2018

Portfolio de projets

REALISATIONS ET INNOVATIONS

QUENTIN GASPAROTTO

Table des matières

Python

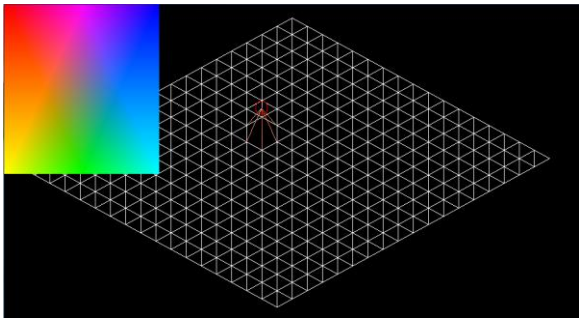
Fil de fer

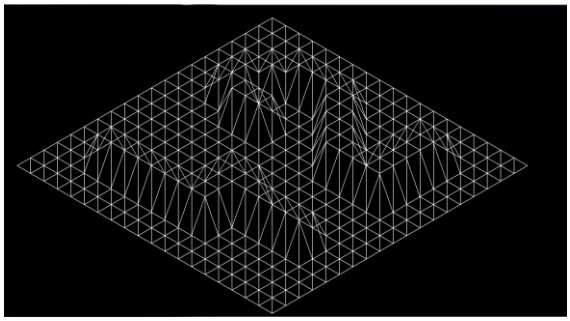
Le fil de fer est un projet de programmation graphique à Epitech. L'objectif était d'afficher, à partir d'un fichier, un terrain en vue isométrique « Fil de fer » (**fdf**), c'est-à-dire afficher seulement les arêtes du terrain.

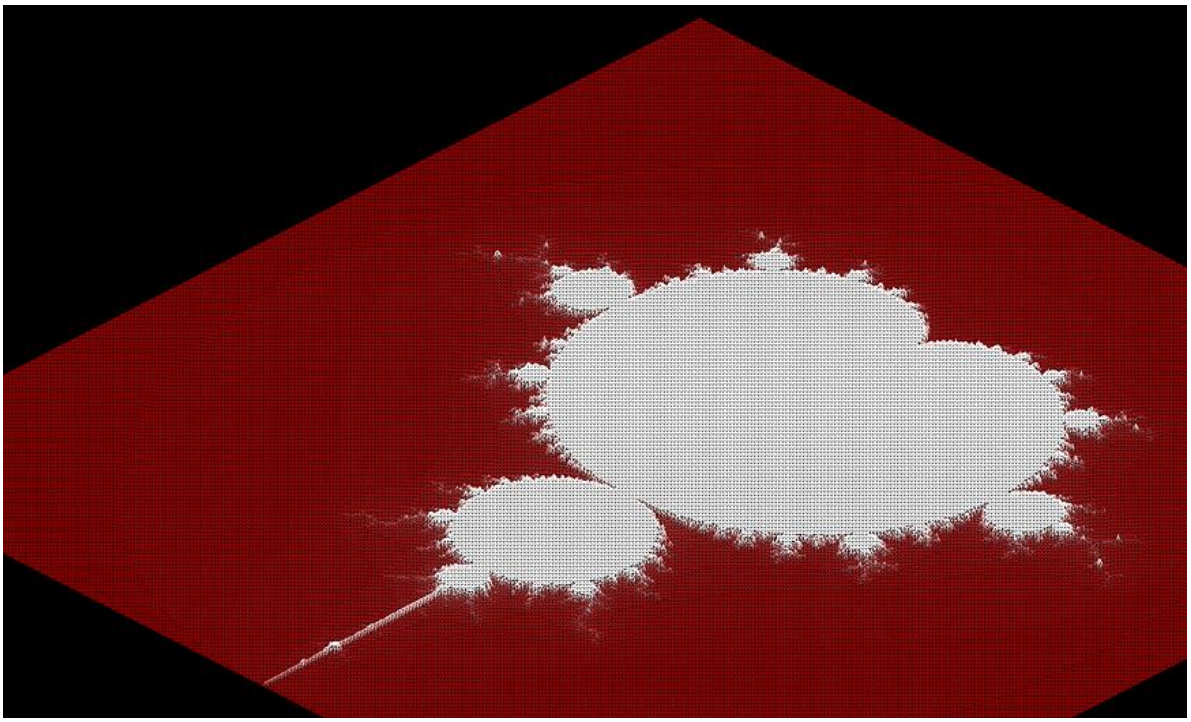
Dans sa forme la plus basique, ce projet reste relativement facile à développer. Voilà pourquoi j'ai décidé d'y incorporer des éléments un peu plus complexes tels que des dégradés de couleur entre les points du terrain ou encore un éditeur interactif.

Vous avez également la possibilité de charger de très grands terrains comme celui représentant la fractale de Mandelbrot.

Ce projet a été réalisé en Python avec la bibliothèque SFML.







Langage C

Jeu de Merienbad

Le jeu de Merienbad est un jeu de réflexion.

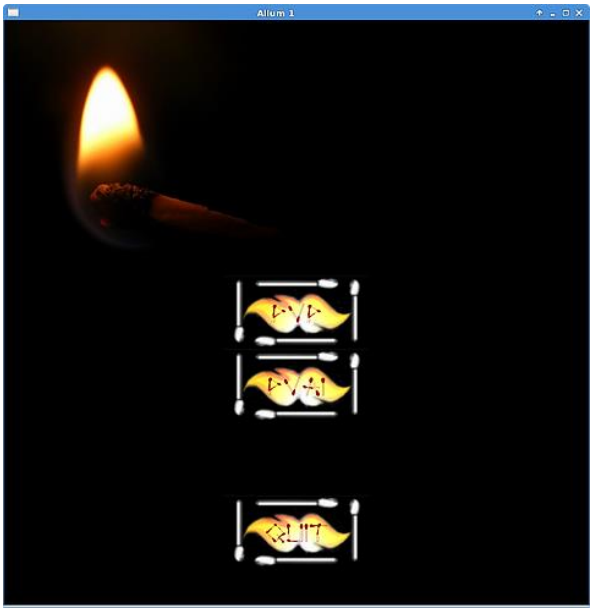
On dispose 16 allumettes en pyramide devant 2 joueurs. A tour de rôles les joueurs doivent récupérer 1 allumette ou plus sur une même ligne, le but du jeu étant de faire prendre la dernière allumette à son adversaire.

