Portfolio Quentin Gasparotto

Introduction :

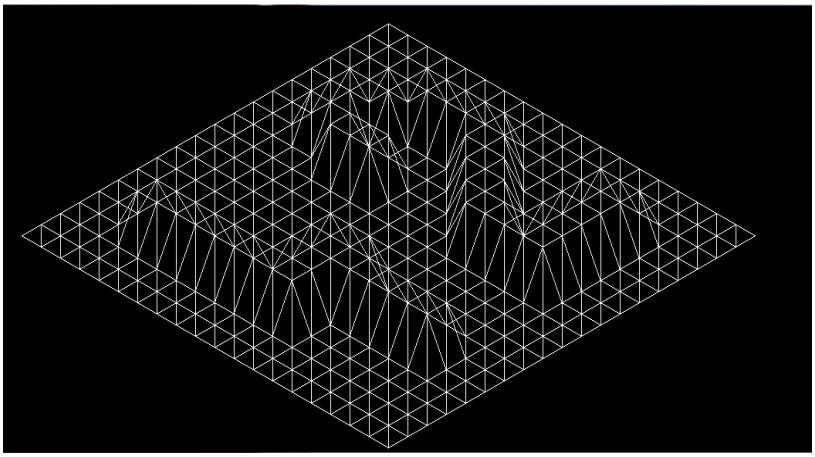
Je vous propose de découvrir mes réalisations dans ce portfolio. J'y parle des projets que j'ai eu plaisir à réaliser au cours de ma scolarité à Epitech dans les technologies que j'ai le plus apprécié.

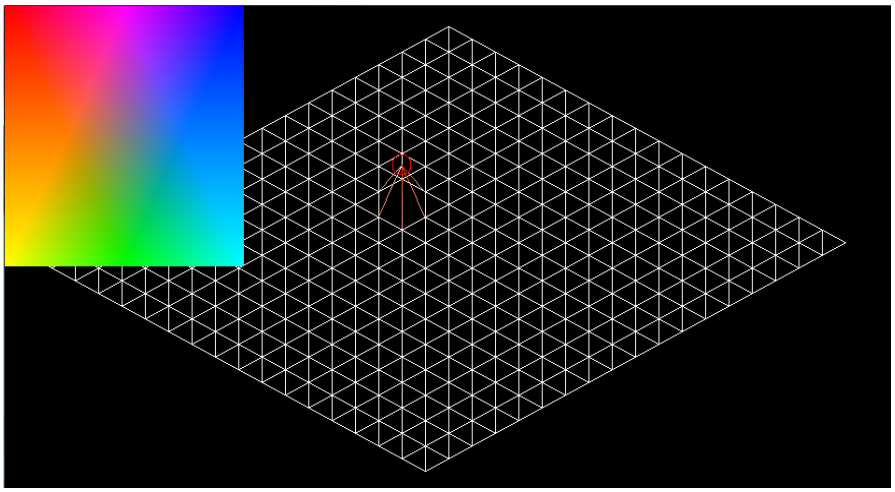
Pour chaque projet, je vous propose une description détaillée de celui-ci. Pour chaque projet, je vous mets à disposition la référence GitHub.

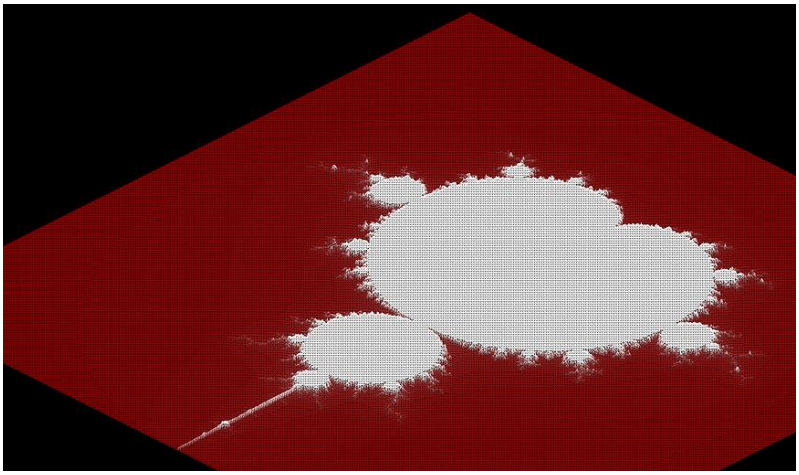
Sommaire :

Python : Fil de fer

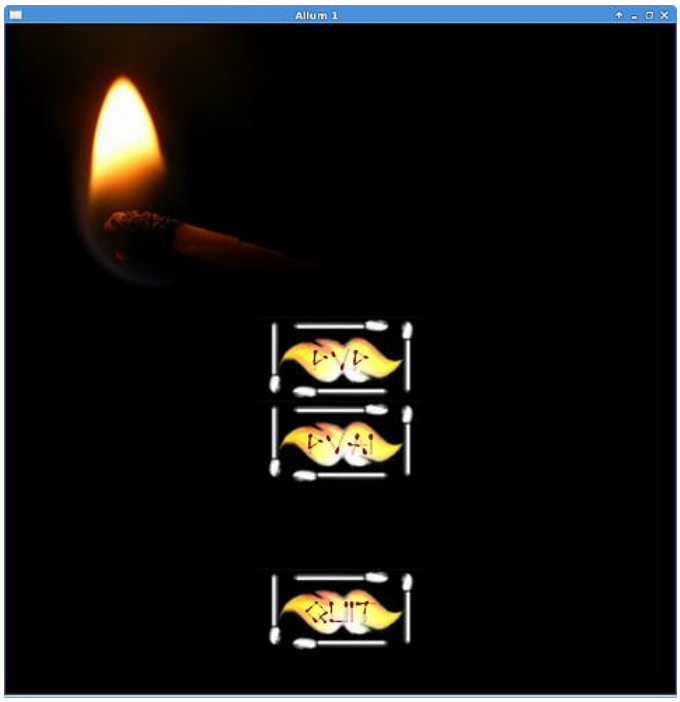
Le fil de fer est un projet de programmation graphique à Epitech. L'objectif était d'afficher un terrain en vue isométrique « Fil de fer », c'est-à-dire afficher seulement les arêtes du terrain. Nous devions lire le contenu d'un fichier fdf afin d'afficher le terrain correspondant. Le projet reste relativement facile à développer dans son utilisation la plus basique, voilà pourquoi j'ai donc décidé d'y incorporer des éléments un peu plus complexes comme des dégradés de couleur entre les points ou encore un éditeur de terrain qui génèrera votre propre terrain au format fdf ainsi que la possibilité de charger de très grands terrains comme la fractale de Mandelbrot. Pour utiliser des termes plus techniques, ce projet a donc été réalisé en Python avec la bibliothèque SFML.



