

C IP AR : ARCORE INITIATION

1. Introduction	2
2. Premier test.	2
2.1 Test de HelloAR.	2
2.2 Test AR + VR	3
2.3 Essai de détecter une image	4
3. Premier programme : Essai du réticule VR	5
3.1 Installer GVR	5
3.2 Installer le réticule.	5
3.3 Créer l'UI.	5
3.4 Créer la scène	6
3.5 Créer les scripts	6
3.6 Associer les scripts	8
3.7 Test du programme	8
3.8 Conclusion	9
3.9 Création d'une app AR	10
4. Second programme : Essai de pointer avec le regard 'gaze timed'	11
4.1 Installer GVR (google VR)	11
4.2 Inclure le réticule	12
4.3 Créer l'image animée	12
4.4 Création du menu :	13
4.5 Créer un nouveau script pour le bouton :	14
4.6 Association des scripts aux déclencheurs :	15
4.7 Action du GVR click (définit dans le script gvr_button)	16
5. Conclusion	18
6. Complément	18
6.1 UI avec Case à cocher (Toggle)	18

1. Introduction

Arcore est le noyau google pour développer la AR. Installer Unity + arcore-unity-sdk-1.16.0.unitypackage + GoogleVRForUnity_1.200.1.unitypackage

Lors du build : soit on utilise Arcore et/ou VR (avec les outils GVR google VR) Matériel : un smartphone compatible Arcoore : liste ici : <u>https://developers.google.com/ar/discover/supported-devices</u> Mes tests sont effectués sur un samsungS7 (il faut au moins Android7.0 d'installé)

Utilisation avec unity : <u>https://developers.google.com/ar/develop/unity</u>

2. Premier test

2.1 Test de HelloAR

Installer le package ARcore pour unity Ouvrir la scène HelloAR située dans Example/HelloAR/scenes Compiler : build avec la configuration :

XR Settings	
Virtual Reality Supported 🔲	
ARCore Supported 🛛 🗹	
Vuforia Augmented Reality	

Tester sur le smartphone

Résultat : OK

La camera s'allume, visualisant l'environnement, la détection des surfaces planes se fait (plane) et en cliquant dessus une figurine AR apparaît :



2.2 Test AR + VR

Rebuild avec AR + VR cochée sur 'player setting'

XR Settings
Virtual Reality Supported 👿
Virtual Reality SDKs
😑 🔻 Cardboard
Depth Format 16-bit depth +
Enable Transition View
± -
Stereo Rendering Mode* Multi Pass
ARCore Supported
Vuforia Augmented Reality

Il faut ajouter un VR SDK on utilise 'cardboard'. Vuforia semble possible aussi mais comme ARcore permet les mêmes actions que Vuforia inutile de le prendre.

Ça marche : la VR et l'AR fonctionne.



2.3 Essai de détecter une image

L'example : AugmentedImage permet de tester la reconnaissance d'image.

Un cadre apparaît au dessus de l'image.

Les images sont stockées dans 'images'. Ce dossier contient aussi les informations sur la database image.



Résultat : ça marche

3. Premier programme : Essai du réticule VR

3.1 Installer GVR

On installe le googleVR (GVR) :



3.2 Installer le réticule

On installe dans la camera un réticule en utilisant le prefab de GVR : GrvreticulePointer



3.3 Créer l'UI

On crée un canvas avec un bouton :

Placer le canvas dans 'World space' à 0,0,0.

Placer le bouton dans le canvas à 0,0,0 : le déplacer afin qu'il rentre dans le champs de vision de la camera.

