TimAR

An AR powered watch trial app

Bandaru Hrithik	17XJ1A0508
Piyush Raj Pandey	17XJ1A0534
Supritha Reddy	17XJ1A0556
Viswa Janith	17XJ1A0565

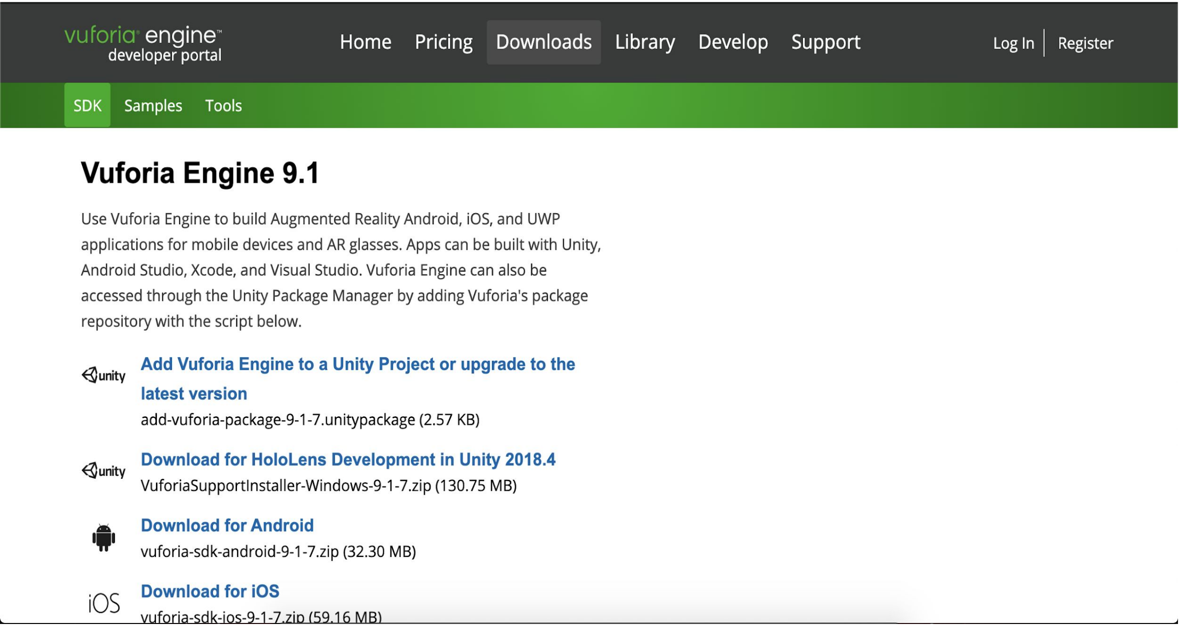
Introduction

What is TimAR?

How is TimAR different from other applications available online right now?

Current limitations of TimAR.

Database and image targets



Vuforia Engine 9.1

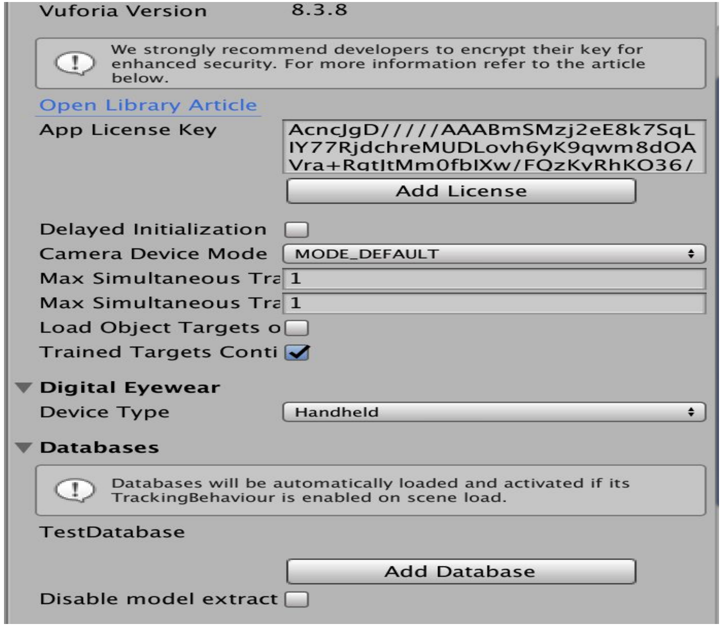
Use Vuforia Engine to build Augmented Reality Android, iOS, and UWP applications for mobile devices and AR glasses. Apps can be built with Unity, Android Studio, Xcode, and Visual Studio. Vuforia Engine can also be accessed through the Unity Package Manager by adding Vuforia's package repository with the script below.

unity **Add Vuforia Engine to a Unity Project or upgrade to the latest version**
add-vuforia-package-9-1-7.unitypackage (2.57 KB)

unity **Download for HoloLens Development in Unity 2018.4**
VuforiaSupportInstaller-Windows-9-1-7.zip (130.75 MB)

Download for Android
vuforia-sdk-android-9-1-7.zip (32.30 MB)

iOS **Download for iOS**
vuforia-sdk-ios-9-1-7.zip (59.16 MB)



Vuforia Version 8.3.8

We strongly recommend developers to encrypt their key for enhanced security. For more information refer to the article below.

Open Library Article

App License Key AcncJgD/////AAABmSMzj2eE8k7SqL IY77RjdchreMUDLovh6yK9qwm8dOA Vra+RqtItMm0fbIXw/FQzKvRhKO36/

Add License

Delayed Initialization □
Camera Device Mode MODE_DEFAULT
Max Simultaneous Tra 1
Max Simultaneous Tra 1
Load Object Targets o □
Trained Targets Conti ☑

Digital Eyewear
Device Type Handheld

Databases

Databases will be automatically loaded and activated if its TrackingBehaviour is enabled on scene load.

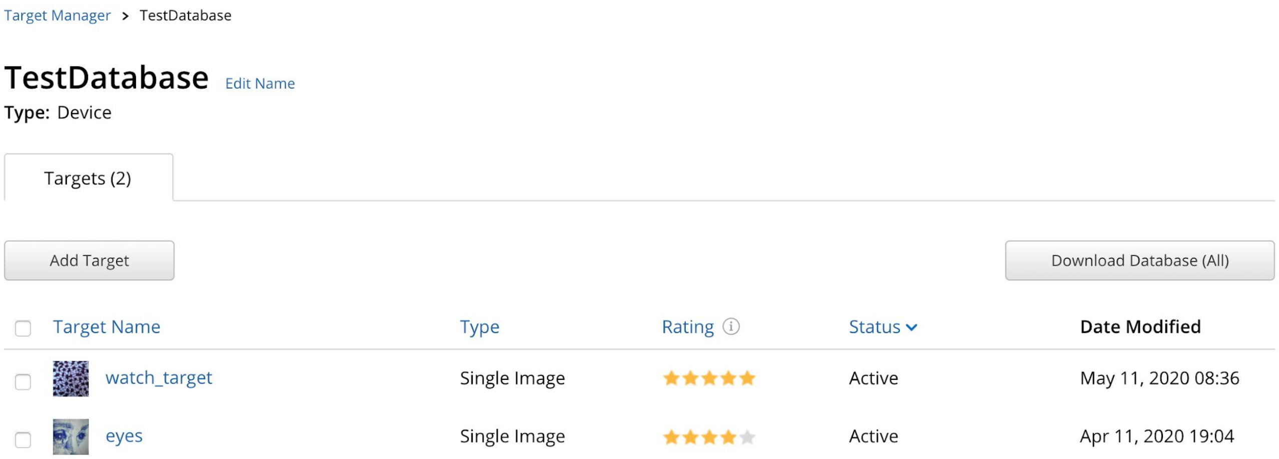
TestDatabase

Add Database

Disable model extract □

What is Vuforia?
How do we connect Vuforia and Unity?

Database and the package



Target Manager > TestDatabase

TestDatabase

Edit Name

Type: Device

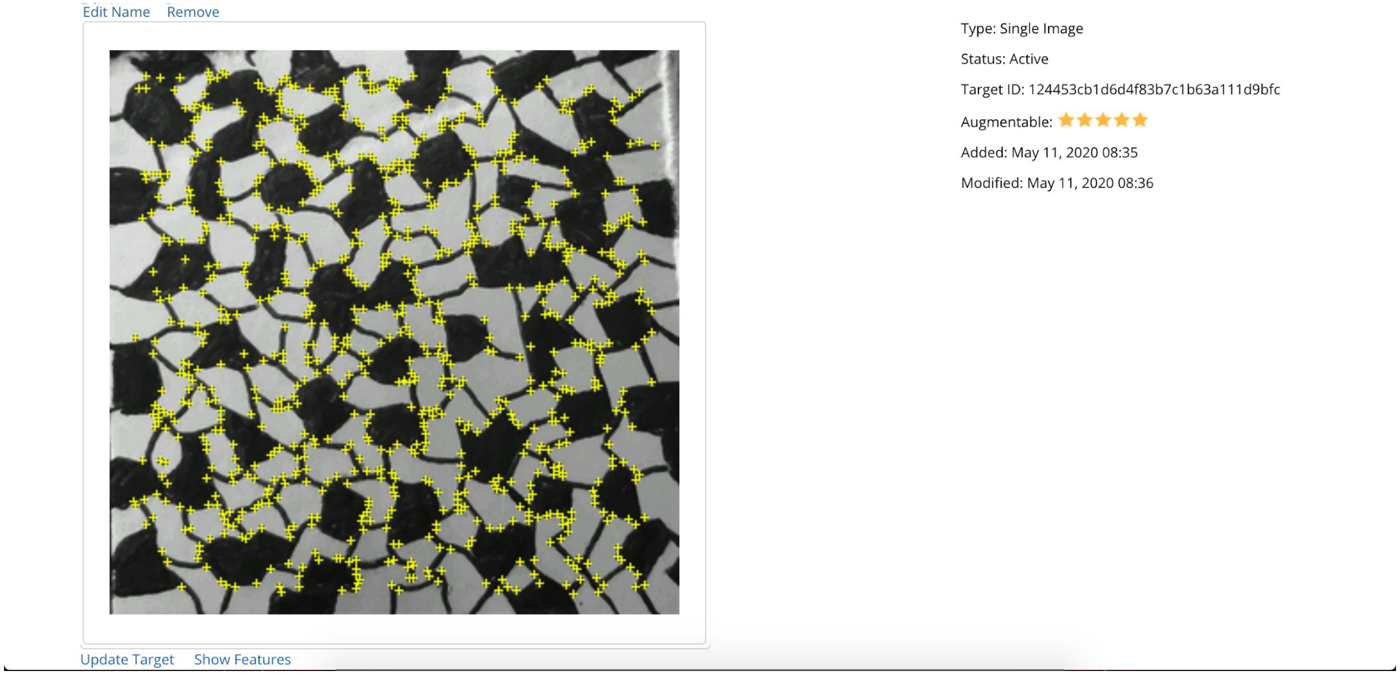
Targets (2)

