



PROJET PROGRAMMATION S2

RAPPORT DE SOUTENANCE 2

G.O.A.T.

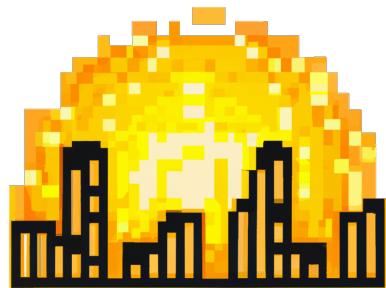
EPITA

INFORMATIQUE PRATIQUE

---

**Go On Another Town**

---



*Auteurs :*

Denis MIROCHNIKOV,  
Zachée DESCAMPS,  
Dorian FOROT,  
Marc DESSEVRE

*Enseignant référent :*  
Martin Van Laere

17 Avril 2023

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Origine et nature du projet . . . . .	2
1.2	Présentation des membres du groupe . . . . .	3
1.2.1	Denis Mirochnikov . . . . .	3
1.2.2	Zachée DESCAMPS . . . . .	3
1.2.3	Dorian FOROT . . . . .	4
1.2.4	Marc DESSEVRE . . . . .	4
<b>2</b>	<b>Réalisation, du point de vue de :</b>	<b>5</b>
2.1	Zachée Descams . . . . .	5
2.1.1	La Grande Refonte . . . . .	5
2.1.2	Le Multiplayer . . . . .	6
2.1.3	Les Niveaux Aquatiques . . . . .	6
2.2	Mark Dessevre . . . . .	7
2.2.1	Le monde du jeu . . . . .	7
2.2.2	Le menu . . . . .	9
2.3	Dorian Forot . . . . .	10
2.3.1	Les entités . . . . .	10
2.3.2	Les Items . . . . .	11
2.3.3	L'IA . . . . .	12
2.4	Denis Mirochnikov . . . . .	13
2.4.1	La décoration . . . . .	13
2.4.2	Les Mini Jeux . . . . .	14
2.4.3	Site-Web . . . . .	20
<b>3</b>	<b>Diverses retards</b>	<b>24</b>
3.1	Quels retards ? . . . . .	24
3.2	Tableau d'avancement . . . . .	24
<b>4</b>	<b>Conclusion</b>	<b>25</b>

## 1 Introduction

### 1.1 Origine et nature du projet

Dans le cadre de notre projet InfoSup, nous avons décidé de créer un jeu vidéo. Celui-ci, intitulé "Go On Another Town", invite le joueur à progresser dans un univers labyrinthique en 2D, avec pour objectif d'atteindre la ville de GOAT, un présumé el-dorado de la survie. En effet, ce monde s'est mué en un environnement hostile où la survie devient un enjeu crucial.

Dans ce rapport, nous avons le plaisir de vous présenter l'origine de nos idées et de vous expliquer comment elles ont été concrétisées.



## 1.2 Présentation des membres du groupe

### 1.2.1 Denis Mirochnikov

Depuis mon plus jeune âge et jusqu'à maintenant j'adore la créativité, ça se ressentait par mes centres d'intérêt qui étaient assez différents de ceux des jeunes de mon âge. Au lieu de jouer aux jouets comme la majorité des enfants de 6 ans, je passais des journées à essayer de bricoler des objets dans le garage avec mon père. Il m'apprenait d'ailleurs énormément de choses sur la mécanique, et surtout sur le fonctionnement des différents objets comme par exemple un moteur électrique, ou même une pile. En bref on faisait pas mal d'expériences scientifiques ensemble afin que je puisse comprendre par la pratique le fonctionnement de ce monde.

C'est à partir de ce moment-là que j'ai compris que j'avais envie de lier ma vie à la science... Quelque chose de si vaste et complexe, mais tellement passionnant !

Et puis en 2014 tout s'est ruiné, on a dû quitter notre pays et commencer une nouvelle vie... Pour essayer de compenser les problèmes, je me suis intéressé à quelque chose qui était nouveau pour moi - la programmation. Elle m'a passionné, et j'ai commencé de plus en plus à m'intéresser et de découvrir de nouveaux langages de programmation, à participer à des concours ou même des expositions à travers des associations de mon lycée. À ce jour mon langage préféré reste le C++, même si j'apprécie énormément Python, Java et bien sûr C# .

Je pense que ce projet me permettra d'approfondir mes connaissances en création du jeu vidéo et surtout m'apprendra à travailler en équipe.

### 1.2.2 Zachée DESCAMPS

Depuis l'aube de l'humanité , l'humain est mauvais pour faire des introductions c'est pourquoi je vais passer dans le vif du sujet. J'ai toujours aimé créer des objets ou logiciels depuis tout petit , ça a commencé par créer des montes charges sur ma mezzanine et collé des kaplas ensemble au pistolet à colle ( je regrette cette action ils sont maintenant hors d'usage) puis par l'obtention de mon premier pc j'ai commencé à m'intéresser à la programmation par ce que je voulais parfaire mon image de geek ( c'est la réelle raison ) , j'ai commencé en logo puis j'ai continué en java pour faire des mods minecraft assez douteux (mais java est resté mon langage préféré jusqu'à aujourd'hui) . J'ai au fil des années porté mon attention sur le C et le C++ pour optimiser certains logiciels qui étaient trop coûteux en mémoire et calcul pour du java , sur le python qui m'as fait dire mes premières phrases de rabat joie tel que "c'est un langage pour ceux qui ne veulent pas programmer " ou encore "mélangé la mise en page et la syntaxe c'est une hérésie" ( oups le dernier est vrai ) le tout en passant par le powershell qui m'a laissé des séquelles ( plus jamais je ne veux voir des réseaux hyper V ) et encore d'autres comme le batch , la SQL , le javascript , le php et rapidement de l'assembleur x86 et 6502 pour du reverse engineering.

Bref j'aime la programmation parce que grâce à elle et avec juste un pc l'on peut créer des outils utiles au quotidien contrairement à l'électronique , la chimie et l'impression 3D qui demande des ressources coûteuse ( mes trois autres passions ).Pour ce qui est du jeux vidéo , je n'ai quasiment jamais eu de console .Pour jouer tous ce que j'avais sous la main était un vieil ordinateur portable qui laguait sur club penguin mais j'ai découvert assez tôt le monde de l'émulation c'est pourquoi les jeux de mon enfance sont les jeux les mieux notés sur Romstation ( Mario 64 , Ocarina of Time , Chrono Trigger , Banjo Kazooie ,etc .. ). Ce projet me permettra d'apprendre à travailler en groupe , ce à quoi je suis vraiment mauvais(cf. l'appréciation Parcoursup de mon professeur de NSI de terminale )

### 1.2.3 Dorian FOROT

Éminemment beau et spécialiste Ocaml et sortant de terminale, le bac en poche avec les spécialités Math-NSI, je me suis lancé dans l'aventure EPITA. Je ne dirais pas que je suis passionné par l'informatique, mais j'aime programmer. Depuis le collège je m'intéresse à ce domaine sans pour autant m'y lancer, par manque de temps et bien sûr de motivation. La spécialité NSI au lycée m'a initié à certains langages de programmations tel que le Python, mon amour de toujours, le php , la SQL ou encore le html css. Mais c'est oublier le précurseur de tout cela, le meilleur langage de programmation all time, celui avec lequel tout a commencé, j'ai nommé, Scratch. Comment pourrais-je ne pas rendre hommage au logiciel grâce auquel tout a commencé, celui qui m'a fait découvrir et aimer la programmation (même si c'est un bien grand mot pour ce que c'est). J'étais très fort en Scratch croyez le ou non, dans mon collège c'était moi le pro associable qui allait au cdi comme un sans ami faire du Scratch.

