

# UNITY AND AUGMENTED REALITY

---



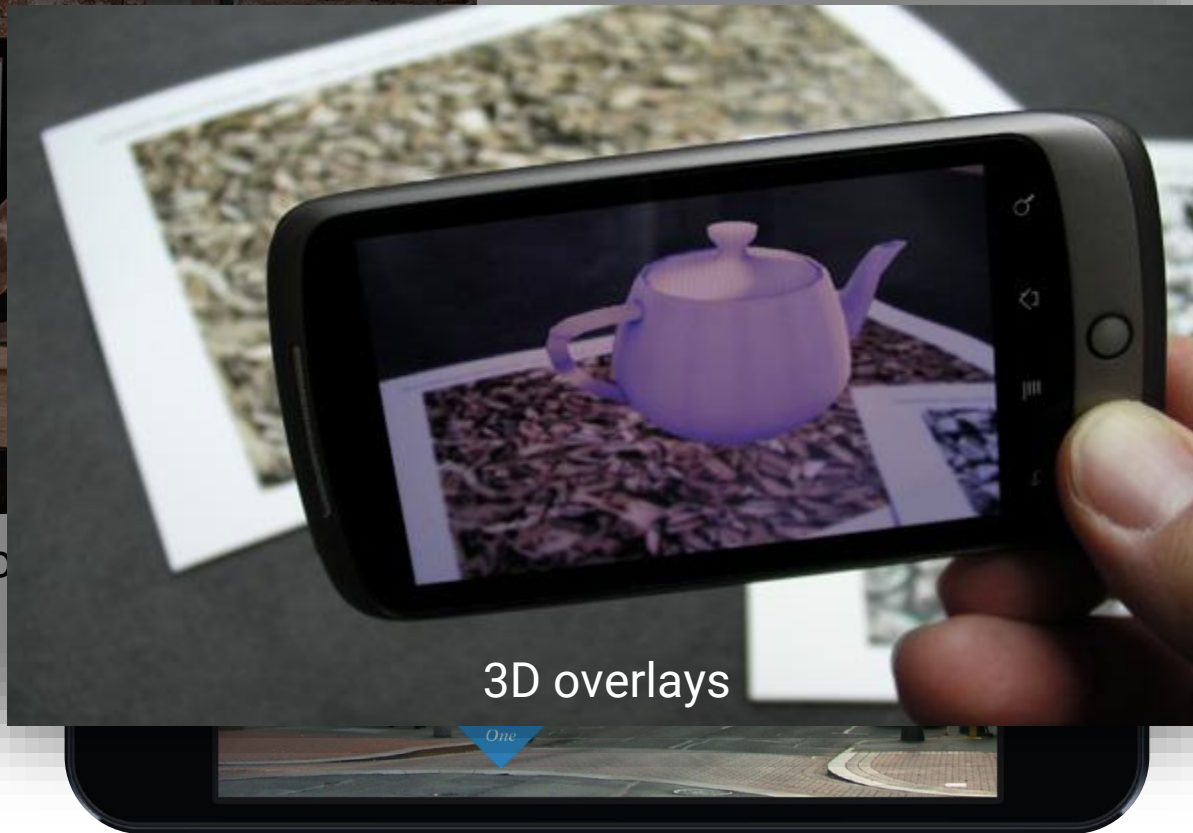
vuforia™

# AUGMENTED REALITY

## Examples



Holo

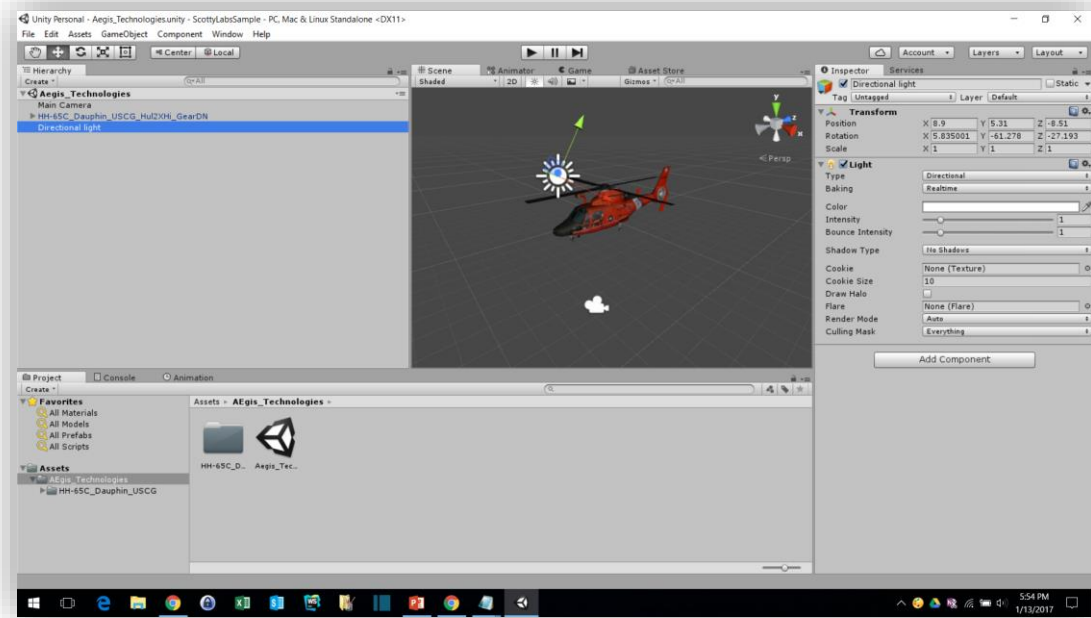


3D overlays

2D overlays



Oculus Rift  
(not AR)



- 3D Game Engine
- Renders 3D graphics, particle effects, animations
- Built in physics engine
- Can export to Android, iOS, OS X, Windows, etc.



# vuforia™



- Unity asset package for AR
- Uses physical tracking targets to position virtual objects
- Targets can be pre-determined or user-defined
- Uses image “features” for recognition

Google



Play Store



Maps



Gmail



Dropbox



Calculator



Photos



Automate



Camera



YouTube



Rocker Locker



Flipt



ScottyLabs Pri...



Default page



# Demo

<https://youtu.be/g9U9WKgqPLQ>



# **TUTORIAL 1**

## **Configuring Unity and Vuforia**

**9:35 AM**



# SCRIPTS

```
1  using UnityEngine;
2  using System.Collections;
3
4  public class spin : MonoBehaviour {
5
6      // Use this for initialization
7      void Start () {
8
9      }
10
11     // Update is called once per frame
12     void Update () {
13
14     }
15 }
16
```

- Unity Scripts can be written in C# or JavaScript
- Scripts are attached to GameObjects (things in the scene)