Add Target

Download Database (All)

□	Target Name	Type	Rating	Status	Date Modified
□	watch_target	Single Image	★★★★★	Active	May 11, 2020 08:36
□	eyes	Single Image	★★★★☆	Active	Apr 11, 2020 19:04

Image Target



Type: Single Image

Status: Active

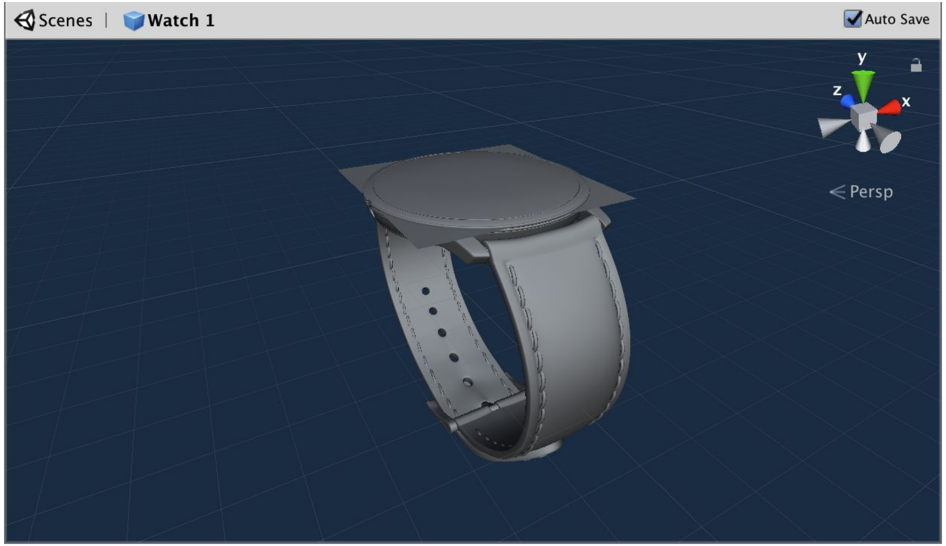
Target ID: 124453cb1d6d4f83b7c1b63a111d9bfc

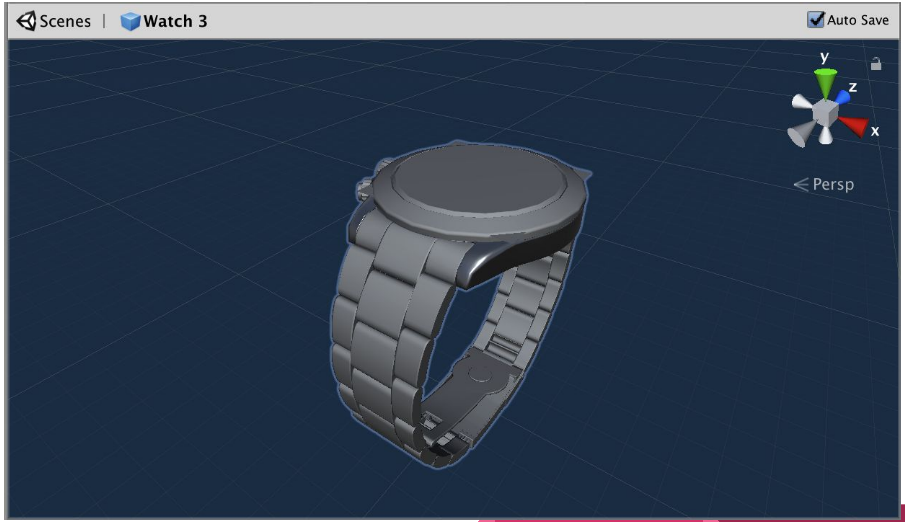
Augmentable: ★★★★★

Added: May 11, 2020 08:35

Modified: May 11, 2020 08:36

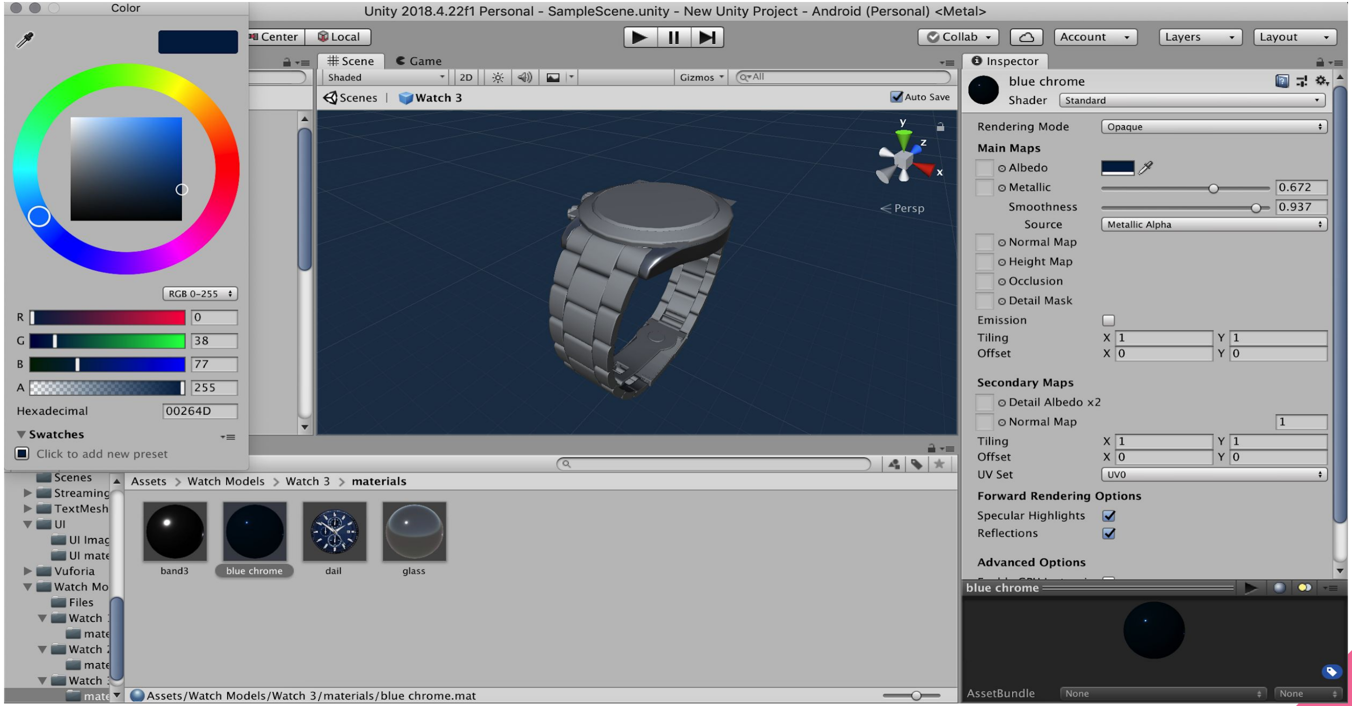
Adding and Setting up Watch Models





A 3D watch models when imported into unity looks like this, all the materials, reflections and textures will be lost.

-We can use any colour and material for the mentioned parts of the watch to make the model.


