**中原大學**

**資訊工程學系**

**108學年度專題實驗期末報告**

**AR平面中的人物布置**

組員 資訊四乙 陳宏溢 10527245

資訊四乙 林威廷 10527239

資訊四甲 彭義翔 10527142

指導教授：蘇志文 老師

**中華民國一○八年十一月二十八日**

目次

目次................................................................1

圖目錄..............................................................2

第一章 序論

1.1 前言........................................................3

1.2 簡介........................................................3

第二章 使用工具介紹

2.1 Unity 3D 介紹..........................................4

2.2 AR Foundation 介紹..........................................4

2.3 C# 介紹..........................................5

2.4 Blender 介紹..........................................5

第三章 使用流程及方法實作

3.1 使用者輸入介面.............................................19

3.2 Plane Manager..............................................20

3.3 光源及陰影設定.............................................21

3.4 物件與腳本.................................................23

第四章 遇到問題與解決方法

4.1 軟體支援度.................................................27

4.2 硬體支援度.................................................27

4.3 技術問題 .................................................27

第五章 專題結論與未來展望 .........................................28

參考資料...........................................................29

圖目錄

圖一 程式使用流程圖................................................6

圖二 開始&注意事項 ................................................7

圖三 等待平面偵測 ................................................8

圖四 選好物體後關掉選單 ...........................................9

圖五 點選平面任意處即可創建 ......................................10

圖六 可創建多個 ..................................................11

圖七 可以移動它 ..................................................12

圖八 點開選單可以縮小它 ..........................................13

圖九 也可以放大 .................................................14

圖十 關掉陰影 ....................................................15

圖十一 關掉燈光 ..................................................16

圖十二 全部的模型 ................................................17

圖十三 延伸圖八並取消平面偵測 ....................................18

**第一章 序論**

**1.1前言**

手機對現在人而言已經是非常的普及，隨處可見，放眼望去每個人都在對著手機操作輸入語音聊天，而最近一波大熱潮—AR擴增實境，緣起於1935年的小說、1970年代有雛形、1990確定名稱AR(Augmented Reality)、直到2013年開始有產品市場，漸漸的人們開始有了概念——我要利用手機或是平板電腦，來預覽我想要的產品結果或是追求娛樂活動。而市面上的AR軟體沒有整合調整物件的功能來有更多不同的選項去預覽物件，所以我們打算著手研究更多功能的整合在AR程式上面有更好的使用同一個物件之體驗。

**1.2簡介**

我們透過Unity強大的整合和AR Foundation的套件，並結合C#的程式設計修改，可以讓使用者選擇預設物品放在偵測到的平面上面，之後可以調整燈光，或是移動位置到最適合的地方，還可以放大縮小該物體，直到使用者滿意，放置定點後使用者可以移動腳步或是鏡頭旋轉，物體還是會在同一個定點不動，可以觀察不同的角度，同時也可以創建多個不同的物體，可以看到理論上他們在現實中的表現，當然數量的多寡與使用者所持的硬體設備能力有相關，還請多加留意。

