

EPITA STRASBOURG



RAPPORT DE SOUTENANCE N°1

PROJET S2

---

# God of Stirmgaard : Hell Arises

---



C#RACALS

*Élèves:*

Gaspard SOUVAY

Ethan LEFEVRE

Ruben MAURIZIO

Alexandre PELLEREAU

8 mars 2023



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Présentation du projet . . . . .	2
1.2	Objectifs pour la première soutenance . . . . .	3
1.3	Avancée du travail de groupe . . . . .	3
1.4	Objectifs pour la prochaine soutenance . . . . .	3
<b>2</b>	<b>Gaspard</b>	<b>4</b>
2.1	Début du projet . . . . .	4
2.2	Conception du personnage . . . . .	4
2.3	Animation du personnage . . . . .	5
2.4	Objectifs pour la prochaine soutenance . . . . .	7
<b>3</b>	<b>Ethan</b>	<b>8</b>
3.1	Début du projet . . . . .	8
3.2	Netcode . . . . .	8
3.3	Assets . . . . .	8
3.4	Intégration des VFX . . . . .	9
3.5	Git . . . . .	10
3.6	Objectifs pour la prochaine soutenance . . . . .	10
<b>4</b>	<b>Ruben</b>	<b>11</b>
4.1	Début du projet . . . . .	11
4.2	Implémentations . . . . .	11
4.2.1	Les déplacements latéraux . . . . .	11
4.2.2	Le saut . . . . .	11
4.2.3	Les dash . . . . .	11
4.2.4	La notion d'énergie . . . . .	12
4.2.5	L'attaque . . . . .	12
4.2.6	Les animations . . . . .	12
4.2.7	Le menu principal . . . . .	13
4.2.8	Le menu pause . . . . .	13
4.3	Objectifs pour la prochaine soutenance . . . . .	13
<b>5</b>	<b>Alexandre</b>	<b>14</b>
5.1	Début du projet . . . . .	14
5.2	Génération procédurale . . . . .	14
5.3	Site Web . . . . .	15
5.4	AI et pathfinding . . . . .	16
5.5	Objectifs pour la prochaine soutenance . . . . .	16
<b>6</b>	<b>Conclusion</b>	<b>17</b>
6.1	Cohésion du groupe . . . . .	17
6.2	Avancée du projet . . . . .	17

# 1 Introduction

## 1.1 Présentation du projet

Rappel sur la nature du projet, extrait de notre cahier des charges :

Le personnage n'a initialement pas de barre de vie ou de coeurs, comme habituellement implémenté dans les jeux de ce type, mais il possède un nombre de vies limité. Si le joueur se fait toucher par un ennemi, ce dernier perd une vie et doit reprendre la partie depuis le dernier *checkpoint*, ou point de sauvegarde. Si le joueur arrive à 0 vies, alors la partie se termine. Pour tout de même encourager la prise de risque et la rapidité, le temps de completion de chaque salle se verra chronométré, et le joueur sera récompensé (en or, et pour les temps exceptionnels en *power-ups* occasionels) proportionnellement au temps qu'il a mis à franchir la salle. Le personnage commence sa partie avec une épée et une capacité de mouvement limitée par une barre, appelée le *dash*. Utiliser le dash réduira la barre, mais frapper un ennemi la restaurera partiellement. L'utilisation astucieuse de cette capacité extrêmement polyvalente sera au coeur du gameplay. Le joueur pourra au fur et à mesure du jeu trouver des *power-ups*, et des objets modifiant son comportement. Le jeu se divisera en niveaux, chacuns divisés en salles. Chaque niveau présentera un *checkpoint* au début, permettant au joueur de reprendre son avancée en cas de mort. Chaque salle contiendra plusieurs éléments, tels que des ennemis, des personnages non joueurs, ou d'autres mécaniques permettant de faire progresser le joueur dans l'aventure, par exemple des items d'amélioration.