- Each script has a Start function (called once) and an Update function (called every frame)



# TRANSFORM

```
1  using UnityEngine;
2  using System.Collections;
3
4  public class spin : MonoBehaviour {
5
6      // Use this for initialization
7      void Start () {
8          // local movement
9          gameObject.transform.localPosition = new Vector3(1, 2, 3);
10         gameObject.transform.localScale = new Vector3(1, 2, 0.5f);
11
12         // movement relative to world
13         gameObject.transform.position = new Vector3(1, 2, 3);
14
15         // vectors can be added, subtracted, and scaled
16         Vector3 v = new Vector3(4, 5, 6) * 2 + new Vector3(7, 8, 9);
17     }
18 }
```

- GameObjects can be translated, rotated and scaled using `gameObject.transform`
- Movement can be relative to screen or parent object
- Movement can be combined with `Time.deltaTime` for smooth movement

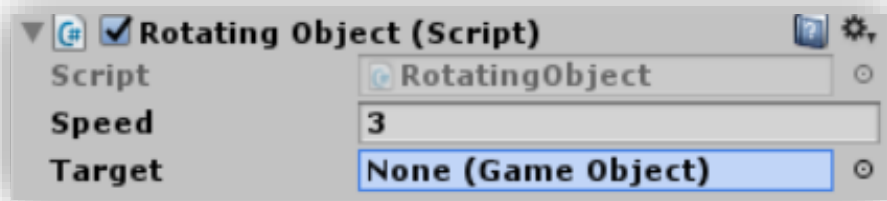
# TUTORIAL 2

## Moving GameObjects

9:35 AM



# SCRIPT PARAMETERS



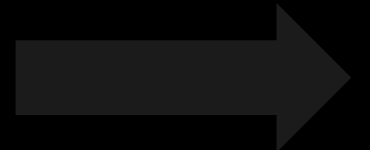
```
1  using UnityEngine;
2  using System.Collections;
3
4  public class RotatingObject : MonoBehaviour {
5      public float speed = 3;
6      public GameObject target;
7      // Use this for initialization
8      void Start () {
9
10     }
```

