Année 2016 / 2017

Rapport de projet tuteuré 4

Rapport du projet de réalité augmentée

Delienne Flavien
IUT SAINT-DIE DES VOSGES

TABLE DES MATIERES

INTRODUCTION

La réalité augmentée :

La réalité augmentée est de plus en plus présente dans notre vie quotidienne. Celle-ci permet de superposer des objets réels / scènes réelles (une photographie dans le cadre de mon projet) à des éléments virtuels comme des sons, des images 2D ou 3D, des vidéos, … dans le but de faire en sorte que l'élément virtuel rajouté s'intègre le mieux possible dans son décor.

Le projet Hololens

Cette image explique bien ce principe, en effet la grande firme américaine Microsoft a pour projet avec ses lunettes à réalité augmentée nommée Hololens de pouvoir utiliser un ordinateur sans clavier ni souris.

Mon projet :

Mon projet consiste à ajouter un objet 3D animé (un ballon de football pour ma part) à une photographie en y ajoutant une source de lumière à l'aide des bibliothèques PoLaR (Portable Library for augmented Reality) et OpenSceneGraph. Pour se faire, j'ai utilisé l'API Qt et le langage de programmation C++.

I) PREPARATION DU PROJET

A) INSTALLATION

Dans un premier temps pour commencer ce projet il a fallu installer PoLaR, la bibliothèque que j'ai utilisée.

Il a fallu d'abord installer ses dépendances en entrant la commande "apt-get install <nom du paquet>" qui sont :

- Qt, où j'utilise les 7 paquets suivants : qt5-default**,** qttools5-dev-tools**,** libqt5opengl5-dev**,** qtmultimedia5-dev, libqt5multimedia5-plugins, stdeclarative5-dev, libqt5svg5-dev).
- OpenSceneGraph : openscenegraph, libopenscenegraph-dev
- OpenGL : freeglut3, freeglut3-dev, libxmu-dev, libxi-dev
- Gstream : libgstreamer0.10-0 , libgstreamer0.10-dev, gstreamer-tools, gstreamer0.10-plugins-good, gnome-media
- Doxygen (pour la documentation) : doxygen, graphviz

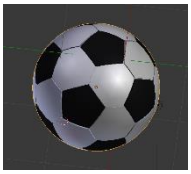
Ensuite, il faut télécharger PoLaR soit via un répertoire git (via la commande : git clone https://scm.gforge.inria.fr/anonscm/git/polar/polar.git) où sois directement sur le site.

L'installation de PoLaR étant maintenant effectuée, il est possible de faire des tests via les exemples fournit dans le dossier Examples pour vérifier l'intégrité de l'installation.
Dans ce dossier, on trouve un fichier texte README qui indique la marche à suivre pour tester les différents exemples avec la commande permettant les exécuter.

B) CREATION DE L'OBJET 3D

Tout d'abord pour commencer le projet, j'ai dû créer un ballon de football en 3D via le logiciel Blender.

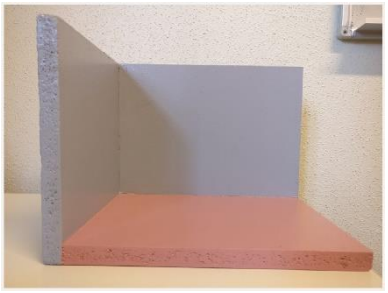
J'ai dû comprendre comment le logiciel fonctionne ainsi que comment j'allais pouvoir faire mon Objet3D, le ballon.



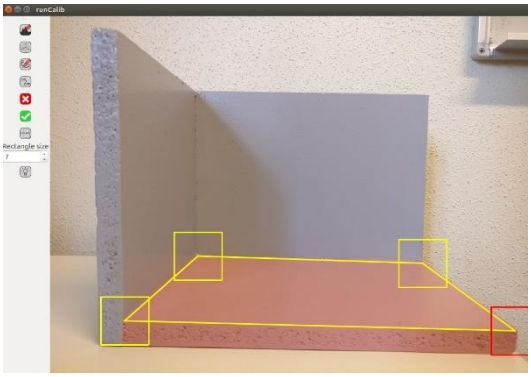
J'ai commencé en créant une sphère puis, au fil de la modélisation j'ai obtenu ce résultat.
Cela m'a pris du temps car, ne connaissant pas ce logiciel auparavant, je ne comprenais pas le fonctionnement de la caméra car je ne connaissais pas les raccourcis qui la rendait bien plus aisée.

