

Rapport du projet Polytech Wars

Raphaël Bresson, Mahdi Hammouche

19 janvier 2016

Table des matières

Description du projet	3
L'objectif global	3
Les tâches réalisées	3
Fonctionnement du programme	4
Manuel d'utilisation et d'installation	5
Bilan	7
Nos fiertés	7
Axes d'améliorations	7
Rendre à César ce qui est à César	8

Description du projet

L'objectif global

Implémentation d'une maquette de jeu pouvant simuler la navigation à bord du célèbre vaisseau Falcon Millenium de Yan Solo.

Les tâches réalisées

Gestion de plusieurs caméras (points de vue sur le monde 3D). Une de type FPS (jeu de shoot) libre orientée à la souris et déplacée au clavier gérant le zoom. L'autre de type simulateur de vol pour contrôler le vaisseau. On peut passer de l'une à l'autre à l'aide de la touche V.

Gestion sommaire de la GUI avec l'affichage du texte.

Gestion des sons (musiques et effets sonores.

Importation de meshes externes.

Implémentation de formes basiques (sphères et cubes) pour représenter les planètes et la skybox.

Utilisation de bibliothèques open source uniquement.

Fonctionnement du programme

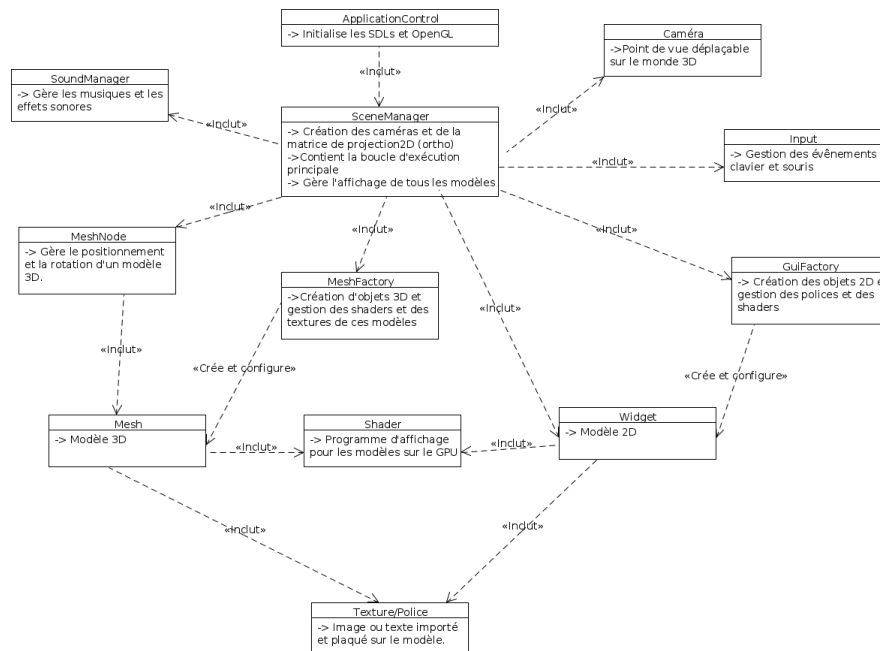


FIGURE 1 – Diagramme fonctionnel global

Pour plus d'information et des diagrammes de classe détaillés : se référer à la documentation du projet.

Manuel d'utilisation et d'installation

Bibliothèques utilisées Ce programme utilise les bibliothèques SDL 2.0 (gestion de la fenêtre et création du contexte OpenGL), SDL2.0-image (importation des textures), SDL2.0-TTF (gestion du texte), SDL2.0-mixer (gestion des sons), Lib3DS pour l'importation des modèles 3D et enfin OpenGL pour le rendu et GLM (OpenGL Mathématiques) pour les calculs géométriques.

Installation Des dossiers contenant les codes sources de ces bibliothèques sont disponibles dans l'archive du projet. Un script lui-aussi présent permet de toutes les compiler et les installer en une seule commande (droit de super-utilisateur requis).

Test Un programme minimal SDL + OpenGL (affiche deux triangles et quitte) permet de générer un fichier d'erreur afin d'éliminer celles qui viennent des pilotes de carte graphique (surtout ATI-RADEON) et de la SDL2. Il est situé dans le dossier test.

Compilation et utilisation Une fois les dépendances installées vous pouvez compiler le programme. L'utilisation de celui-ci est simple :

Commandes clavier ne dépendant pas de la caméra Tapez sur V pour changer de caméra. Espace pour déclencher un tir (juste le son pour l'instant).

Caméra FPS En mode caméra FPS vous pouvez vous orienter à l'aide de la souris et vous déplacer à l'aide du clavier : Z pour avancer, S pour reculer, Q pour se déplacer sur la gauche, D pour se déplacer sur la droite. On peut aussi zoomer + pour zoomer, - pour dézoomer.

Caméra simulateur de vol En mode simulateur de vol vous pouvez déplacer le vaisseau. Vous pouvez contrôler la vitesse à l'aide des touches directionnelles Haut pour accélérer et Bas pour ralentir. Vous pouvez orienter le vaisseau à l'aide des touches Z,A,Q,S,D,E (3 degrés de liberté).

Bilan

Nos fiertés

Avoir réussi à maîtriser suffisamment OpenGL pour pouvoir faire un petit moteur 3D basique en assez peu de temps.

Importation de modèles 3DS.

La gestion des caméras

Axes d'améliorations

Gestion des collisions.

Déplacement des planètes en accord avec les lois de Kepler.

Amélioration de la partie GUI pour pouvoir gérer des boutons et des panneaux d'affichage et de configuration.

Ajout des particules pour pouvoir gérer les explosions en cas de collision ainsi que les tirs.

Gestion des sources de lumière.

Rendre à César ce qui est à César

Images de planètes : NASA

[http : //maps.jpl.nasa.gov/](http://maps.jpl.nasa.gov/)

Modèle du vaisseau Falcon Millenium

[http : //www.scifi3d.com/details.asp?intGenreID = 10&intCatID = 10&key = 4](http://www.scifi3d.com/details.asp?intGenreID=10&intCatID=10&key=4)