

CUSTOS CARCERIS

PROJET DE S2

# Deadly Science



*Léandre Perrot*  
*Yann Boudry*  
*Steve Suissa*  
*Célian Rimbault*

Un projet EPITA

17 janvier 2020

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Le Projet . . . . .	2
1.2	Objectifs . . . . .	2
<b>2</b>	<b>Fonctionnement</b>	<b>3</b>
2.1	Première phase . . . . .	3
2.2	Seconde phase . . . . .	3
2.3	Origine et nature du projet . . . . .	4
<b>3</b>	<b>Présentation des membres du groupe</b>	<b>5</b>
3.1	<i>Léandre Perrot</i> alias Majoran . . . . .	5
3.2	<i>Yann Boudry</i> . . . . .	5
3.3	<i>Célian Raimbault</i> . . . . .	5
3.4	<i>Steve Suissa</i> a.k.a SteveNoobGeek (chef de projet) . . . . .	6
<b>4</b>	<b>Étude du projet</b>	<b>6</b>
4.1	Objet de l'étude . . . . .	6
4.2	État de l'art . . . . .	6
<b>5</b>	<b>Découpage du projet</b>	<b>8</b>
5.1	Le moteur 3D . . . . .	8
5.2	Le labyrinthe . . . . .	8
5.3	Les joueurs . . . . .	8
5.4	Les objets . . . . .	8
5.5	Les sons . . . . .	8
5.6	Le gameplay . . . . .	9
5.7	Le réseau . . . . .	9
<b>6</b>	<b>Plannifications et outils</b>	<b>10</b>
6.1	Répartition des tâches . . . . .	10
6.2	Logiciels et outils informatiques . . . . .	10
6.3	Coût de production . . . . .	11
6.4	Avancement des tâches . . . . .	12
<b>7</b>	<b>Conclusion</b>	<b>13</b>

# 1 Introduction

## 1.1 Le Projet

Notre projet intitulé Deadly Science est un jeu 3D et multijoueur, il est de type FPS. Nous incarnons alors une personne infectée dans un laboratoire. Lors de chaque partie, 4 joueurs sont enfermés dans un laboratoire - une sorte de labyrinthe - et, étant infecté, chaque joueur est amené à rechercher un remède afin de survivre. Bien évidemment, il n'y aura pas assez de remèdes pour tout le monde. Le dernier joueur infecté doit alors se venger sur les joueurs guéris. Notre jeu se décompose alors en deux phases, la première étant destinée à rechercher le remède et la seconde quant à elle consiste en la survie des joueurs guéris face au dernier joueur tentant de se venger. L'ambiance du jeu sera sombre et lugubre afin d'immerger au maximum le joueur dans le gameplay.

## 1.2 Objectifs

Ce projet a pour but de plonger le joueur dans un jeu 3D et multijoueur. De plus, ce jeu apportera une expérience pour chaque membre du projet, en effet, il nous permettra de renforcer notre esprit logique en faisant face à des algorithmes complexes. Nous allons également améliorer nos compétences secondaires comme la modélisation 3D, le game / UI / UX design, les illustrations et également la façon dont nous pouvons nous projeter sur un grand projet.

## 2 Fonctionnement

### 2.1 Première phase

Les joueurs sont des personnes ayant été infectées par une maladie. Ils doivent alors récupérer les échantillons de sérum ayant été dissimulés dans les souterrains du laboratoire qui étudie la maladie. Seul problème : il s'agit d'un véritable labyrinthe, et il n'y aura pas de sérum pour tout le monde...

**Objectif :** Trouver un échantillon de sérum et se soigner avant que les autres joueurs ne s'emparent de tous les remèdes.

**Comment jouer :**

- Se déplacer
- Diriger la caméra
- Coup de boule
- Coup de pied
- Se baisser (Esquive de coup de boule)
- Sauter (Esquiver de coup de pied)

**Le labyrinthe :**

- Le labyrinthe est généré aléatoirement, ainsi que la position des sérums et des joueurs.
- Il est constitué de murs évoquant un laboratoire. (Murs uniformes blancs, vitrines avec tubes à essai, écrans avec statistiques, cuves de plexiglas remplies d'un liquide trouble...)
- Réussir à frapper quelqu'un l'assomme momentanément.