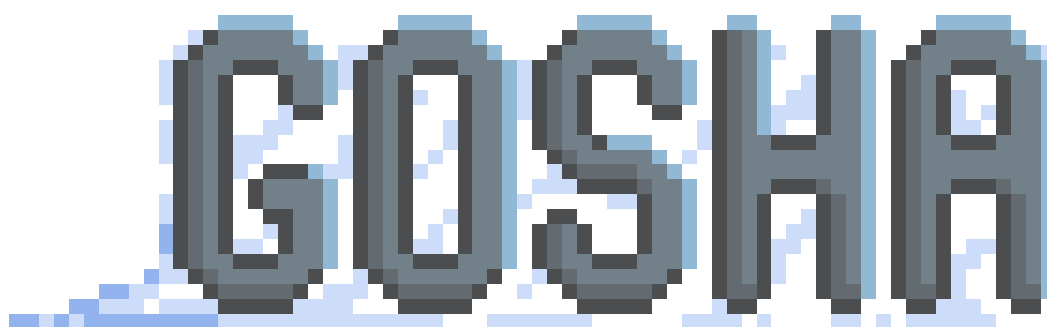


FIGURE 1 – Le logo du jeu

## 1.2 Objectifs pour la premiere soutenance

L'objectif le plus important pour cette soutenance était la conception d'un prototype entièrement fonctionnel pour les mouvements du personnage joueur. Ce prototype devait être complet avec des assets graphiques pour le personnage joueur et quelques effets visuels. Les actions suivantes devaient être implémentées :

- Les déplacements horizontaux
- Le saut
- L'attaque, au sol et en l'air
- Le dash vers la gauche, la droite, le haut, et les diagonales, qui dépend d'une barre d'énergie
- Le saut contre un mur, ou walljump

## 1.3 Avancée du travail de groupe

Malgré quelques lenteurs au début du projet, le groupe travaille ensemble et se fréquente souvent. Il n'y a pas eu de dispute ou de conflit pour le moment. Même si la répartition des tâches actuelles fait que certains travaillent plus que d'autres, l'avancement du projet est satisfaisant pour tous et la productivité du groupe ne fait qu'augmenter avec le temps.

## 1.4 Objectifs pour la prochaine soutenance

Nous avons pour but lors d'ici la prochaine soutenance de commencer à intégrer notre personnage dans des environnements. Ces environnements seront des pièces créées à la main, puis assemblées de manière procédurale pour créer des niveaux qui changent à chaque partie. La création d'obstacles et d'ennemis sera aussi nécessaire. Le multijoueur devra fonctionner pleinement, permettant aux joueurs de jouer en parallèle et de percevoir l'autre joueur et son avancée au second plan. Comme dans tout jeu de type Rogue-Like, le joueur pourra acquérir des capacités supplémentaires, il faudra les concevoir et les implémenter. Voici une liste de capacités qui seront implémentées :

- Une attaque chargée, qui aura plus de portée et infligera plus de dégâts aux boss
- Un double saut
- Un projectile lancé par l'attaque de base
- Une attaque qui élèvera le joueur vers le haut
- Une attaque aérienne qui couvrira tous les alentours du joueur
- Une réduction du coût en énergie du dash

## 2 Gaspard

### 2.1 Début du projet

Depuis le début du projet, j'ai supervisé l'avancée du projet et conçu des assets graphiques pour le projet, en plus d'avoir assisté à la résolution de problématiques de programmation sur le logiciel Unity, notamment l'implémentation des animations. Il fut au début compliqué de motiver le groupe (moi compris) à s'atteler au projet, mais tous les membres travaillent bien plus depuis la fin des examens et sont bien plus présents dans les espaces que nous avons créés pour le projet (notamment un serveur discord). L'ambiance au sein du groupe a toujours été calme et détendue, sans aucun conflit entre les membres du groupe, et j'ai hâte de voir où nous emmènera ce projet aux cours des mois prochains.