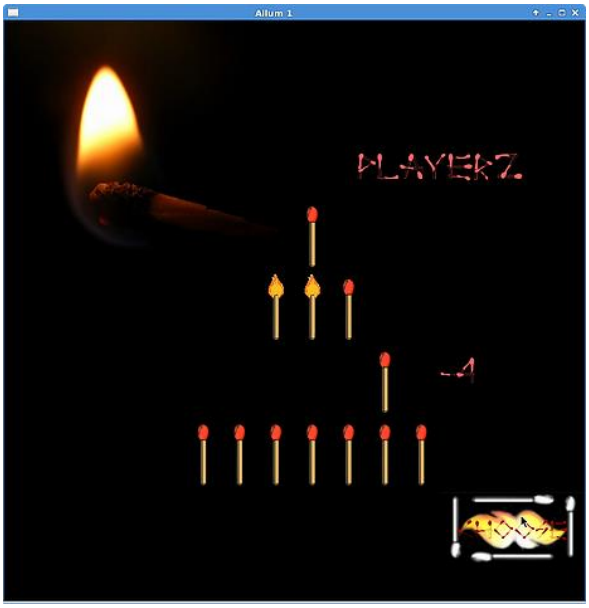
Nous devions réaliser ce jeu sur un terminal en proposant plusieurs niveaux de difficulté.

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 pour déterminer 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.

De plus, un mode multijoueur vous permet de défier vos amis dans vos soirées Fort Boyard endiablées.





Wolf 3D

Le Wolf 3D est un projet dont l'objectif était de nous familiariser avec les premières techniques de rendu graphique en 3 dimensions utilisé notamment dans le jeu Wolfenstein.

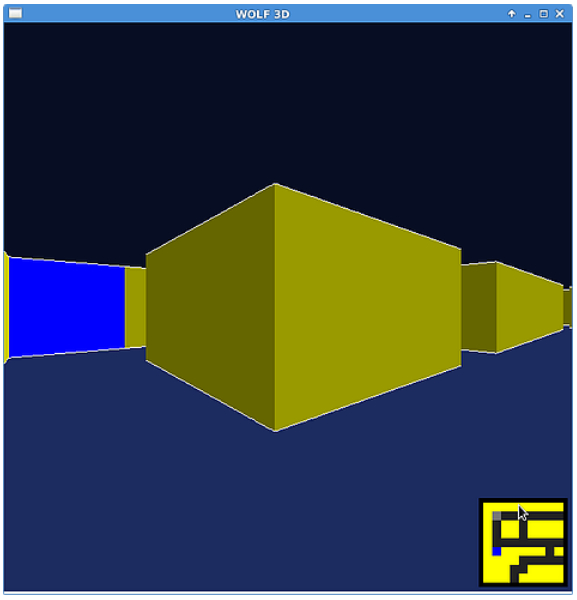
Il s'agit en réalité d'une fausse 3D utilisant la technique du Raycasting permettant d'afficher des murs plus ou moins grands en fonction de leur distance.

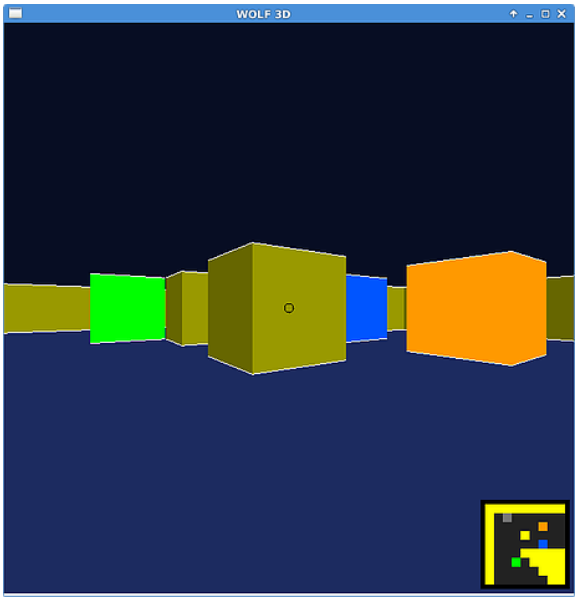
Il nous était demandé d'afficher un labyrinthe, de gérer les collisions avec les murs ainsi que le glissement du personnage sur les murs.

Le projet s'en retrouva vite fini et je trouvais dommage de ne pas avoir de but du jeu. A ce titre j'ai donc décidé d'implémenter 2 modes.

Un mode labyrinthe où le but est de retrouver la sortie du labyrinthe et un mode Sokoban.

J'ai également ajouté une minicarte afin de faciliter la visualisation de l'espace dans le labyrinthe.