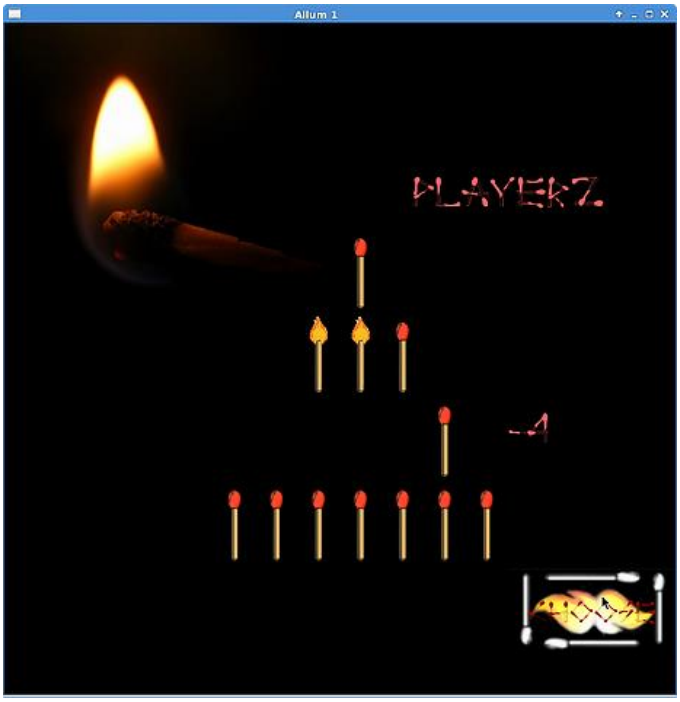




C : Jeu de Merienbad



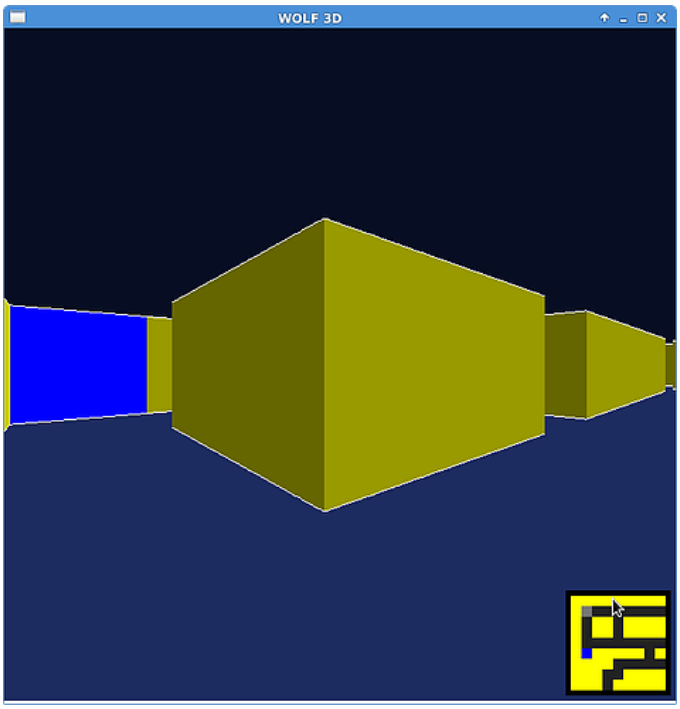
Le jeu de Merienbad est un jeu de réflexion. On dispose 16 allumettes en pyramide devant 2 joueurs. A tour de rôles les joueurs doivent récupérer 1 allumette ou plus sur une même ligne, le but du jeu étant de faire prendre la dernière allumette à son adversaire. Il nous était demandé de réaliser ce jeu sur un terminal en proposant plusieurs niveaux de difficulté. J'ai donc réalisé une intelligence artificielle sur l'algorithme du XOR, mathématiquement invincible, et une intelligence artificielle sur