### 2.2 Seconde phase

Les sérums ont tous été consommés. Les joueurs ont été soignés, sauf un, dont l'état devient critique. Ce dernier devient alors le condamné : le temps qu'il lui reste avant de succomber à la maladie n'est donc plus qu'une question de minutes, et son contact devient léthal. Et comme si cela ne suffisait pas, le système de surveillance du laboratoire s'est activé, réveillant des pièges dans le labyrinthe...

**Objectif :**

- **Condamné :** Histoire de ne pas être le seul à y laisser sa peau, le condamné décide de se venger des autres joueurs. Un coup de boule, un coup de pied ou même une simple collision (mouvement) suffisent à terrasser un joueur. Une jauge est visible sur l'écran du condamné, et il se distingue par une coloration rouge.
- **Guéris :** Leur but est évidemment de survivre. Leur seule solution contre le condamné est de fuir, puisque lui donner un coup tue le joueur (cela assomme quand même le condamné). Ils peuvent jouer individuellement ou en équipe : seuls les vivants pourront gagner, ou s'il n'y a aucun survivant, c'est le condamné qui remporte la victoire.

**Le labyrinthe :** Les lumières tamisées blanches sont désormais en mode "intrusion", donc plutôt rouges. Les cadavres des joueurs constituent des obstacles,

la seule solution est de sauter par dessus (si cadavre il y a).

### **2.3 Origine et nature du projet**

On s'est rencontré très rapidement vu qu'on fait tous partie de la même classe. On a rapidement vu qu'on avait de nombreux points communs et puis on est devenu amis. À la base, le groupe était formé que de Léandre et moi(Steve). Je voulais faire à la base un Doom-Like, car j'adore le jeu originel et Léandre était fortement intéressé par un jeu de labyrinthe. Puis on a eu la bonne idée de mélanger nos deux idées pour créer un seul jeu qui s'appellera par la suite Deadly Science. Il nous fallait deux autres camarades pour créer le groupe alors on a premièrement opté pour Yann vu qu'on a déjà fait un exposé en techniques d'expression et cela s'est très bien passé puis on a choisi Célian pour son expérience dans le domaine de l'informatique. Comme dit précédemment, nous voulons faire un jeu qui se différencie de tous les autres projets de S2 et même des jeux vidéos. On a pris le côté horreur de Doom qui était à la base l'idée de Steve et les labyrinthes de Léandre avec son algorithme de générateur de labyrinthes. Le jeu que nous voulons vous proposer est un jeu en 3D dans un labyrinthe en multijoueur et possiblement solo aussi. Le jeu se scinde en deux parties.

## 3 Présentation des membres du groupe

### 3.1 *Léandre Perrot* alias Majoran

Enchanté, on me nomme Léandre. J’ai découvert l’informatique il y a deux ans avec l’option “ICN”. Je suis passionné depuis tout petit par les domaines de la logique, notamment les labyrinthes ou la conception de jeux. Je n’ai pas beaucoup d’expérience, étant donné que la majorité de mes programmes se trouvent sur calculatrice... Même si je connais maintenant assez bien le langage Python. En ce qui concerne le projet, c’est la première fois que je vais développer un jeu avec d’autres personnes, en 3D, et en multijoueur... J’espère donc apprendre plein de choses durant la progression de Deadly Science. Par ailleurs, j’aime beaucoup les jeux où il est ardu de prévoir ce qui va arriver... dont les labyrinthes. Heu... Ah, mince, je l’ai déjà dit ? Oups... Bon, peu importe. Quoiqu’il en soit, je souhaite que vous preniez du plaisir en vous égarant dans notre dédale... Voire en y périssant, qui sait ?

### 3.2 *Yann Boudry*

Bonjour, je suis Yann. J’ai découvert l’informatique l’année dernière en terminale et cela m’a immédiatement plu. Je suis également un fervent adepte des jeux vidéos depuis de nombreuses années (un moment de silence pour la DS et Asteroids). J’ai maintenant la chance de pouvoir combiner ces deux passions pour à mon tour créer des jeux et peut-être partager ces passions à mon entourage. C’est pourquoi ce projet m’attire, car il combine amusement, apprentissage et travail de groupe. J’aime voir le résultat de mon travail et des échecs (dédicace spéciale aux “;”). Cela permet de m’améliorer et de créer ce dont j’ai envie. Je connais le langage C, qui se rapproche du C#, que j’ai utilisé pour réaliser une bataille navale. Réaliser un jeu vidéo est une bonne manière de découvrir de nombreux éléments à gérer comme le son, le réseau ou les graphismes. Je souhaite participer de mon mieux au développement de ce jeu pour le rendre le plus complet possible.