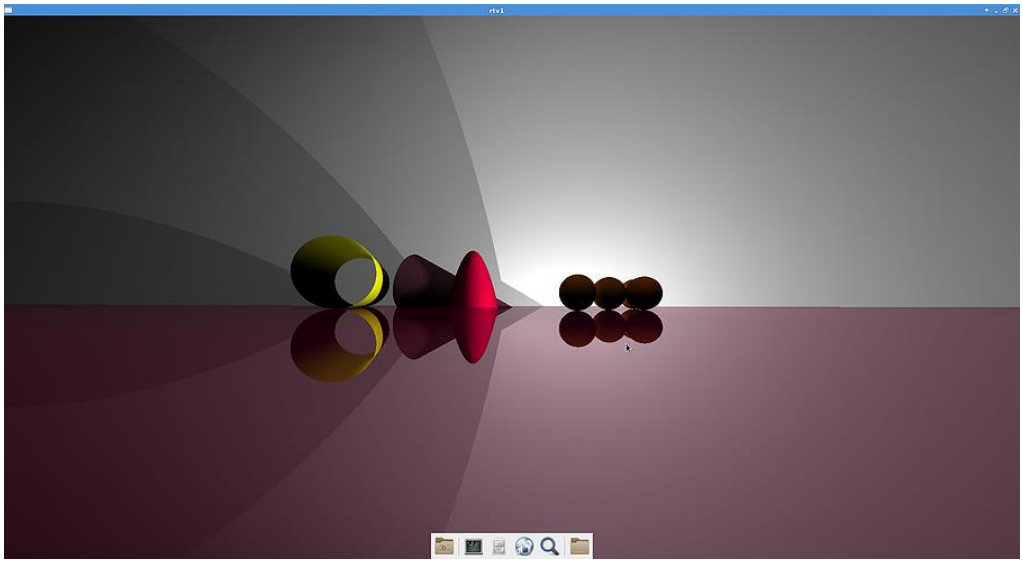




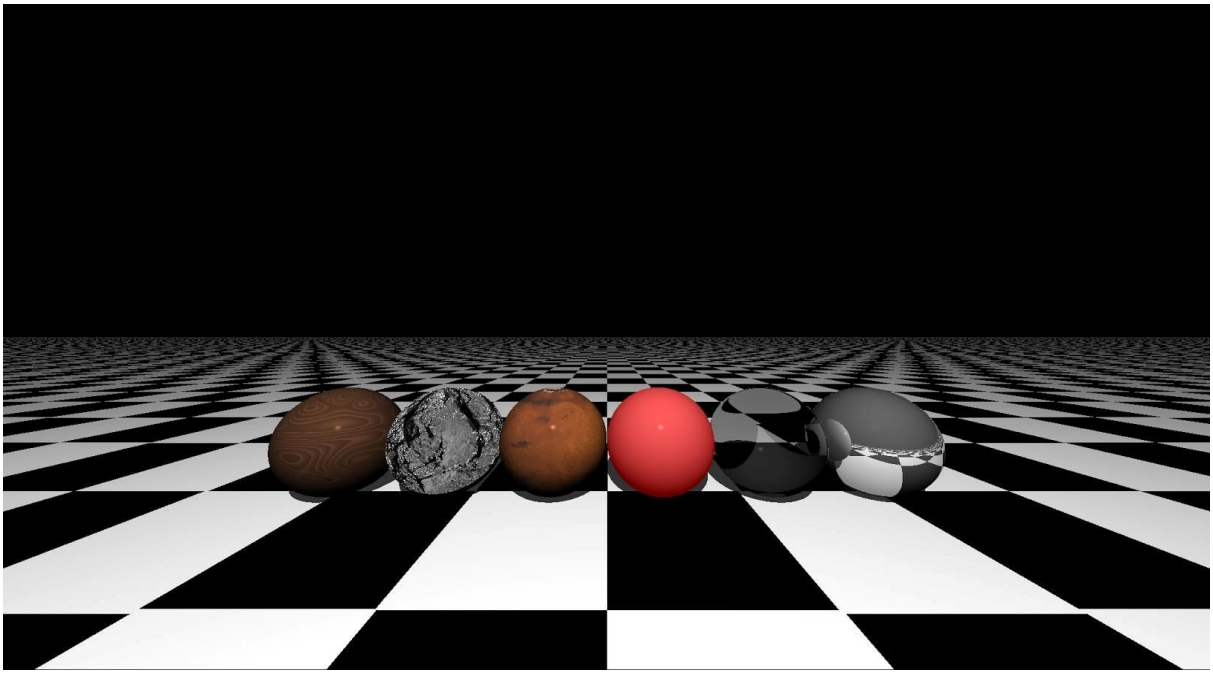
Raytracer

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

Nous devions implémenter l'affichage de formes mathématiques simples telles qu'une sphère, un cylindre ou un cône, pouvant être transformée dans l'espace à n'importe quelle position et rotation, et sur lesquels il était possible d'appliquer avec des effets de lumière et d'ombrage. Cependant la partie obligatoire ne rapportait aucun point à la notation finale.