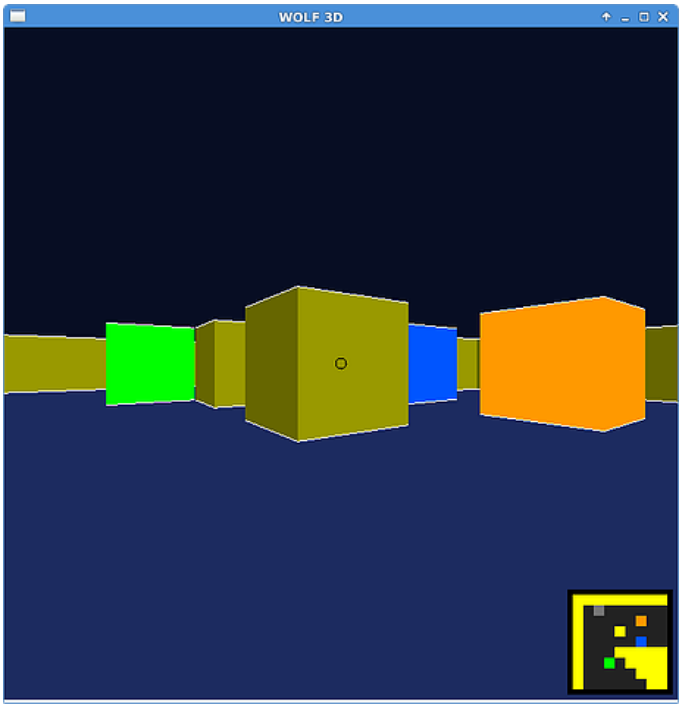


l'algorithme de Monté Carlo, faisant des simulations de parties aléatoires afin de définir avec des statistiques quel sera le meilleur coup. Cependant, l'affichage dans un terminal n'était pas suffisamment graphique à mon goût, j'ai donc décidé de me former sur la bibliothèque SDL afin de réaliser ce programme dans un environnement graphique plus adapté pour un jeu. Et quitte à faire un jeu, pourquoi ne pas y jouer à plusieurs, j'ai donc décidé d'implémenter un mode joueur contre joueur.

C : Wolf 3D

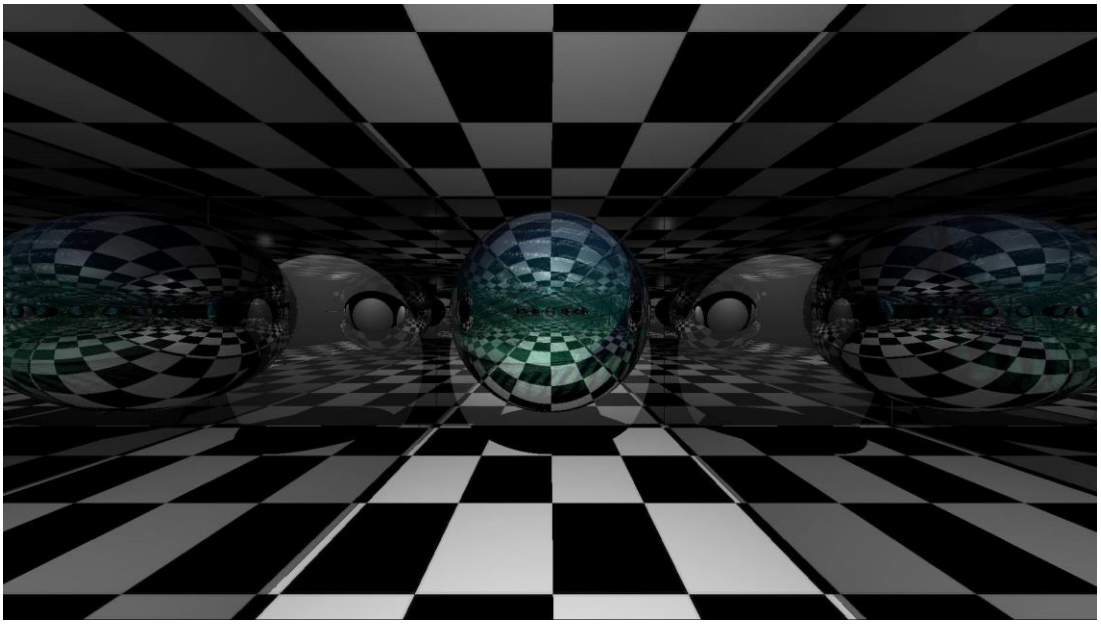
Le Wolf 3D est un projet dont l'objectif était de nous familiarisé avec les premières technique de rendu graphique en 3 dimensions utilisé notamment dans le jeu Wolfeinstein. Il s'agit en réalité d'une fausse 3D où nous allons juste calculer la distance nous séparant des murs afin d'afficher des murs plus ou moins grands. Les principales fonctionnalités apportées étaient la collision avec les murs, le glissement sur les murs, l'affichage d'une couleur différente en fonction de l'orientation du mur. Aucun but du jeu n'était demandé ce que je trouvais fort dommageable, à ce titre j'ai donc décidé d'implémenter 2 modes de jeu. Un mode labyrinthe où le but est de retrouver la sortie du labyrinthe et un mode Sokoban (un petit jeu de réflexion). J'ai aussi ajouté une minicarte afin de faciliter la visualisation dans le labyrinthe.



