int layerIndex = LayerMask.NameToLayer("ARGameObject");
19          collisionLayerMask = 1 << layerIndex;
20
21          // Start the GameObject as hidden if we provided one
22          if (placementIndicator != null) {
23              placementIndicator.SetActive(false);
24          }
25      }
26
```

ARPlacementTarget.cs

```
26
27      // Update is called once per frame
        0 references
28      void Update () {
29
30          // This is a point in the center of the screen (z is ignored for ScreenPointToRay)
31          Vector3 center = new Vector3(Screen.width / 2f, Screen.height / 2f, 0.0f);
32
33          // Create a ray from the camera through the screen x/y pixel coordinates (z is ignored)
34          Ray ray = Camera.main.ScreenPointToRay(center);
35          // `hit` will hold the results of the raycast collision if there is one
36          RaycastHit hit;
37
38          // Perform the raycast using our ray, and checking for collisions against ARGameObject layer
39          if (Physics.Raycast(ray, out hit, 30.0f, collisionLayerMask)) {
40
41              // Update the position, rotation and visibility of the GameObject if we provided one
42              if (placementIndicator != null) {
43                  placementIndicator.transform.position = hit.point;
44                  placementIndicator.transform.rotation = hit.transform.rotation;
45                  placementIndicator.SetActive(true);
46              }
47          } else {
48              // Update the visibility of the GameObject if we provided one
49              if (placementIndicator != null) {
50                  placementIndicator.SetActive(false);
51              }
52          }
53      }
54  }
```

ObjectPlacement

ARObjectPlacement.cs

```
20    void Update () {
21        // if touch, then update with TryPlaceObject
22        if (Input.touchCount > 0 && Input.GetTouch(0).phase == TouchPhase.Ended)
23            {
24                TryPlaceObject();
25            }
26
27    }
      1 reference
28    void TryPlaceObject () {
29        // This is a point in the center of the screen (z is ignored for ScreenPointToRay)
30        Vector3 center = new Vector3(Screen.width / 2f, Screen.height / 2f, 0.0f);
31        // Create a ray from the camera through the screen x/y pixel coordinates (z is ignored)
32        Ray ray = Camera.main.ScreenPointToRay(center);
33        // `hit` will hold the results of the raycast collision if there is one
34        RaycastHit hit;
35
36        // Perform the raycast using our ray, and checking for collisions against ARGameObject layer
37        if (Physics.Raycast(ray, out hit, 30.0f, collisionLayerMask)) {
38            GameObject obj = Instantiate(objectPrefab, transform);
39            // Set the object's position to that of the raycast hit
40            obj.transform.position = hit.point;
41            // Update the rotation to "look at" the camera (so it's always "facing you")
42            obj.transform.LookAt(Camera.main.transform.position);
43            // Use the hit point's x and z axis rotation so that the object is oriented logically
44            obj.transform.eulerAngles = new Vector3(hit.transform.rotation.x, obj.transform.eulerAngles.y, hit.transform.rotation.z);
45        }
46    }
47 }
```

THE FUTURE: WebAR

Questions?

Thanks

Brandon is @brandon_mn on Twitter and elsewhere.
Max is @thorsonmscott on GitHub and nowhere else.

Catch us milling about if you'd like to chat!

Resources

- ARCore and ARKit platform documentation
- Get Unity from unity3d.com
- Unity modules used here include: XR, Input, Touch
- Unity's ARInterface repository and example project
- WebAR documentation, example article, and ARKit/ARCore projects
- Amazon Sumerian's AR documentation