= Hierarchy		i≡ Hierarchy 🔒 📲	O Inspector 🔒 +≡
Create * Q*All	🔽 🔽 Canvas 🗌 Static 🔻	Create * Q*All	
▼ SbTestVRseul +≡	Tag Untransd the Laver (up the	▼ 🕄 SbTestVRseul -=	
GvrEventSystem >		GvrEventSystem >	Tag Untagged + Layer UI +
GvrEditorEmulator >	🛛 🔻 🖓 Rect Transform	GvrEditorEmulator	Rect Transform
Directional Light	Pos X Pos Y Pos Z	Directional Light	center POS X POS Y POS 7
🔻 🜍 Main Camera	0 0 185.3	Main Camera	
GvrReticlePointer >	Width Height	GvrReticlePointer	Width Height
💭 Cube	2560 1440 🔛 R	Cube	
MesScriptsGameObject	▶ Anchors	MacScriptsComeObject	h Anghara
MesImages	Pivot X 0.5 Y 0.5	Mesimages	Plant V DE V DE
🔻 🧊 Canvas		Capyas	Pivot X 0.5 Y 0.5
V O Button	Rotation X 0 Y 0 Z 0	Ruttop	Rotation X 0 Y 0 Z 0
🗍 Text	Scale X 1 Y 1 Z 1	Dutton	Scale X 1 X 1 7 1
		U Text	
	Pender Mode Wedd Grees		🕨 🕑 🛛 Canvas Renderer 🛛 🔛 🛱 🔅
	Event Comera		🕨 🍢 🗹 Image (Script) 🛛 🚺 🖈 🦊
	Event Camera (None (Camera) 0		🕨 🗹 Button (Script) 🛛 🔲 🚽 🔅
🖻 Project 🔒 📲	A World Space Canvas with no specified	Project	V - Cyr All Events Trigger 7
Create + Q 🐴 🔖	correctly.		Script GyrAllEventsTrip 0
Editor		Editor	On Deinten Clieb (Come Chiest, Deinten Fu
Fonts	Sorting Layer Default +	▶ 🚔 Fonts	On Pointer Click (GameObject, PointerEvi
Materials	Order in Layer 0	▶ 🚞 Materials	List is Empty
Models	Additional Shader Ch Nothing +	Models	
Plugins		▶ 🔤 Plugins	
Prefabs	V 🔛 🗹 Canvas Scaler (Script) 🛛 🔛 🐺	▶ 🚔 Prefabs	+ -
Scripts	UI Scale Mode World +	Scripts	On Pointer Down (GameObject, PointerE
Shaders	Dynamic Pixels Per U 1	▶ 🚔 Shaders	List is Empty
🕨 🚞 Textures	Reference Rivels Per 100	▶ 🚰 Textures	
GVRVideoPlayer		GVRVideoPlayer	
	🛭 🕨 🔣 Graphic Raycaster (Scrit 🛛 📓 🖈 🏟		+ -
			On Pointer Up (GameObject, PointerEver
🔻 🚞 images	Add Component	v 🚔 images	List is Empty
V Creticule02		v reticule02	
Teticule02		Terticule02	Putton 4

3.4 Créer la scène

On crée un cube dans la scène :



Placer le cube afin qu'il soit dans le champs de vision de la caméra.

3.5 Créer les scripts

On crée les scripts afin que :

le cube apparaisse si on vise le bouton et

le cube disparaît quand on sort du bouton avec le réticule :

Afin d'organiser la hiérarchie, je place mes scripts dans un GameObject que j'appelle : MesScriptsGamesObjects :

Ξ.	'⊞ Hierarchy		Inspector area
y	Create T Q*All		MesScriptsGameObject 🗌 Static 🔻
	GvrEventSystem		Tag Untagged + Layer Default +
	GvrEditorEmulator	>	🔻 🙏 Transform 🛛 🔯 🛱 🔅
	🔘 Directional Light		Position X 0 Y 0 Z 1
	🔻 问 Main Camera		Rotation X 0 Y 0 Z 0
	GvrReticlePointer	>	Scale X 1 Y 1 Z 1
	MesScriptsGameObject		🕨 🖃 🗹 Gvr_button (Script) 🛛 📓 🤿
	Cube		🔻 🖷 🛛 Affiche Cube Off (Script 🛛 🗐 🗐 🏶
	🔘 MesImages		Script afficheCubeOff 0
			Mon Cube 💿 Cube 🎯
			🔻 🝙 Affiche Cube (Script) 🛛 🔯 🤿
			Script afficheCube Ø
			Mon Cube 🔍 Cube Ø

3.5.1 Script AfficheCube :

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class afficheCube : MonoBehaviour
{
    public GameObject monCube;
    public void affichageCube()
    {
        monCube.SetActive(true);
        Debug.Log("afficheCube activé");
    }
}
```

3.5.2 Script AfficheCubeOff :

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
```

```
public class afficheCubeOff : MonoBehaviour
{
    public GameObject monCube;
    public void affichageCubeOff()
    {
        monCube.SetActive(false);
        Debug.Log("afficheCubeOff activé");
    }
}
```

3.6 Associer les scripts

Les scripts doivent se déclencher lorsque l'on vise le <mark>bouton</mark>. Dans le <mark>bouton,</mark> on ajoute un composant du type 'gvr all events trigger'. Ce dernier contient tous les déclencheurs qui nous intéressent.



Illustration 1: Script lorsqu'on rentre sur le bouton

Illustration 2: Script lorsqu'on quitte le bouton

3.7 Test du programme

Lorsque l'on vise le bouton le cube apparaît.



Illustration 3: Test en visant le bouton

Lorsque l'on quitte le bouton, le cube disparaît.



Illustration 4: Test en quittant le bouton

3.8 Conclusion

Ca marche !