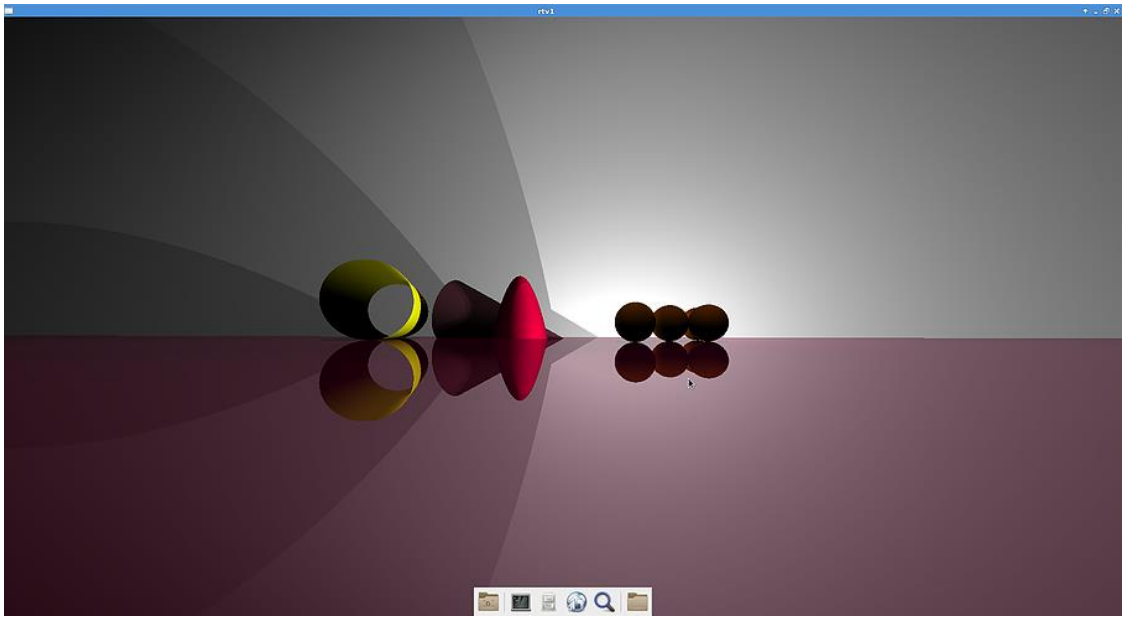


C : Raytracer

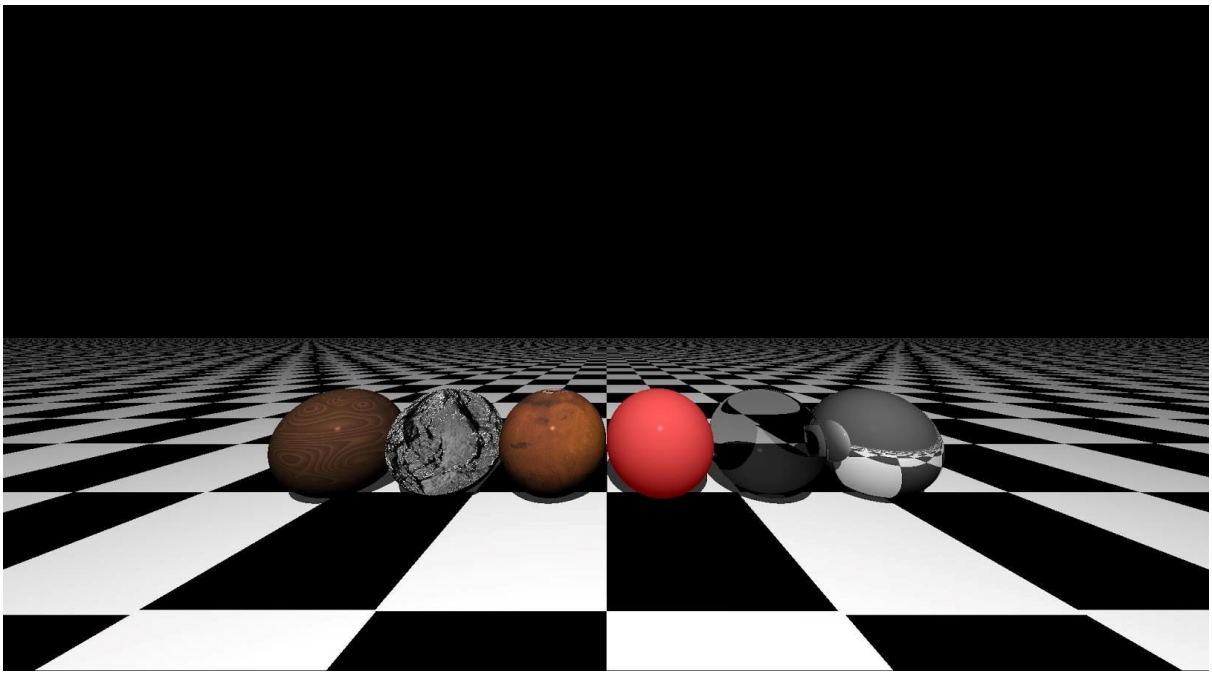
Le Raytracer est le dernier projet de la première année à Epitech. Son objectif était de nous familiariser aux techniques de rendu 3D en ray-casting qui est une technique de rendu très gourmande en performances mais qui a l'avantage de proposer une qualité de rendu supérieure.