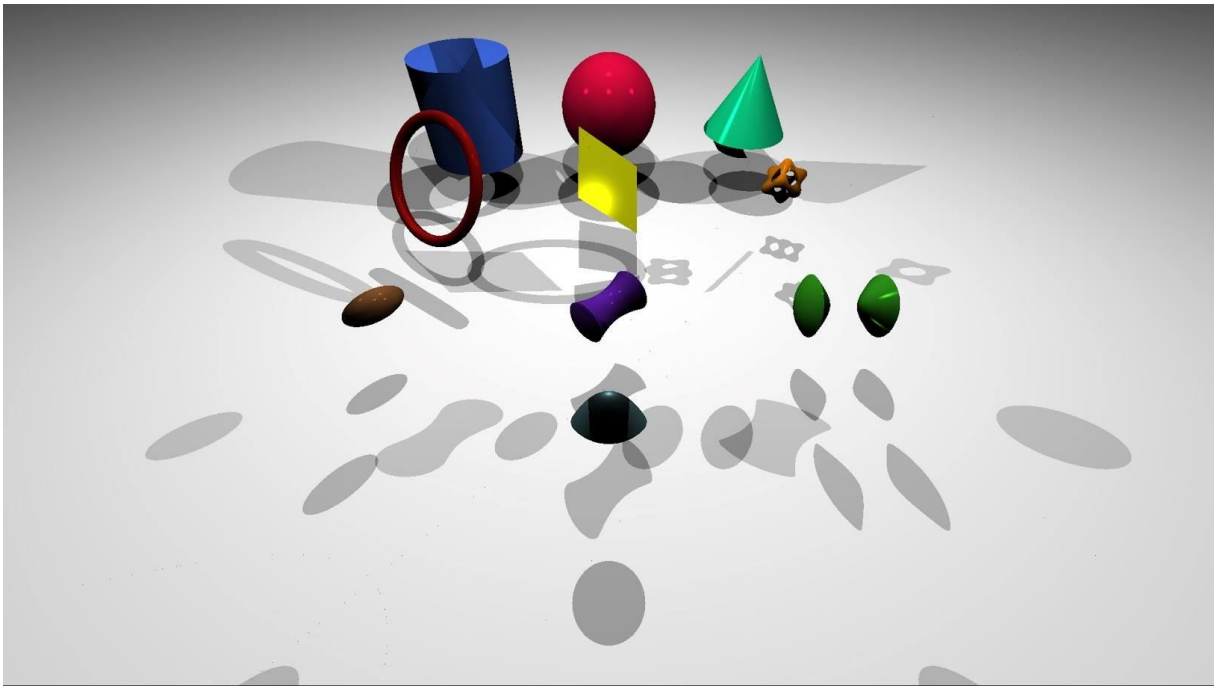
Les points étaient donc rapportés sur les fonctionnalités supplémentaires que nous pouvions apporter. J'ai donc décidé d'implémenter des effets supplémentaires comme la réflexion et la réfraction. Mais aussi l'application de texture selon plusieurs techniques (texture mapping, texture bumping, textures procédurales).

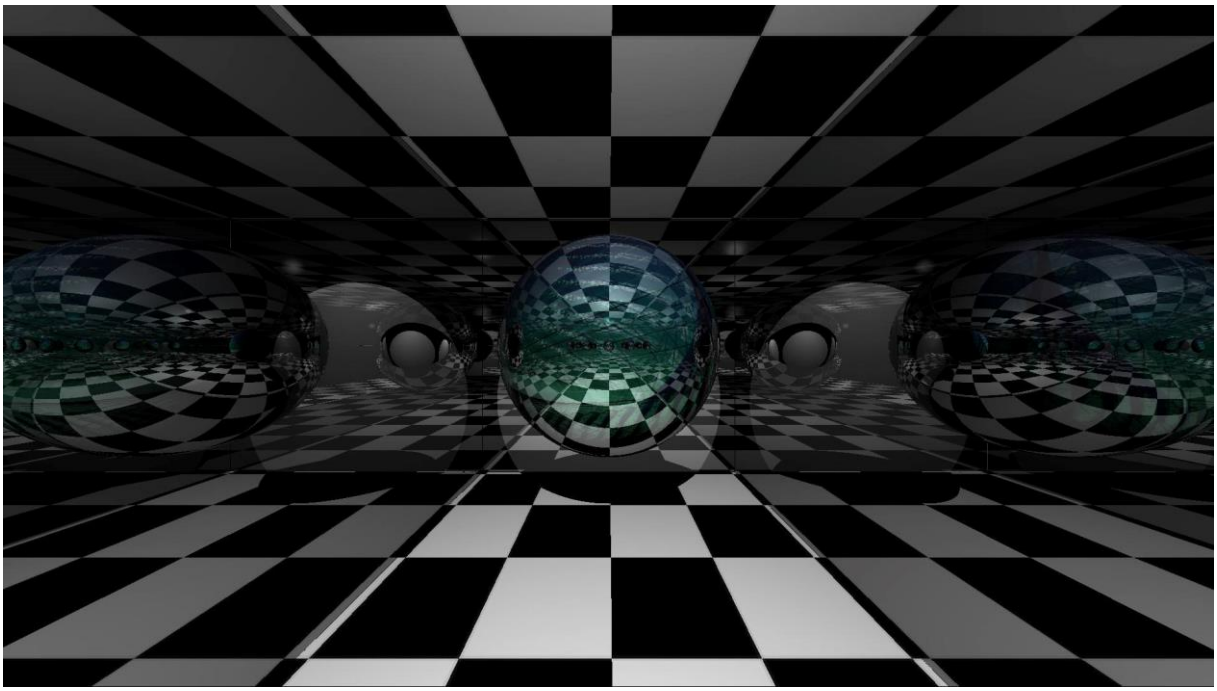


J'ai également implémenté le post-processing d'images avec l'application de filtres de couleurs tels que sépia ou nuance de gris ainsi que la détection de bords.

J'ai aussi décidé d'implémenter des formes mathématiques un peu plus complexes comme le tore (beignet circulaire), et le cube troué.

Les objets étaient d'ailleurs soumis à l'éclairage de toutes les sources de lumière présent dans la scène.





Langage C++

Arcade

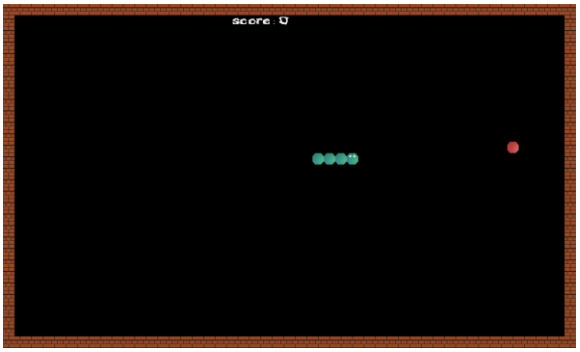
L'arcade est un projet de deuxième année à Epitech ayant pour vocation de nous faire appréhender la notion de modularité.