C) L'IMAGE

J'ai choisi cette image pour réaliser mon projet, il s'agit d'un décor fait pour la réalité augmentée fournit par mon tuteur M. Pierre-Frédéric Villard.



J'ai par la suite utilisé le logiciel runCalib de PoLaR pour calculer la matrice de projection pour pouvoir intégrer mon objet 3D (mon ballon).



Je clique sur le bouton « Edit markers » et ensuite je fais un clic-molette sur les quatre coins de ce qui représente le sol du décor.

Puis je clique sur « Print infos » qui va générer un fichier texte contenant la matrice de projection.

II) REALISATION DU PROJET

A) ADAPTATION DE L'INTERFACE

Pour réaliser mon projet, j'ai réutilisé en grande partie l'interface réalisée pour notre projet tuteuré 3 qui présentait ce dont j'avais besoin pour réaliser mon projet.

J'ai donc gardé tous les éléments de l'interface sauf la check box (case à cocher) pour colorier l'objet en rouge.

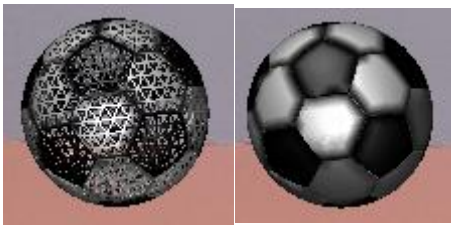
La combobox (liste déroulante) sert à choisir un objet, cela permet de pouvoir ajouter un nouvel objet3D en ayant une interface toujours fonctionnelle qui permet les modifications sur tous les objets.

Les 2 sliders permettent d'effectuer une rotation selon 2 angles.

La checkbox Visible permet de cacher l'objet quand on la décoche, pour l'afficher à nouveau lorsqu'on la coche à nouveau.

Après avoir décoché la checkbox voici le résultat :

La checkbox Mesh permet de modifier le style d'affichage de l'objet (maillage ou plein).



B) AJOUT DE L'OBJET 3D

L'ajout de mon modèle se fait de la façon suivante :

```
// Charge la matrice de projection
osg::Matrixd P = PoLAR::Util::readProjectionMatrix("../Examples/data/matrix.txt");
inter.setProjection(P, pt);


// Ajout du modèle
if(!PoLAR::Util::fileExists("../Examples/data/ballon.obj"))
{
    std::cerr << "Unable to load model" << std::endl;
    exit(0);
}
else
{

    osg::ref_ptr<PoLAR::Object3D> ballon = new PoLAR::Object3D("../Examples/data/ballon.obj", true, true, true);
    osgUtil::SmoothingVisitor smooth;
    osgUtil::Optimizer optimizer;
    optimizer.optimize(ballon.get(), osgUtil::Optimizer::ALL_OPTIMIZATIONS);
    ballon->accept(smooth);
    ballon->setName("ballon");
    ballon->translate(3, 2, -1);
    ballon->scale(0.6f);
    inter.addNode(ballon);
    inter.getViewer()->setTrackNode(ballon.get()); // set this object to be tracked by the manipulator
```

On charge d'abord la matrice de projection, puis on vérifie si le fichier existe.
Si non, on obtient l'erreur stipulant « Unable to load model », si oui, on charge celui-ci avec la dite matrice et on le positionne de façon cohérente à l'échelle désirée.

C) REALISATION DE L'OMBRE

On ajoute un sol fantôme qui permet l'affichage de l'ombre en fonction d'une lumière que l'on crée.