### 2.2 Conception du personnage

La première étape de ma tâche était de concevoir le personnage. Il a fallu prendre en compte les graphismes en pixel-art du projet pour établir un design assez sobre pour notre personnage. L'idée de départ était celle d'un ninja cybernétique se battant avec une épée laser, justifiant l'aspect rapide et nerveux du jeu. Le premier croquis établi du personnage fut celui-ci :

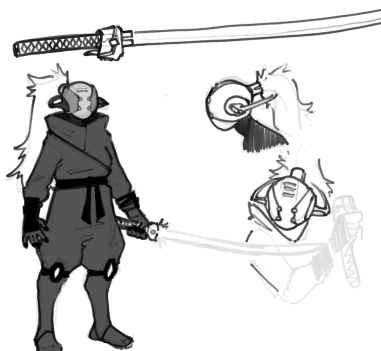


FIGURE 2 – Premier croquis du personnage joueur

M'étant vite rendu compte que bien que je sache dessiner et animer, je ne possédais pas les compétences pour animer des sprites à haute définition, j'ai décidé d'épurer encore ce croquis pour en faire un personnage représenté par un sprite de 32 pixels de hauteur et largeur, pour qu'il puisse être animé dans les temps.



FIGURE 3 – Rendu final du personnage

J'ai donné à la 'chevelure' la même couleur et animation que l'épée du personnage, car ces deux objets seront soumis à des mouvements brusques et afin d'uniformiser les couleurs et animations des effets visuels.

## 2.3 Animation du personnage

Toutes les animations ont été réalisées avec le logiciel Aseprite.

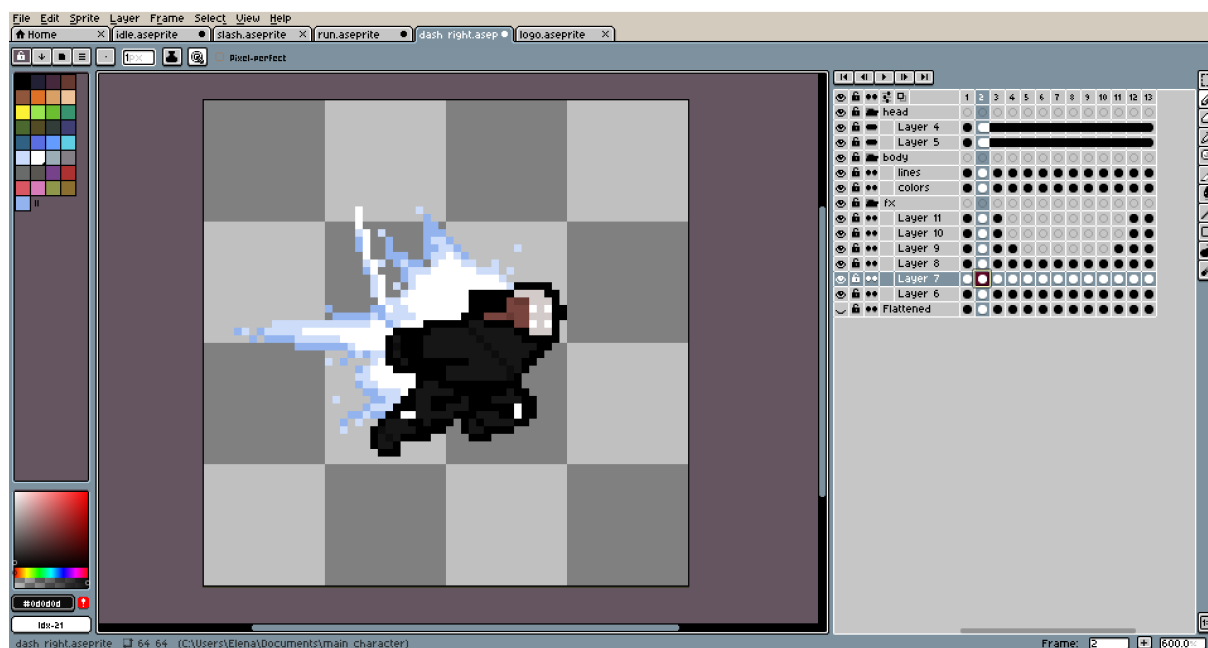


FIGURE 4 – Capture d'écran du logiciel Aseprite

La première animation réalisée fut l'idle, celle qui joue quand le personnage se tient sur place. Elle est basique et peu mouvementée, mais les 'cheveux' du personnage sont toujours animés pour garder une notion de dynamisme.