### 3.3 *Célian Raimbault*

Bonjour, moi c’est Célian. Je suis passionné depuis plus de 5 ans par la programmation et l’informatique en général, je suis alors ravi de participer à ce projet en groupe. J’aime particulièrement les algorithmes et les intelligences artificielles. Mes points forts sont d’une part mon expérience, ayant déjà utilisé Unity, mais aussi d’autres moteurs de jeux lors de Game Jams j’ai déjà pensé à l’architecture du projet. De plus, j’aime réaliser des algorithmes complexes et la programmation système, ce qui me permet de comprendre globalement le fonctionnement d’Unity. Je me débrouille également artistiquement, je pourrais composer et produire la musique du jeu, réaliser des illustrations et modéliser des personnages en 3D. Les programmes que je code ne sont pas des jeux vidéos, mais réaliser un jeu vidéo est pour moi la meilleure option, car elle permet

d'avoir tout type d'algorithmes dans un programme et, car nous allons utiliser des compétences secondaires pour produire les ressources du jeu. Je connais divers langages, mais j'utilise beaucoup le Python ainsi que le C++, ce qui se rapproche du C# que nous allons utiliser pour la conception de tous nos algorithmes.

### 3.4 *Steve Suissa* a.k.a SteveNoobGeek (chef de projet)

Bonjour, je suis Steve ! Comme mon pseudonyme l'indique, j'ai toujours été un geek (comme une bonne majorité des membres d'EPITA) et un noob (même avec la pratique, j'étais pas très bon petit aux jeux vidéos, mais ça a changé, enfin pas trop non plus) . Fan des jeux vidéos depuis la très petite enfance, j'ai eu ma première console et mon premier ordi à 3 ans, mais même si je ne savais pas encore les utiliser, je les aimais déjà ! Le temps est passé, mais ma passion pour l'informatique, les sciences et les jeux vidéos n'a quant à elles pas changé !

Le Java, ça me connaît. J'ai commencé à programmer quand j'avais 14 ans en Basics et Lua, puis j'ai vu l'immensité du Java. J'ai commencé à apprendre pour faire des modes sur l'un de mes jeux préférés : Minecraft (je sais c'est pas très connu). Puis j'ai pris le goût à la programmation et c'est pour ça que je suis aujourd'hui à l'EPITA.

Lorsque j'ai pu regarder les différents types de projets disponibles, dès que j'ai vu l'option pour faire un jeu vidéo, je savais ce que je voulais faire. Je voulais utiliser toutes mes heures de jeu de mon enfance à des fins pratiques en utilisant mes connaissances en informatique et en jeu vidéo pour en créer un différent du déjà vu. Le défi est grand, mais je pense que notre équipe va surmonter les problèmes et que l'on arrivera à un projet merveilleux.

## 4 Étude du projet

### 4.1 Objet de l'étude

Ce projet a pour nous plusieurs buts et intérêts. Il nous permet premièrement de nous entraîner à la POO (Programmation Orienté Objet) en utilisant un langage de programmation qui est le C#. Il nous permet aussi de travailler sur une interface différente de celle que nous avons précédemment utilisée, car l'ingénieur doit savoir vivre avec son temps. Le projet a aussi pour but de nous faire travailler en équipe et savoir gérer les différentes tâches, car un ingénieur travaille souvent en équipe.

### 4.2 État de l'art

Comme dit précédemment, on s'est inspiré de Doom pour le côté horreur et de Portal, pour le côté labyrinthique et l'esthétique du labyrinthe. Même si le jeu est très différent du reste, on pourrait l'assimiler à Garry's Mod ou sur le principe à Pac-Man, voire même Bomberman. Chacun a des avantages. Garry's Mod a la particularité d'être très modulable et personnalisable par la

communauté. Bomber-Man lui avait un gameplay très innovant lorsqu'il est sorti et est un jeu multijoueur dans un labyrinthe. Pac-Man quant à lui est un pionnier pour son époque et une référence dans le domaine du jeu d'arcade. Il est vrai que notre jeu se base principalement sur des "vieux" jeux, par exemple, Pac-Man et Bomber Man sont sortis dans les années 1980 bien que Garry's Mod soit sorti en 2006.



