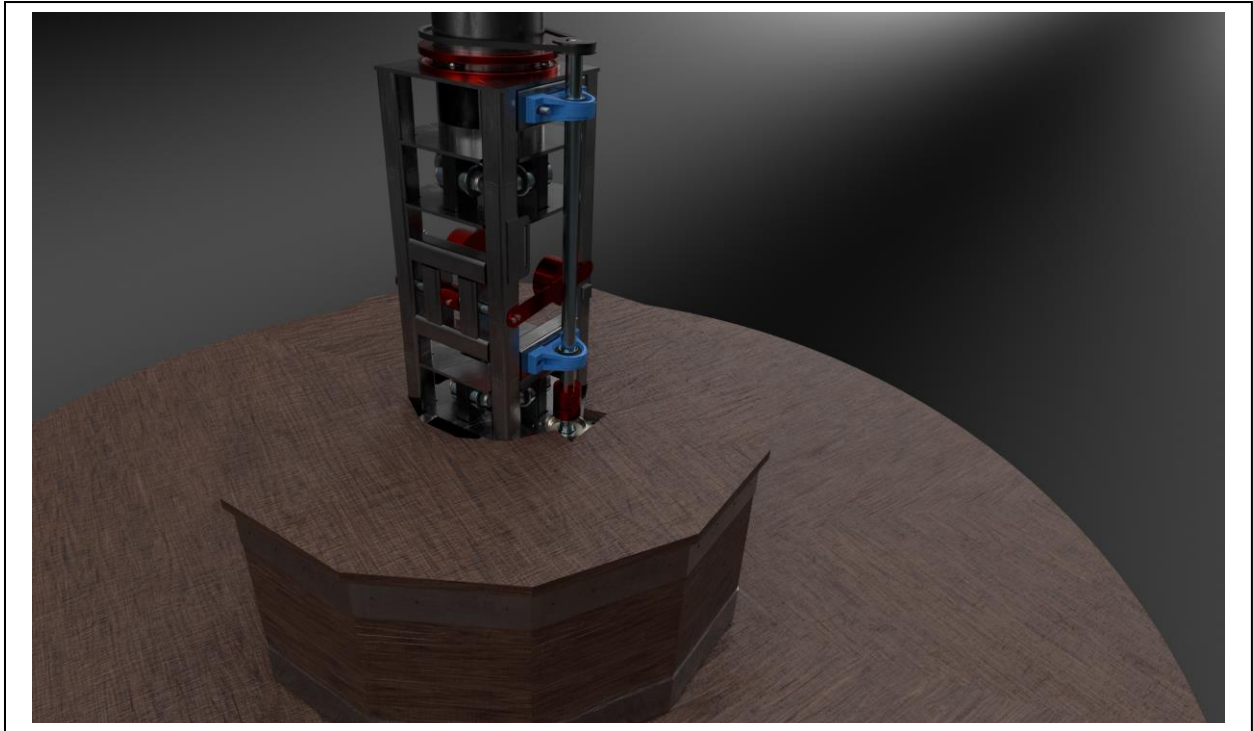


Exploitation de la Réalité augmentée en chaudronnerie



Exploitation de la réalité augmentée grâce à Unity et Vuforia :