Unity 2018.4.22f1 Personal - SampleScene.unity - New Unity Project - Android (Personal) <Metal>

Color

RGB 0-255

R 0

G 38

B 77

A 255

Hexadecimal 00264D

Swatches

Click to add new preset

Inspector

blue chrome

Shader Standard

Rendering Mode Opaque

Main Maps

Albedo

Metallic 0.672

Smoothness 0.937

Source Metallic Alpha

Normal Map

Height Map

Occlusion

Detail Mask

Emission □

Tiling X 1 Y 1

Offset X 0 Y 0

Secondary Maps

Detail Albedo x2

Normal Map 1

Tiling X 1 Y 1

Offset X 0 Y 0

UV Set UV0

Forward Rendering Options

Specular Highlights ☑

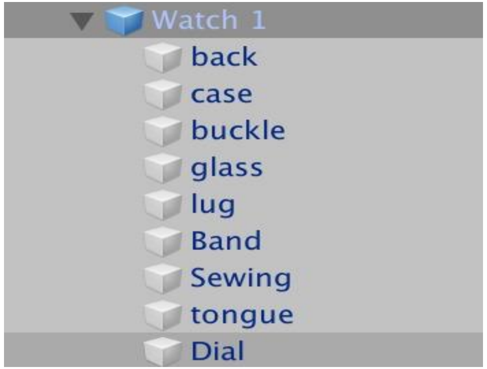
Reflections ☑

Advanced Options

AssetBundle None None

Assets > Watch Models > Watch 3 > materials

Assets/Watch Models/Watch 3/materials/blue chrome.mat


Watch 1

back

case

buckle

glass

lug

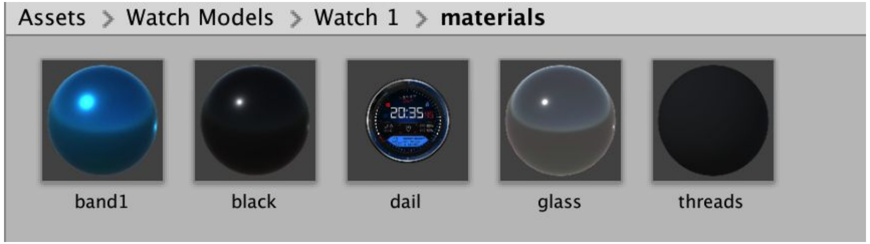
Band

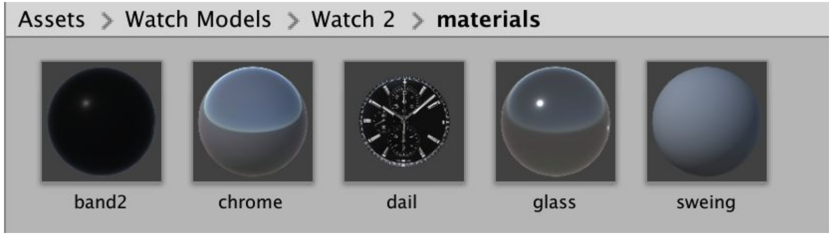
Sewing

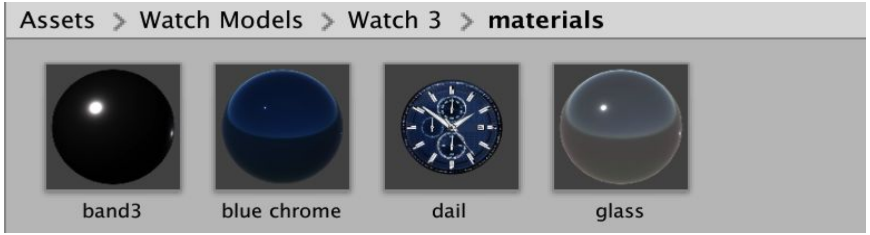
tongue

Dial

Created materials can be applied to the different parts watches.

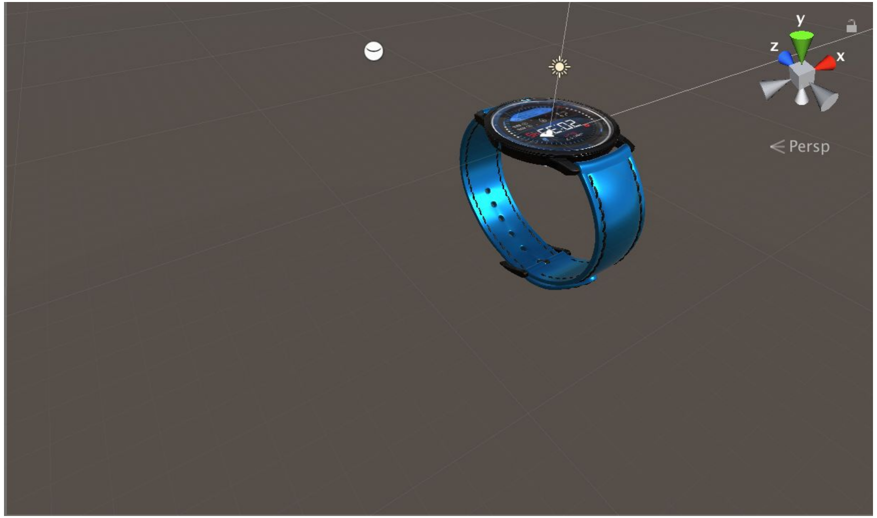


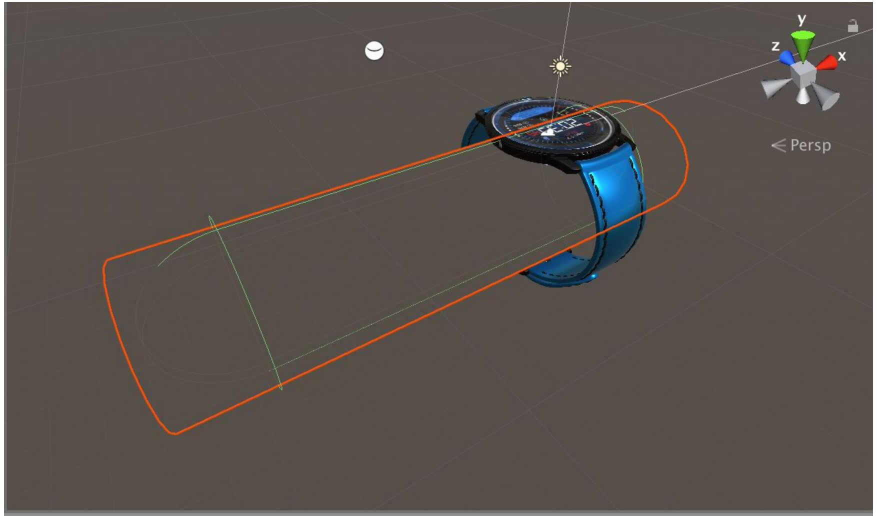




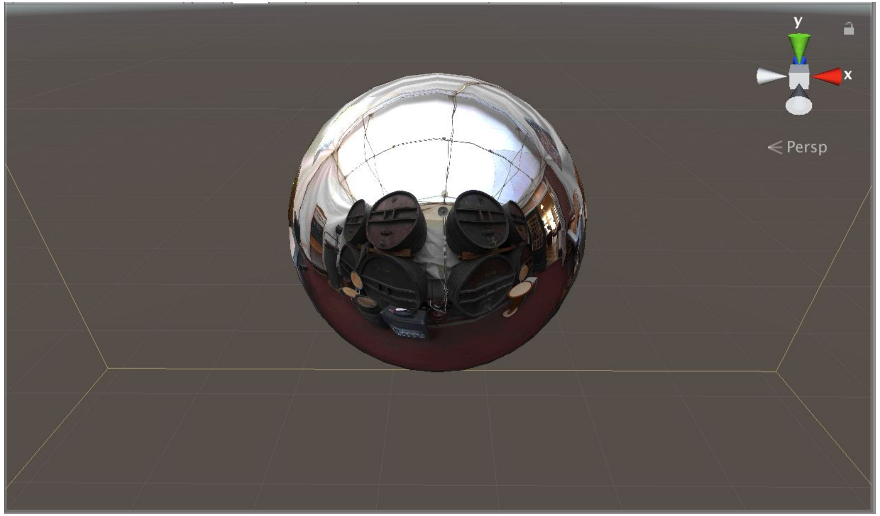
Jan 17
Sun
20:35
45

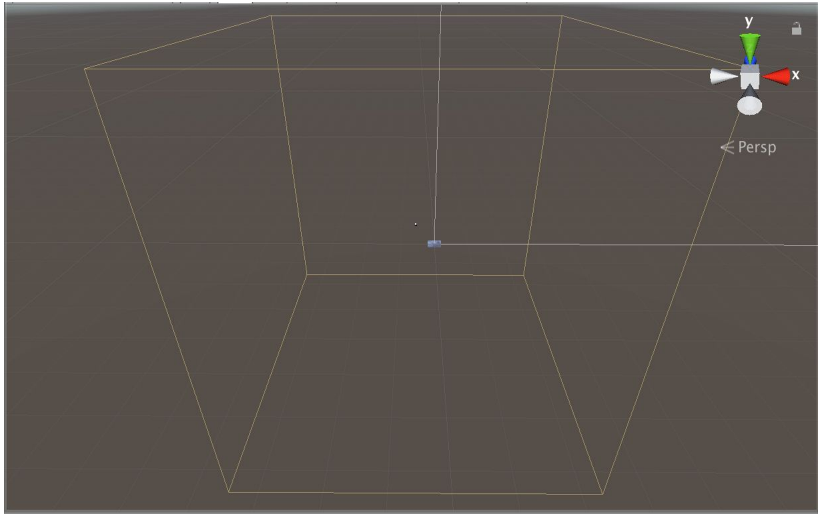
Occlusion





Probe lighting and Reflections



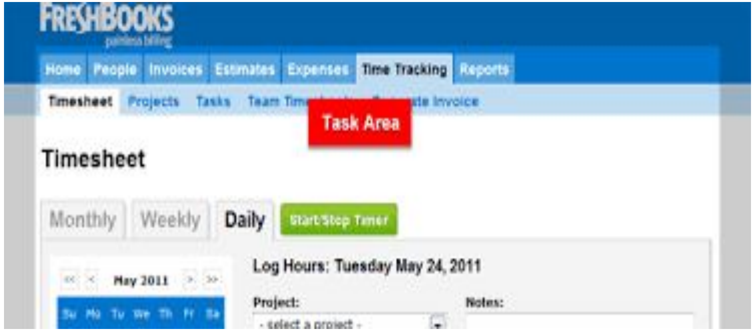


Introduction of UI

-User interface (UI) design is the process of making interfaces in software or computerized devices with a focus on looks or style.