## 5 Découpage du projet

### 5.1 Le moteur 3D

Pour réaliser ce projet, il nous est nécessaire de posséder un moteur graphique permettant d'afficher un labyrinthe en trois dimensions. Pour ce faire, nous avons opté pour le logiciel Unity, qui permet de générer aussi bien des labyrinthes que des personnages et permet également de gérer les collisions, la gravité, et ainsi de suite. En outre, Unity est un outil que certains d'entre nous ont déjà manipulé.

### 5.2 Le labyrinthe

L'un des points les plus importants du projet réside dans le labyrinthe. En effet, le jeu perdrait tout son intérêt si les joueurs se retrouvaient à chaque fois dans les mêmes couloirs : ceux qui auraient déjà joués seraient donc largement valorisés que les nouveaux joueurs, de par leur connaissance du terrain, notamment avec la position des sérums. Il nous faut donc mettre au point un générateur de labyrinthes aléatoires fermés, permettant cependant d'accéder à toutes les zones du labyrinthe pour éviter que les sérums ne soient hors d'atteinte. De plus, le labyrinthe doit permettre de fuir un danger : il ne doit donc pas y avoir qu'un seul chemin correct pour se rendre d'un point à un autre du dédale.

### 5.3 Les joueurs

Chaque joueur est censé pouvoir se déplacer, mais également interagir avec les autres joueurs et les objets. Il nous faut donc permettre à quatre ordinateurs de se connecter à un même réseau afin de permettre à quatre joueurs de jouer en même temps. De plus, il nous faut prendre en charge les animations des déplacements des joueurs, la caméra qui sera en première vue, les collisions entre joueurs, notamment avec le joueur malade, et les objets qu'ils transportent.

### 5.4 Les objets

L'un des points clés du jeu, c'est la récupération de sérums. Il nous faut donc mettre au point un système de récupération d'objets. Un objet s'active automatiquement lorsque le joueur le ramasse. Pour l'instant, nous nous concentrerons sur les sérums, qui seront matérialisés par des fioles. Mais si nous en avons le temps, il se pourrait que nous ajoutions d'autres objets dans le labyrinthe, avec des effets positifs (augmentation de la vitesse) aussi bien que négatifs (immobilisation temporaire).

### 5.5 Les sons

Les sons seront importants dans le jeu : ils peuvent donner de précieux indices sur la présence d'autres joueurs dans le labyrinthe (bruits de pas). L'ambiance

sonore doit également permettre au joueur de se plonger dans l'action, il n'y aura donc peut-être pas de grandes musiques, mais plutôt de petits bruitages reproduisant l'ambiance d'un laboratoire (bouillonnement, craquement, bruit de récipients qui s'entrechoquent) afin de rendre une atmosphère à la fois inquiétante et pesante.

## 5.6 Le gameplay

Deadly Science est un jeu qui se jouera au clavier-souris ; la souris permettra de gérer la caméra et de ramasser des objets, le clavier permettra de se déplacer. Ainsi, il sera possible au joueur de regarder derrière lui simultanément, et de regarder autour de lui tout en se déplaçant, de manière fluide et intuitive. Il va être important d'adapter les mouvements des joueurs au style du jeu, pour cela, Unity fournit un système très efficace pour changer chaque paramètre d'un objet.

## 5.7 Le réseau

Afin que plusieurs joueurs puissent jouer à notre jeu, il va falloir implémenter une gestion de réseau, c'est-à-dire contrôler qu'elles modifications un joueur peut apporter aux autres ou recevoir par les autres. Nous allons utiliser la bibliothèque Photon qui permet de gérer les événements liés au réseau. Il est important que notre jeu se construise autour du réseau, car c'est un des éléments primaires du jeu, il est alors important de commencer le projet en créant un "lobby" où chaque joueur pourrait se connecter correctement.