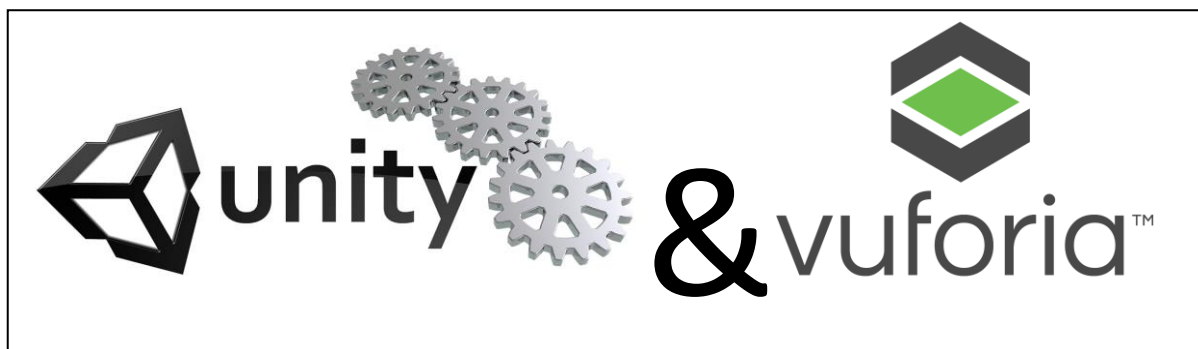


Table des matières

Table des matières	2
1-Préambule	3
2-Principe	3
3-Export de modèle SolidWorks en Objet « .Obj »	4
4-Installation de Blender	4
5-Objectif pédagogique	4
6-Vuforia.....	5
7-Les scripts :	5

LA REALITE AUGMENTEE EN CHAUDRONNERIE

1-Préambule

L'objectif de ce tutoriel est de vous permettre de développer vos propres applications en réalité augmentée sur Android. Ceci est une trame générale mais elle peut être adaptée à ce que l'on veut. J'ai acquis ces connaissances grâce à Internet donc je ne prétends pas avoir tout inventé. Je vous invite à faire vos propres recherches afin de répondre aux besoins qui sont les vôtres.

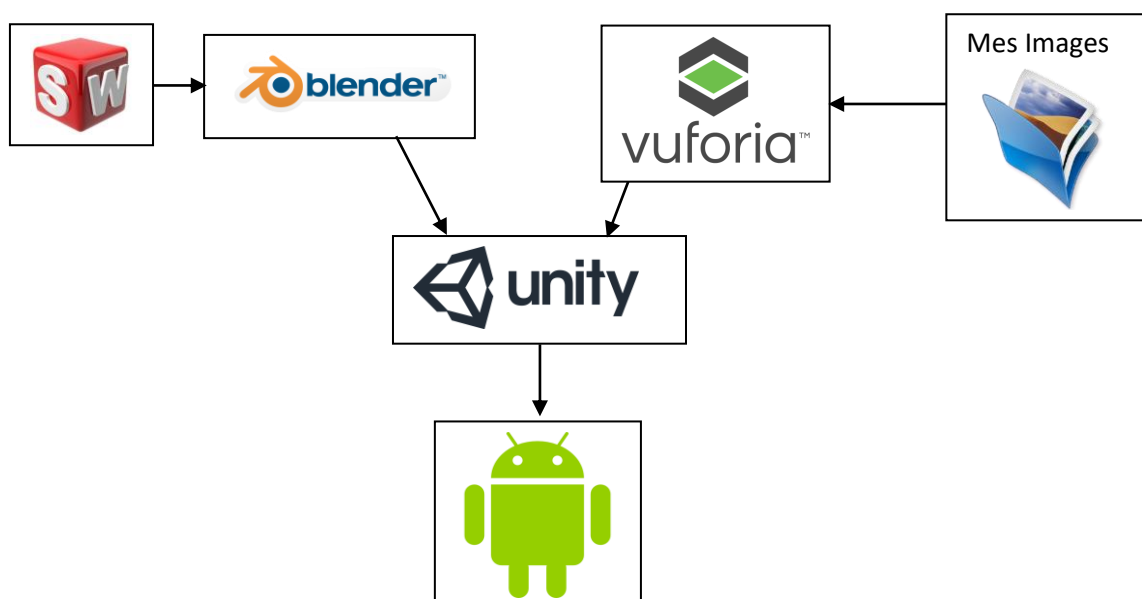
En me basant sur mon application, je me propose de vous expliquer, étape par étape, comment réaliser une application semblable. Dans un souci de concision, je me concentrerai sur les sept points suivants :

- Présentation de mon application en réalité augmentée
- Les bases de la réalité augmentée via Vuforia
- L'intégration de Vuforia dans Unity3D
- La gestion des modèles numériques depuis SolidWorks
- Les animations dans Unity3D
- Les « UI » dans Unity3D
- La gestion des scènes
- La compilation pour Android

A chaque point correspond un tutoriel, ces tutoriels doivent être fait en respectant leur chronologie. Les différents « packages » que j'ai fait de la Vis volante, et que j'utilise dans ces tutoriels, sont disponible via le Qr code Ci-contre :



2-Principe



3-Export de modèle SolidWorks en Objet « .Obj »

L'export des modèles numériques de SolidWorks en « .Obj » se fait par l'intermédiaire d'une macro créée par **Neil Larsen**. Elle est disponible par l'intermédiaire du Qr-Code ci-dessous.



