

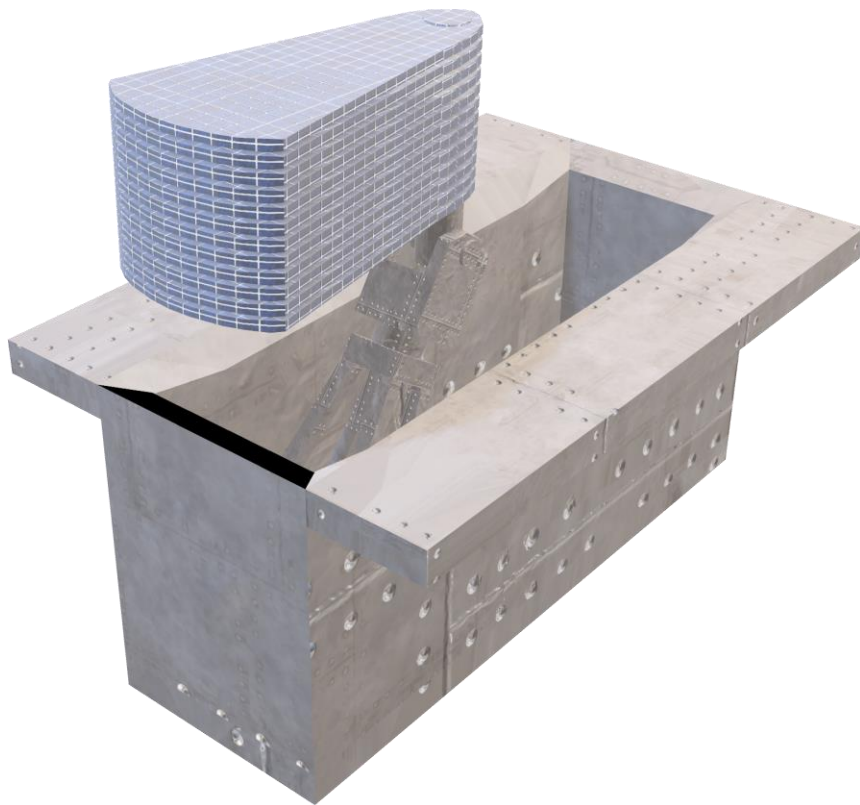
# Computer Graphics

**Etudiant:** Minh-Hoang DANG

**Matiere:** Computer Graphics

**Responsable:** Thierry BOUWMANS

**Date:** 23/03/2019



# Sommaire

---

Introduction .....	2
Modélisation.....	2
Simulation .....	3
Interface.....	3
Animation.....	3
Conclusion.....	4

## Introduction

Le projet consiste à démontrer le fonctionnement du SmartFlower, une solution photovoltaïque qui optimise son rendement en tournant vers le soleil, comme les tournesols. Avec le projet, j'ai eu l'opportunité d'expérimenter les connaissances sur plusieurs domaines, notamment la modélisation et la simulation. Ce rapport va vous guider à travers tous les étapes du développement du projet : modélisation, animation, simulation.

Le code source du projet est entièrement disponible sur GitHub, ainsi qu'une application pour la simulation sous Windows (64 bits).

**Mots clés : Modélisation, Animation, Infographisme, Unreal Engine**

## Modélisation

On distingue 3 composants : le conteneur, la tige et les pétales.

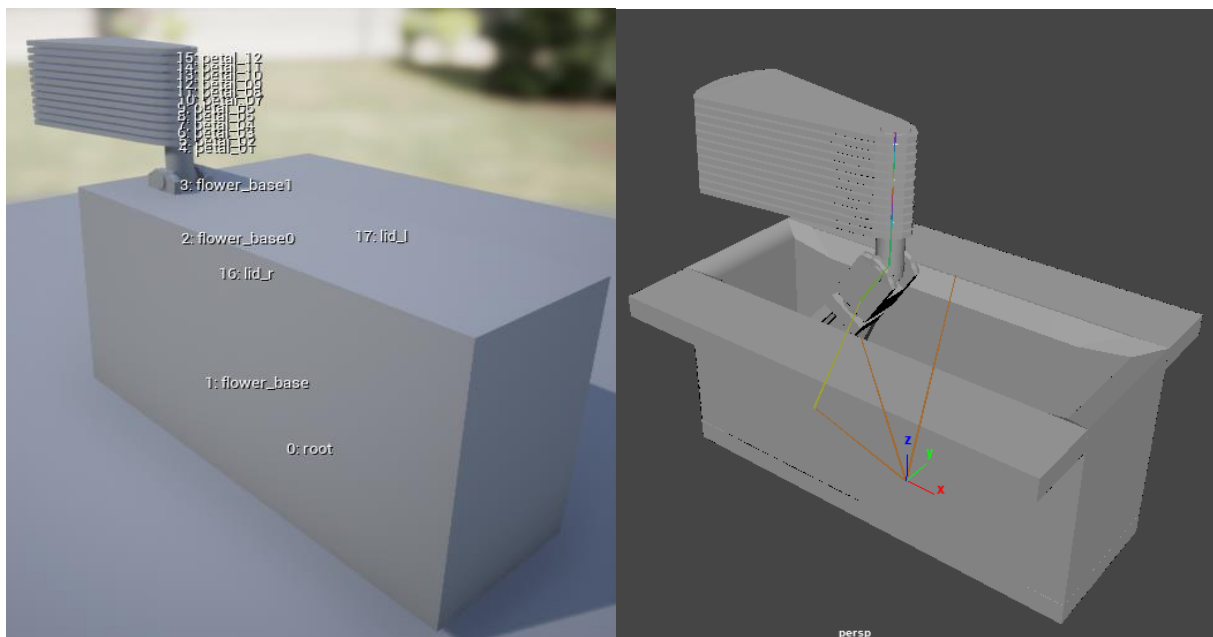
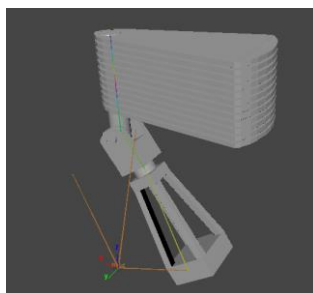
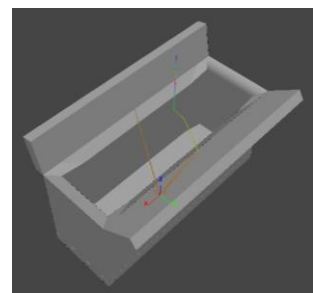