J'ai eu la chance (ou pas) d'avoir déjà eu l'occasion de faire un projet de ce type en NSI durant ma première avec mon professeur de NSI (cœur sur vous Mr.Lebret). Avec mon équipe de choc (non) nous avions codé de toute pièce un fps multijoueur (sans multijoueur mais c'est une histoire à part). Je connais donc la difficulté d'un tel projet, l'organisation et la rigueur qu'il faut avoir. J'ai très hâte de commencer le projet et de me donner à fond pour réussir à réaliser les objectifs que nous nous sommes fixés.

### 1.2.4 Marc DESSEVRE

Tout juste sortie de terminale en spécialité Math-Physique, comme beaucoup d'autres, j'ai décidé de me tourner vers EPITA car l'informatique m'a toujours remplie d'admiration. Je n'avais jamais osé m'y lancer pour faute de matériel et par peur de me perdre dans cet univers aussi grand. Cependant, aujourd'hui je suis à l'EPITA, en plein apprentissage du C#, du python et d'Ocaml, et la création d'un jeu vidéo est une très grande étape à franchir dans mon parcours.

D'autant plus que les jeux vidéo me passionnent (j'ai fait mon stage de troisième à Quantic Dream par exemple), ceux-ci m'apparaissent de plus en plus complexes au fil de mon apprentissage. J'ai par exemple appris l'existence d'une boucle de jeu très récemment.

Ainsi, ce projet est à la fois un défi important dans mon apprentissage, mais aussi une manière de mieux comprendre les fondations et la structure d'un jeu vidéo.

## 2 Réalisation, du point de vue de :

### 2.1 Zachée Descams

#### 2.1.1 La Grande Refonte

Pour cette nouvelle soutenance, j'ai commencé par faire ce que j'avais annoncé lors de la première : changer de bibliothèques graphiques. Et oui, à la base, j'utilisais Winforms car c'est ce qui se rapproche le plus de Jswing en C# (ayant toujours programmé en Java, je pouvais reprendre d'anciens codes). Mais malheureusement, l'utilisation de Winforms s'est avérée être un frein, en effet, Winforms a la fâcheuse tendance à ne pas utiliser le GPU, ce qui nuit gravement aux performances du jeu : plus l'écran sur lequel on jouait était grand, plus le jeu laguait. Je me suis donc mis à la recherche d'une alternative : Raylib. Raylib est une bibliothèque écrite en C et, à vrai dire, faite pour le C et C++, mais on peut l'utiliser depuis du C# en tant que librairie dynamique (DLL). Raylib est exactement ce que je recherchais : une bibliothèque simple, une sorte de SDL simplifiée, parfaite pour mes besoins, à la fois en simplicité de code mais aussi de performances. J'ai donc passé plusieurs jours à transformer ce que j'avais fait pour Winforms de manière à ce que Raylib puisse l'utiliser. Le résultat est le suivant : le jeu n'a plus aucun lag et peut tourner à 200% de sa vitesse de base. Mais malheureusement, la gestion des images s'est avérée plus complexe car les images deviennent des Texture2D de Raylib, qui sont donc des structures à la place d'objets. Mais étant donné les outils de développement et les classes faites pour encapsuler ces manipulations, cela n'a pas eu d'impact sur le travail de mes camarades. Mais subsiste un dernier problème, Raylib est du C et donc cela veut dire pas de Garbage Collector qui me mâcherait le travail. J'ai donc dû trouver une solution pour éviter de trop grandes fuites de mémoire tout en faisant en sorte que mes camarades n'aient pas à s'en soucier. J'ai donc rajouté à chaque type d'entité une méthode d'autodestruction propre appelant à leur mort ou à un changement de niveau.

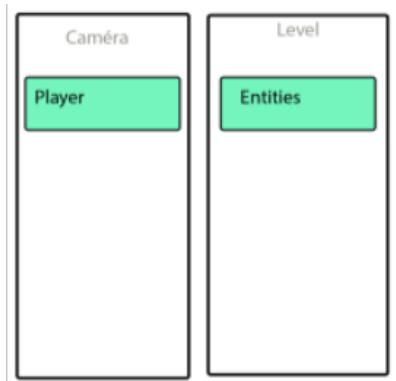


FIGURE 1 – Avant

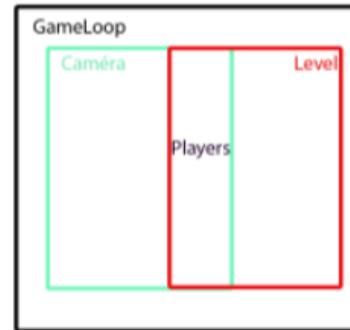


FIGURE 2 – Apres

### 2.1.2 Le Multiplayer

Le multijoueur est ce qui est le plus simple : deux joueurs, deux caméras, un écran splitté. Cependant, même si cela semble simple en théorie, j'ai créé le moteur pour un seul joueur et j'ai mis beaucoup de variables en statique, comme la caméra ou le joueur, car il ne pouvait n'y en avoir qu'un. Passer d'un jeu solo à un jeu multijoueur est donc devenu une tâche difficile, j'ai dû refaire l'ensemble de la structure du moteur pour permettre à plusieurs joueurs de jouer simultanément.

Une fois que la structure a été modifiée, j'ai dû adapter le comportement de toutes les entités pour qu'elles puissent interagir entre elles et avec le joueur à travers cette nouvelle structure. Lorsque je crée une structure, je regarde d'abord quelles données doivent être facilement accessibles en fonction de l'endroit où je me trouve dans le code.



### 2.1.3 Les Niveaux Aquatiques

Quelque chose de plus simple mais aussi plus visuel pour ajouter de la variété dans le gameplay, j'ai ajouté la possibilité de nager. En utilisant les zones dangereuses (celles qui tuent le joueur s'il y entre), j'ai modifié le code pour avoir un moyen simple et peu coûteux en performance pour savoir si le joueur est dans l'eau.



## 2.2 Mark Dessevre

### 2.2.1 Le monde du jeu

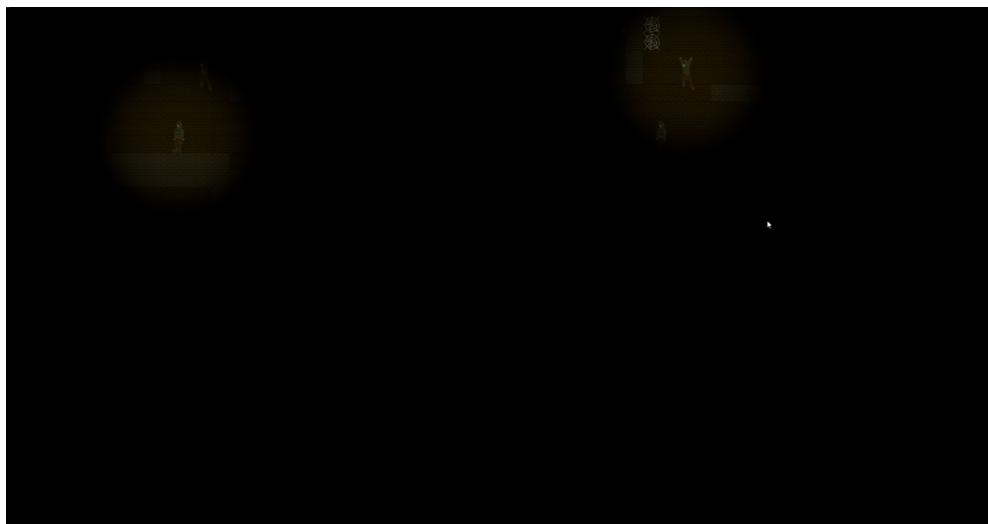
Avant même de pouvoir dessiner nos niveaux, nous avons dû nous décider sur l'univers dans lequel évoluerait le joueur. Nous avons choisi de nous situer dans un univers post-apocalyptique, ce qui nous permet de créer des niveaux dans un monde qui nous est relativement familier. En effet, nous n'avons pas besoin de créer un monde plausible de toute pièce, le nôtre nous offrant déjà tout ce dont nous avons besoin en termes de design.