**第二章 使用工具介紹**

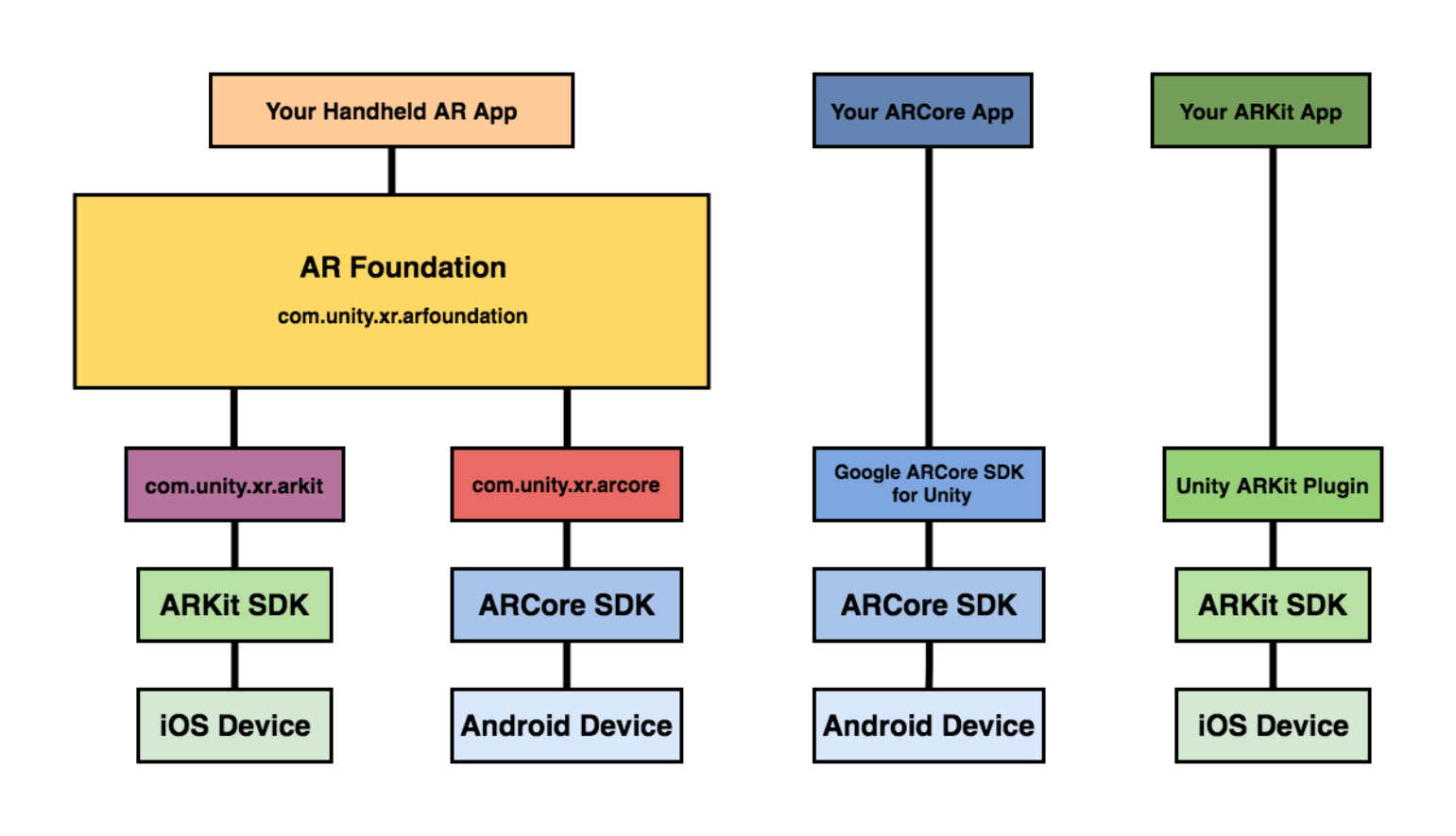
**2.1 Unity 3D**

****Unity 是一款由 Unity Technologies 研發的跨平台2D / 3D [遊戲引擎](https://zh.wikipedia.org/wiki/%E6%B8%B8%E6%88%8F%E5%BC%95%E6%93%8E)，可用於開發 Windows、MacOS 及 Linux 平台，PlayStation、XBox和Swich [遊戲主機](https://zh.wikipedia.org/wiki/%E9%9B%BB%E5%AD%90%E9%81%8A%E6%88%B2%E6%A9%9F)平台，或是 [iOS](https://zh.wikipedia.org/wiki/IOS" \o "IOS)、[Android](https://zh.wikipedia.org/wiki/Android" \o "Android) 行動裝置。除可以用於研發電子遊戲之外，Unity 還是被廣泛用於建築視覺化、即時[三維動畫](https://zh.wikipedia.org/wiki/%E4%B8%89%E7%BB%B4%E8%AE%A1%E7%AE%97%E6%9C%BA%E5%9B%BE%E5%BD%A2" \o "三維電腦圖形)等類型互動內容的綜合型創作工具。多人連線是由第三方套件提供。

Unity 最初於2005年在[蘋果公司](https://zh.wikipedia.org/wiki/%E8%98%8B%E6%9E%9C%E5%85%AC%E5%8F%B8" \o "蘋果公司)的[全球開發者大會](https://zh.wikipedia.org/wiki/%E8%8B%B9%E6%9E%9C%E5%85%A8%E7%90%83%E5%BC%80%E5%8F%91%E8%80%85%E5%A4%A7%E4%BC%9A)上對外公布並開放使用，當時只是一款面向 [OS X](https://zh.wikipedia.org/wiki/MacOS) 平台的遊戲引擎。時至2018年，該引擎所支援的研發平台已經達到27個。

因為它有層級式的綜合開發環境，視覺化編輯，詳細的屬性編輯器和動態的遊戲預覽。所以常常用來快速的製作遊戲或者開發遊戲原型。

**2.2 AR Foundation**



AR Foundation 屬於Unity 提供的API，它支援Google開發在Android平台運行的 ARCore與Apple開發在iOS 平台運行的ARKit，以及預期可能的頭戴頭盔平台，令程式開發人員可以使用一組程式碼庫就可以創建iOS和Android的AR應用程式。

主要提供的套件功能有平面偵測、裝置位置、方向追蹤、光線測量、參考點及其設定、人臉追蹤、圖像追蹤、物體追蹤。

**2.3 C#**

**C#**是微軟推出的一種基於.NET框架的、物件導向的進階程式語言。C#以.NET框架類別館作為基礎，擁有類似Visual Basic的快速開發能力。微軟在2000年發布了這種語言，希望藉助這種語言來取代Java。C#已經成為Ecma國際和國際標準組織的標準規範。