On peut rapidement créer une app VR avec visée.

3.9 Création d'une app AR

En partant de la scène précédente, ajoutons le mode AR avec ARcore.

On inclut l'asset ARcore de google.

On remplace la caméra classique par une camera AR fournit par googleAR.



On garde tout le reste en faisant un copier coller entre la scène précédente et la nouvelle.

On coche ARcore dans player setting (et VR cardboard).

On 'build and run'

3.9.1 Résultats :

Essai 1 : NON

Ca marche sur le PC mais sur le smartphone le réticule ne suit pas la camera, il reste figé au centre !!!

Solution : En cochant 'ARCore required' dans 'Edit' + 'Project setting' ça marche !!

O Project Settings			
	Q		
Audio Editor Google ARCore Graphics Input Physics Physics 2D Player Preset Manager Quality Script Execution Order Tags and Layers TextMesh Pro Time	Google ARCore ARCore Required Instant Preview Enabled OS Support Enabled Cloud Anchor API Keys Android iOS		

Maintenant on va ajouter un script qui affiche une barre de progression sur le réticule.

4. Second programme : Essai de pointer avec le regard 'gaze timed'

A partir de la scène AR précédente on ajoute un canvas et une image de cercle.

On visant le bouton on déclenche l'affichage progressif du cercle ce qui sert d'indicateur de temps. On fixe 1s pour faire apparaître le cercle. A la fin le clic se fait afin de lancer l'action désirée.

Nous allons afficher le cube précédent au bout de 1s de visée sur le bouton. Une seconde visée efface le cube.

Remarque :

En travaillant à partir des déclencheurs 'gvr all events trigger' précédents le fonctionnement est aléatoire. En utilisant les 'Events triggers' classiques ça marche mieux. Par la suite nous les utiliserons.

Des tutos existent sur YouTube : Gaze timed.

Il faut :

4.1 Installer GVR (google VR)

L'installation se fait à partir du UnityHub.

Il est possible aussi de le télécharger

Les outils sont alors disponibles dans les Assets (démo et utilitaires)

Dans la scène on installe des éléments du prefab de GVR :



4.2 Inclure le réticule



4.3 Créer l'image animée

L'image animée est une image fixe (un cercle) qui va apparaître le temps du clic.

Create * Q*All	Image		Static 🔻
▼ 🚭 SbTestARgazetimed* →=	Tag	t Laver Lu	
GvrEventSystem >	indg Untagged		
GvrEditorEmulator >	▼35 Rect Transfor	m	🔛 🖬 📯
Directional Light	center	Pos x Pos y	Pos Z
V ARCore Device >		U U	0
Gurp aticle Deinter			
GarvesCaze		100	
	Plant		_
	Pivot	X 0.5 Y 0.5	
	Rotation	X O Y O	Z 0
▼ Button	Scale	X 0.1 Y 0.1	Z 1
Text			
			·····
		Station leads 3	
	Material	None (Material)	0
🖹 Project 🔒 📲	Raycast Target		
Create T Q 4 💊	Image Type	Filled	\$
	Fill Method	Radial 360	\$
	Fill Origin	Bottom	\$
	4 Fill Amount	O	0.474
	Clockwise		
	Preserve Aspect		
afficheCube		Set Native Si	ze
afficheCubeOff		- he siel	
gvr_button	Default OI Ma	ateriai	
▶ 🚔 PlayServicesResolver	► Shader UI/D	efault	¥

L'image 'reticule02' est créée avec Inskape pour profiter de la transparence.

On réalise juste un cercle vide.

On l'inclut dans le projet. Attention il faut lui donner la texture type 'Sprite (2D UI)'

On le place dans un canvas sous la camera.(2)+(3)

On contrôlera avec un script la propriété 'Fill amount' afin de faire apparaître le disque. (4)

4.4 Création du menu :

Ajouter un canvas et un bouton dedans.

Sur certain site on indique de changer le graphic raycatser n'en faites rien.



Repositionner le bouton dans 'world space'.

4.5 Créer un nouveau script pour le bouton :

Sélectionner le Button + Add component + New script + choisir un nom + Create and add

Editer le script...