Nous avions alors un monde. Il nous restait maintenant à l'exploiter. J'ai fait une première ébauche du monde, avec ses différents environnements et leurs particularités. Par exemple, les cryptes seraient sombres et impraticables sans lampe ou torche, ou encore les montagnes seraient infranchissables sans matériel d'escalade. Le monde semblait alors plutôt cohérent.

Mais alors que je dessinais les niveaux et les implémentais dans le jeu, nous nous sommes rendu compte que ces niveaux étaient bien trop linéaires et ne ressemblaient en rien à un metroidvania. Il y avait bien quelques allers-retours par-ci par-là, mais rien de vraiment labyrinthique. J'ai alors repensé le monde, en dispersant les objets obligatoires un peu partout, mais surtout en retirant les zones inutiles. De 11 environnements, nous sommes passés à 7. De plus, cette baisse du nombre total d'environnements a permis de renforcer la cohérence du monde, en retirant la présence d'un désert qui n'avait rien à faire là, par exemple.



Ce monde sera sensiblement le même, en jouant seul ou à plusieurs. Les seules différences se feront au niveau des salles et des différentes fonctionnalités. Jouer seul sera davantage un jeu de plateforme, tandis qu'à deux, le jeu proposera quelques énigmes (en déplaçant certains blocs, en appuyant sur des interrupteurs, ou encore avec des plateformes qui ne bougent que si le nombre de joueurs dessus est le bon). Les tailles des salles seront différentes afin de permettre à la caméra de notre jeu de ne pas avoir de problèmes d'affichage : les salles en solo peuvent être plus petites, car il n'y a qu'un seul joueur à afficher, en duo le ratio doit être de 8 :9 et les salles bien plus grandes.

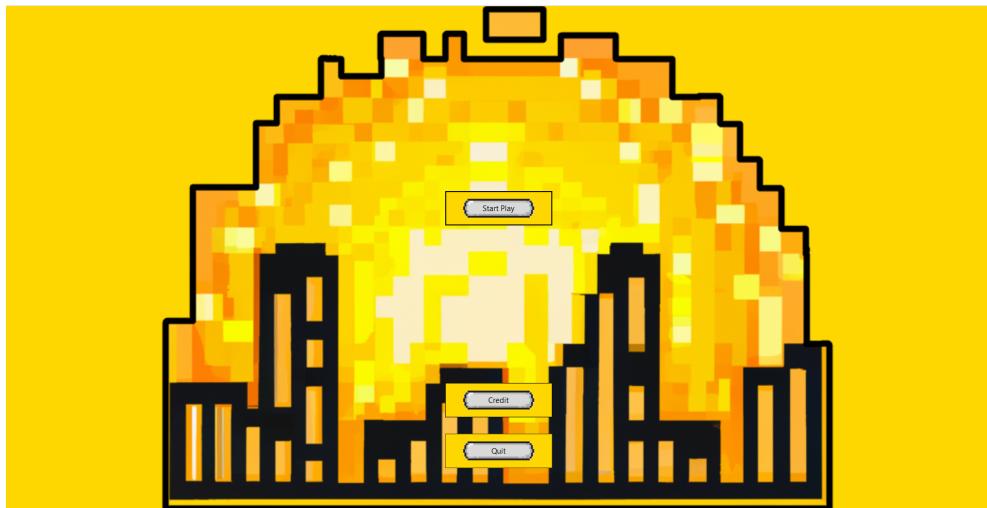


### 2.2.2 Le menu

Lors de la première soutenance, j'ai été chargé de créer un menu de lancement pour le jeu. Pour ce faire, j'ai dû me renseigner sur les différentes manières de manipuler Winform. Après quelques recherches et essais infructueux, j'ai identifié le problème : la méthode de lancement du projet utilisait également Winform, rendant impossible l'utilisation de `Process.Start()`. J'ai expérimenté diverses méthodes, telles que passer le projet en référence ou créer une variable de type `Process` pour ajouter des arguments lors du lancement du .exe. Finalement, la dernière méthode a fonctionné, bien qu'avec quelques difficultés.

Une fois l'aspect technique résolu, il était temps de se concentrer sur l'aspect esthétique. Dans l'ensemble, cette étape s'est déroulée sans encombre. Les seules difficultés rencontrées étaient dues à une connaissance insuffisante de Winform : les boutons s'affichaient mal en plein écran et l'image d'arrière-plan ne se centrait pas correctement. J'ai donc opté pour un style Pixel Art aux teintes jaunes-oranges.

Depuis la première soutenance, nous avons réussi à intégrer le menu de lancement au jeu tout en conservant le système Winform, bien que le jeu tourne sur Raylib. Les raisons de la migration de WinForms sur Raylib seront expliquées par mon collègue en charge de cette tâche.

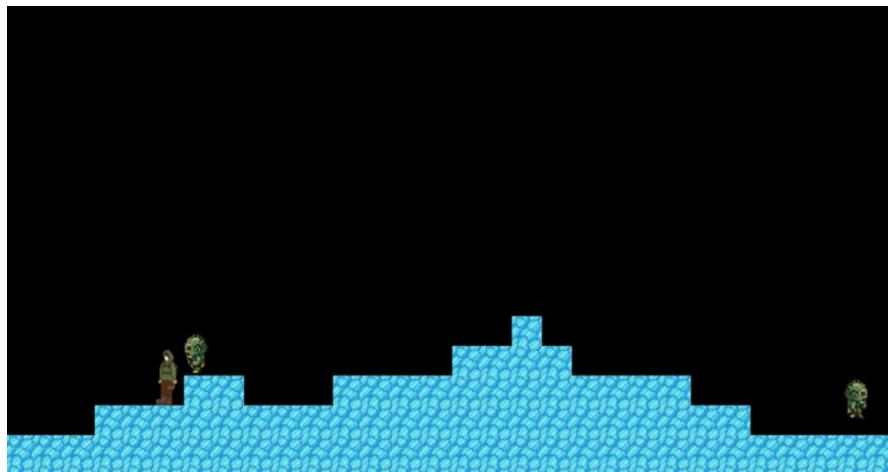


## 2.3 Dorian Forot

### 2.3.1 Les entités

Pour ce qui est des entités, comme lors de la première soutenance j'ai continué l'implémentation de nouvelles entités et surtout j'ai perfectionné celles déjà existantes. C'est notamment à coup de débogage que j'ai pu obtenir des monstres qui interagissent vraiment avec le niveau et le joueur sans bug. L'un des problèmes principaux était de réussir à faire sauter les entités au-dessus des obstacles. Cela peut paraître simple mais sans Unity ça se complique ! En effet pour avoir les collisions et les faire sauter il fallait d'abord pour différencier si la collision se fait sur l'axe X ou l'axe Y, à ça s'ajoute de nombreux tests, comme par exemple si l'entité est déjà en l'air ou non, si elle interagit avec le joueur. De plus nous avons décidé, afin de pouvoir mettre à jour le niveau, de créer une fonction update, celle-ci comme son nom l'indique assez explicitement update le niveau. Le problème étant que si l'on met à jour le niveau trop rapidement de nombreux problèmes peuvent se produire, notamment donc au niveau du passage des murs. En effet le problème qui se posait était que les entités n'avaient pas le temps de passer le mur car le niveau et les entités elles-mêmes s'updateait trop vite, et donc elles ne réagissaient pas comme convenu. Pour remédier à cela il a fallu créer une nouvelle fonction update, elle dépendante uniquement des entités pour permettre de débloquer certains bugs. Toutefois cela a créé d'autres problèmes en effet vu que les entités de s'updateaient pas à la fréquence que les autres éléments du niveau cela a créé des bugs notamment le fait que les entités ne prenaient pas les dégâts, ne disparaissaient pas et plus généralement interagissaient mal avec le joueur. Ce problème a été en partie réglé, en revanche il manque un petit perfectionnement au niveau des classes entités notamment pour les IA. Je reviendrais plus en détail sur ce point dans la partie IA.