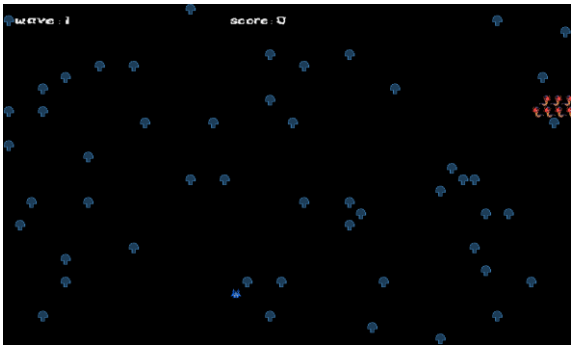
Lors de ce projet, le but était de réaliser une borne d'arcade sur laquelle pouvaient se greffer des jeux et des environnements graphiques.

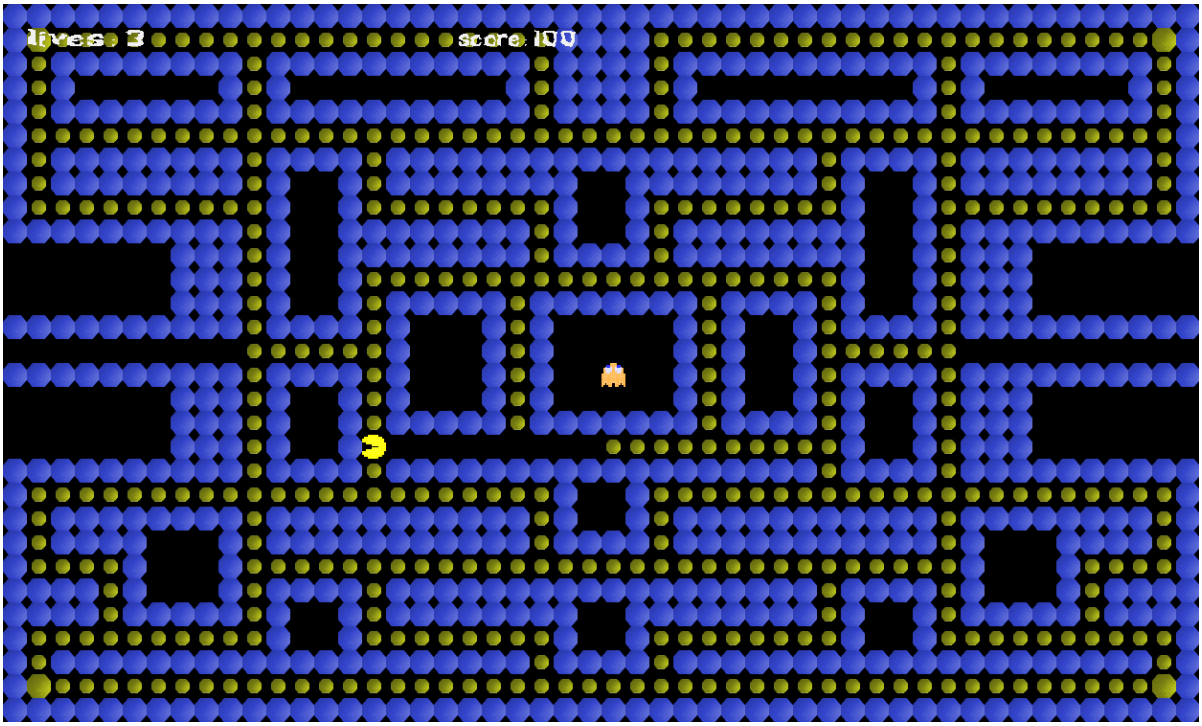
Ainsi nous avons réalisés 6 modules pour ce projet dont 3 jeux (Pacman, Snake et Centipède) ainsi que 3 environnements graphiques (ncurses, SDL2 et OpenGL).

L'aspect modulaire du système nous permettait de modifier les jeux et les environnements graphiques au cours de l'exécution du programme. Il vous était alors possible de passer d'un Pacman en 2D vers un Snake en 3D.

Le plus gros avantage qu'avait notre borne d'arcade par rapport aux autres était qu'elle jouait la musique de Musclor en boucle. De quoi jouer au Snake toute la nuit.