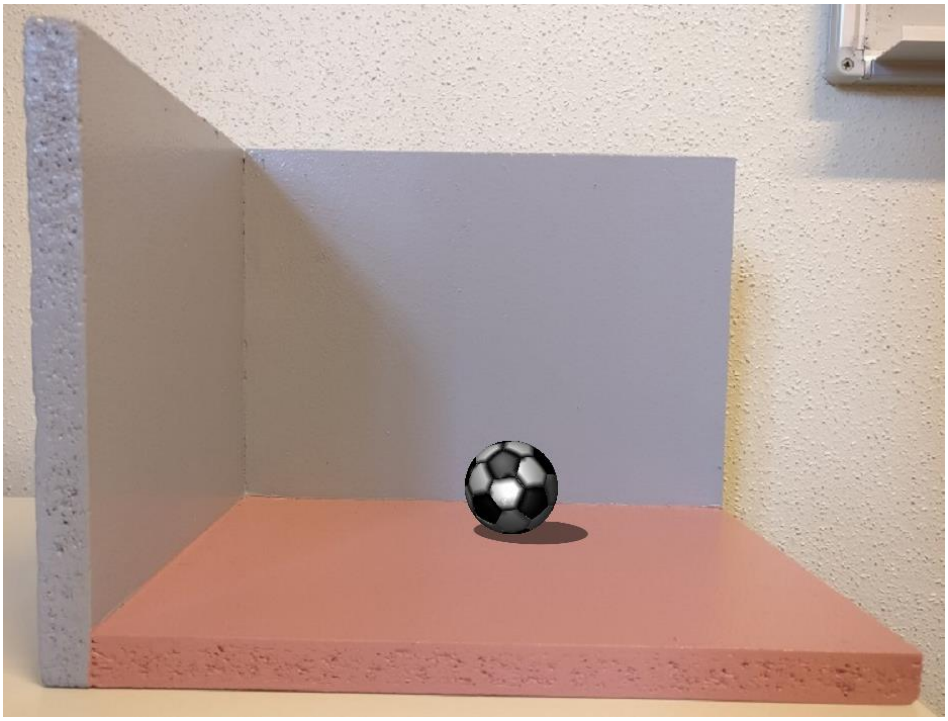
```
// Create a Viewer with shadows management
viewer = new PoLAR::Viewer(this, "viewer", 0, true);
viewer->addEventHandler(new osgViewer::StatsHandler);// add stats handler

//Ajout du sol pour les ombres
osg::ref_ptr<osg::Geode> sol = new osg::Geode;
osg::ref_ptr<osg::ShapeDrawable> shape;
shape = new osg::ShapeDrawable(new osg::Box(osg::Vec3(5.5f, 4.0f, 0.0f), 10.2f, 6.0f, 0.00001f));
sol->addDrawable(shape.get());
osg::ref_ptr<PoLAR::Object3D> object3D = new PoLAR::Object3D(sol.get());
object3D->setName("sol");
object3D->setPhantomOn();
viewer->addObject3D(object3D.get());
object3D->setEditOn();

//ajout de la source de lumière
viewer->addLightSource(2,2,-5, true);
viewer->show();
```

On crée le Viewer auquel on ajoute le sol fantôme.

On ajoute une lumière en hauteur et on obtient le résultat suivant :



CONCLUSION

Pour conclure, dans la continuité de mon projet tuteuré 3 sur la cage thoracique, ce projet m'a permis de mieux comprendre le C++ que je n'avais pas utilisé auparavant ainsi que de découvrir toutes les possibilités qu'offre PoLaR dans le domaine de la réalité augmentée.
Ce projet m'a aussi permis d'approfondir mes connaissances avec les différents outils utilisés et d'en découvrir de nouveaux.

Si je devais le refaire, je ne ferais pas les mêmes erreurs qui m'ont coûté beaucoup de temps et, avec les connaissances acquises, j'aurais pu faire la totalité du projet plus rapidement et aussi rajouter des fonctionnalités quant à la source de lumière.