Pour cette deuxième soutenance nous avons décidé de ne pas implémenter beaucoup plus de nouvelles entités mais plutôt de perfectionner celles déjà existantes, seul un squelette a été ajouté en tant que réelles entités, je dis cela car comme mentionné à la soutenance 1 nous avons décidé de passer plusieurs Item en entités pour avoir une bien meilleure interaction avec. Pour ce qui est des designs nous n'y avons pas touché car ils nous conviennent amplement. En effet pour nous il est inutile d'avoir un nombre incalculable d'entités diverses-et-variées, selon nous un Metroid-Vania est avant tout un jeu simple qui doit nous faire réfléchir et non pas nous montrer toutes sortes de créatures diverses et variées.

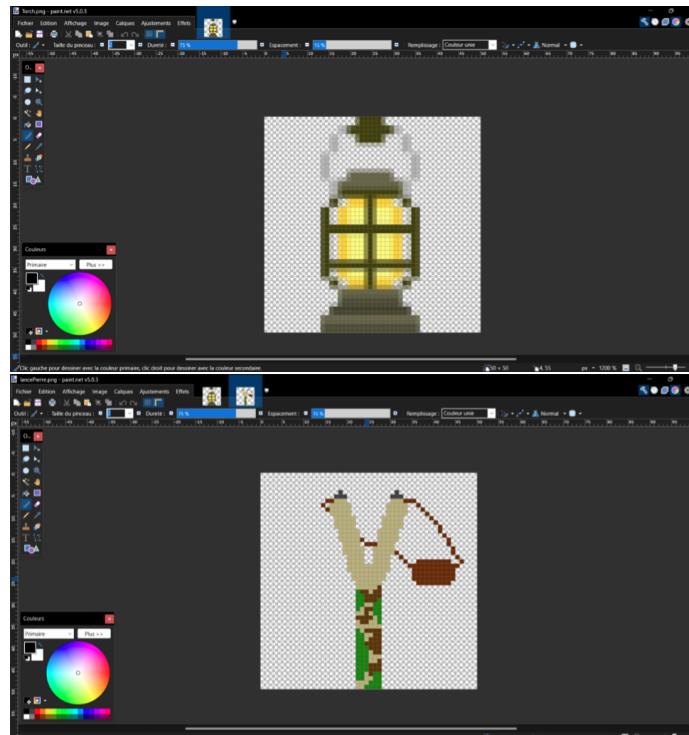


### 2.3.2 Les Items

A fortiori, pour ce qui est des items, un grand travail avait déjà été fait lors de la première soutenance. Pour cette deuxième soutenance, nous avons simplement implémenté d'autres items. Avec l'implémentation de l'inventaire et grâce notamment à LINQ que nous avons continué d'utiliser, il était assez simple de manipuler les items, les supprimer du niveau, pouvoir les réutiliser et gérer les stocks.

Assez simplement, nous avons donc ajouté des items apportant chacun une nouvelle mécanique au jeu. Notamment et non des moindres, un lance-pierre récupérable sur la carte permet de repousser les monstres qui nous attaquent. Le lance-pierre fonctionne grâce à des munitions elles aussi récupérables sur la carte, par petits stocks. Ce lance-pierre ainsi que les munitions sont stockés dans l'inventaire et le joueur peut tirer sur les monstres tant qu'il possède des balles. Le lance-pierre fonctionne comme un item ; néanmoins, les munitions fonctionnent comme des entités, car il était nécessaire de les faire déplacer pour obtenir un meilleur rendu esthétique. Ainsi, nous avons créé deux classes distinctes : une pour spécifier que c'est un item et pouvoir le stocker dans l'inventaire, et une autre classe pour pouvoir effectuer toutes les modifications nécessaires sur la balle en jeu, comme si c'était une entité. De cette manière, les balles tirées peuvent interagir avec les décors, les éventuels autres joueurs et, bien sûr, les entités.

Grâce aux items, nous avons également pu perfectionner le système de sortie, et notamment les portes. Avec l'ajout des clés et des interrupteurs, cela permet une plus grande réflexion au joueur dans le jeu et offre davantage de possibilités de gameplay. Ces deux items sont donc stockés dans l'inventaire du joueur et peuvent être utilisés à tout moment sur une porte correspondante.

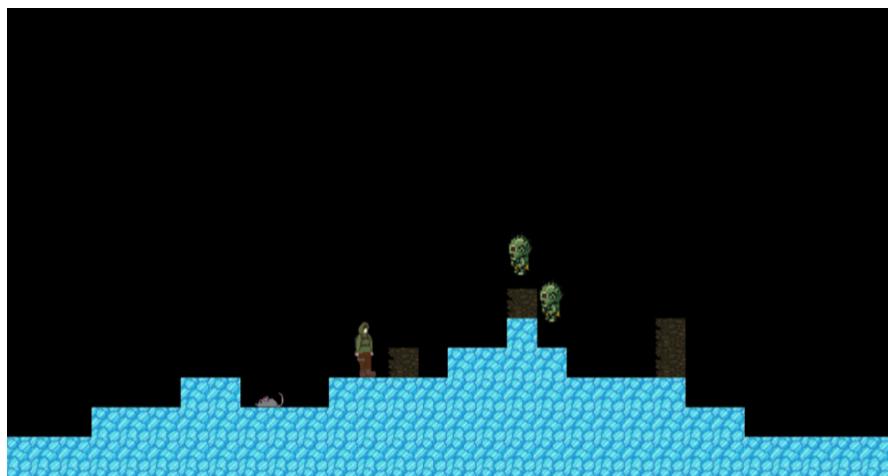


### 2.3.3 L'IA

Il est nécessaire de créer une IA, à la fois parce que cela est demandé, mais surtout parce qu'un jeu vidéo serait ennuyeux s'il ne se passe rien. C'est pourquoi nous avons implémenté des entités. Bien que certaines n'aient rien d'IA, d'autres sont capables d'interagir avec le niveau et le joueur et d'effectuer des actions de manière fonctionnelle et aléatoire. Ainsi, nous avons implémenté le zombie, une entité qui hérite de la classe `TriggerEntity`. Tant que le joueur n'est pas dans sa "zone de déclenchement", le zombie se déplace aléatoirement selon son environnement. Lorsque le joueur entre dans cette zone, le zombie court vers lui. J'ai fait en sorte que, tant que le joueur et le zombie ne sont pas suffisamment proches, le zombie réagisse en fonction du niveau et du décor, mais de manière aléatoire. Par exemple, si le zombie rencontre un mur devant lui, il choisira aléatoirement de franchir le mur ou de faire marche arrière. La décision du zombie est ainsi déterminée à l'aide d'un générateur de nombres aléatoires pour éviter qu'il fasse toujours la même chose.

Le problème qui se posait, comme mentionné dans la partie sur les entités, était de leur permettre de franchir les murs. La solution pour pallier ce problème a été la refonte complète des entités dites "IA". Ainsi, nous avons modifié la structure de déplacement de celles-ci, ce qui leur permet d'avoir des actions dépendantes de leur fonctionnement.