FIGURE 5 – Animation d'idle

Ne me sentant pas prêt à la tâche d'une animation de course j'ai décidé d'animer l'attaque basique du personnage, mais le résultat était mou. Cela était dû à l'anticipation excessive du mouvement, à la pose de fin molle, et à l'arc trop linéaire décrit par l'épée.



FIGURE 6 – Première animation d'attaque

J'ai donc décidé d'abandonner cette animation d'attaque et d'en faire une nouvelle, en partant sur des bases différentes, notamment en oubliant l'idée de donner une anticipation prononcée à une action qui devrait être rapide et satisfaisante à exécuter. Le résultat est plus rapide, décrit une trajectoire plus prononcée, et se finit par une pose plus forte, qui tient l'épée plus bas. Les jambes du personnage ont été modifiées pour créer l'animation d'attaque aérienne, de manière à ce qu'elle puisse transitionner vers l'animation au sol si le joueur atterrit pendant celle-ci.



FIGURE 7 – Animations d'attaque actuelles

Le saut a été la prochaine action à être animée. J'ai décomposé celui-ci en trois animations :

- L'animation d'élévation, qui se joue quand le joueur appuie sur le bouton de saut jusqu'à ce qu'il atteigne l'apogée de son saut.
- L'animation d'apogée, qui se joue quand celle-ci est atteinte.
- L'animation de chute, qui se joue quand le joueur retombe vers le sol, ou bien lorsqu'il se trouve en l'air par d'autres moyens qu'un saut, par exemple en courant par dessus le bord d'une plateforme.



FIGURE 8 – Animations de saut

Les animations de dash utilisent la 'chevelure', qui s'étend pour prendre la forme d'un propulseur le temps d'un instant.



FIGURE 9 – Animations de dash





FIGURE 10 – Les poses clé de l’animation de course

L’animation de course a été construite à partir de trois ‘keyframes’, des poses clé, qui représentent les poses les plus importantes du mouvement. Des poses intermédiaires légèrement différentes des poses clé ont été ajoutées pour fluidifier le mouvement.



FIGURE 11 – L’animation de course finale

## 2.4 Objectifs pour la prochaine soutenance

Mon objectif principal pour la prochaine soutenance sera la conception de salles et la création d’ennemis. Les salles posséderont 2 à 4 portes, et seront assemblées entre elles de manière procédurale pour constituer des niveaux. J’aurai pour but de créer et entièrement animer au moins un type d’ennemi, qui pourra se déplacer et attaquer le joueur. Je devrai aussi animer les nouvelles capacités du personnage joueur qui seront ajoutées.

## 3 Ethan

### 3.1 Début du projet

A dater de cette première soutenance, j'ai eu l'occasion de prendre en main les outils de Unity, particulièrement les fonctionnements des scripts, de l'animation et des particules grâce à de nombreux tutoriels Unity2D, particulièrement ceux de Antarsoft, mais aussi des forums et autres sources nombreuses. J'ai donc commencé mon travail en confiance.