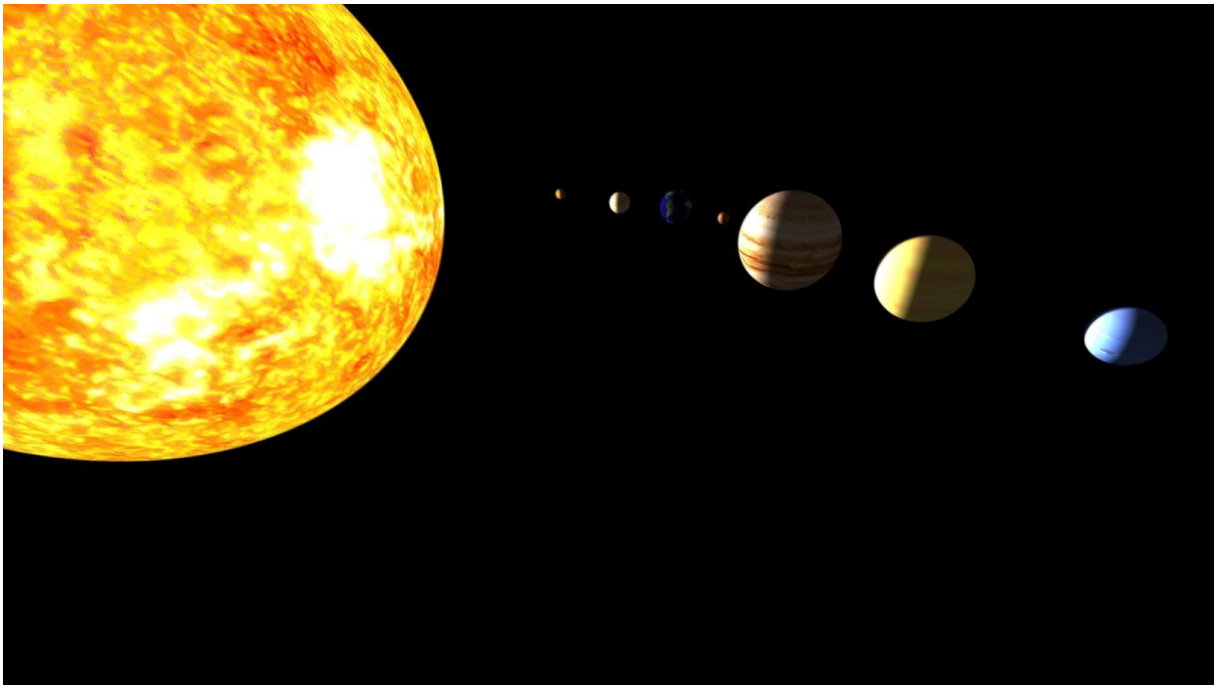
Le cahier des charges nous demandait d'effectuer l'affichage de formes mathématiques simples telles qu'une sphère ou un cylindre avec des effets de lumière et d'ombre sur lesquels nous pouvions appliquer des translations et des rotations. Cependant la partie obligatoire ne rapportait aucun point à la notation.



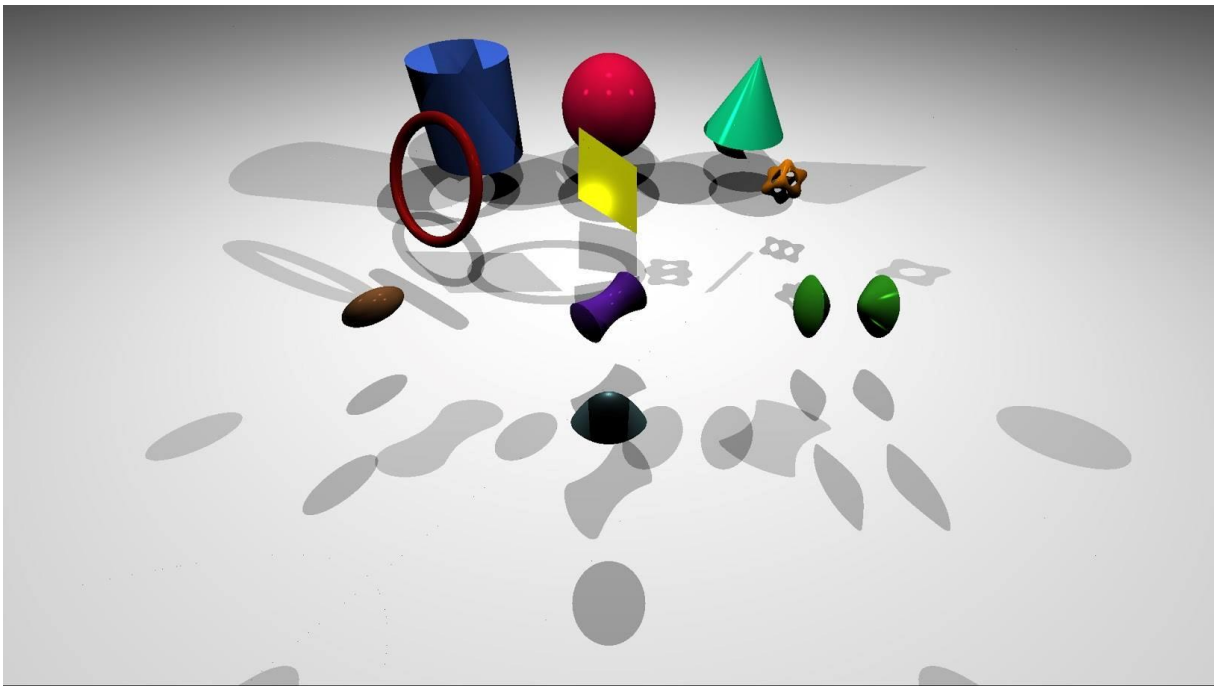
Les points étaient donc rapportés sur les fonctionnalités supplémentaires que nous pouvions apporter. J'ai donc décidé d'implémenter des effets supplémentaires comme la réflexion et la réfraction.