Un autre débat concernant les entités IA était de savoir si elles devaient suivre le joueur entre les niveaux. Nous avons décidé que ce n'était pas une bonne idée pour deux raisons. La première était la difficulté de l'implémentation d'une telle modification. En effet, les entités sont stockées dans les niveaux à l'aide d'un "sac" (bag), qui est une liste, et ces sacs sont chargés directement à partir du chargement du niveau avec le moteur graphique. Ainsi, ajouter des entités traversant les niveaux s'avérait quelque peu compliqué, mais pas impossible. L'autre raison était simplement que, pour nous, cela n'avait aucune utilité. En effet, nous avons tous déjà joué à un jeu où l'on se fait poursuivre par un monstre pendant une demi-heure sans pouvoir rien faire pour l'arrêter, et nous savons tous à quel point cela peut être ennuyeux. Ainsi, pour nous, il n'y avait aucun avantage à réaliser cette manipulation.



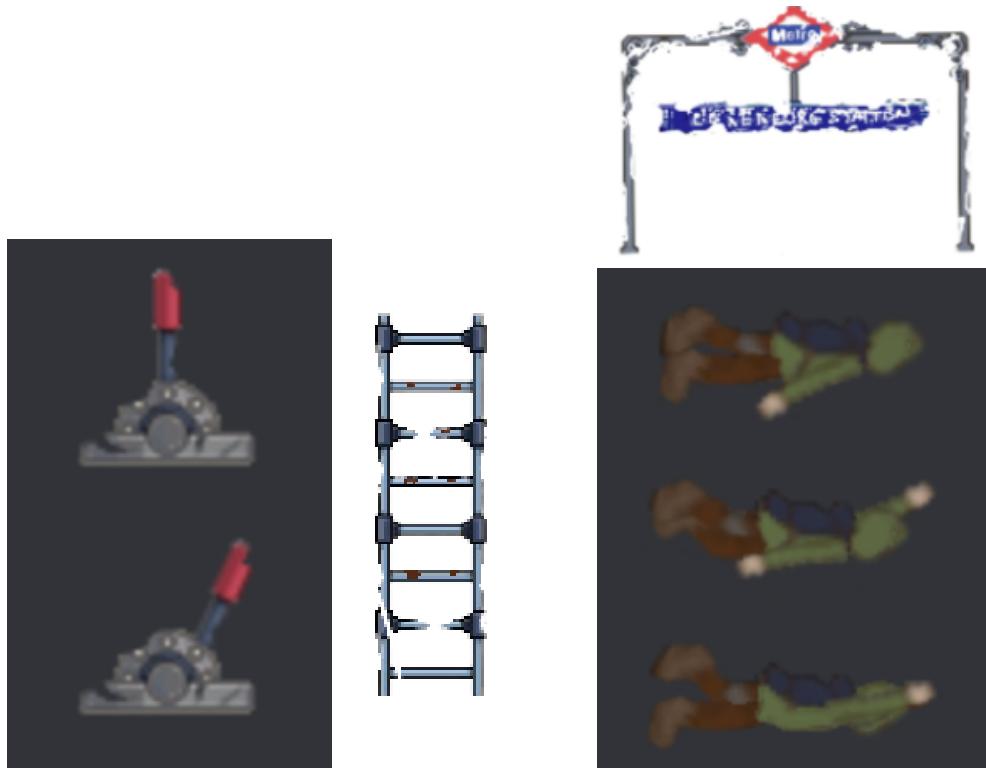
## 2.4 Denis Mirochnikov

### 2.4.1 La décoration

Lors de la première soutenance, nous avions constaté que l'ambiance de notre jeu vidéo laissait à désirer. Les niveaux semblaient vides et dénués de vie, ce qui nous a poussés à travailler sur l'intégration d'éléments de décor pour donner plus de caractère à notre jeu. Mes collègues et moi-même avons étudié chaque niveau afin de comprendre son thème et son ambiance, puis j'ai créé des objets de décoration adaptés à l'aide de "ProCreate". Ainsi, le niveau "Métro" a été enrichi de poubelles, de wagons de train cassés et en marche, de panneaux d'affichage et autres éléments qui ont rendu le niveau plus intéressant visuellement et immersif pour les joueurs.

Depuis lors, nous avons continué à améliorer l'ambiance de notre jeu en ajoutant progressivement de nouveaux niveaux et fonctionnalités. Par exemple, nous prévoyons d'introduire un niveau aquatique, dont le sprite du joueur a été réalisé par mes soins. De plus, nous envisageons d'ajouter un levier d'aiguillage pour rendre le jeu plus compétitif, surtout en mode multijoueur comme Mark l'a précisé précédemment.

Ainsi, notre objectif est de continuer à travailler sur l'ambiance de notre jeu, en ajoutant de nouveaux niveaux et éléments pour créer une expérience de jeu mémorable et captivante pour nos utilisateurs.



## 2.4.2 Les Mini Jeux

Lors de la première soutenance, lorsque nous avions commencé à travailler sur Go On Another Town, nous étions convaincus que le jeu principal offrait une expérience suffisamment engageante pour maintenir l'intérêt des joueurs. Toutefois, avec le temps, il est devenu évident que notre jeu avait besoin d'un élément supplémentaire pour le rendre plus attrayant.

En tant que joueur moi-même, j'avais constaté que les transitions d'une pièce à l'autre manquaient de défi et de complexité lors de cette première phase du projet. Cela n'offrait pas de véritable challenge aux joueurs, et pouvait même diminuer l'esprit de compétition chez certains d'entre eux. C'est ainsi que, pour la première soutenance, l'idée d'introduire des mini-jeux lors de chaque passage entre les pièces est née. Le but était de proposer aux joueurs une expérience de jeu variée et amusante, qui rompt la monotonie du jeu principal tout en conservant leur attention.

Nous avons décidé de migrer le jeu de Windows Forms vers Raylib pour profiter de la simplicité, de l'optimisation et de la compatibilité multiplateforme. J'ai donc converti les deux mini-jeux existants (OSU et le jeu de puzzle) en utilisant Raylib. Ce choix s'est avéré judicieux, car leur développement a été plus rapide et plus simple qu'avec Windows Forms. J'ai également pu créer deux nouveaux mini-jeux : Speed-Click et le décodeur.

L'utilisation de Raylib a également résolu mon problème de développement sur Mac, puisque je n'ai plus eu besoin d'utiliser une machine virtuelle peu performante, ce qui ralentissait mes travaux.

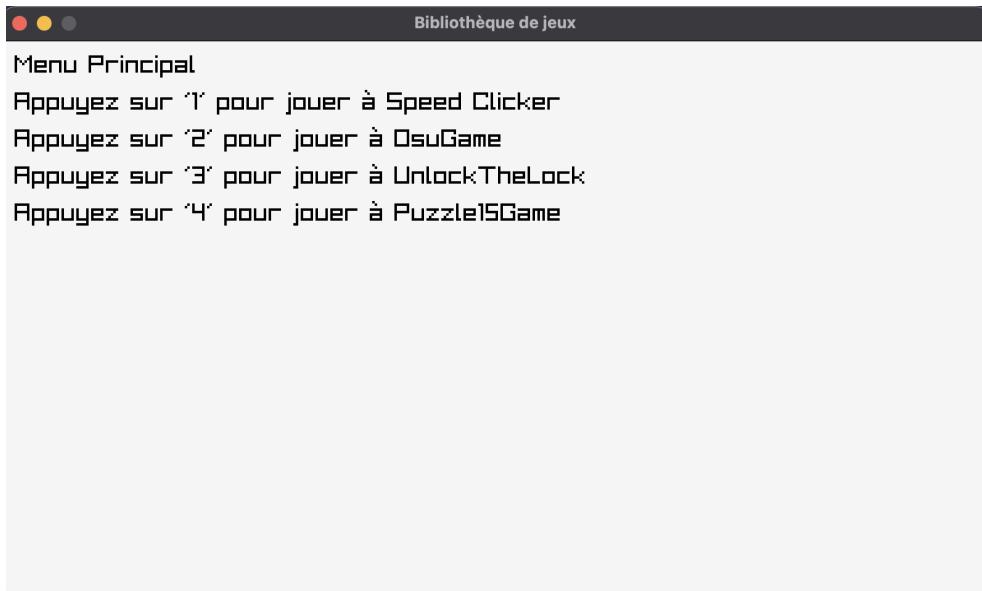
Pour rendre le jeu encore plus amusant et stimulant, nous prévoyons d'intégrer les mini-jeux dans le mode multijoueur. Cela permettra aux joueurs de s'affronter intellectuellement, car ils devront anticiper leurs coups et résoudre des énigmes pour progresser dans les niveaux. En somme, l'ajout de mini-jeux et la migration vers Raylib ont contribué à améliorer notre projet et à offrir une expérience de jeu plus riche et plus divertissante pour nos utilisateurs.