C#旨在設計成為一種「簡單、現代、通用」，以及物件導向的程式設計語言。

提供以下支援：強型別檢查、陣列維度檢查、未初始化的變數參照檢測、自動垃圾收集（Garbage Collection）。

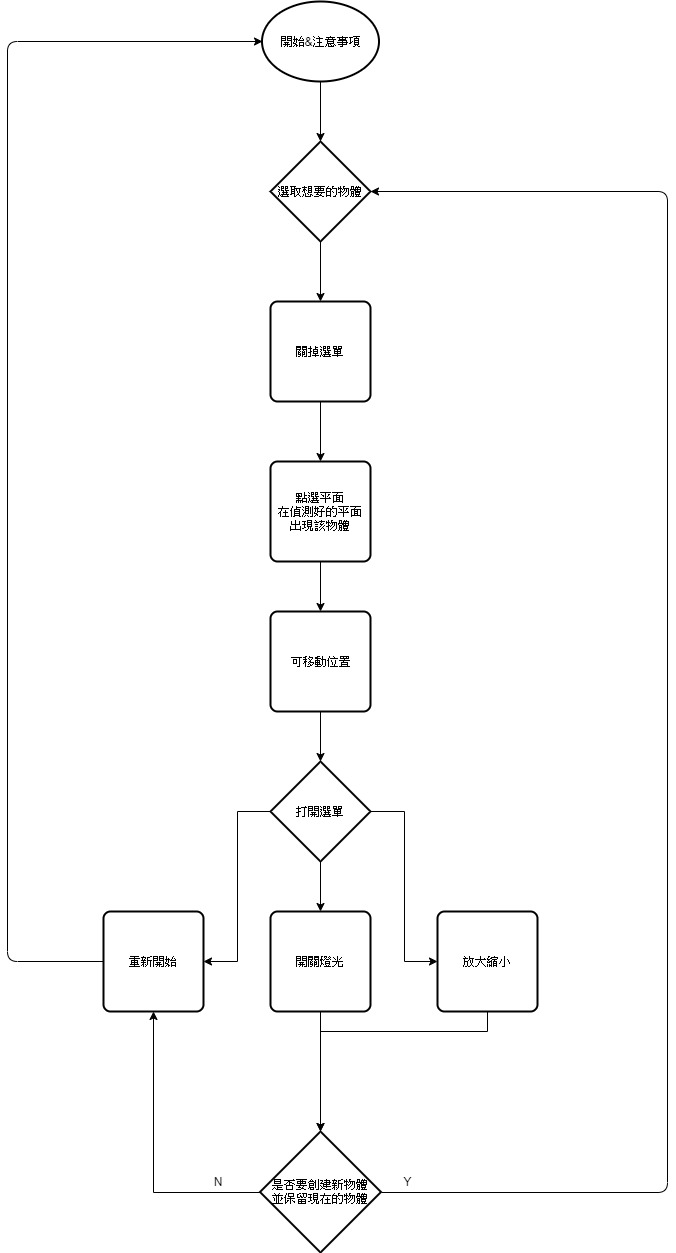
**2.4 Blender**

****主要程式設計者是荷蘭人Ton Roosendaal於1998年6月成立NaN公司，將其進一步發展，並以共享軟體的形式對外發布這個程式，以GNU通用公眾授權條款發布。Blender現在是自由軟體，並由Blender基金會維護與更新。

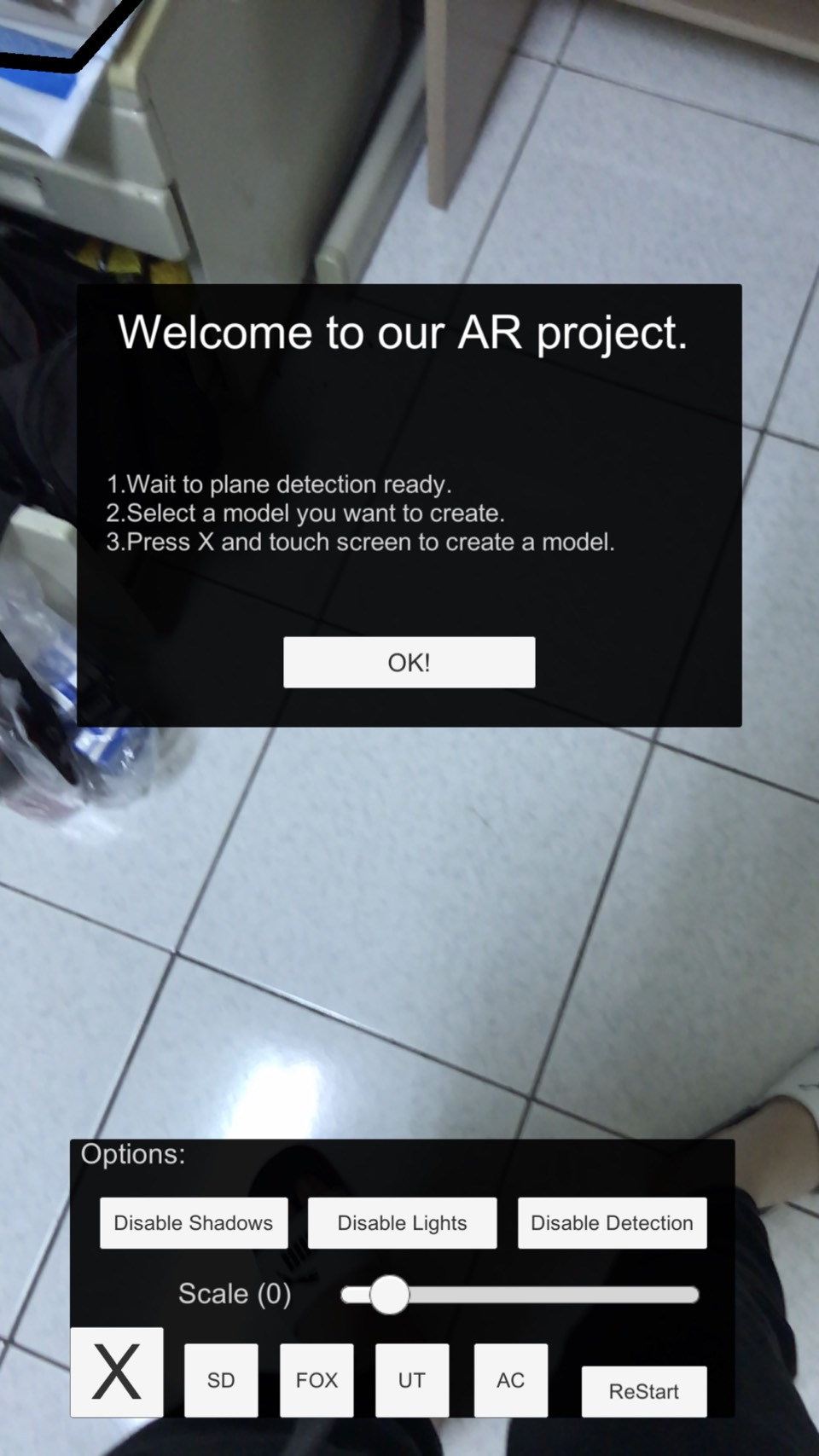
支援不同的幾何圖元，包括多邊形網紋，快速表層塑模，曲線及向量字元。

內建cycle光線追蹤彩現引擎，可提供真實的畫面，同時也有eevee實時彩現引擎，可即時提供極佳的預覽畫面，他的動畫工具，包括了反向動作組件，可設定骨幹，結構變形，關鍵影格，時間線，非線性動畫，系統規定參數，頂點量重及柔化動量組件，包括網孔碰撞偵察和一個具有偵察碰撞的粒子系統。使用Python語言來創作及製作遊戲及工作自動化指令碼。在Blender中，物體與資料是分離的，可以快速塑模。可以將多個場景合併至單一檔案（.blend）。所有blender所生成的「.blend」檔案均有完善的向前、向後版本的相容性，同時亦具有完整的跨平台支援，源自於可以匯入匯出多種通用的3D模型。