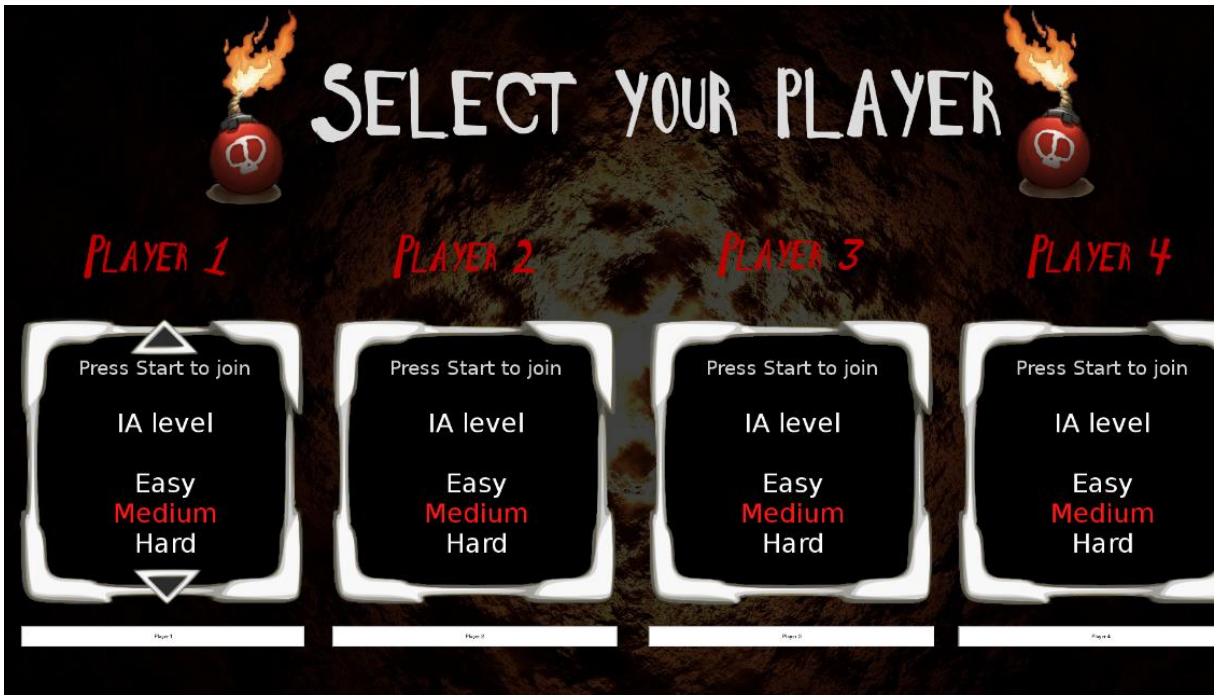






Bomberman

Le Bomberman est un projet de deuxième année à Epitech pour nous initier à la conception de jeux avec moteur les moteurs de jeu Ogre3D ou Irrlicht (que nous avons choisi).



Le jeu du Bomberman consiste à tuer ses adversaires en posant des bombes explosant de façon linéaire. A noter qu'il est possible de se tuer soi-même et que les joueurs ne peuvent pas passer au travers des bombes lorsqu'elles sont posées. Une bombe explose quelques secondes après avoir été posée.

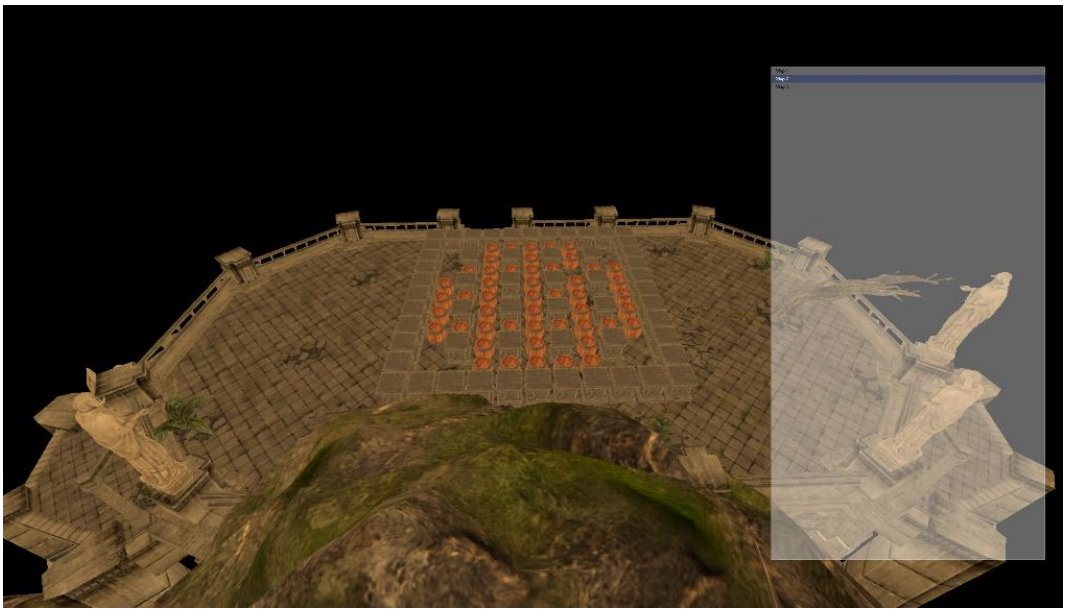
Ma mission pour ce projet était de réaliser des intelligences artificielles de niveaux différents. J'ai donc mis en place un système permettant d'intégrer des comportements intelligents en Lua (notamment utilisé pour World of Warcraft).

L'IA possédait 3 niveaux de difficulté, facile, moyenne et difficile.

Dans le niveau facile, l'IA fait le minimum, c'est-à-dire qu'elle pose des bombes dans un endroit tout en se laissant une position de replis.

Le niveau moyen, lui, posera plus de bombes que le niveau facile et en pose une dès qu'il croise un joueur.

Enfin, le niveau difficile va se focaliser sur la récupération des nombreux bonus du jeu qui offrent des avantages allant jusqu'à vous sauver d'une explosion, permettant ainsi de prendre significativement l'avantage. Afin de permettre aux IA de se déplacer d'un endroit à l'autre, j'ai mis en place l'algorithme A* en Lua.



R-type

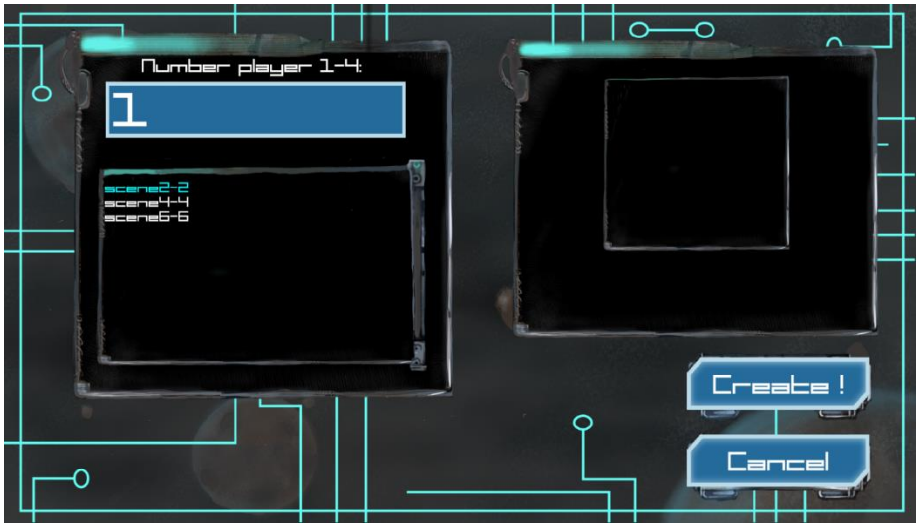
Le R-type est le dernier gros projet graphique de C++ des 3 premières années confondues à Epitech.