### L'application

Lors de la migration vers Raylib, nous avons dû repenser notre approche pour intégrer les mini-jeux de manière optimale dans les niveaux solo et en duo de Go On Another Town. Cette transition nous a amenés à explorer de nouvelles techniques pour gérer les changements de scène et adapter la structure des classes afin de simplifier l'utilisation des mini-jeux.

Dans ce contexte, j'ai développé une application qui fait office de bibliothèque de mini-jeux, comprenant un menu principal et une option permettant de sélectionner le mini-jeu à tester. Cela nous a permis d'expérimenter différentes manières de gérer les transitions entre les scènes et d'identifier la structure de classe la plus adaptée pour offrir une expérience de jeu fluide et facile d'accès.

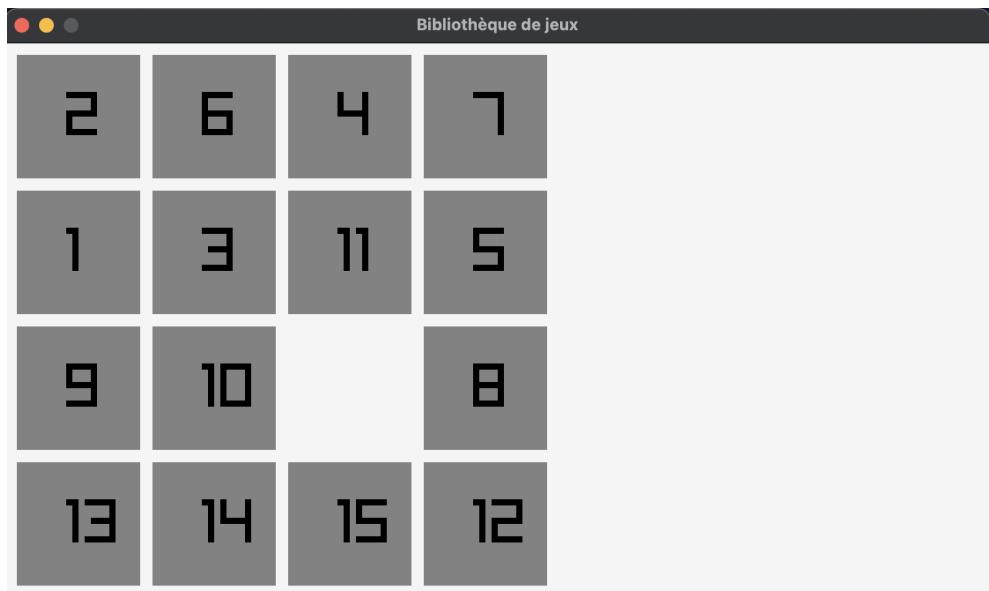
Pour l'instant, cette application n'est pas directement intégrée au jeu principal, mais nous travaillons activement pour finaliser cette intégration dans les meilleurs délais. Notre objectif est de proposer une expérience de jeu complète et cohérente, enrichie par ces mini-jeux qui apporteront une touche de diversité et de challenge supplémentaire.



Les quatre mini-jeux que j'ai développés se divisent en deux catégories distinctes. La première catégorie regroupe les jeux de logique et de réflexion, qui ajoutent un aspect énigmatique qui manquait tant à notre jeu. Je pense que ces jeux enrichissent l'expérience des joueurs en leur offrant de nouveaux défis à relever et en les stimulant intellectuellement.

**Puzzle15Game**

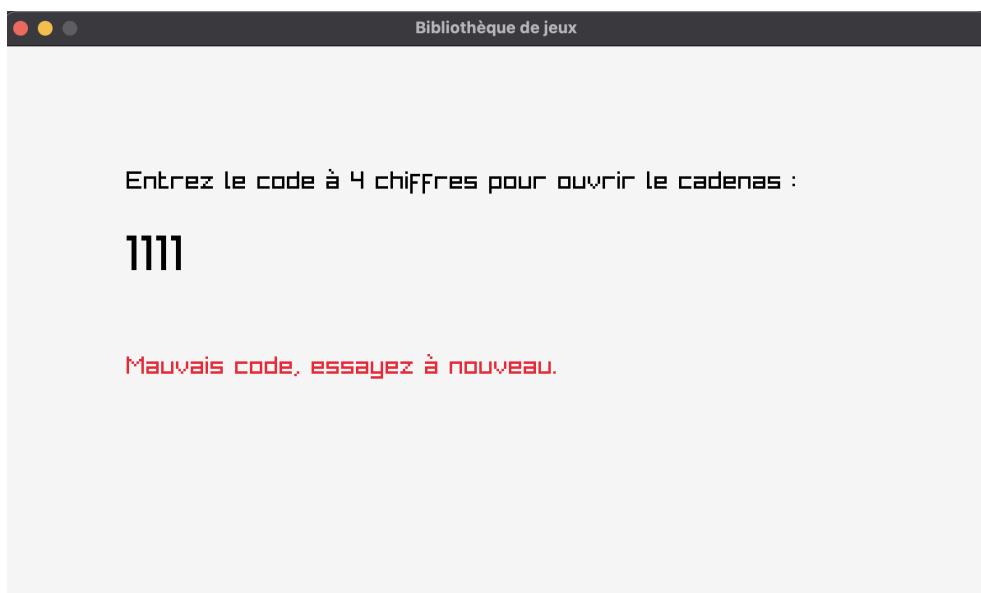
Le premier jeu, Puzzle15Game, est un grand classique dans le monde des énigmes. Il s'agit d'un jeu de taquin où les joueurs doivent réorganiser les pièces pour reconstituer une image ou un motif. Ce jeu sera placé dans des espaces cachés pour limiter l'accès au joueur, ce qui ajoutera une dimension de découverte et d'exploration au jeu. À mon avis, cela rendra le passage d'un niveau à l'autre plus engageant et intéressant pour les joueurs.



### Décodeur

Le deuxième jeu est un décodeur, qui fonctionne un peu comme un cadenas. Ce jeu nécessite de trouver une combinaison de quatre chiffres pour déverrouiller l'accès à la suite du niveau. Si le joueur éprouve des difficultés à trouver le bon code, divers indices seront disponibles pour l'aider à progresser. Ces indices prendront la forme de questions ou d'informations qui guideront le joueur vers la bonne combinaison. Le joueur devra donc faire preuve de réflexion et d'analyse pour interpréter correctement les indices qui lui sont donnés.