Une vidéo décrivant l'installation de la macro sur SolidWorks est disponible par l'intermédiaire du Qr-Code ci-dessous :



4-Installation de Blender

Blender est logiciel de modelage numérique très puissant et qui a l'avantage d'être gratuit. Vous pouvez le télécharger à l'adresse suivant : <https://www.blender.org/download/>.

Nous ne nous servons de ce logiciel que pour convertir nos « .obj » en « .fbx » exploitable par Unity3D.

Remarque : D'autres logiciels de modélisation sont également utilisables. Je pense à **3Dmax** d'Autodesk, disponible gratuitement pour les enseignants, ou **Maya** qui est une solution payante.

5-Objectif pédagogique

Compétence visée : C4 Interpréter et vérifier les données de définition de tout ou partie d'un ensemble chaudronné

Niveau : Classe de 1^{ère} TCI, début voire milieu d'année, après les liaisons mécaniques

Type : Travaux dirigé : l'objectif ici est de prendre le temps d'expérimenter les différents mouvements d'un schéma cinématique, de prendre conscience du rapport entre la portée d'une liaison et sa représentation schématique ainsi que l'influence du jeu en fonction de la portée.

6-Vuforia

Vuforia permet la reconnaissance d'image et de volume en réalité augmentée. Vuforia est disponible à l'adresse suivantes :

<https://developer.vuforia.com/>

Utiliser Vuforia exige de créer un compte, cette solution est gratuite mais elle est soumise à certaines conditions décrites dans le tutoriel.

7-Les scripts :

Script de navigation depuis la scène « page de garde » :

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

public class Menu : MonoBehaviour
{
    public void Page1()
    {
        SceneManager.LoadScene("DessinDensemble");
    }

    public void Sortir()
    {
        Application.Quit();
    }
}
```

Script de gestion de l'animation via un Slider : il peut être utilisé plusieurs fois il suffit de renseigner les valeurs suivantes et de mettre ce script dans le GameObject possédant l'Animator :

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

public class SliderAnimator : MonoBehaviour
{
    public Slider slider;
    private Animator animator;

    [SerializeField]
    string nomAnim;

    void Start()
    {
        animator = GetComponent<Animator>();
    }

    public void OnValueChanged()
    {
        animator.Play(nomAnim, -1, slider.normalizedValue);
    }
}
```

Script de mise en pause :

<pre>using System.Collections; using System.Collections.Generic; using UnityEngine; using UnityEngine.UI; public class Pause : MonoBehaviour { private bool pause; void Start() { pause = false; } public void PauseButton() { if (pause) { pause = false; Time.timeScale = 1; } else { pause = true; Time.timeScale = 0; } } }</pre>	<p>Création d'une variable booléenne « pause » prenant la valeur soit vraie soit fausse :</p> <p>« pause » = true (animation en pose)</p> <p>« pause » = false (animation en</p>
	<p>Au démarrage de l'application la variable pause passe à faux</p>
	<p>La méthode « PauseButton » vient vérifier :</p> <ul style="list-style-type: none"> - si pause est égale à vraie, si c'est le cas alors il passe pause à faux et met l'échelle de temps à 1 - si pause est égale à faux alors il passe pause à vraie et met l'échelle de temps à 0

Script de retour à la scène « Page de garde », ce script est à placer dans chaque scène hors scène « Page de garde ».

<pre>using System.Collections; using System.Collections.Generic; using UnityEngine; using UnityEngine.SceneManagement; public class RetourMenu : MonoBehaviour { void Update() { if (Input.GetKeyDown(KeyCode.Escape)) { SceneManager.LoadScene("Page de garde"); } } }</pre>	<p>Cette méthode « Update » est une méthode qui se réalise à chaque image, si votre application tourne à 60 images par seconde alors la méthode se fera 60 fois par seconde. Ces méthodes sont gourmandes en ressources et doivent être utilisées avec parcimonie. Ici on vient détecter à chaque image si la touche « escape » est sollicitée.</p>
--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------