On crée le script suivant :

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
using UnityEngine.Events;
public class gvr_button : MonoBehaviour
{
    public Image imgCircle; //l'image du cercle qui s'affiche
    public UnityEvent GVRClick; //l'évènement sur le bouton
    public float totalTime = 2; //le temps total pour afficher l'image
    bool gvrStatus; //l'état du processus
    public float gvrTimer; // le temps direct
    // Update is called once per frame
    void Update()
    {
        if (gvrStatus)
         {
            gvrTimer += Time.deltaTime;//increment le temps
            imgCircle.fillAmount = gvrTimer / totalTime;//affiche progressivement l'image
         }
        if (gvrTimer>totalTime)
         {
            GVRClick.Invoke();//si le temps den atteint on lance une action
         }
    }
    public void GvrOn()
    {
        gvrStaus = true;
    }
    public GvrOff()
```

14/20

```
{
    gvrStatus = false;
    gvrTimed = 0;//reinitialise le temps
    imgCircle.fillAmount = 0;//réinitialise l'apparition image
  }
}
```

On retourne sur Unity et on place les paramètres (toutes les valeurs public dans le script du bouton : l'image...)

4.6 Association des scripts aux déclencheurs :

Il faut maintenant ajouter un nouveau composant au bouton : 'add component' : un 'events trigger' (1) et on ajoute les triggers désirés par : 'add new event type'

On ajoute : 'pointer enter' (2) et 'pointer exit' (5)

Pour les deux, on ajoute un script par + et on place le 'button' sous 'runtime only' et on choisit le script dans le menu déroulant.(3-4) et (6-7)

On place la fonction : gvrOn() dans 'pointer enter' et gvrOff() dans 'pointer exit'.

Ces scripts ont pour effet de lancer l'affichage de l'image 'circle' lorsque le pointeur entre sur le bouton et de l'effacer quand il en sort. Si le pointeur reste le temps indiqué (10) alors un clic sur le bouton est généré (gvrClick())



Maintenant il faut définir l'action du GVRClick.

4.7 Action du GVR click (définit dans le script gvr_button)

Création d'un script simple d'affichage d'un cube si on clique :

```
//script afficheCube() by SB.
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
public class afficheCube : MonoBehaviour
{
    public GameObject monCube;
    public void affichageCube()
    {
        if (monCube.activeSelf)
         {
            monCube.SetActive(false);
            Debug.Log("afficheCube activé");
         }
        else
         {
            monCube.SetActive(true);
            Debug.Log("afficheCube activé");
         }
    }
}
```

On ajoute ce script au bouton :



Lancer le Run ...

Une erreur survient :



Pour la régler il faut ajouter la 'camera' dans le 'GVRReticule pointer'.

Ça marche sur le PC en configurant le 'player setting' avec le 'cardboard' et ARcore.

		Development Build		svrunitySakversion	Publishing securitys
	₽JE4 PS4	Autoconnect Profiler		ose3D iders	XR Settings
6.6	Universal Windows Platform	Scripts Only Build		:tures ₹∨ideoPlayer	Virtual Reality Supported 🖌 🖌 Virtual Reality SDKs
Clea	WebGL	Compression Method SDKs for App Stores	Default +	ENSE ADME	■ Cardboard 4 Depth Format 16-bit depth +
	Facebook	Xiaomi Mi Game Cente	r Add	iel cripts theCube	Enable Transition View
_	Player Settings 1	В	Learn about Unity Cloud Build uild Build And Run	irces s ningAssets utton	Stereo Rendering Mode* <u>Multi Pass</u> ARCore Supported Vuforia Augmented Reality
			🕨 🚔 Packa	ges 🔻	

Le résultat permet d'afficher le cube quand on vise le bouton un certain temps. Puis efface le cube quand on le revise.



5. Conclusion

La technique du 'gaze timed' permet de créer des UI complètements contrôlables par le réticule sans utiliser de contrôleur externe.

6. Complément

UI complet avec : case à cocher et slider...à venir

6.1 UI avec Case à cocher (Toggle)

Inclure une case à cocher dans l'UI

Inclure les scripts précédents pour gérer le réticule animé (image circle) : gvrOn()(2-3), gvrOff() (4-5).

Ajouter le script suivant qui va gérer l'action du clic :

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
public class gestionToggle : MonoBehaviour
{
    public Toggle myToggle;
    public void controlToggle()
    {
        if (myToggle.isOn) myToggle.isOn = false;
        else myToggle.isOn = true;
    }
}
```

```
Placer ce script dans le gvrclick()(6-7)
```



Résultat :

ok.

7. Test des fonctions reconnaissances photo de ARcore.

https://developers.google.com/ar/develop/unity/augmented-images/guide

Créer une base de donnée des images cibles.

Inclure les images dans le projet

Prévoir un dossier spécifique.

Sélectionner les images souhaitées et cliquez droit puis + Create > Google ARCore > AugmentedImageDatabase.

Une base de donnée est crée avec les images. Un pourcentage de qualité est présent pour chaque image. Eliminer les plus mauvaises (<75/100)



Indiquer les tailles (largeur) de l'image réelle afin de faciliter la reconnaissance.