

## **Rapport Projet OpenGL**



Bébé lapinou qui désirait conquérir le monde

<b>Introduction</b>	<b>2</b>
<b>Comment utiliser le jeu</b>	<b>3</b>
<b>Compilation</b>	<b>3</b>
<b>Fonctionnalités implémentées</b>	<b>3</b>
<b>Sources</b>	<b>4</b>

## Introduction

Notre projet se passe dans un musée d'art moderne composé de tableaux et de sculptures. Les modèles 3D ont été réalisés sur Blender ou bien sont openSources.

Notre personnage principal, un lapin curieux, peut se balader dans ce a pied ou alors en se plaçant dans un petit wagon qui lui fera faire le tour de la pièce.

Notre musée et nos rails sont en forme de lapin comme nous pouvons le voir sur la figure 1. Le but initial du lapinou était de conquérir le monde en étant maître de la culture et des art, mais le manque de temps lui résigna à n'être qu'un visiteur de musée.

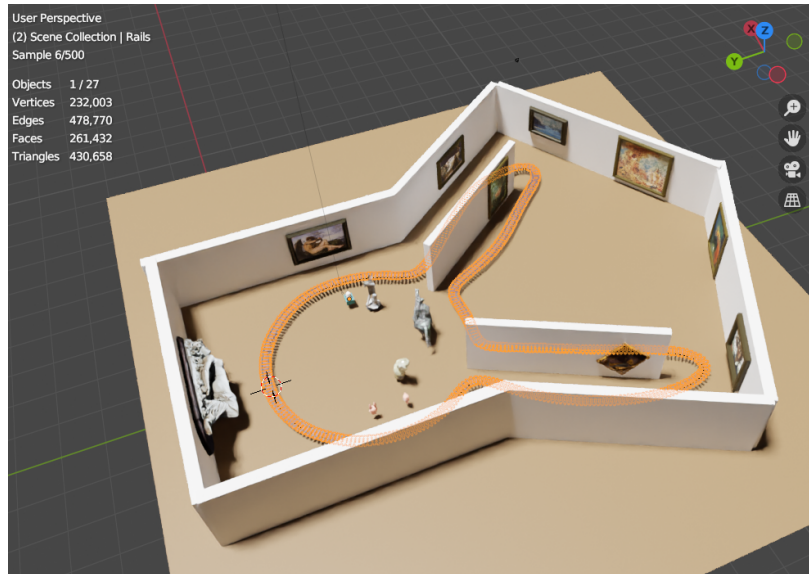


Figure 1 : La scène du jeu modélisée sur Blender

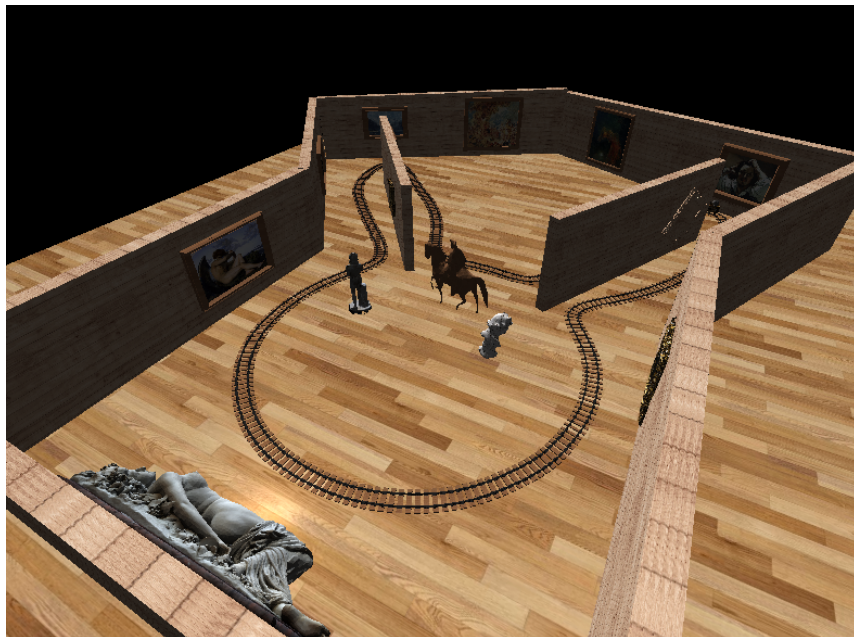


Figure 2 : La scène du jeu sur OpenGL

## Comment utiliser le jeu

- Lorsque le lapin est dans le wagon, il ne peut pas se déplacer. Par contre, il peut tourner la tête dans le sens qu'il veut pour regarder autour de lui.
- Pour sortir du wagon ? cliquer sur **H** pour bouger librement.
- Pour entrer dans le wagon ? cliquer à nouveau sur **B** pour être attaché.
- Pour passer en mode jour, appuyer sur **J**, et en mode nuit sur **N**.
- Lorsque le lapin n'est pas dans le wagon il peut se déplacer normalement grâce à Freefly (**Q**, **Z**, **S**, **D**) et donc peut voler et regarder la scène d'en haut.

## Compilation

Pour lancer le projet il suffit de se placer dans le répertoire build et d'écrire à partir d'un terminal les commandes suivantes :

```
cmake ..  
make  
./ProjetBase/ProjetBase_FinalProject
```

Notre projet est organisé de la même manière que dans les TP en général mais nous avons ajouté dans le dossier *assets* le dossier *models* qui regroupe tous nos modèles 3D utilisés dans le projet.

## Fonctionnalités implémentées

Nous avons besoin de faire certaines fonctionnalités pour réaliser ce projet :