-UI design typically refers to graphical user interfaces.

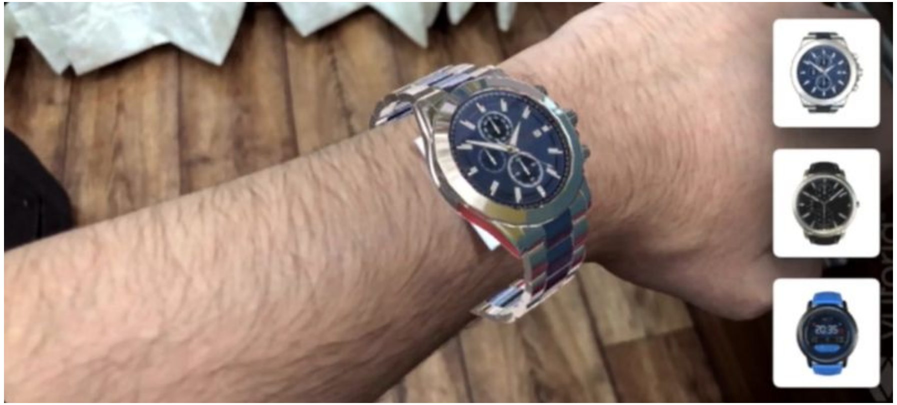
-The aim is to users find easy to use and pleasurable.

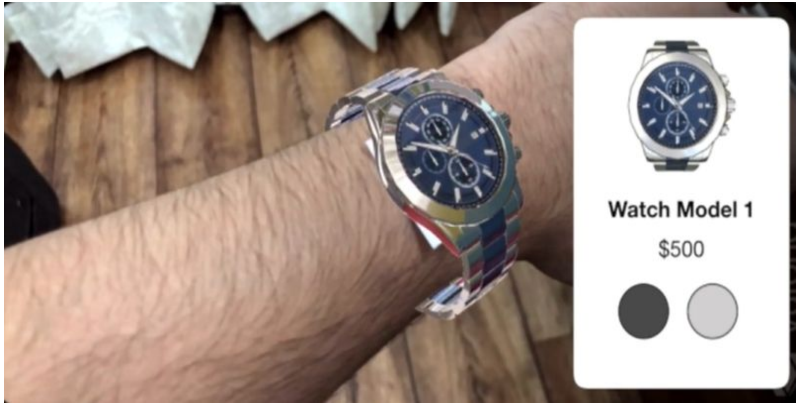


User Interface in UNITY

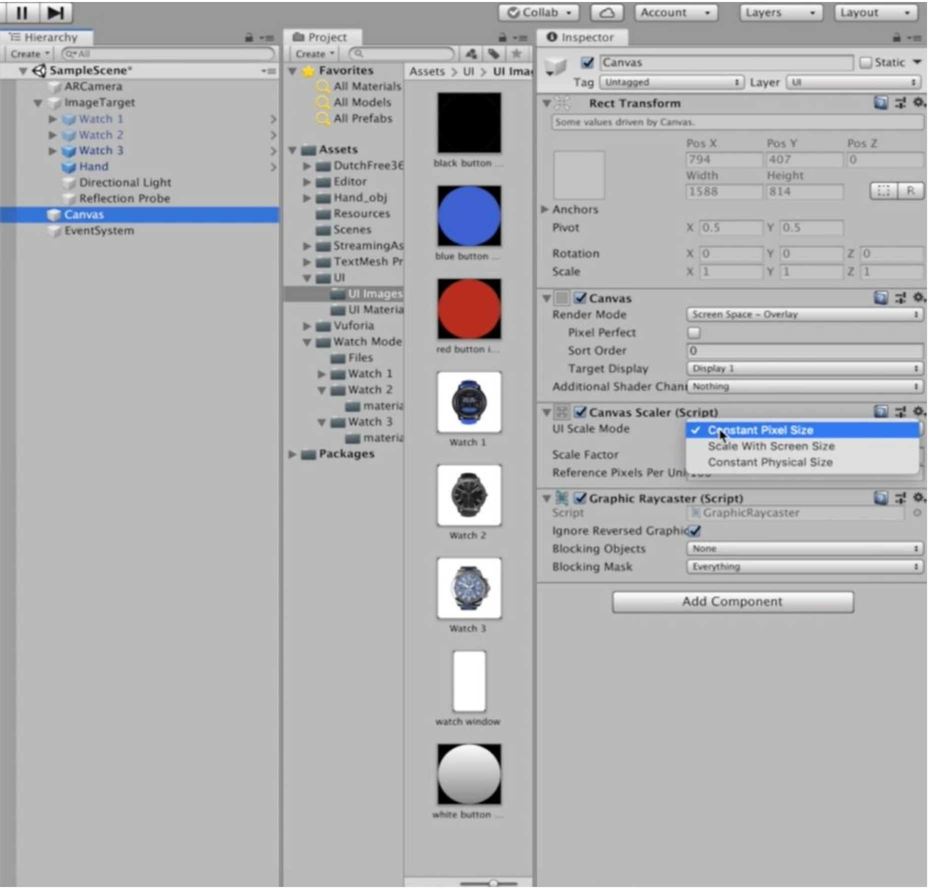
- UI components
- Canvas: The **Canvas** is the area that all UI elements should be inside. The Canvas is a Game Object with a Canvas component on it, and all UI elements must be children of such a Canvas.
- The Canvas has a **Render Mode** setting which can be used to make it render in screen space or world space.

CREATING USER INTERFACE





Part 1



Inspector

☑ Canvas ☐ Static

Tag Untagged Layer UI

Rect Transform

Some values driven by Canvas.

Pos X	Pos Y	Pos Z
794	407	0
Width	Height	
1588	814	

Anchors

Pivot X 0.5 Y 0.5

Rotation X 0 Y 0 Z 0

Scale X 1 Y 1 Z 1

☑ Canvas

Render Mode: Screen Space – Overlay

Pixel Perfect: ☐

Sort Order: 0

Target Display: Display 1

Additional Shader Chann: Nothing

☑ Canvas Scaler (Script)

UI Scale Mode: ✓ Constant Pixel Size / Scale With Screen Size / Constant Physical Size

Scale Factor

Reference Pixels Per Uni

☑ Graphic Raycaster (Script)