Figure 1. Le modèle et la squelette



**Le conteneur** est une simple boîte rectangulaire avec les portails pour cacher la fleur. L'intérieur est modélisé également.

**La tige** est séparée en 3 bases. De bas en haut avec l'élément du squelette qui les gouverne : le support (**flower\_base**), le moteur pour la rotation en lacet (**flower\_base0**) et le moteur de rotation en tangage (**flower\_base1**).



Les pétales sont copiés à partir du premier pétale : on conserve donc la structure et peut appliquer une même texture. Lors de l'animation de déploiement, la boîte s'ouvre, la fleur s'émerge et chaque pétale tourne 30 degrés. **L'animation** de retraction est une copie exacte de l'autre, mais avec la clé gouvernant le temps inversé. Finalement, **les matériaux** sont fait rapidement grâce à **Substance Painter**. Le UV Mapping est fait par projection de 6 planes, qui donne un résultat propre.

# Simulation

Unreal Engine 4 est un outil puissant pour ce faire. Le système Blueprint permet de réaliser rapidement un prototype sans passer par C++. Cependant, celui-ci devient peu présentable quand le code devient complexe. Le découpage fonctionnel est donc important.

## Interface

L'interface est simple : un menu permettant de modifier la date, l'heure et la géolocalisation de l'observateur.

## Animation

Pendant la phase „Modélisation”, on a généré deux animations statiques. Pendant la simulation, on peut modifier les données (rotation) via le Animation Blueprint (SmartFlower\_ABP). On veut cibler deux composants, **flower\_base0** pour la rotation en lacet et **flower\_base1** pour la rotation en tangage.

Il y a une multitude de variables qu'on souhaite mettre à jour à chaque tick. Ces variables vont influencer directement le comportement des composants visés.

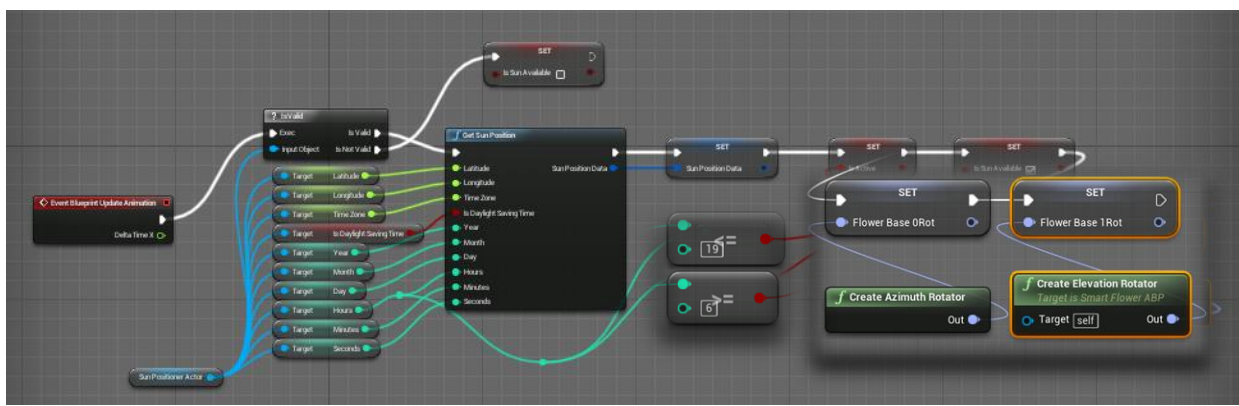


Figure 2. Extrait du EventGraph: l'heure active sera de 6h-20h.