Le but était de réaliser le jeu d'arcade R-type de 1987 mais en multijoueur.

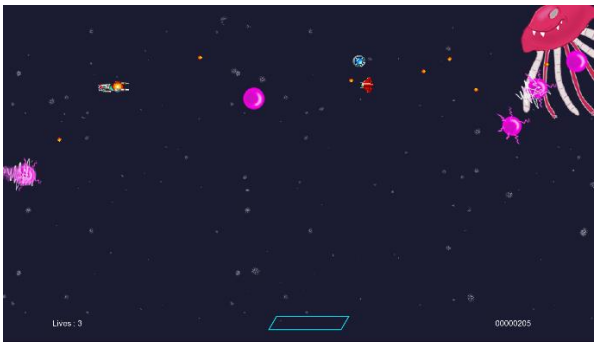
Ainsi, nous avons mis en place une architecture sur le modèle des serveurs dédiés, avec plus de 1200 révisions, 4 programmes différents et plus d'une dizaine de bibliothèques dynamiques.

Nous avons également mis en place au cours de ce projet un moteur de jeu complet ainsi qu'une bibliothèque de gestion réseau en C++.

Nous avons décidé de mettre en place un serveur administrant les Salles de jeu sur lequel se connectent les joueurs. Ainsi, lorsqu'un client lancera une partie, le serveur de salle lancer un serveur de Jeu sur lequel se connecteront les joueurs.



Ce projet nous a demandé 1 mois et demi de travail acharné soit près de 300 heures. 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.



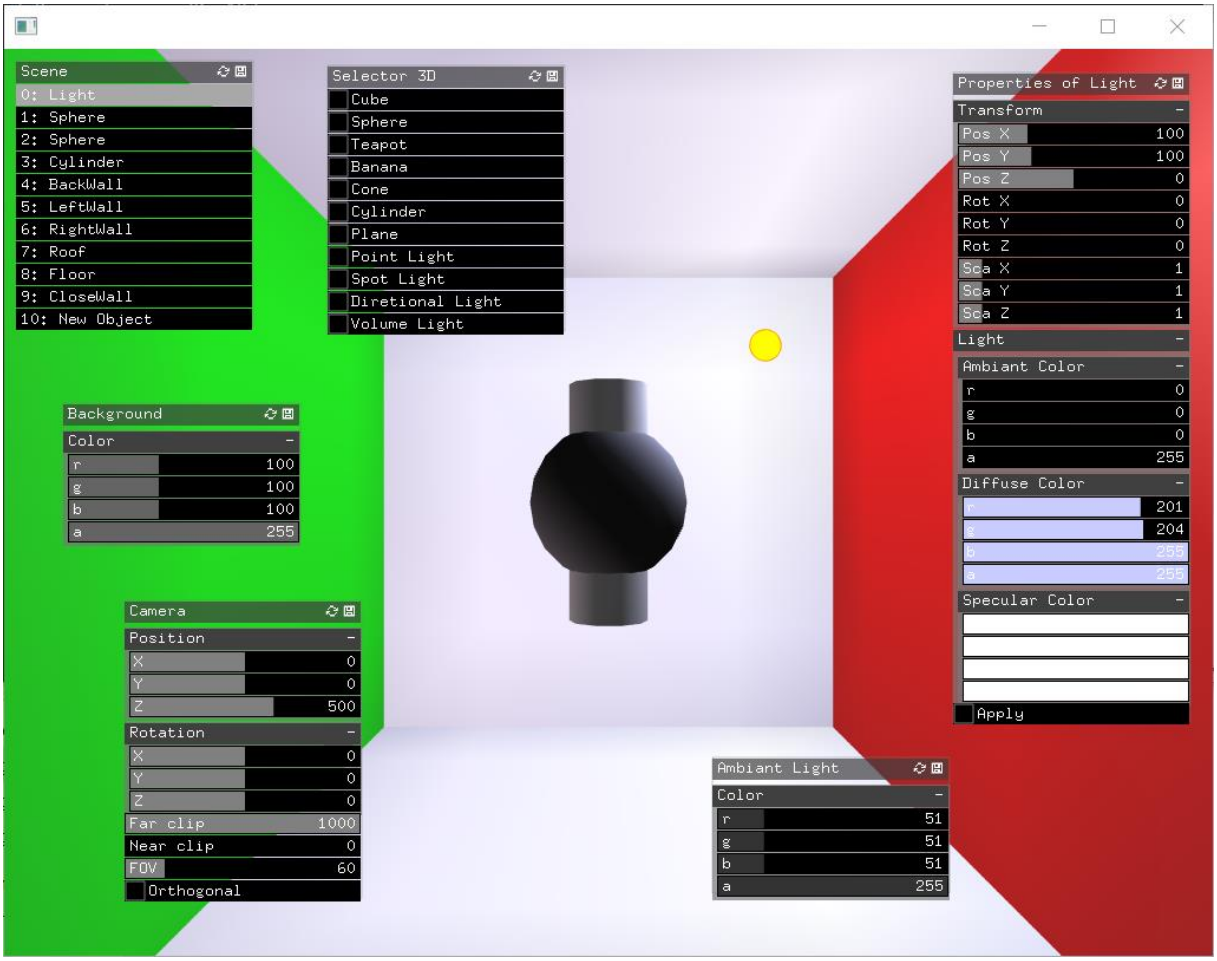
Editeur de scène 2D/3D

Au cours de mon année à l'Université Laval de Québec au Canada, j'ai eu l'occasion d'assister au cours d'infographie, essentiellement centré sur les techniques de rendu 2D et 3D existant de nos jours en infographie.

Dans le cadre de ce cours, j'ai pu réaliser un éditeur de scène en 2D et 3D à l'aide de la bibliothèque OpenFrameworks. Il permet la manipulation de primitives géométriques 2D et 3D et de maillages triangulaires.