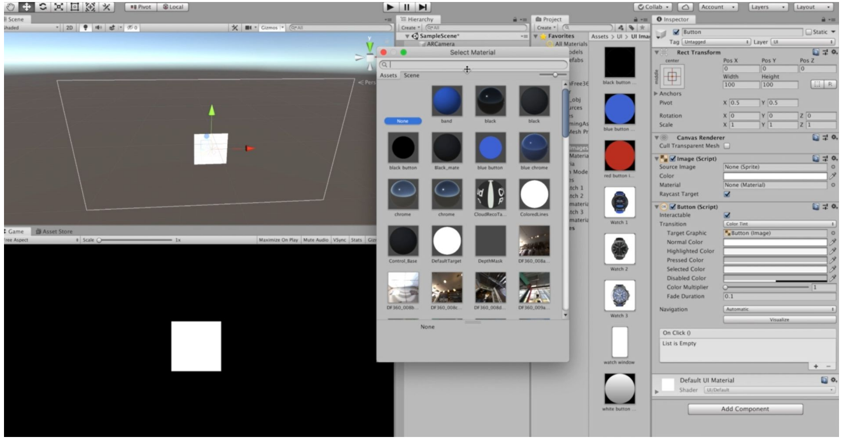
Script: GraphicRaycaster

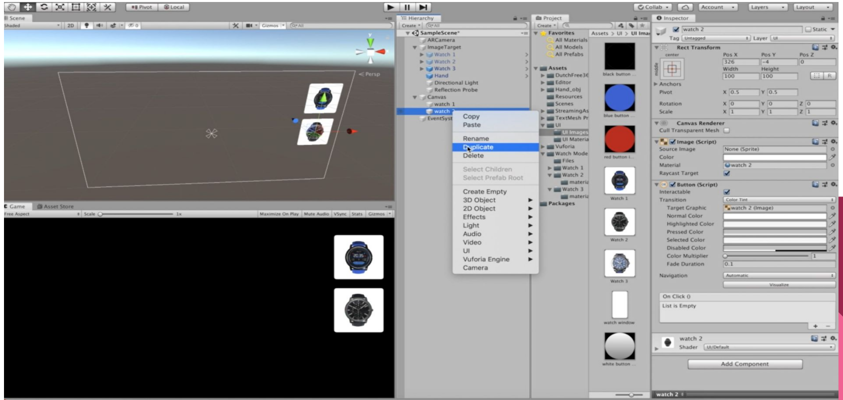
Ignore Reversed Graphic ☑

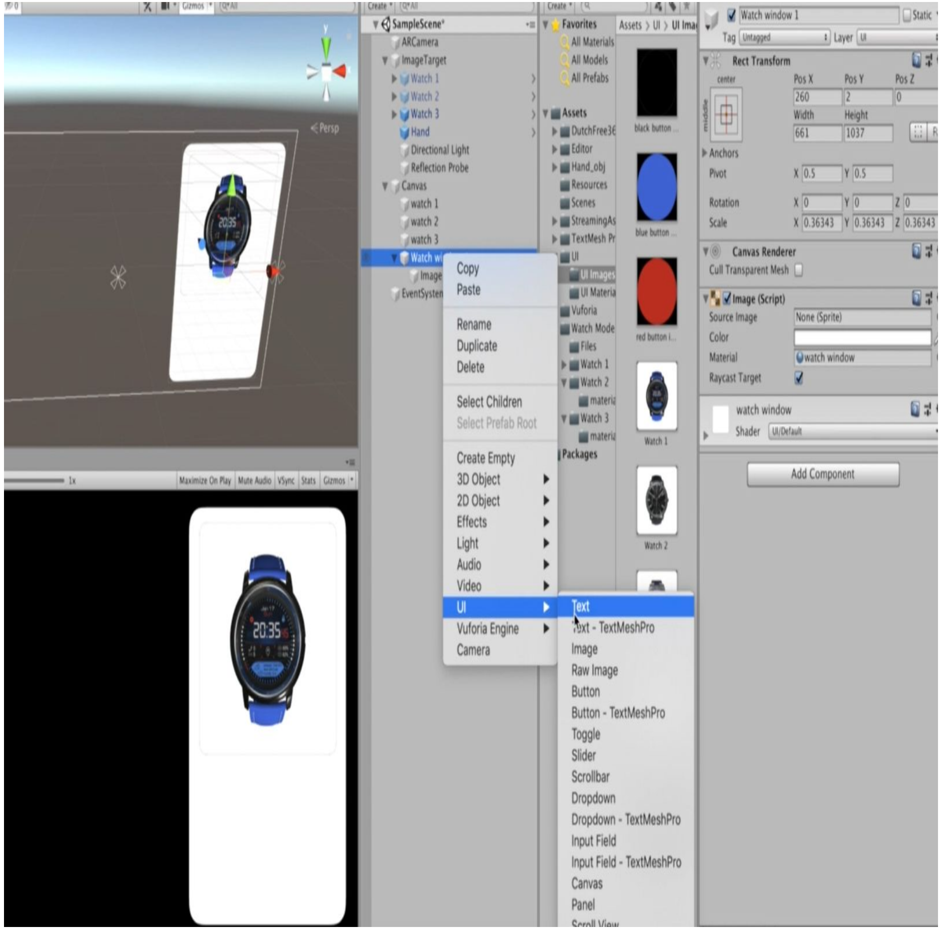
Blocking Objects: None

Blocking Mask: Everything

Add Component







Watch window 1 □ Static
Tag Untagged Layer UI

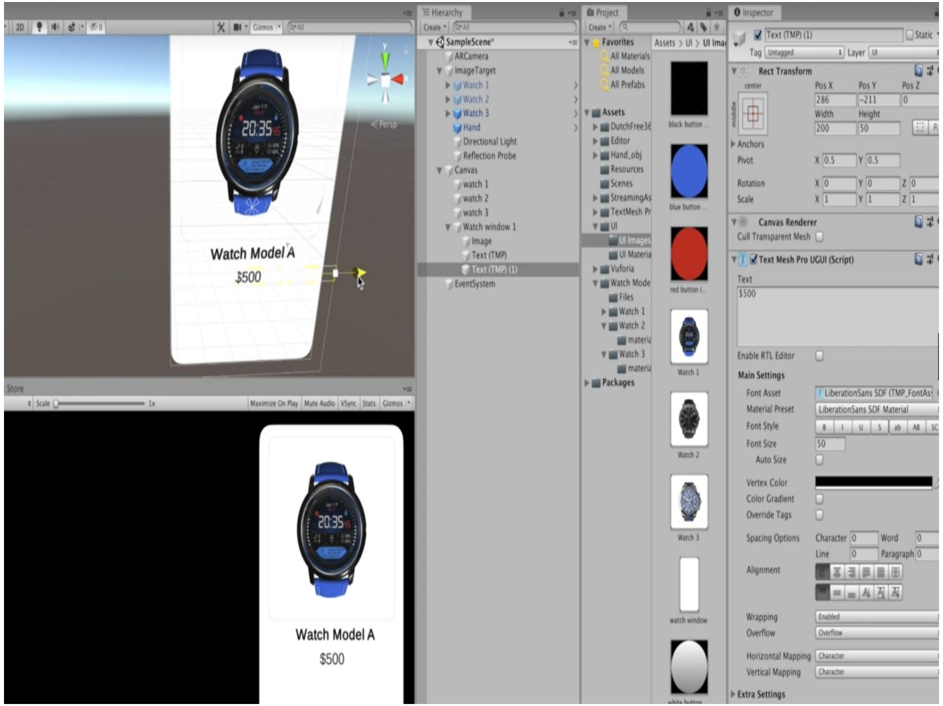
Rect Transform
center middle
Pos X 260 Pos Y 2 Pos Z 0
Width 661 Height 1037
Anchors
Pivot X 0.5 Y 0.5
Rotation X 0 Y 0 Z 0
Scale X 0.36343 Y 0.36343 Z 0.36343

Canvas Renderer
Cull Transparent Mesh □

☑ Image (Script)
Source Image None (Sprite)
Color
Material watch window
Raycast Target ☑

watch window
Shader UI/Default

Add Component



Inspector
☑ Text (TMP) (1) □ Static
Tag Untagged Layer UI

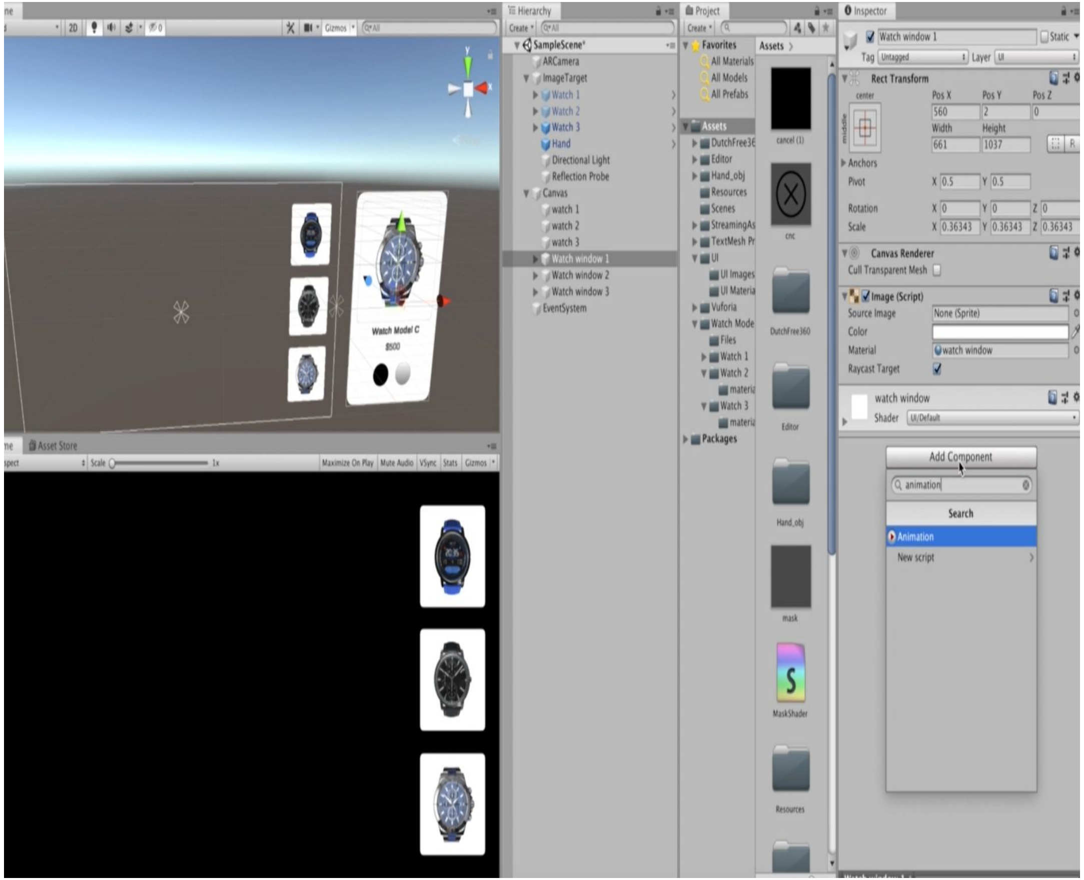
Rect Transform
center middle
Pos X 286 Pos Y -211 Pos Z 0
Width 200 Height 50
Anchors
Pivot X 0.5 Y 0.5
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

Canvas Renderer
Cull Transparent Mesh □

☑ Text Mesh Pro UGUI (Script)
Text
$500

Enable RTL Editor □
Main Settings
Font Asset LiberationSans SDF (TMP_FontAs
Material Preset LiberationSans SDF Material
Font Style B I U S ab AB SC
Font Size 50
Auto Size □
Vertex Color
Color Gradient □
Override Tags □
Spacing Options Character 0 Word 0
Line 0 Paragraph 0
Alignment
Wrapping Enabled
Overflow Overflow
Horizontal Mapping Character
Vertical Mapping Character
Extra Settings