- Scripts can have options which show up in Unity's inspector pane. Simply make a public variable
- References to GameObjects and other scripts can also be passed by dragging that object to the field in the inspector
- We can then call functions inside the other script

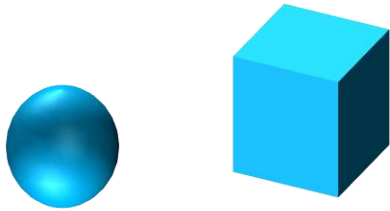
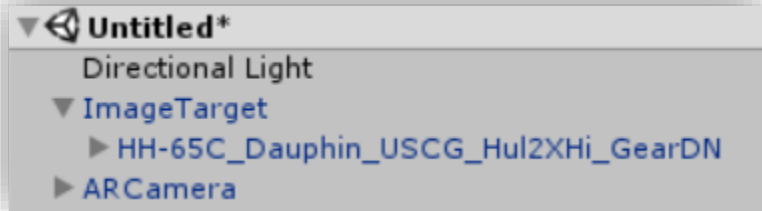
# **TUTORIAL 3**

## **Calling scripts from other scripts**

**9:35 AM**

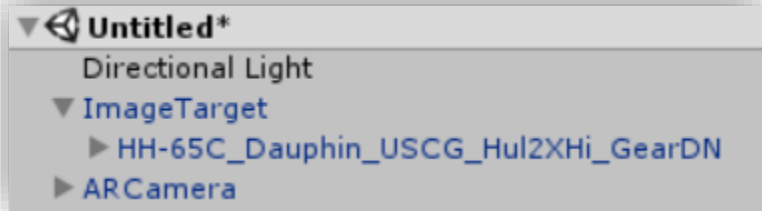


# OBJECT HIERARCHY



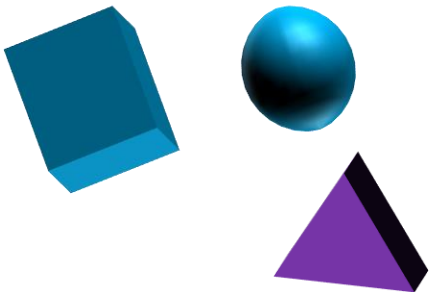
- GameObjects move with parent objects
- A GameObject can change its parent with:  
`gameObject.transform.parent=[newparent].transform`
- ImageTargets automatically move, show, and hide their children when a target is detected

# OBJECT HIERARCHY

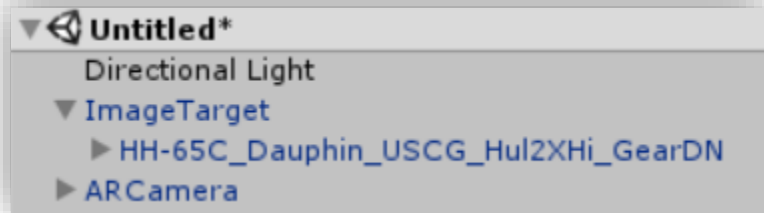


- GameObjects move with parent objects
- A GameObject can change its parent with:  

```
gameObject.transform.parent=[newparent].transform
```
- ImageTargets automatically move, show, and hide their children when a target is detected

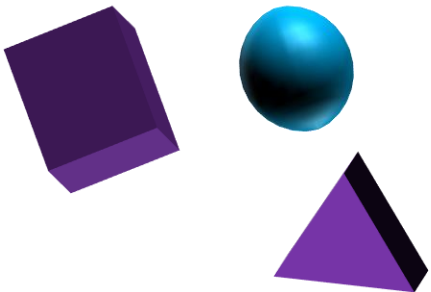


# OBJECT HIERARCHY

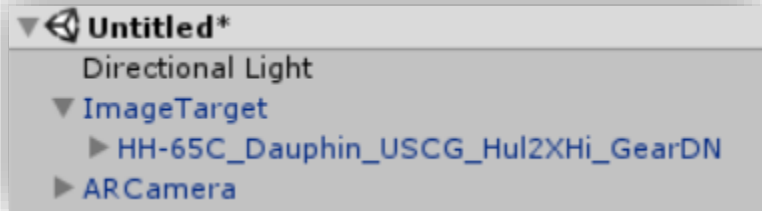


- GameObjects move with parent objects
- A GameObject can change its parent with:  

```
gameObject.transform.parent=[newparent].transform
```
- ImageTargets automatically move, show, and hide their children when a target is detected