La rotation de chaque composant est calculée avec une fonction :

- Flower\_base0: Create Azimuth Rotator
- Flower\_base1: Create Elevator Rotator

Unreal Engine 4 par défaut offre un plugin permettant de positionner le Soleil (en effet une source lumière directionnelle dans l'environnement) en fonction de la date et la géolocalisation de l'observateur puis calculer les angles d'élévation et d'azimut entre l'observateur et le soleil.

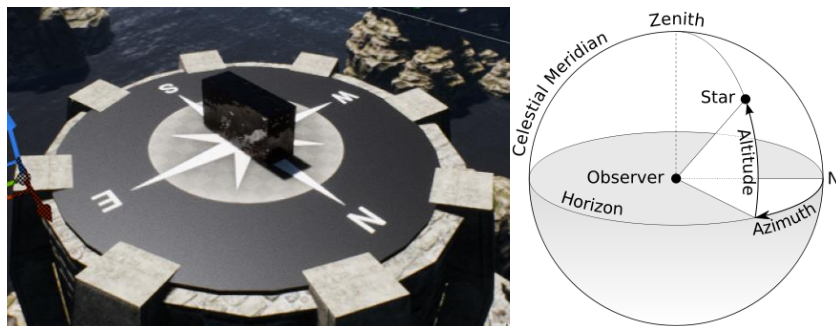


Figure 3. L'observateur (la boussole) est placé au même endroit que la fleur

On va ensuite distinguer 2 moments : active – la fleur est déployé et pose – la fleur est retractée.

Les modifications sur les composants ciblés seront utilisées ici pour modifier la séquence d'animation elle-même.

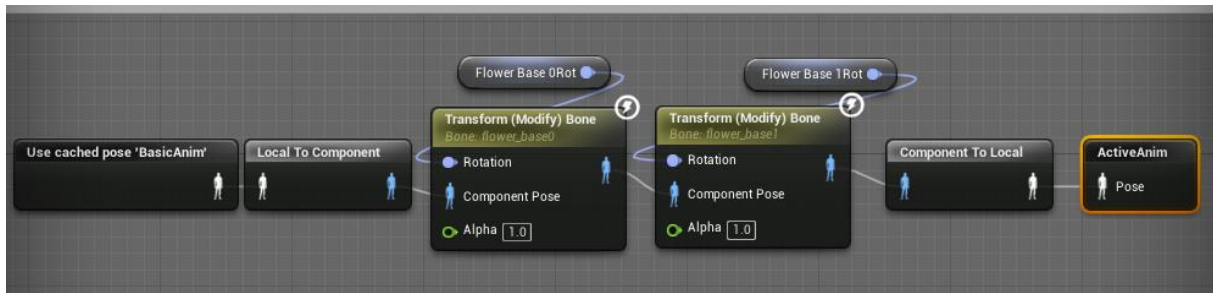


Figure 4. Les deux cibles sont bien dirigées par les variables calculées

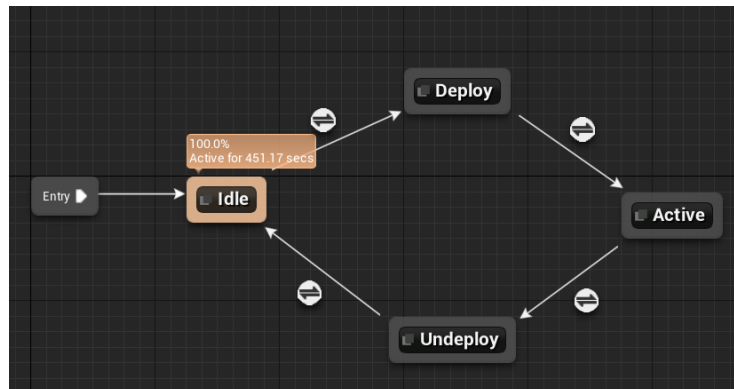


Figure 5. La machine à état décrivant la cycle d'animation

## Conclusion

Grâce à ce projet, j'ai eu le temps pour m'améliorer dans l'utilisation des outils : Autodesk Maya, Unreal Engine, Substance, etc.. que je n'utilise que pour le hobby. Ayant participé aux projets personnels de ce type précédemment, je n'ai pas de difficulté pour me servir des outils. Dans la démonstration, vous allez voir que la fleur de se synchronise pas parfaitement son angle d'élévation. Il s'agit d'un problème mathématique mais le temps ne me permet pas d'aller jusqu'au bout du débogage (autres projets en cours). Beaucoup de chose reste à améliorer pour la partie simulation. Par exemple, la partie programmation aurait pu être réalisé avec C++ au lieu de Blueprint, qui est peu présentable quand le projet grandit. En outre, une séquence de vidéo (avec Unreal Engine 4) n'a pas pu être finalisé à temps.