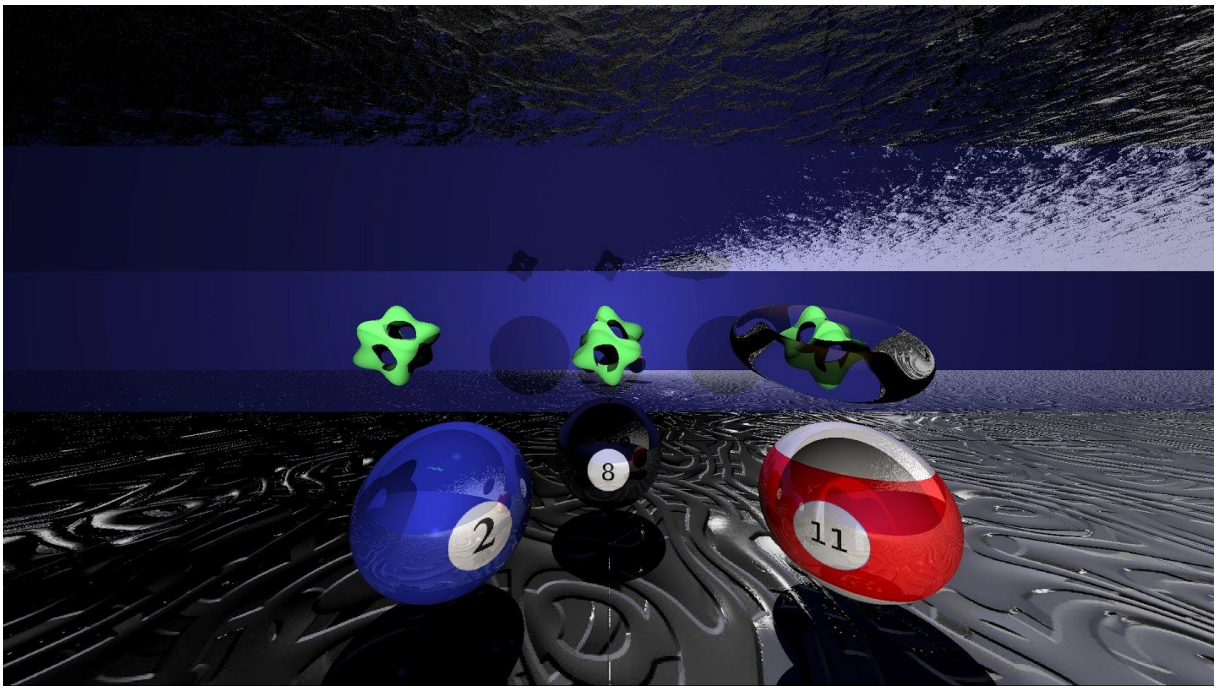


Mais aussi l'application de texture selon plusieurs techniques (texture mapping, texture bumping, textures procédurales)



Le post-processing d'images avec l'application de filtres couleurs (sépia, nuance de gris, …), détection de bords. J'ai aussi décidé d'implémenter des formes mathématiques un peu plus complexes comme le tore (sorte de chambre à air), et le cube troué.

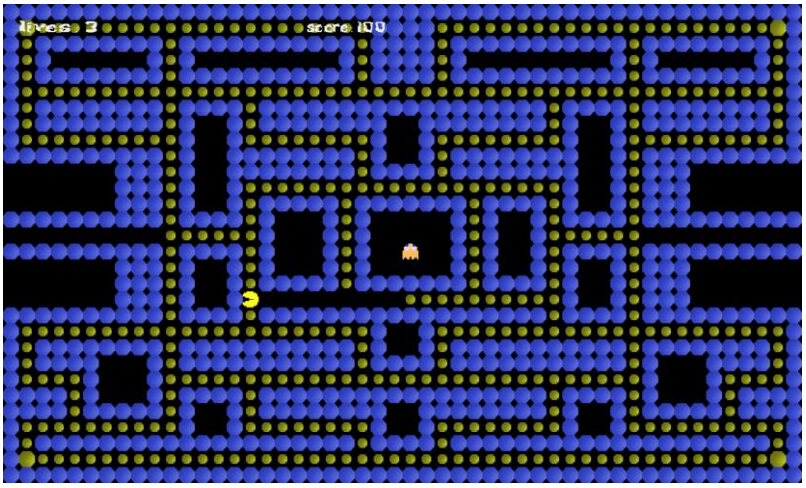




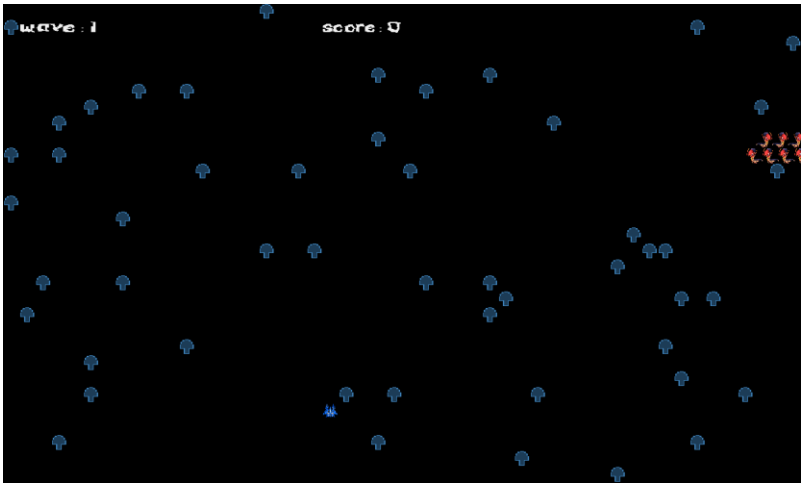
C++ : Arcade

L'arcade est un projet de deuxième année à Epitech. Il nous introduit à la notion de plugin et à leur mise en place. Lors de ce projet, le but était de réaliser une borne d'arcade sur laquelle pouvaient se greffer des jeux et des environnements graphiques.

Ainsi nous avons réalisés 6 plugins pour ce projet. 3 jeux (Pacman, Snake et Centipède) et 3