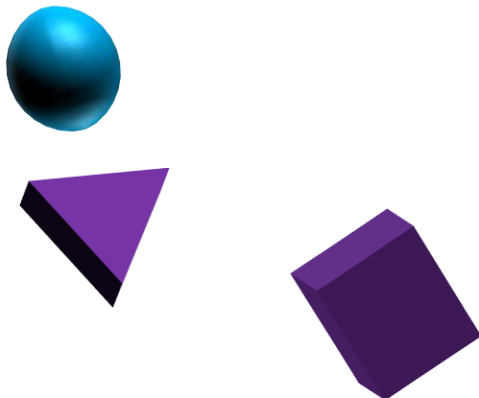


# OBJECT HIERARCHY



- GameObjects move with parent objects
- A GameObject can change its parent with:  

```
gameObject.transform.parent=[newparent].transform
```
- ImageTargets automatically move, show, and hide their children when a target is detected

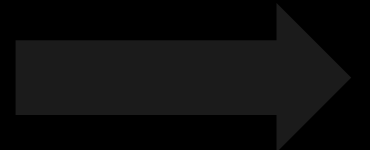




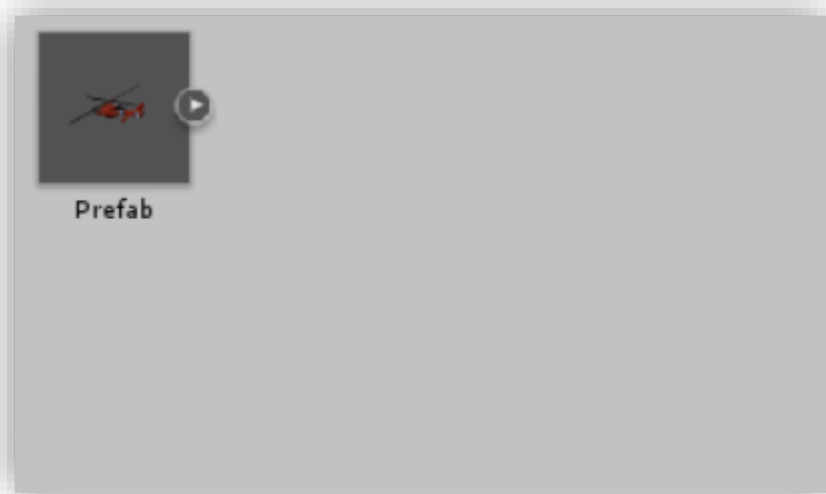
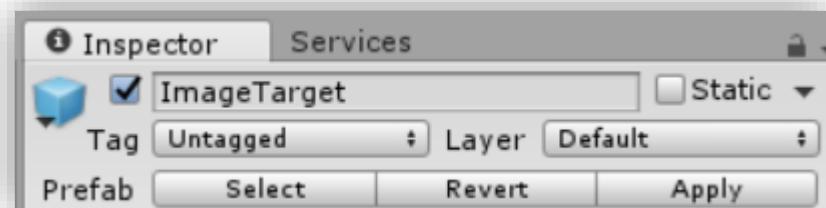
# **TUTORIAL 4**

## **Anchoring to the camera**

**9:35 AM**



# PREFABS

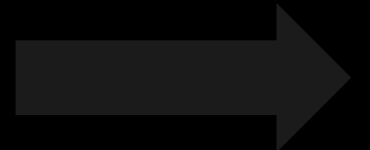


- Prefabs are collections of GameObjects, scripts, and other parts which are grouped together as one unit for reuse
- Modifying one instance of a prefab can be used to change all the others

