



# Enzo Hugonnier

DÉVELOPPEUR

enzohugonn@gmail.com

www.darleanow.github.io

Etudiant de 19 ans, passionné par le développement.  
J'apprécie tout particulièrement le C++, langage sur lequel  
je me forme un peu plus chaque jour.

## WORK EXPERIENCES

01

2021 – Current

### IAM/SYSTEM CONSULTANT

Cerberis, Grenoble

- Installation et support de logiciels pour salles de réunions / écrans de salles.
- Formation sur AD / AAD / Exchange / MSSQL / Jumpcloud / Sécurité
- Développement Python d'outils diverses tels que un générateur de rapports de sécurité

## SKILLS

02

- Python
- C++
- Développement Web (HTML/CSS/JS)
- Sécurité (Pentest / Compliance)
- Cours en C, Data mining

Python

C++

C

Web

Sécurité



Numéro mobile  
+33 652003257

Adresse  
Grenoble

## EDUCATION

03

### Ecole Supérieure en Génie Informatique

2021 – 2026 (En cours)

Master en développement logiciel

### Lycée Pierre Beghin

2018–2021

Baccalauréat Général avec mention

## HOBBIES

04



Développement

## CERTIFICATIONS

05

### Python

100 % | Codingame

2022

### C++

100 % | Codingame

2022

### Javascript

84 % | Codingame

2022

### C

75 % | Codingame

2022

# PROJECTS

## Project : Alcia

### C++ - 2022

Alcia est un RPG textuel qui comporte un peu plus de 4000 lignes de code. C'est mon plus gros projet jusqu'à présent. Le but du jeu est de tuer Azeael, le boss final.

Tout au long de votre aventure, il est possible de récupérer de l'équipement, de monter en niveaux, il y a aussi une boutique, et un forgeron qui vous permettra de crafter (échanger des loots de monstres contre de l'équipement) tout les items du jeu !

J'ai utilisé la POO, l'héritage, le polymorphisme (presque l'entièreté du jeu se base dessus), l'encapsulation (champs privés de classes et méthodes pour y accéder).

Ce projet à été réalisé dans le but de m'affiner sur tout ce qui est POO et les méthodes / concepts qui y sont associées.

J'ai aussi fait face a des difficultés, comme l'affichage des couleurs sur terminal, et aussi pour rendre le jeu multiplateformes.

Il y aussi un système de hasard. J'ai essayé d'éviter d'utiliser le connu `srand()` (C++) qui génère des chiffres pseudo-aléatoires en fonction de l'horloge système. J'ai donc prit la décision d'utiliser un système un peu plus complexe; un générateur de nombres distribués uniformément, couplé au Mersenne Twister (mt19937) qui m'a permit d'avoir de très bons résultats.

La saga Alcia a pour but de comporter 3 volets, les deux restants seront des évolutions 2D / 3D du jeu initial, avec amélioration des mécanismes et du fonctionnement interne.

## Space invaderz

### C++ - 2022

Ma toute première initialisation a la POO en C++.

Le jeu possède un système beaucoup plus trivial que Alcia.

J'ai surtout voulu travailler le contrôle possible du terminal avec mon code.

Au lieu d'effacer l'écran a chaque déplacement de monstres / joueur, le curseur est positionné sur l'entité a déplacer, est effacée (par un char vide), puis à X+1, le monstre est affiché a nouveau. Cela permet d'éviter d'avoir a effacer tout l'écran, et donc rend le jeu plus fluide.

Pour les déplacements du joueur, l'intérêt ici était de ne pas arrêter le déplacements des autres entités si le joueur ne fournissait aucune entrée ( e.g. ne touchait pas a son clavier).

Il était possible de créer un deuxième thread sur lequel gérer les déplacements du joueur uniquement, mais je voulais garder la simplicité du space invaders.

C'est pourquoi j'ai choisit d'utiliser des fonctions présentes sur l'API de Windows, comme "WaitForSingleObject".

Dans ce contexte la, j'accepte une entrée user pendant 1000 ms. Sans entrée utilisateur, un cas TIMEOUT est déclenché et on passe a la suite.

Niveau difficulté, il fallait aussi faire en sorte que le jeu devienne plus dur au fil du temps, ce pourquoi le monstre change en fonction des ticks au moment du spawn (si le reste de la division du nombre de ticks actuel par 200 est a 0, on fait apparaitre un monstre intermédiaire, et un monstre fort pour 400 ticks).

Le jeu s'accélère aussi, il y a des variables a certaines valeurs au début, qui vont réduire au fil du temps, car tout le système de mise a jour des objets / balles / vaisseau se fait lorsque les ticks a un moment n peuvent diviser les valeurs et que le reste de cette division est de 0.