**第三章 使用流程及方法實作**



圖一 程式使用流程圖



圖二 開始&注意事項

****

圖三 等待平面偵測



圖四 選好物體後關掉選單



圖五 點選平面任意處即可創建



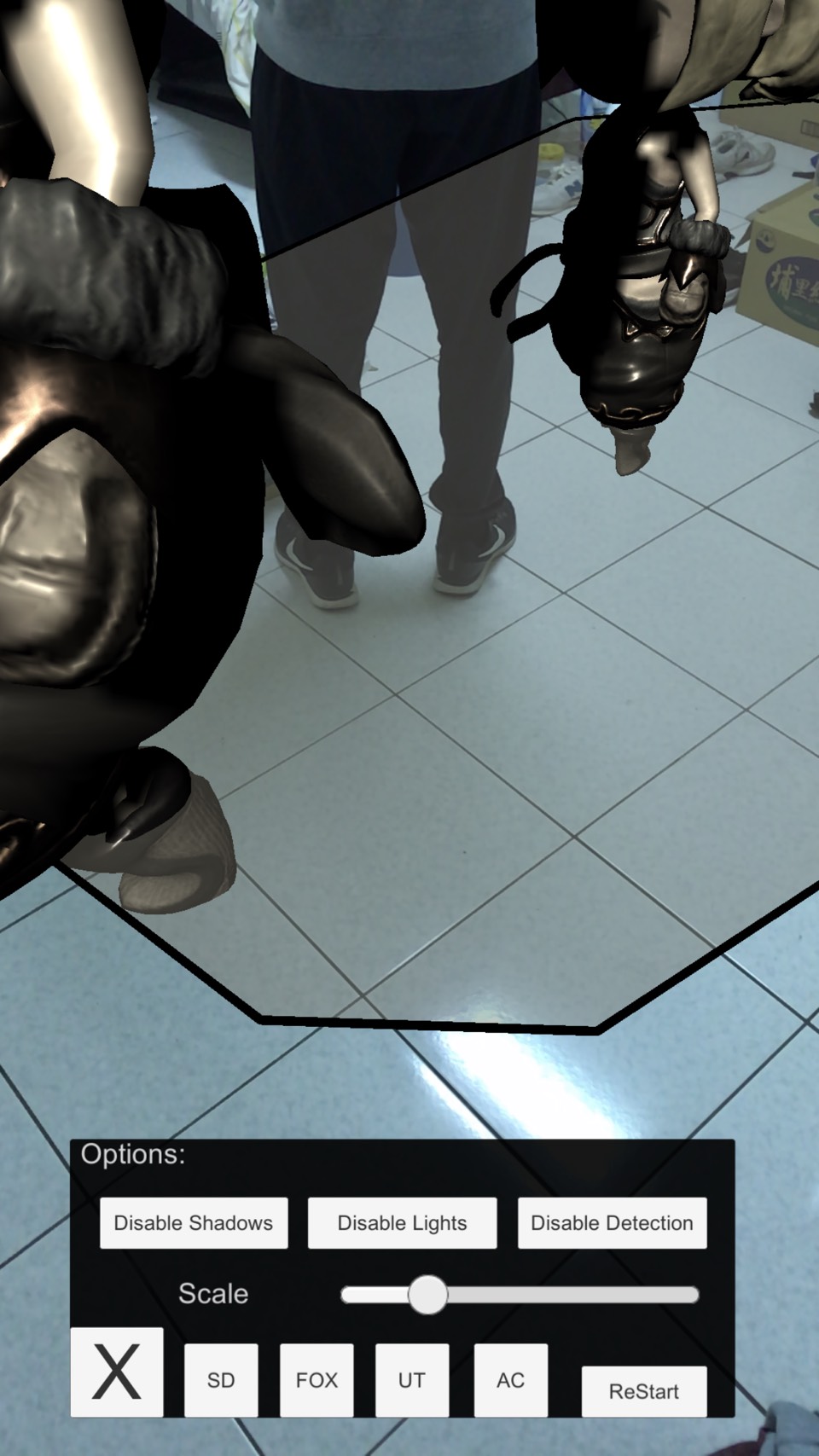
圖六 可創建多個



圖七 可以移動它



圖八 點開選單可以縮小它



圖九 也可以放大



圖十 關掉陰影



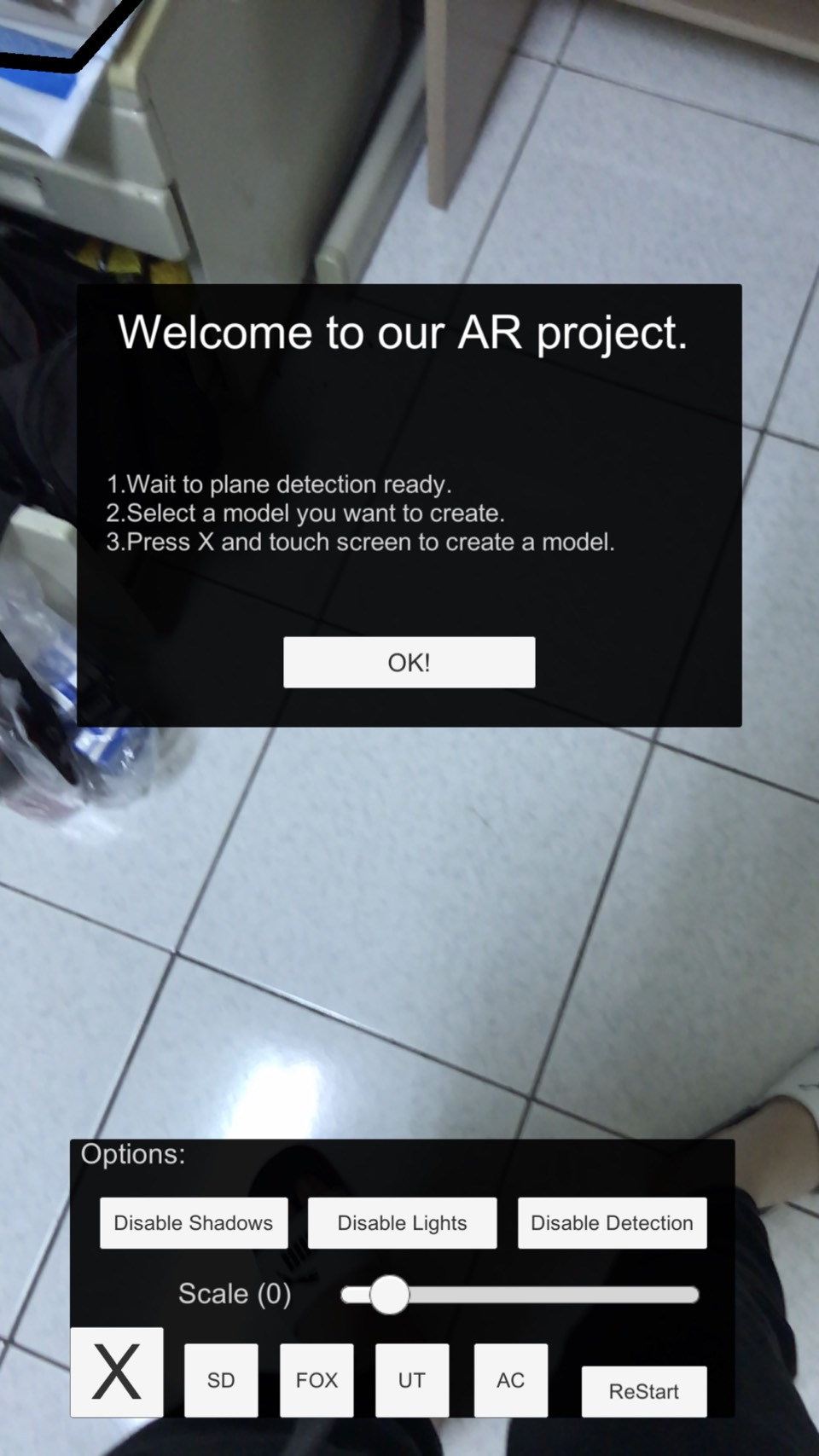
圖十一 關掉燈光



圖十二 全部的模型



圖十三 延伸圖八並取消平面偵測

**3.1 使用者輸入介面**

圖二 開始&注意事項

# 3.2 Plane Manager

平面偵測會將環境中偵測到的平面創建Gameobj ，由尺寸和邊界典表示，環境中可以檢測的平面包括水平桌子、地板、櫃子、垂直牆壁…等。

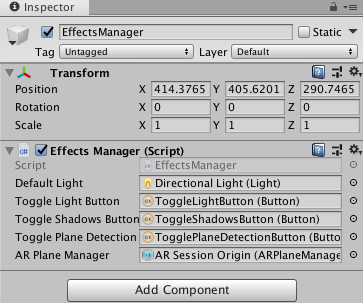
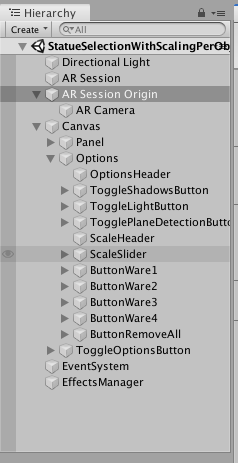
當檢測到平面的時候，會用新創物件來選取表達範圍。Plane prefab可以是null，但GameObj會有ARPlane的物件，事實上就是平面的數據。