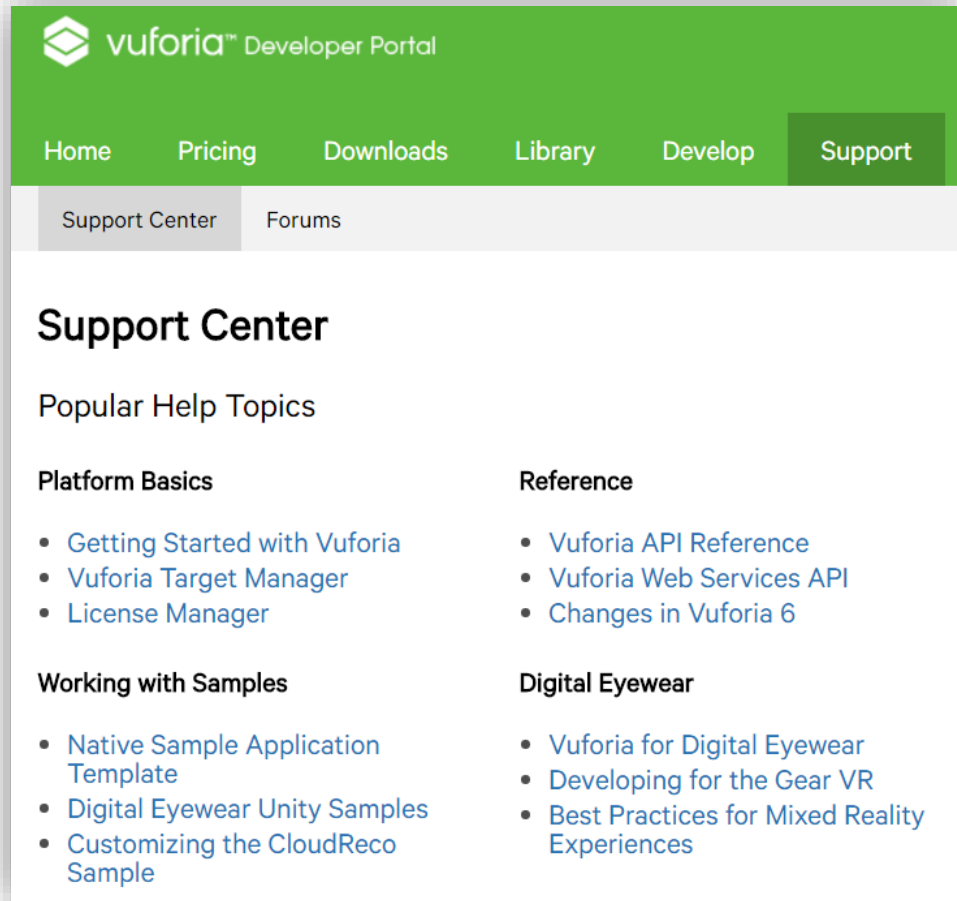
# TUTORIAL 5

## Saving Prefabs

9:35 AM



# OTHER RESOURCES



The screenshot shows the Vuforia Developer Portal's Support Center. The page has a green header with the Vuforia logo and the text 'vuforia™ Developer Portal'. Below the header is a navigation menu with links for Home, Pricing, Downloads, Library, Develop, and Support. The Support link is highlighted. Underneath the navigation menu, there are two sub-links: Support Center and Forums. The main content area is titled 'Support Center' and features a section for 'Popular Help Topics'. This section is divided into four columns: Platform Basics, Reference, Working with Samples, and Digital Eyewear. Each column contains a list of links to various help topics.

**vuforia™ Developer Portal**

Home Pricing Downloads Library Develop **Support**

Support Center Forums

## Support Center

Popular Help Topics

**Platform Basics**

- [Getting Started with Vuforia](#)
- [Vuforia Target Manager](#)
- [License Manager](#)

**Reference**

- [Vuforia API Reference](#)
- [Vuforia Web Services API](#)
- [Changes in Vuforia 6](#)

**Working with Samples**

- [Native Sample Application Template](#)
- [Digital Eyewear Unity Samples](#)
- [Customizing the CloudReco Sample](#)

**Digital Eyewear**

- [Vuforia for Digital Eyewear](#)
- [Developing for the Gear VR](#)
- [Best Practices for Mixed Reality Experiences](#)

- Just Google it!
- [docs.unity3d.com/ScriptReference/](https://docs.unity3d.com/ScriptReference/)
- [developer.vuforia.com/support](https://developer.vuforia.com/support)