## 6 Plannifications et outils

### 6.1 Répartition des tâches

Chaque membre du projet a des compétences et une expérience différentes, il est alors obligatoire de scinder ce projet en diverses tâches réparties entre les membres du groupe. Voici comment nous nous répartissons les tâches (E pour Esthétique, G pour Gameplay, Const pour Construction, TXT pour Texte et Gen pour Génération).

	Léandre	Yann	Célian	Steve
Caméra			S	R
G Joueur		S	R	
G Jeu	S			R
Réseau	S			R
Map Const	S			R
Menu		R		S
Chrono/GUI	R			S
Site TXT	S	R		
Site E		R	S	
Map Gen	R	S		
Musique	S		R	
Sons	R		S	
Effets			R	S

#### Répartition des tâches entre responsables et suppléants

*R* = Responsable, *S* = Suppléant

### 6.2 Logiciels et outils informatiques

Pour réaliser ce projet, il est évident que nous allons devoir nous munir des outils les plus adaptés. Unity : Le moteur de jeu qui permet également d'éditer tout type de propriétés pour construire notamment les niveaux du jeu.

- **Photon** : Que serait le multijoueur Unity sans Photon, un plugin simple d'emploi pour la synchronisation multijoueur.
- **Git et GitHub** : Git est un outil formidable que nous avons beaucoup utilisé cette année, il permet de synchroniser et de gérer les changements que chacun a apportés. Pour que celui fonctionne en réseau, nous allons utiliser GitHub qui permettra d'héberger un serveur Git, de plus nous pourrons voir les changements directement sur le site.
- **Rider / Visual Studio** : Ces deux environnements de développement intégrés permettent l'édition du code, mais aussi apportent plein de fonctionnalités très utiles pour de grands projets comme le debugging.
- **Blender** : Ce logiciel sera utilisé pour tout ce qui est 3D, c'est-à-dire la modélisation, mais aussi l'animation. Nous pourrons par la suite tester nos textures ou shaders avant de les intégrer dans Unity.

- **Photoshop** : Permettant le traitement d'images, Photoshop sera utilisé pour les textures du jeu, en effet il faudra apporter un traitement supplémentaire pour que les textures soient adaptées à la forme d'un objet 3D.
- **The GIMP** : Pour les pauvres qui ne peuvent pas se payer Photoshop.
- **Illustrator** : Celui-ci pourra servir lors de la décoration pour ajouter des illustrations comme des panneaux par exemple.
- **FL Studio** : Le traitement audio sera réalisé par ce logiciel pour que l'ambiance sonore soit la plus adaptée au jeu. Il permettra également de composer la musique.
- **LaTeX** : Il sera utilisé afin de faire la mise en page de nos rapports.

### 6.3 Coût de production

Ce projet n'a pas pour but d'être un jeu AAA, c'est pour cela que nous allons utiliser au maximum des logiciels gratuits ou open source. Il nous faut néanmoins certains produits, les voici :

Nom	Prix (euros)
Nourriture	4340
Ordinateurs	4000
EPITA	22800
Volonté	Cher
Antidépresseurs	250

#### Coût du projet

## 6.4 Avancement des tâches

Notre projet présentera trois versions majeures, chacune étant associée à une soutenance. Afin de pouvoir nous situer dans le projet, nous avons créé un planning d'avancement.

Tâches / Soutenances	Première	Seconde	Dernière
Caméra	50%	75%	100%
G Joueur	30%	75%	100%
G Jeu	30%	60%	100%
Réseau	50%	75%	100%
Map Const	30%	60%	100%
Menu	15%	55%	100%
Chrono/GUI	30%	60%	100%
Site TXT		75%	100%
Site E		50%	100%
Map Gen		50%	100%
Effets		30%	100%
Musique			100%
Sons			100%

**Planning d'avancement**

## 7 Conclusion

Certes, on pourrait croire à premier abord que nous vous promettons un jeu de labyrinthe comme on en voyait déjà dans les salles d'arcade 30 ans plus tôt. Mais notre jeu a le grand défi de se démarquer de ce genre et de vous montrer avant tout nos compétences en programmation et ce que l'on peut faire de mieux. Nous concevons aussi l'idée que ce cahier des charges semble être assez simple, mais c'est pour qu'on puisse ajouter des fonctionnalités supplémentaires durant les 7 mois de développement de notre jeu. S'il y a bien une chose à se souvenir sur Deadly Science, c'est que nous ferons le maximum !