Part 2



Watch window 1 — Static
Tag: Untagged — Layer: UI

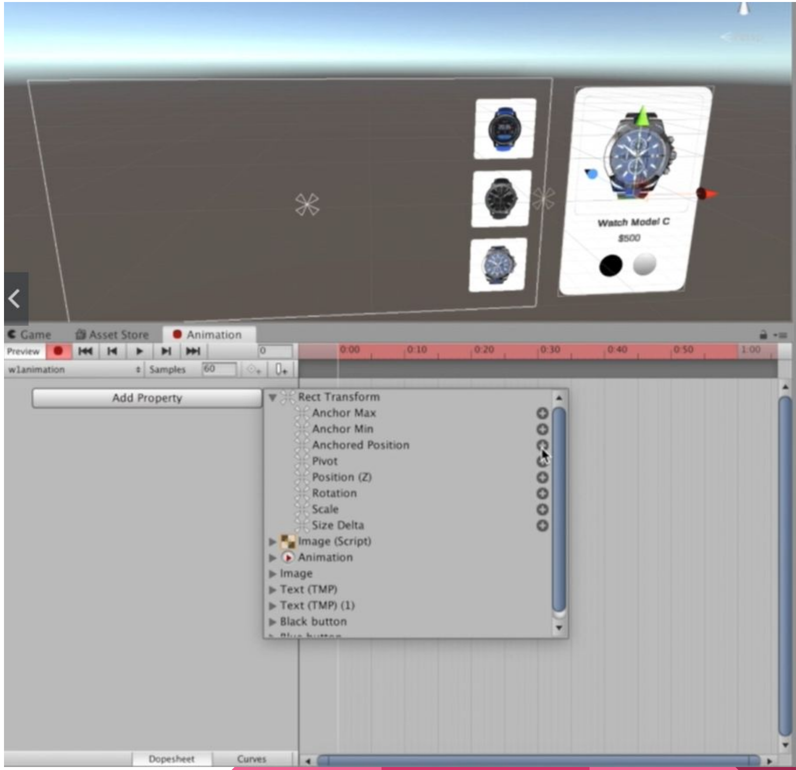
Rect Transform (center / middle)
Pos X: 560 — Pos Y: 2 — Pos Z: 0
Width: 661 — Height: 1037
Anchors
Pivot: X 0.5, Y 0.5
Rotation: X 0, Y 0, Z 0
Scale: X 0.36343, Y 0.36343, Z 0.36343

Canvas Renderer
Cull Transparent Mesh: □

Image (Script)
Source Image: None (Sprite)
Color
Material: watch window
Raycast Target: ☑

watch window
Shader: UI/Default

Search: animation



w1animation — Samples: 60

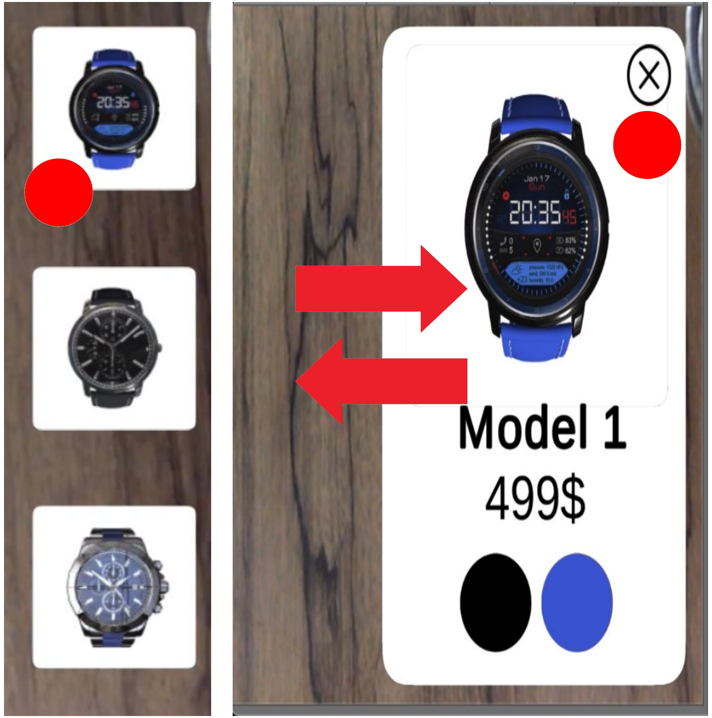
Rect Transform
- Anchor Max
- Anchor Min
- Anchored Position
- Pivot
- Position (Z)
- Rotation
- Scale
- Size Delta

Image (Script)
Animation
Image
Text (TMP)
Text (TMP) (1)
Black button

C# Scripts

Two Components of Scripting in the program, namely:

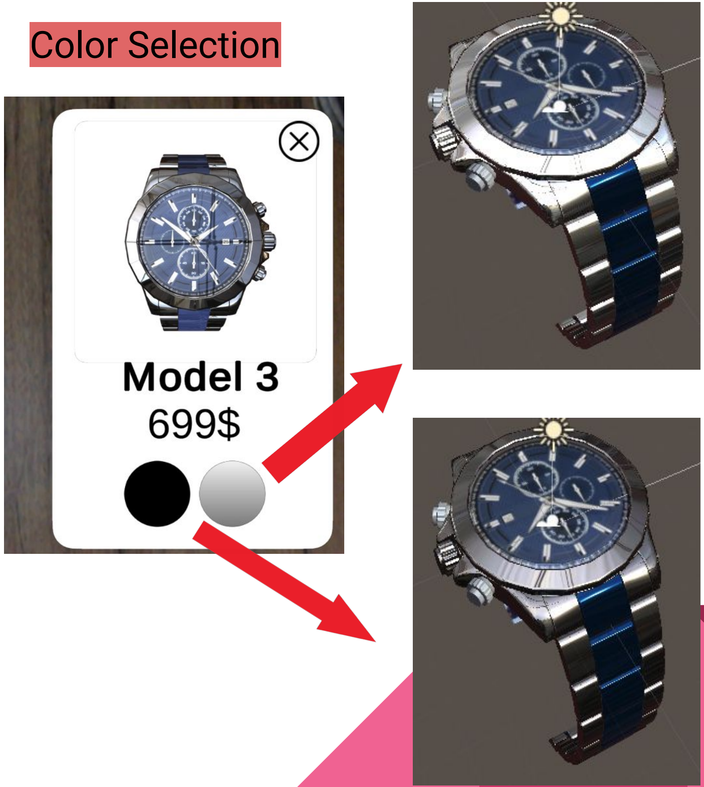
Watch Selection



Model 1

499$

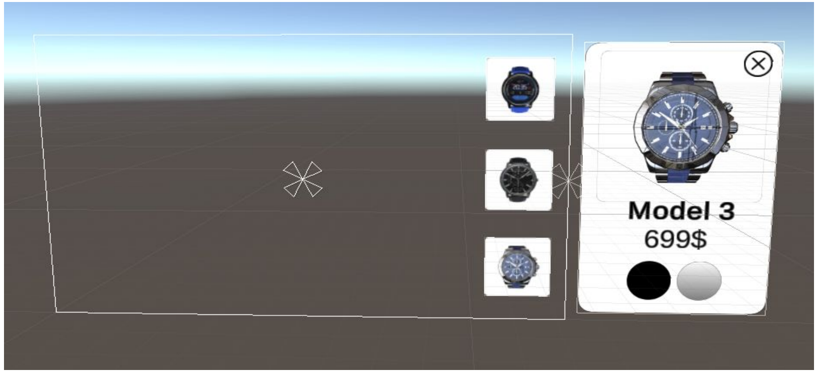
Color Selection

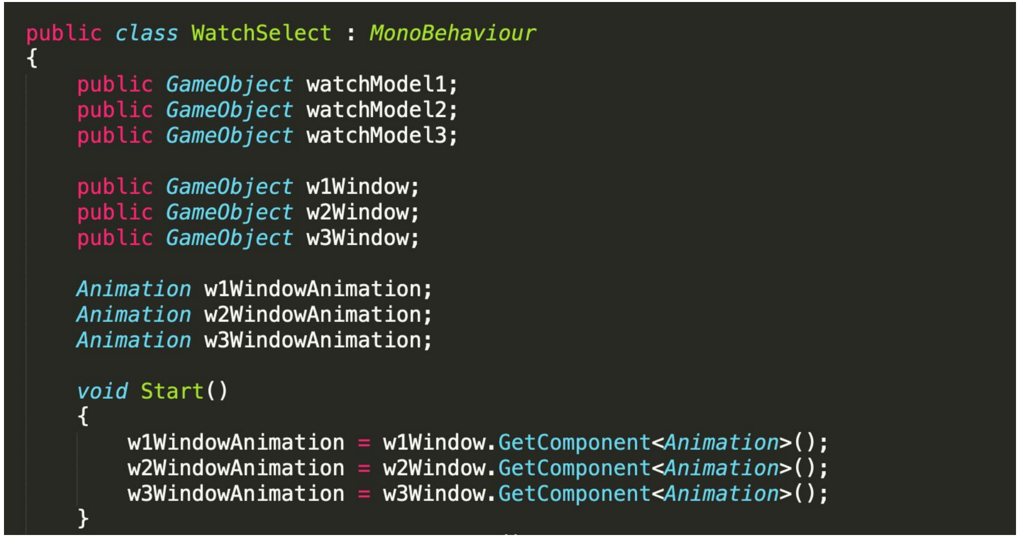


Model 3

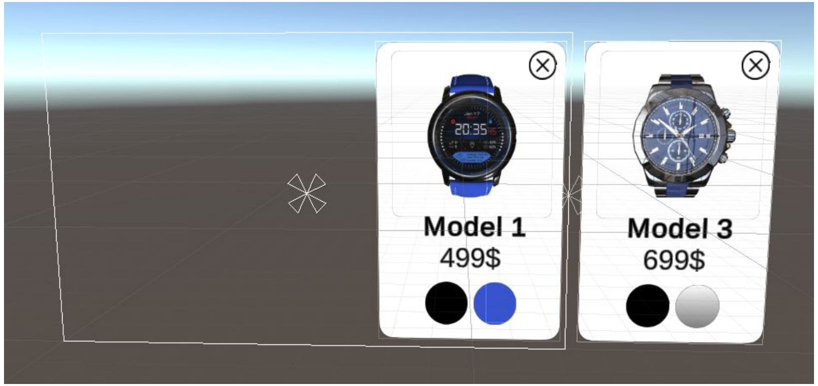
699$

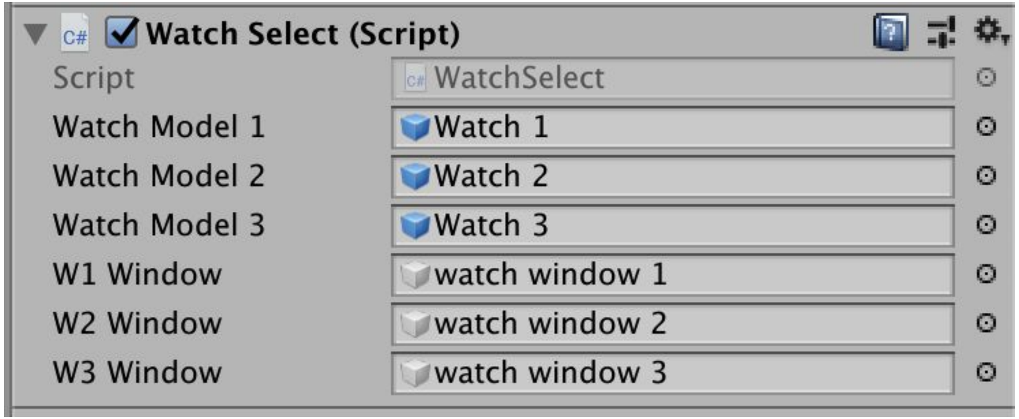
Part 1: Watch Selection





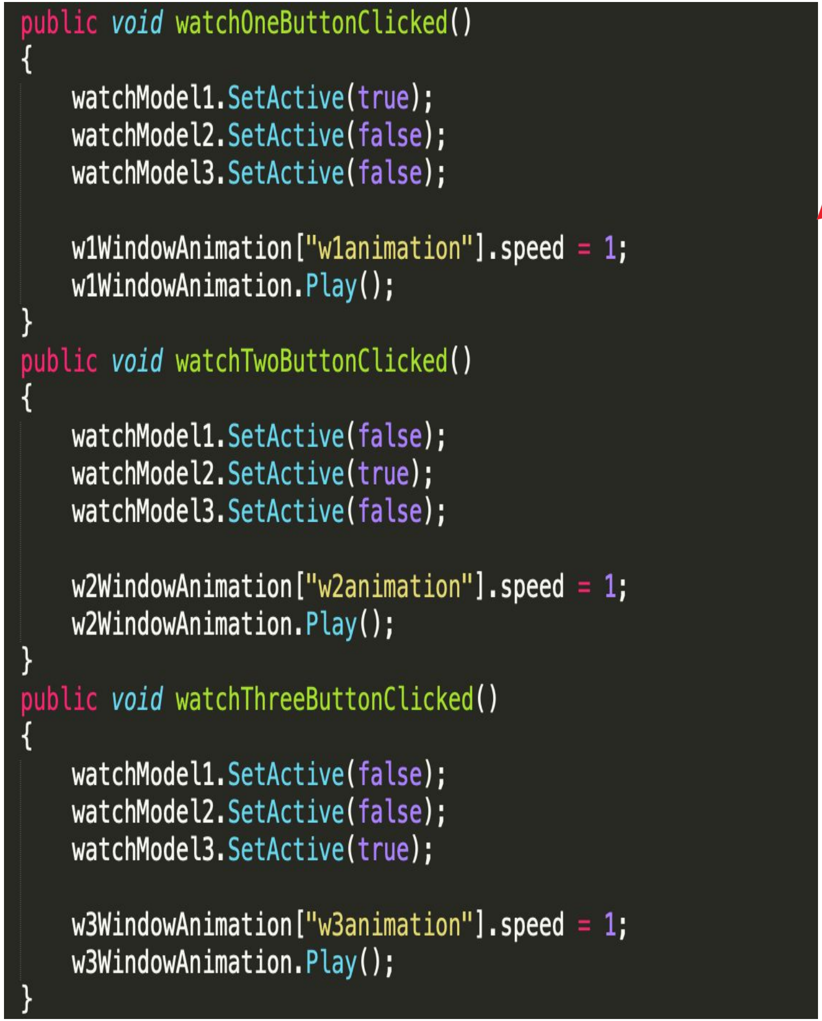
```
public class WatchSelect : MonoBehaviour
{
    public GameObject watchModel1;
    public GameObject watchModel2;
    public GameObject watchModel3;

    public GameObject w1Window;
    public GameObject w2Window;
    public GameObject w3Window;

    Animation w1WindowAnimation;
    Animation w2WindowAnimation;
    Animation w3WindowAnimation;

    void Start()
    {
        w1WindowAnimation = w1Window.GetComponent<Animation>();
        w2WindowAnimation = w2Window.GetComponent<Animation>();
        w3WindowAnimation = w3Window.GetComponent<Animation>();
    }
```





Watch Select (Script)

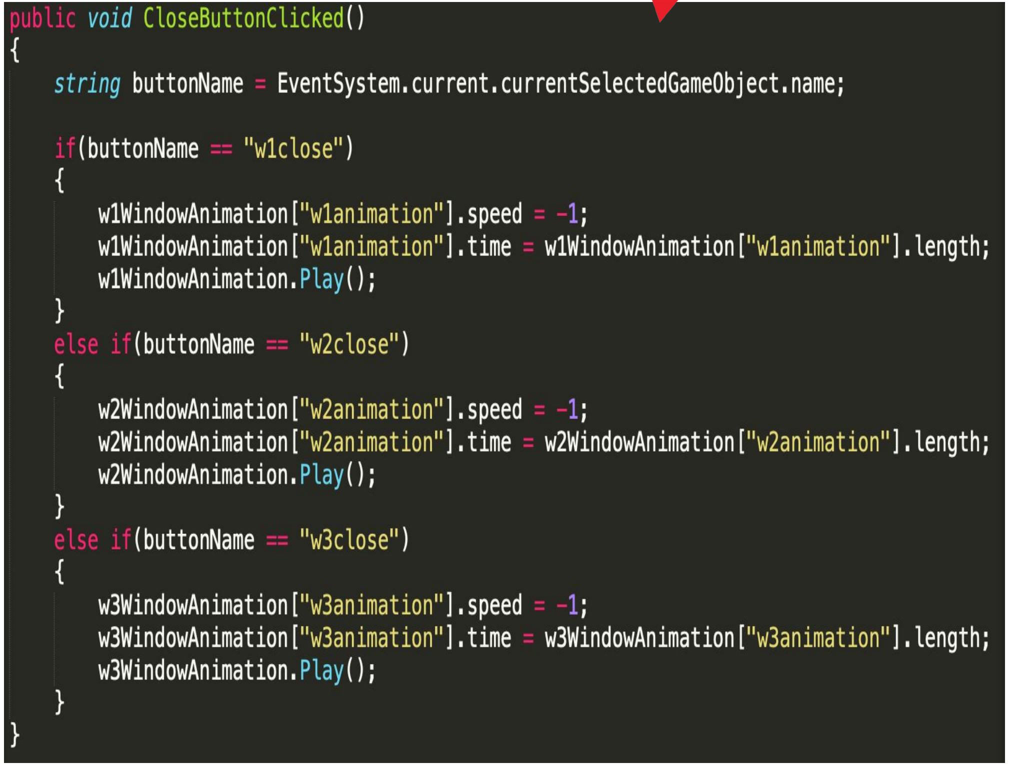
Script: WatchSelect
Watch Model 1: Watch 1
Watch Model 2: Watch 2
Watch Model 3: Watch 3
W1 Window: watch window 1
W2 Window: watch window 2
W3 Window: watch window 3



```
public void watchOneButtonClicked()
{
    watchModel1.SetActive(true);
    watchModel2.SetActive(false);
    watchModel3.SetActive(false);

    w1WindowAnimation["w1animation"].speed = 1;
    w1WindowAnimation.Play();
}
public void watchTwoButtonClicked()
{
    watchModel1.SetActive(false);
    watchModel2.SetActive(true);
    watchModel3.SetActive(false);

    w2WindowAnimation["w2animation"].speed = 1;
    w2WindowAnimation.Play();
}
public void watchThreeButtonClicked()
{
    watchModel1.SetActive(false);
    watchModel2.SetActive(false);
    watchModel3.SetActive(true);

    w3WindowAnimation["w3animation"].speed = 1;
    w3WindowAnimation.Play();
}
```