可視化平面有助於我們的開發，要有GameObj裡面的ARPlane宣告，就可以使用boundary來表示範圍。



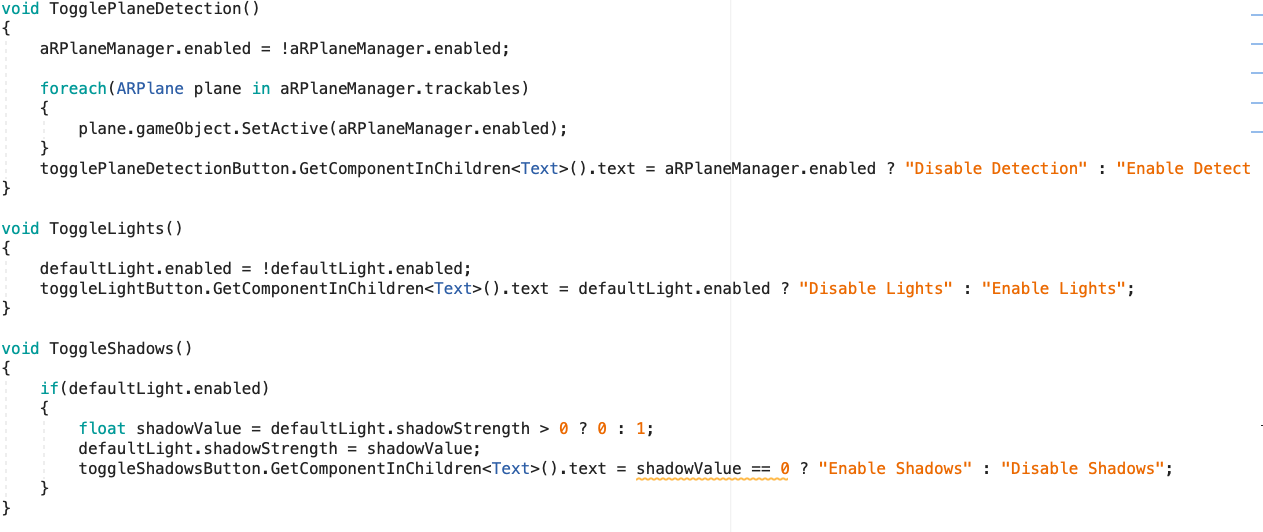
圖四 選好物體後關掉選單

**3.3 光源及陰影設定**



**此程式為Effect Manager,主要為控制燈光、陰影、開關平面偵測之功能，並與Light、AR Session Origin 連接去做調整**

****



前面兩個if為防止沒有設定toggleLightButton、toggleShadowsButton、togglePlaneDetectionButton (為null)的情形。

之後按鈕觸發將事件加入list中，後面的三個function分別為toggleLightButton、toggleShadowsButton、togglePlaneDetectionButton三個案件之事件觸發。

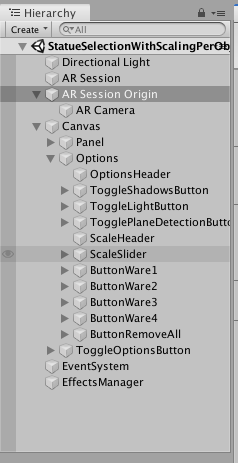
第一個為平面偵測enable的功能，若呼叫function會將aRPlaneManager enable

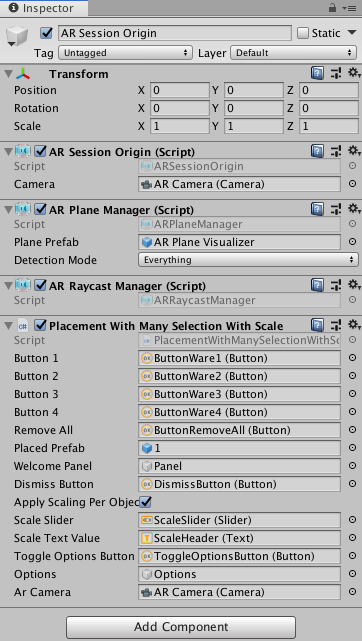
的狀態設為相反，若打開則關閉，反之若關閉則打開，並將每個ARPlane中偵測出來的平面隱藏起來。

第二個為燈光enable的功能，若呼叫function會將 defaultLight enable的狀態設為相反，若打開則關閉，反之若關閉則打開。

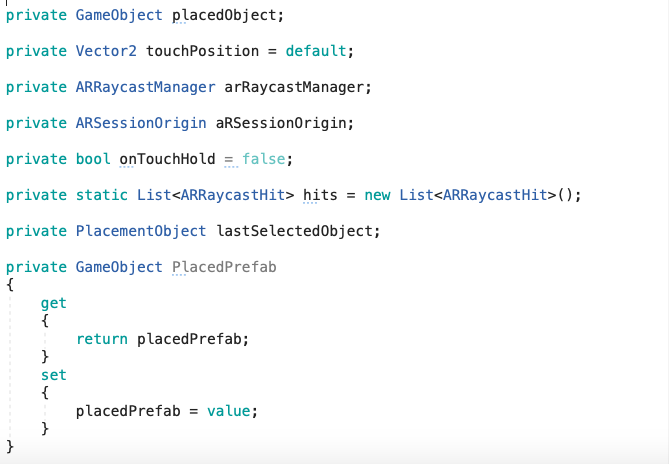
第三個為陰影enable的功能，若呼叫function會將 defaultLight的shadow enable的狀態設為相反，若打開則關閉，反之若關閉則打開。