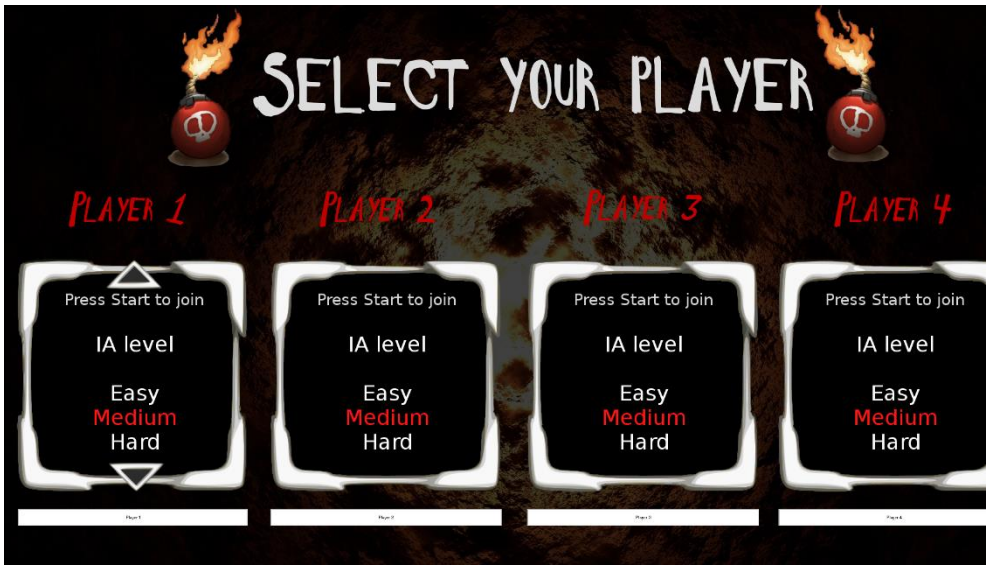


environnements graphiques (ncurses, sdl2, openGL). Nous avons implémenté le changement de jeu et d'environnement pendant l'exécution du programme, un menu principal vous permettant de choisir graphiquement vos modules et un système de highscore pour chaque jeu.

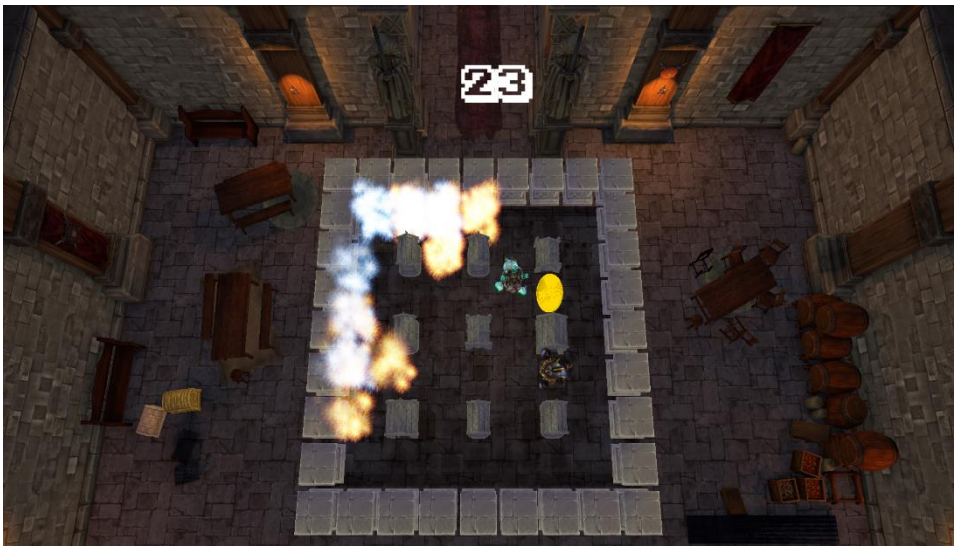


C++ : Bomberman

Le Bomberman est un projet de deuxième année à Epitech. L'objectif pédagogique de ce projet était de créer un jeu en utilisant une Engine entre Ogre3d et Irrlicht. Nous avons choisi d'utiliser Irrlicht pour sa simplicité d'utilisation et d'installation.

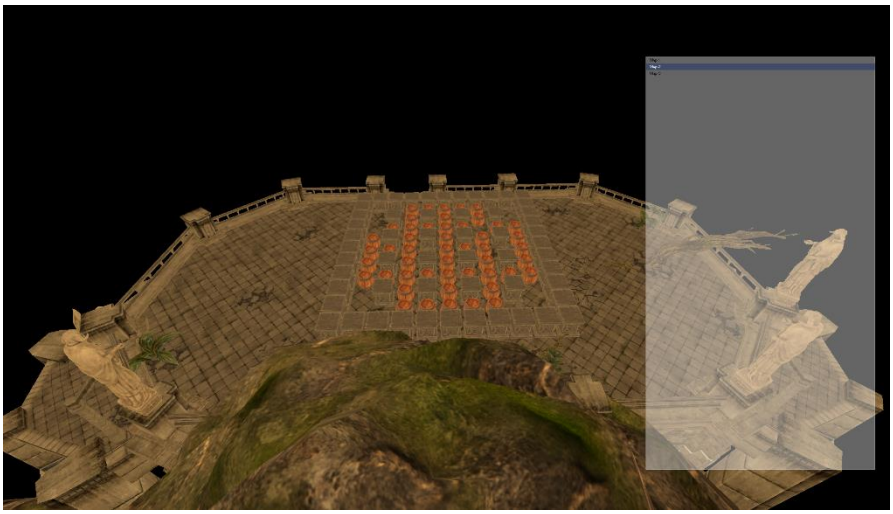


Au cours de ce projet j'ai mis en place un système permettant de réaliser des intelligences artificielles en Lua ainsi qu'un système de communication entre les données C++ du jeu et le Lua. L'IA avait 3 niveaux de difficulté, sachant que pour le Bomberman vous pouvez vous tuer à tout moment. Un niveau facile où l'IA fait le minimum, pose des bombes dans un endroit tout en se laissant une position de replis. Le