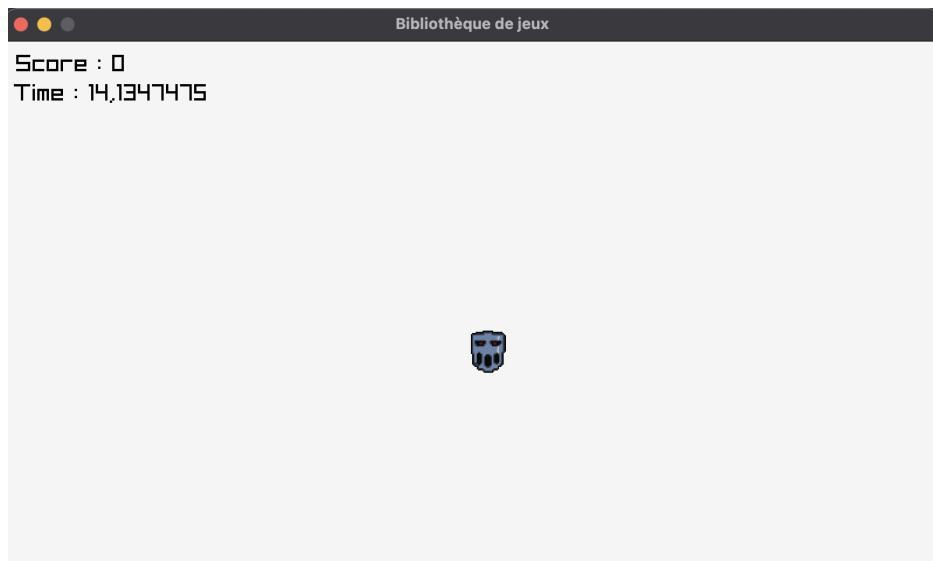
Le défi de ce jeu réside dans le fait que le joueur n'a que trois tentatives pour trouver la bonne combinaison. Si les trois tentatives sont épuisées, le jeu s'arrête et le code est réinitialisé, obligeant le joueur à recommencer depuis le début. Ce système ajoute une dimension de tension et d'urgence au jeu, incitant le joueur à réfléchir soigneusement avant de tenter une combinaison.



### OsuGame

L'autre moitié des jeux est dédiée aux jeux de réaction et de rapidité, offrant ainsi une expérience plus intense et compétitive par rapport aux jeux de logique et de réflexion (à mon avis, cela ajoute une excellente variété au gameplay et permet de garder les joueurs engagés). Le premier jeu de cette catégorie est un jeu de tir sur cibles, où des cibles se déplacent aléatoirement et le joueur doit en toucher 15 en 15 secondes. Cela signifie que le joueur dispose d'une seconde pour toucher chaque cible, ce qui crée un défi de rapidité et de précision.

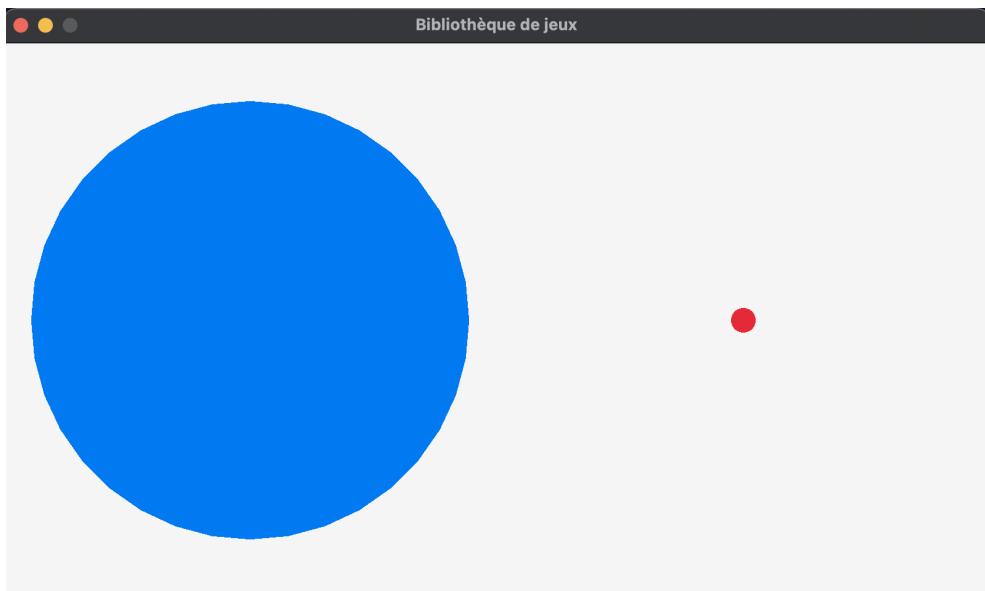
Ce jeu met l'accent sur les réflexes et l'agilité du joueur, ajoutant une dimension d'action et de compétition au gameplay. Si le joueur parvient à toucher les 15 cibles dans le temps imparti, il remporte la partie et peut poursuivre son aventure. Dans le cas contraire, il devra recommencer l'épreuve jusqu'à la réussir. Personnellement, je pense que cette épreuve ajoute une excellente variété et un défi bienvenu qui permet de casser la monotonie du jeu principal.



### SpeedClicker

Enfin, le dernier jeu à ce jour est un "Speed Clicker", un jeu qui sera présent en mode multijoueur afin de créer une compétition directe entre les joueurs. En effet, le joueur qui clique le plus rapidement et parvient à toucher le rond de l'autre joueur remportera la partie, ce qui lui conférera des avantages spécifiques par la suite.

À mon avis, ce mini-jeu est une excellente addition au mode multijoueur, car il ajoute une dimension compétitive qui peut renforcer l'engagement et l'enthousiasme des joueurs. Il permet de tester les réflexes et la rapidité des participants, tout en offrant une expérience de jeu amusante et stimulante. Le fait de gagner des avantages en réussissant ce mini-jeu rend également la compétition encore plus intéressante et gratifiante pour les joueurs.



### 2.4.3 Site-Web

Lors de la première soutenance, j'ai présenté mon choix d'utiliser le framework Django pour le développement du site web de notre jeu. Ce choix a été motivé par les nombreux avantages de Django, notamment sa simplicité, son organisation et sa modularité, ainsi que par la grande communauté active qui l'entoure. Grâce à Django, j'ai pu mettre en place rapidement une structure de projet claire et bien définie, ce qui facilite la maintenance et l'évolution du site web à long terme. J'ai également partagé les défis initiaux que j'ai rencontrés en travaillant avec Django, tels que la compréhension de sa logique de modèles et son organisation stricte. Néanmoins, surmonter ces défis m'a aidé à développer mes compétences en matière de développement web, de résolution de problèmes et de gestion de projet.

Dans ce rapport de deuxième soutenance, je vais décrire en détail les fonctionnalités spécifiques que j'ai implémentées ou prévois d'implémenter dans le site web, et comment elles se rapportent aux besoins de notre équipe et de notre jeu. Cela permettra à l'équipe de mieux comprendre la valeur ajoutée du site web et les avantages qu'il offre en termes de promotion du jeu, d'engagement des utilisateurs et de support technique. Les fonctionnalités développées visent à créer une expérience utilisateur complète et engageante, favorisant ainsi une communauté solide autour de notre jeu et soutenant notre équipe dans son évolution.

En mettant l'accent sur ces fonctionnalités spécifiques et en expliquant comment elles répondent aux besoins de notre entreprise et de notre jeu, je souhaite donner une vision claire de la direction que prend notre projet. Cette approche permettra non seulement de garder l'équipe informée des progrès réalisés, mais aussi de mettre en évidence les domaines dans lesquels nous avons été particulièrement efficaces et ceux qui nécessitent une attention supplémentaire.