**3.4 物件與腳本**



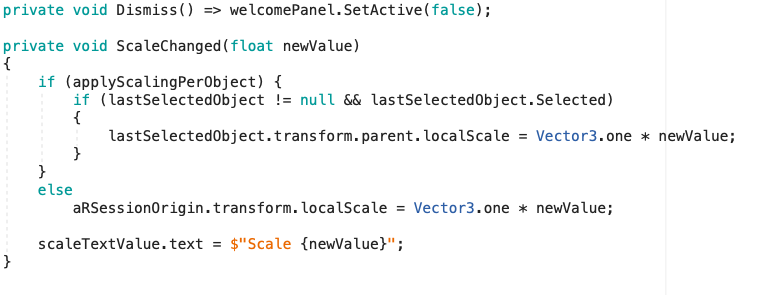
****

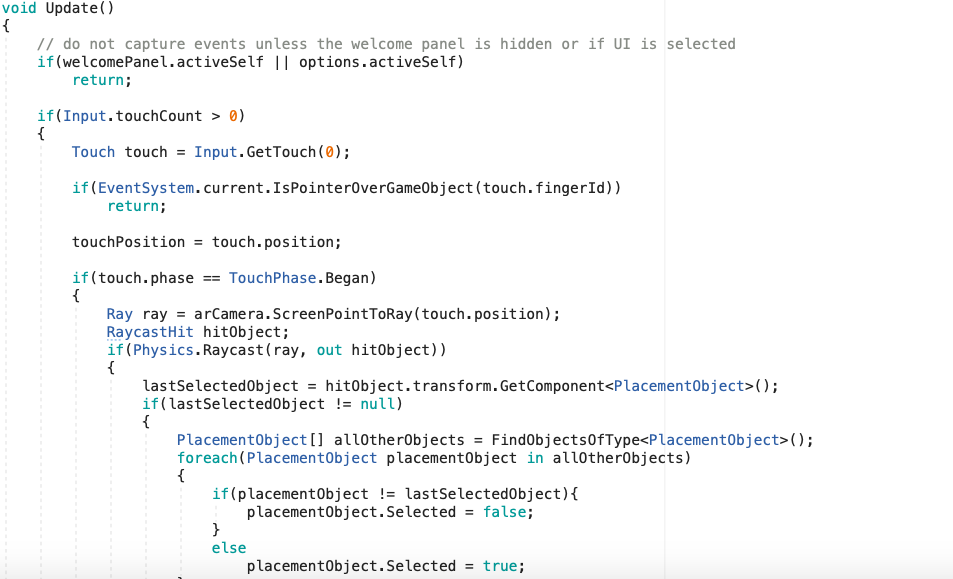
此Script為AR Session Origin中的C#程式碼:Placement With Many Selection With Scale

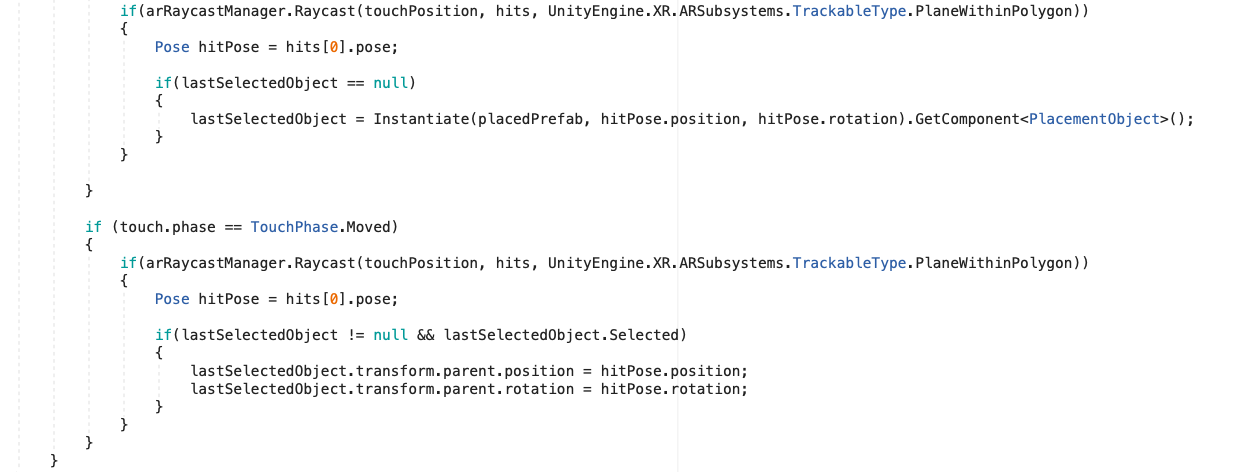


****

****

****

****

****

[SerializeField]

private Button Button1, Button2, Button3, Button4;

為選取人物的四個按鈕

[SerializeField]

private Button RemoveAll;

為重新啟動app，原本為清除所有物件改成直接重啟整個場景

[SerializeField]

private GameObject placedPrefab;

此為新增人物的指標，當手指觸碰螢幕時，若沒有選取到物件且在平面爭測的範圍時，新增placedPrefab所指向人物。

[SerializeField]

private GameObject welcomePanel;

此為初始畫面的簡介提醒背景。

[SerializeField]

private Button dismissButton;

此為OK!按鈕，在使用者看完流程說明後，點擊即可關閉提醒。

[SerializeField]

private Slider scaleSlider;

此為調整選取3D人物大小的調整條。

[SerializeField]

private Text scaleTextValue;

此為大小調整條的文字說明欄。

[SerializeField]

private Button toggleOptionsButton;

此為開關Option選單的按鈕，開啟選單時只能對物件進行調整(拖動除外)，不能新增物件，關閉選單時只能進行人物的新增或拖動，不能調整其他特徵。

[SerializeField]

private GameObject options;

此為Option物件，裡面包含除了OptionsHeaderd、ToggleShadowsButton、ToggleLightButton、ToggleDetectionButton、ScaleHeader、ScaleSlider、Button1、 Button2、Button3、Button4以及ButtonRemoveAll物件，由toggleOptionsButton管理開關。

private PlacementObject lastSelectedObject;

此為最後被選取的人物，可能為空，為空且在option被關閉時，建立新人物。

void Awake()

{

arRaycastManager = GetComponent<ARRaycastManager>();

aRSessionOrigin = GetComponent<ARSessionOrigin>();

dismissButton.onClick.AddListener(Dismiss);

scaleSlider.onValueChanged.AddListener(ScaleChanged);

toggleOptionsButton.onClick.AddListener(ToggleOptions);

if (Button1 != null && Button2 != null && Button3 != null && Button4 != null)

//&& Button5 != null && Button6 != null && Button7 != null && Button8 != null )

{

Button1.onClick.AddListener(() => ChangePrefabSelection("1"));

Button2.onClick.AddListener(() => ChangePrefabSelection("2"));

Button3.onClick.AddListener(() => ChangePrefabSelection("3"));

Button4.onClick.AddListener(() => ChangePrefabSelection("4"));

}

if (RemoveAll != null)

{

RemoveAll.onClick.AddListener(() => DeleteAllObj());

}

}

當使用者觸發事件時被awake進行即時處理(按鈕觸發)。

private void DeleteAllObj()

{

//Application.LoadLevel(0);

SceneManager.LoadScene(0);