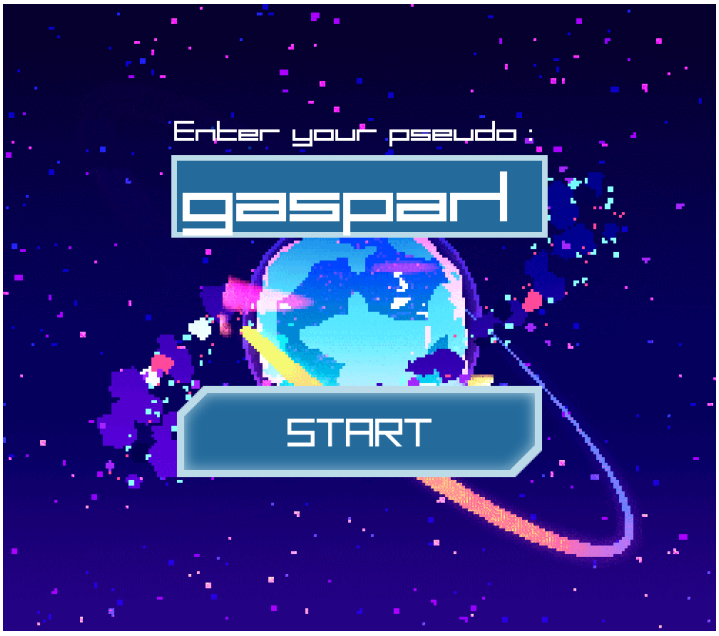
niveau moyen qui lui, aura un ratio plus important pour poser des bombes et en pose dès qu'il croise un joueur. Et enfin un niveau difficile qui va se focaliser sur la récupération des nombreux

bonus du jeu qui offrent des avantages allant jusqu'à vous sauver d'une explosion. Afin de permettre aux IA de se déplacer d'un endroit à l'autre, j'ai mis en place l'algorithme A* en Lua.

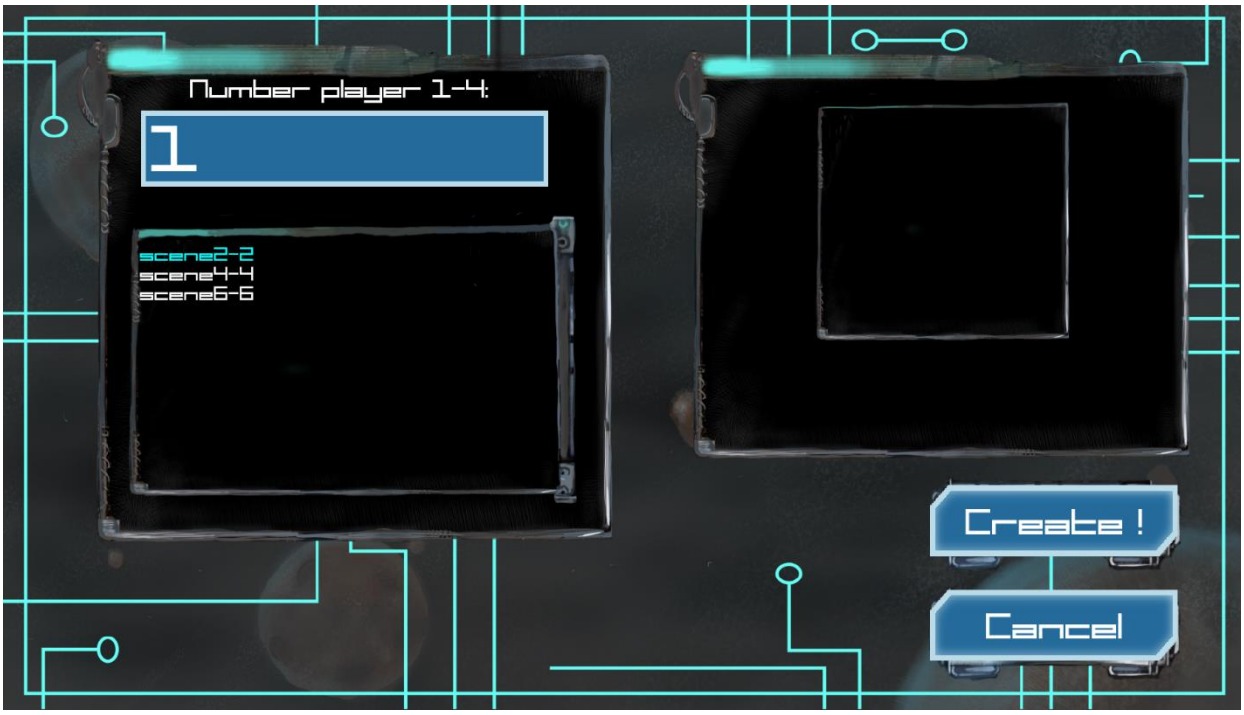


C++ : R-type

Le R-type est le dernier gros projet graphique de C++ des 3 premières années confondues à Epitech et sachant qu'il s'agit des années les plus riches techniquement, on peut dire qu'il s'agit du plus gros projet d'Epitech.



Le but était de réaliser le jeu d'arcade R-type de 1987 mais en multijoueur. Ainsi nous avons mis en place une gigantesque architecture pour ce projet avec plus de 1200 révisions, 4 programmes différents et plus d'une dizaine de bibliothèques dynamiques. Nous avons aussi mis en place au cours de ce projet une Engine



complète ainsi qu'une bibliothèque de gestion réseau en C++. Nous avons décidé de mettre en place un serveur gérant les Salles de jeu sur lequel se connecterons les Clients et les

Répartiteurs de jeu. Ainsi lorsqu'un client lancera une partie, le serveur de salle demandera à un répartiteur de lancer un serveur de Jeu sur lequel se connecteront les Clients. Ce projet nous a demandé 1 mois et demi de travail acharné soit près de 300 heures de travail. Cela nous a beaucoup apporté en compétences, notamment dans la gestion des bibliothèques dynamique sur Windows ou encore la gestion de l'UDP au niveau du réseau.