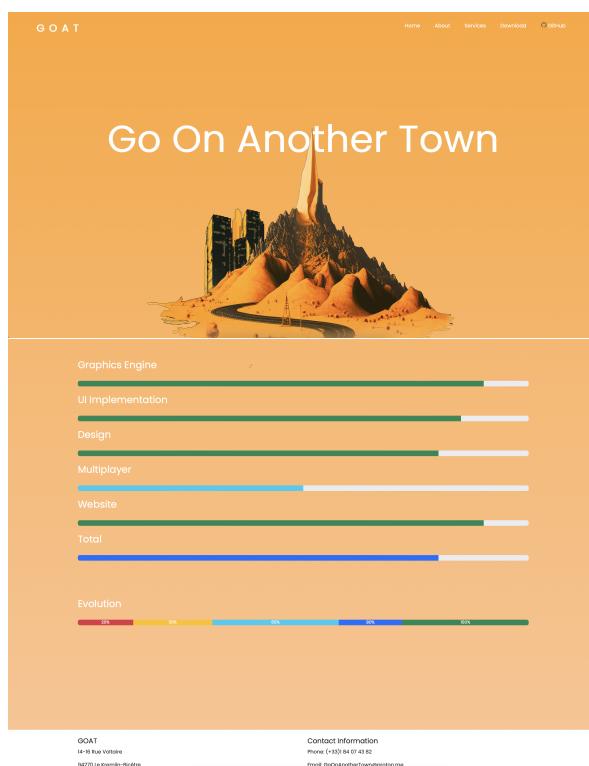
Avant d'aborder les différentes fonctionnalités du site web, je tiens à souligner que j'ai réalisé le design du site web par mes propres soins, avec pour objectif principal de mettre en valeur notre jeu. Le design est très minimaliste, mais performant, et attire l'œil grâce aux formes géométriques et animations présentes sur le site. Les couleurs du site ont été soigneusement choisies pour être en totale corrélation avec le jeu, afin de créer une expérience immersive pour les visiteurs et de renforcer le lien entre le site et le jeu. Ce design épuré et cohérent contribue non seulement à la promotion du jeu, mais facilite également la navigation pour les utilisateurs, offrant ainsi une expérience en ligne agréable et fonctionnelle.

### 1) Page d'accueil

J'ai développé une page d'accueil attrayante et informative qui présente notre jeu de manière visuellement engageante. Étant donné que le jeu n'est pas encore terminé à 100 %, la page d'accueil met en avant le nom de notre jeu ainsi que l'image/logo créée par une IA. Cela donne une bonne impression du jeu et souligne la beauté du monde que nous sommes en train de créer.

Un peu plus bas sur la page d'accueil, j'ai intégré des barres d'évolution pour les tâches principales de notre projet. Ces barres permettent aux visiteurs de suivre l'avancement de chacune des tâches et de mieux comprendre l'état actuel du développement du jeu.

Enfin, tout en bas de la page d'accueil, j'ai inclus des informations concernant notre équipe. Cette section est présente sur toutes les pages du site web, ce qui permet aux visiteurs de se familiariser avec les personnes impliquées dans la création du jeu et de renforcer la confiance en notre projet.



## 2) Pages de présentation du jeu et de l'équipe

La présentation du jeu est divisée en deux parties. La première partie est consacrée à l'explication de l'histoire du jeu, aux exemples de gameplay et aux divers éléments qui le composent. Cette page est conçue avec des transitions fluides entre chaque partie et des animations captivantes pour attirer l'œil des visiteurs et les inciter à explorer davantage le jeu. Les animations et transitions servent également à illustrer les points clés de l'histoire et du gameplay, permettant aux utilisateurs de se familiariser rapidement avec l'univers du jeu et ses mécaniques.

La deuxième partie est une page de présentation de l'équipe de développement, où nous mettons en avant les membres de notre équipe et leur contribution au projet. Les profils LinkedIn des membres sont intégrés pour donner un aperçu de leur parcours et compétences. De plus, une barre de progression animée indique l'état d'avancement des tâches présentées sur la page d'accueil, ainsi que la date à laquelle elles ont été réalisées. Cette approche offre une transparence sur le travail accompli et permet aux visiteurs de suivre le développement du jeu en temps réel.

En guise de pré-conclusion pour cette partie, il convient de mentionner que notre site web comprend également des pages dédiées au téléchargement du jeu et à la consultation de tous les rapports relatifs au projet. Enfin, une page "Sources" est disponible, où nous présentons tous les éléments et ressources utilisés lors du développement du jeu. Cela permet aux visiteurs d'avoir une compréhension complète de notre processus de création et de renforcer la confiance en notre travail.





### 3) Futur proche

Dans un futur très proche, nous prévoyons d'intégrer une version "lite" du jeu directement sur notre site web, permettant aux joueurs de tester et d'expérimenter le gameplay avant même de télécharger le jeu complet. Cette démo offrira un aperçu des mécaniques de jeu, de l'histoire et du style artistique, ce qui encouragera les visiteurs à s'immerger dans notre univers et à envisager de télécharger la version complète du jeu.

À mon avis, cette approche est particulièrement avantageuse pour attirer un public plus large et convaincre les joueurs potentiels de l'intérêt de notre jeu. En offrant une expérience de jeu accessible directement depuis le site web, nous facilitons l'engagement des visiteurs et créons une opportunité pour les joueurs de se familiariser avec notre jeu sans avoir à s'engager immédiatement à télécharger la version complète. Cela pourrait également générer du bouche-à-oreille et des partages sur les réseaux sociaux, contribuant ainsi à la promotion et à la visibilité de notre jeu auprès d'un public plus vaste.

### 3 Diverses retards

#### 3.1 Quels retards ?

Il se trouve que maintenant, nous sommes dans les temps. Nous avons bien un GUI, un moteur graphique et physique fonctionnel, des niveaux et de quoi les implémenter, de même pour les entités et les items. L'ambiance sonore est créée, il nous manque juste des sons et musiques, le site est déjà bien avancé. Le seul bémol pour l'instant sera l'absence de réseaux, mais le multijoueur fonctionnant déjà sur un ordinateur, cela ne devrait pas trop poser de problème.

#### 3.2 Tableau d'avancement

Ayant atteint nos objectifs pour la soutenance 2, voici le planning qui illustre notre progression et les étapes à venir pour continuer sur cette lancée.

Tâche	Première Soutenance	Deuxième Soutenance	Troisième Soutenance
Création du moteur	80%	90%	100%
Programmation du moteur graphique	80%	90%	100%
Programmation du moteur physique	80%	90%	100%
Création de l'UI	80%	90%	100%
Implémentation des entités	50%	90%	100%
Implémentation des items	50%	90%	100%
Définition de la structure des niveaux	50%	90%	100%
Design des niveaux solo	75%	90%	100%
Design des niveaux en duo	75%	90%	100%
Design du personnage jouable	80%	90%	100%
Design des entités	80%	90%	100%
Design des items	80%	90%	100%
Design du GUI	30%	90%	100%
Ambiance sonore	10%	30%	100%
Création du site internet	20%	90%	100%
Implémentation du multijoueur	10%	50%	100%

## 4 Conclusion

Ainsi, ce projet avance non seulement selon le calendrier prévu, mais il dépasse même nos attentes en termes de progrès. Par exemple, le développement du site Web est déjà bien en avance sur notre planning. En outre, le développement du mode multijoueur est également en très bonne voie, avec une implémentation presque complète. Quelques bugs mineurs restent à résoudre et des améliorations sont encore à apporter, mais dans l'ensemble, nous sommes très satisfaits de la progression actuelle.

Actuellement, notre priorité est de continuer à améliorer le jeu en élargissant notre catalogue d'objets et d'entités. Cela permettra d'enrichir l'expérience de jeu et de la rendre encore plus intéressante pour les joueurs.

En somme, en maintenant notre détermination et notre esprit d'équipe, nous sommes convaincus notre jeu donnera des bons souvenirs pour notre communauté.