}

當使用者按下ReStart時，重製整個場景。

private void ChangePrefabSelection(string name)

{

GameObject loadedGameObject = Resources.Load<GameObject>($"Prefabs/{name}");

if (loadedGameObject != null)

{

PlacedPrefab = loadedGameObject;

Debug.Log($"Game object with name {name} was loaded");

}

else

{

Debug.Log($"Unable to find a game object with name {name}");

}

}

當使用者按下人物按鈕選取人物時，觸發事件到Prefabs資料夾中選取相對應人物模型(1~4為檔案名稱)。

private void ToggleOptions()

{

if (options.activeSelf)

{

toggleOptionsButton.GetComponentInChildren<Text>().text = "O";

options.SetActive(false);

}

else

{

toggleOptionsButton.GetComponentInChildren<Text>().text = "X";

options.SetActive(true);

}

}

ToggleOptions觸發時，將圖案改成”O”或”X”，並開(關)Option物件。

private void Dismiss() => welcomePanel.SetActive(false);

關閉welcomepanel物件。

private void ScaleChanged(float newValue)

{

if (applyScalingPerObject) {

if (lastSelectedObject != null && lastSelectedObject.Selected)

{

lastSelectedObject.transform.parent.localScale = Vector3.one \* newValue;

}

}

else

aRSessionOrigin.transform.localScale = Vector3.one \* newValue;

scaleTextValue.text = $"Scale {newValue}";

}

拉動調整條時，處理人物大小的實現。

void Update()

{

// do not capture events unless the welcome panel is hidden or if UI is selected

if(welcomePanel.activeSelf || options.activeSelf)

return;

if(Input.touchCount > 0)

{

Touch touch = Input.GetTouch(0);

if(EventSystem.current.IsPointerOverGameObject(touch.fingerId))

return;

touchPosition = touch.position;

if(touch.phase == TouchPhase.Began)

{

Ray ray = arCamera.ScreenPointToRay(touch.position);

RaycastHit hitObject;

if(Physics.Raycast(ray, out hitObject))

{

lastSelectedObject = hitObject.transform.GetComponent<PlacementObject>();

if(lastSelectedObject != null)

{

PlacementObject[] allOtherObjects = FindObjectsOfType<PlacementObject>();

foreach(PlacementObject placementObject in allOtherObjects)

{

if(placementObject != lastSelectedObject){

placementObject.Selected = false;

}

else

placementObject.Selected = true;

}

}

}

if(arRaycastManager.Raycast(touchPosition, hits, UnityEngine.XR.ARSubsystems.TrackableType.PlaneWithinPolygon))

{

Pose hitPose = hits[0].pose;

if(lastSelectedObject == null)

{

lastSelectedObject = Instantiate(placedPrefab, hitPose.position, hitPose.rotation).GetComponent<PlacementObject>();

}

}

}

if (touch.phase == TouchPhase.Moved)

{

if(arRaycastManager.Raycast(touchPosition, hits, UnityEngine.XR.ARSubsystems.TrackableType.PlaneWithinPolygon))

{

Pose hitPose = hits[0].pose;

if(lastSelectedObject != null && lastSelectedObject.Selected)

{

lastSelectedObject.transform.parent.position = hitPose.position;

lastSelectedObject.transform.parent.rotation = hitPose.rotation;

}

}

}

}

}

}

當welcomepanel和option被關掉後，整個程式運作的實現，包含手指觸碰螢幕時的定位點，是否有可選取物件，沒有的話新增，有的話選取並可拖動。

**第四章 遇到問題與解決方法**

**4.1 軟體支援度**

Unity版本問題，測試最新beta與鄰近較舊版本。

ARFoundation package版本問題，不同版本函數所在位置有些不一樣，要多多查閱官方說明文件。

XCoed 與iOS 版本相容性問題，因為所用的AR為較新技術，iOS只支援13.0以上版本，XCode只支援11.0以後，XCode與iOS 版本須同步更新。

**4.2 硬體支援度**

iOS手機只支援iphone6S以上使用，唯一解決途徑換新手機。

MAC較為老舊，導致與手機連接常常接觸不良連接不上，解決辦法須同時充飽手機和電腦的電池、開啟低耗電模式、換充電線、換連接口、開關手機與MAC網路，以上交叉測試直到可以順利連接。

**4.3 技術問題**

1.圖形辨識以圖形為追蹤目標時，若將目標距離與鏡頭太遠，將會偵測不出特徵，導致物體無法正常移動，解決的辦法，為了能遠距離移動目標，改以平面偵測方式實現，重寫全部程式。

2.製作模型擬真度不足，解決辦法，進unity商店尋找適合物件並對其進行適度修改以替代原本物件。

**第五章 專題結論與未來展望**

**(1)更多模型以供使用者操作及預覽**

目前只有三個可以提供使用者操作的模型樣本，我們專注於功能的整合，建模並使用的技術知識尚未完備，還無法提供使用者我們自行建的模型使用，實為一個缺憾。

**(2)可以根據現實生活中的光影表現在顯示的物件上**

最理想化的結果是可以根據現實的光影來展現更逼真的成像，但是需要複雜的數學模型套件和更強大的硬體來支援這個技術實現，為了避免使用者的設備負擔，我們沒有考慮完善這個能力。

**(3)可以單獨刪除掉選擇的物件**

如果不小心創建的物件太多使得手機效能低落，我們有一個remove all按鈕可以重新來一次，但是應該要有更方便的功能可以取消單個物件來維持住之前的動作可以不用重新來過一遍。

**參考資料**

1. AR Plane Manager :

https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@2.2/manual/plane-manager.html

# 

# AR Session & AR SessionOrigin :

# https://blog.csdn.net/yolon3000/article/details/93914012

1. Dilmer Valecillos’s channel