- Une caméra mobile : Nous avons utilisé la caméra FreeFly pour ce projet. Notre caméra suit le wagon lorsque nous sommes positionnés dedans. Nous avons également la possibilité de bouger la caméra à la main pour que le personnage regarde autour de lui lorsqu'il se trouve dans le wagon. Si le lapin se trouve en dehors du wagon, il peut déplacer normalement dans le terrain.
- Deux objets texturés : Nous avons plusieurs objets dans notre monde comme les différentes œuvres d'art, les personnages principaux et secondaires ainsi que le chariot. Nos objets ont tous une texture.

- Terrain : Nous avons modélisé un terrain très simple qui est notre musée. Il y a donc un décor, des murs et un sol.
- Deux sources de lumières : Nous avons mis deux sources de lumières. Une lumière directionnelle tel que le soleil ainsi qu'une lumière ponctuelle. Nous avons une light directionnelle dirigée vers la sculpture murale Bacchante et une autre lumière ponctuelle qui est positionnée sur la caméra, de sorte à éclairer un peu les zones observées.
- Un Rollercoaster contenant un virage : Nous avons modélisé notre parcours avec des rails en forme de lapin. Il y a donc plusieurs virages dans notre décor. Le rollercoaster est construit à l'aide d'une spline sur blender et est accessible à partir de son fichier .obj.
- Un wagon : Nous avons modélisé sur Blender un wagon qui sera posé sur les rails. Notre wagon suit le parcours et le sens des rails pour avoir une trajectoire réaliste au maximum.
- Un mode jour nuit : Nous avons considéré le mode nuit comme une absence de directional light, et le mode jour comme sa présence.

Pour ajouter toutes ces fonctionnalités nous avons ajouté plusieurs choses. Nous avons par exemple dû ajouter des fichiers dans glimac qui se nomme objloader.cpp et objloader.hpp qui permettent de lire les fichiers .obj que nous avons créés sur Blender et de les ouvrir pour les mettre dans notre décor <sup>[1]</sup>.

#### **Remarques :**

Les modèles 3D et textures sont insérés de façon brute, utilisant un parser implémenté soi-même, en raison de problèmes irrésolus d'utilisation de la librairie Assimp.

## **Sources**

[1] : <http://www.opengl-tutorial.org/beginners-tutorials/tutorial-7-model-loading/>