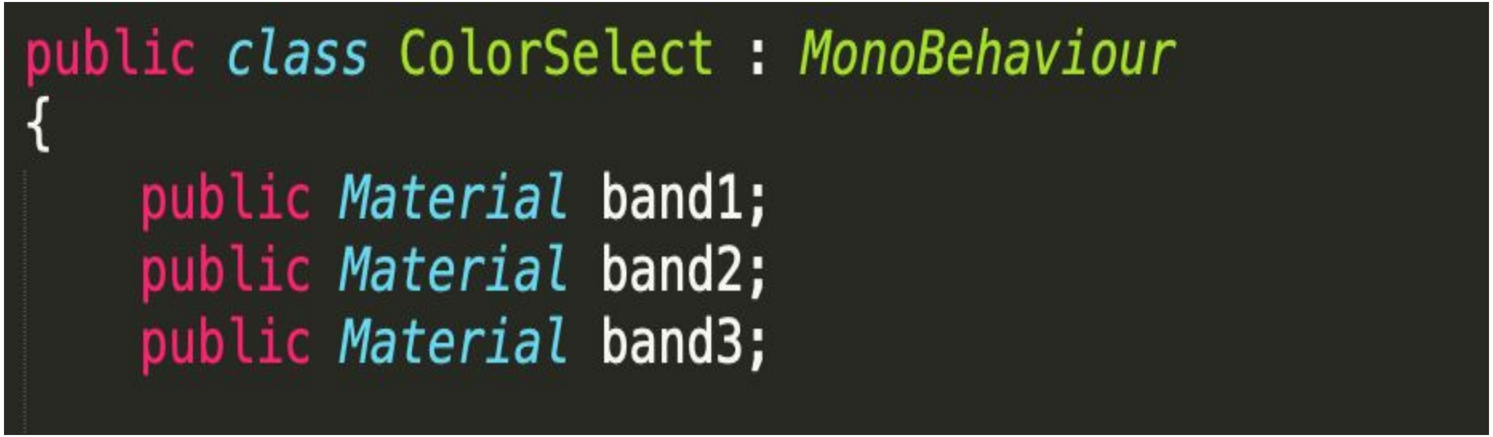
Window Sliding In

Window Sliding Out

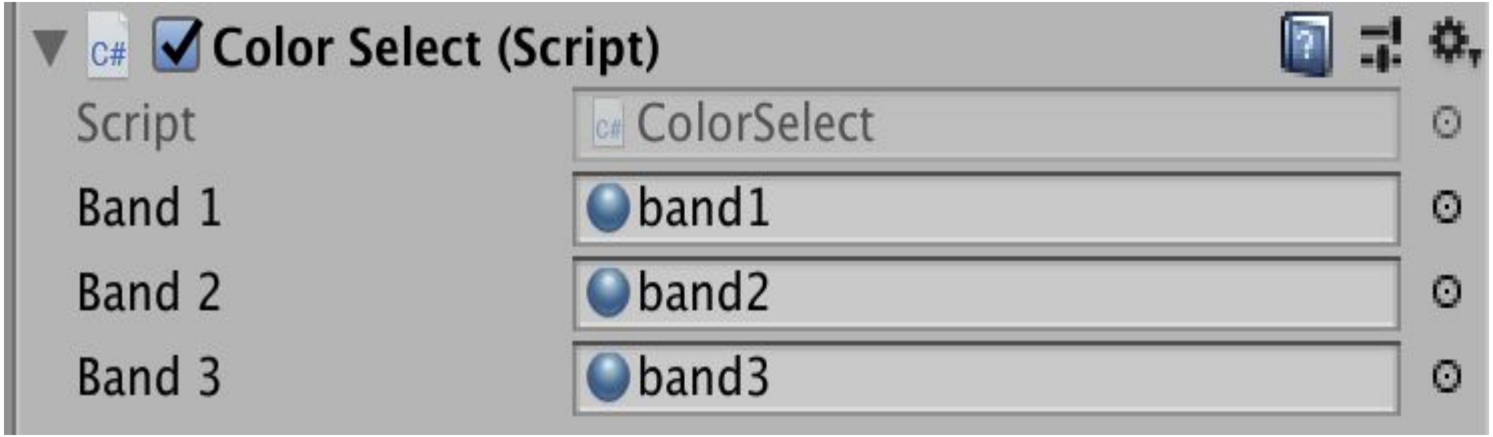


```
public void CloseButtonClicked()
{
    string buttonName = EventSystem.current.currentSelectedGameObject.name;

    if(buttonName == "w1close")
    {
        w1WindowAnimation["w1animation"].speed = -1;
        w1WindowAnimation["w1animation"].time = w1WindowAnimation["w1animation"].length;
        w1WindowAnimation.Play();
    }
    else if(buttonName == "w2close")
    {
        w2WindowAnimation["w2animation"].speed = -1;
        w2WindowAnimation["w2animation"].time = w2WindowAnimation["w2animation"].length;
        w2WindowAnimation.Play();
    }
    else if(buttonName == "w3close")
    {
        w3WindowAnimation["w3animation"].speed = -1;
        w3WindowAnimation["w3animation"].time = w3WindowAnimation["w3animation"].length;
        w3WindowAnimation.Play();
    }
}
```

Part 2: Color Selection



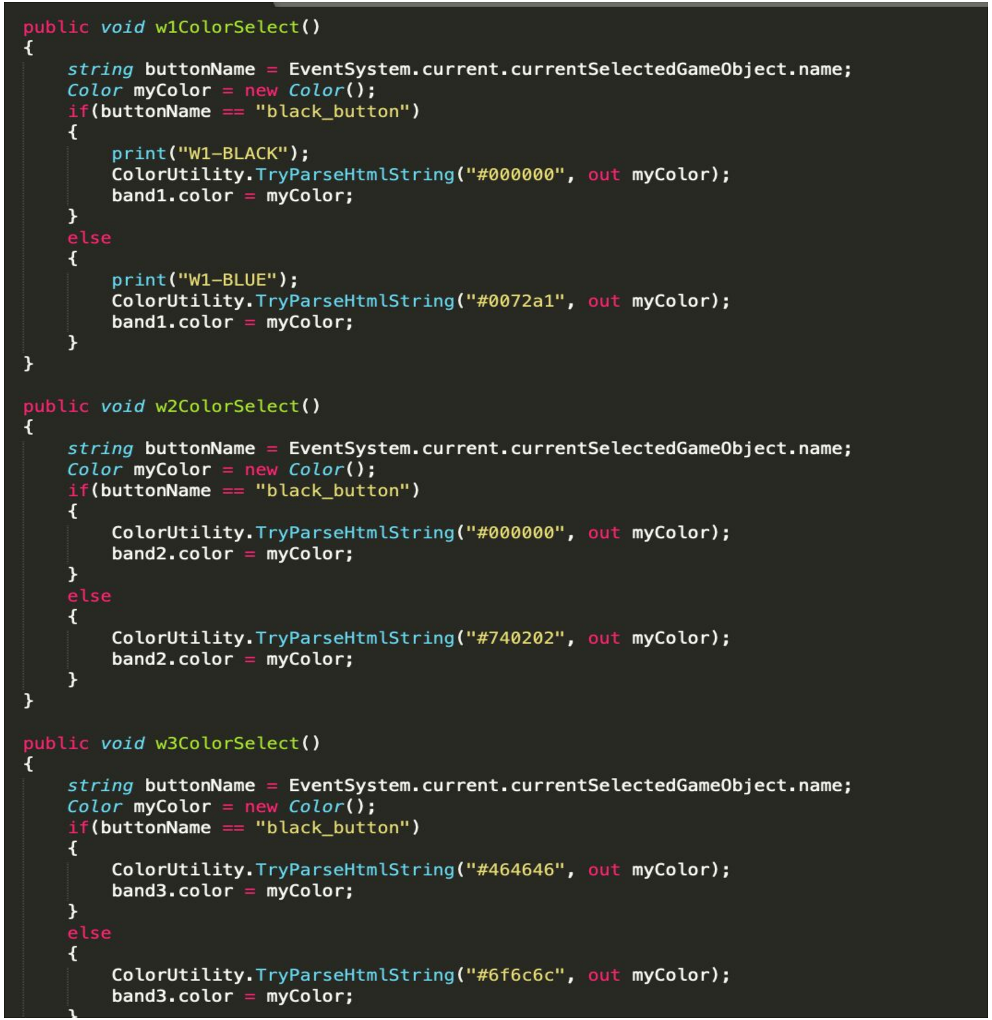
```
public class ColorSelect : MonoBehaviour
{
    public Material band1;
    public Material band2;
    public Material band3;
```



☑ Color Select (Script)

Script	ColorSelect
Band 1	band1
Band 2	band2
Band 3	band3

Code for color selection, for each button



```
public void w1ColorSelect()
{
    string buttonName = EventSystem.current.currentSelectedGameObject.name;
    Color myColor = new Color();
    if(buttonName == "black_button")
    {
        print("W1-BLACK");
        ColorUtility.TryParseHtmlString("#000000", out myColor);
        band1.color = myColor;
    }
    else
    {
        print("W1-BLUE");
        ColorUtility.TryParseHtmlString("#0072a1", out myColor);
        band1.color = myColor;
    }
}

public void w2ColorSelect()
{
    string buttonName = EventSystem.current.currentSelectedGameObject.name;
    Color myColor = new Color();
    if(buttonName == "black_button")
    {
        ColorUtility.TryParseHtmlString("#000000", out myColor);
        band2.color = myColor;
    }
    else
    {
        ColorUtility.TryParseHtmlString("#740202", out myColor);
        band2.color = myColor;
    }
}

public void w3ColorSelect()
{
    string buttonName = EventSystem.current.currentSelectedGameObject.name;
    Color myColor = new Color();
    if(buttonName == "black_button")
    {
        ColorUtility.TryParseHtmlString("#464646", out myColor);
        band3.color = myColor;
    }
    else
    {
        ColorUtility.TryParseHtmlString("#6f6c6c", out myColor);
        band3.color = myColor;
    }
```

TIME FOR A LIVE DEMONSTRATION

Future Possibilities

- Small World Applications
  - Tatoos
  - Designer Wall Painting
- Bigger Ideas
  - Fully Developed Watch App
  - Furniture
- Final Aim
  - Making a whole new shopping experience with all different kinds of products

THE END