Son architecture est extrêmement modulaire ce qui permettait de facilement ajouter des fonctionnalités sans avoir à modifier sensiblement le projet. Ce projet m'a permis de mettre en application le cours de Génie logiciel orienté objet (dispensé dans la même Université) nous initiant à l'intégration de Design Pattern dans nos projets.

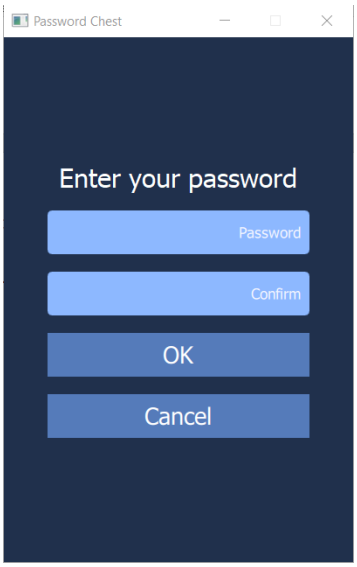
Les éléments d'interface sont directement rendus avec une extension de la bibliothèque.



Framework Qt/QML

Coffre a mot de passe

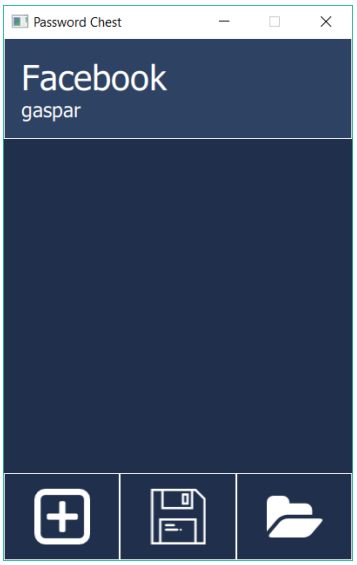
J'ai toujours été attiré par le développement logiciel et j'aime beaucoup le C++. Le framework de développement Qt était donc un passage obligatoire pour moi. Pour apprendre cette technologie, j'ai décider de réaliser un projet de coffre à mot de passe.

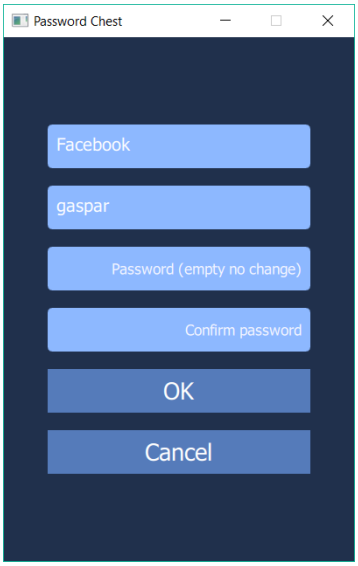


Le principe est simple, il s'agit d'un logiciel vous permettant de sauvegarder vos mots de passe de façon sécurisée.

En effet, ils seront contenus dans un fichier chiffré par un unique mot de passe.

Vous n'aurez plus qu'à vous souvenir de ce mot de passe pour avoir accès aux autres.

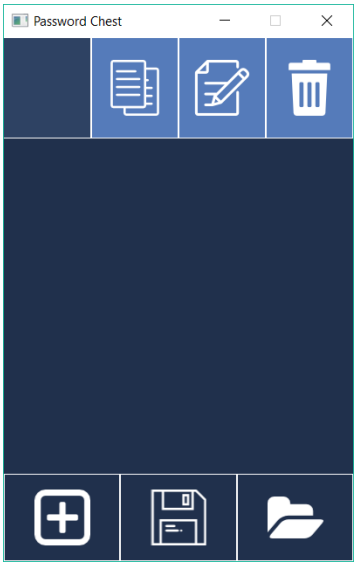




J'ai travaillé l'interface pour qu'elle soit à la fois intuitive mais aussi sécurisée pour que personne ne puisse avoir accès à vos mots de passe.

En effet, lorsque vous utiliserez le logiciel, vous ne verrez jamais écrit vos mots de passe à l'écran mais vous pourrez les copier dans le presse-papier.

Vous pourrez bien évidemment ajouter, modifier et supprimer vos mots de passe comme bon vous semble.



Le projet est disponible sur mon GitHub à l'adresse https://github.com/GasparQ/PassChest où vous pourrez retrouver une mise en production contenant le programme d'installation (sur Windows 10 uniquement).

DNAI

Les deux dernières années à Epitech correspondent au cycle d'expertise et sont marquées par la réalisation d'un projet de fin d'étude, ou EIP, ayant pour principal critère d'être innovant.

Au sein d'une équipe de 8 personnes, j'ai contribué à la réalisation d'un projet visant à simplifier l'accès au domaine de l'intelligence artificiel encore aujourd'hui trop réservé aux spécialistes.

Nous avions pour objectif de fournir une suite logicielle permettant de créer, d'utiliser et de partager des intelligences artificielles dans n'importe quel projet. Nous avons donc réalisé un éditeur graphique permettant de créer des intelligences artificielles, une plateforme communautaire sur laquelle les IA pouvaient transiter avant d'être intégrées dans des extensions logicielles telle que celle que nous avons réalisé sur Unity3D, logiciel d'édition de jeux vidéo.

Pour notre éditeur, nous avons choisi la technologie Qt/QML car elle permettait de créer des interfaces logicielles très performantes et assez poussées. Personnellement, je me suis occupé de la partie modélisation des intelligences artificielles. J'ai donc intégré rapidement le développement de l'éditeur.

J'ai réalisé toute la gestion de la création des fichiers d'IA ainsi que leur représentation dans le logiciel.

Le projet est disponible sur GitHub à l'adresse https://github.com/Nicolas-Constanty/Dnai.Editor et vous pourrez retrouver notre site internet sur https://dnai.io