### 3.2 Netcode

J'ai recherché des méthodes pour faciliter la communication entre les machines et optimiser l'aspect multijoueur du jeu. J'ai considéré les différents protocoles réseau et les différentes méthodes d'implémentation. La première idée était d'utiliser uniquement le protocole TCP, pour assurer la réception des paquets, mais mon inquiétude est que cela risque d'alourdir et ralentir le trafic réseau. Un protocole uniquement UDP peut être envisagé. Cependant, c'est un protocole moins sûr et plus susceptible de perdre des paquets. Il est possible, à l'instar d'anciens jeux de la fin des années 80, d'utiliser un système doublant chaque paquet envoyé, augmentant les chances de réception de chaque paquet. Actuellement, l'idée serait de synchroniser au début, à la fin, et à intervalles réguliers une horloge présente pour les 2 jeux via un paquet TCP, mais de garder les données moins importantes au déroulement du gameplay en UDP tel que les mises-à-jour de la position du personnage sur la carte. Il sera aussi nécessaire selon moi d'assurer un trafic contrôlé et réduit des informations. Je m'appliquerai à assurer une encryption de toutes les données pouvant identifier un utilisateur. Dans un objectif de simplification du processus de fonctionnement réseau, il sera aussi de mon devoir d'essayer au plus possible de réduire le trafic réseau aux moins de ports possible. Cela aidera non seulement la régulation des échanges, mais aussi l'administration d'un réseau dans le cas où ce dernier serait sécurisé par un firewall.

### 3.3 Assets

Il était aussi de mon devoir de trouver les parties de design de second plan. C'est à dire, les assets : Les arrière-plans, le HUD, les sprites d'items, de sols, murs, etc., mais aussi le design sonore, tel que les bruitages et la bande sonore. Cette partie fût assez longue, car la recherche d'assets adaptés au thème fixé n'est pas forcément évidente. Il m'a fallu faire une sélection d'assets large en premier lieu, puis passer par un processus de tri pour déterminer quels assets conviendrait pour la construction du jeu. Mes recherches ont tout d'abord débuté sur le site OpenGameArt, une large librairie d'assets libres de droits et gratuits. J'y ai ici trouvé des arrière-plans, comme joints ci-dessous :

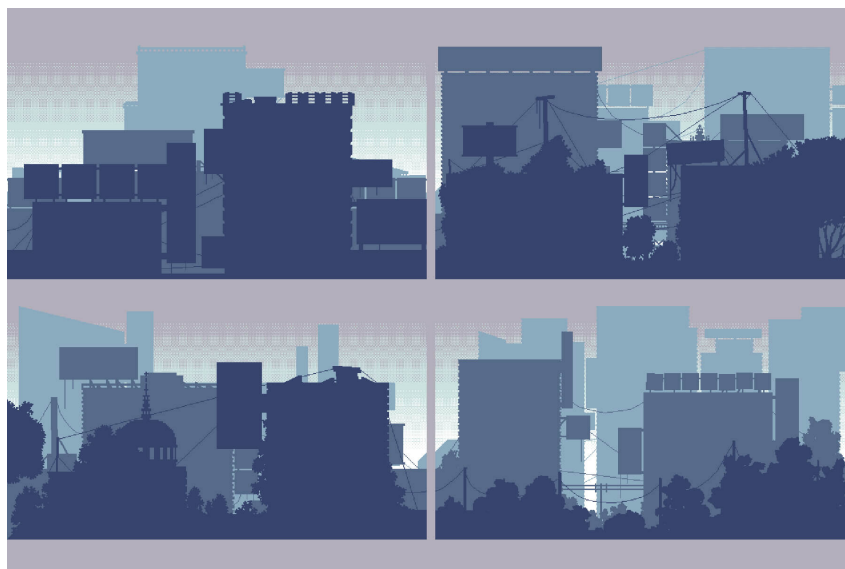


FIGURE 12 – Un exemple de backdrop

La recherche de polices d'écritures, et de détails graphiques fut aussi part de mes tâches. J'ai enfin travaillé sur la recherche des musiques et du son qui jouent dans le menu, et dans le jeu. Savoir chercher les bruitages adaptés et une musique suffisamment forte pour garder l'attention du joueur sont une part importante du design de tout jeu.

### 3.4 Intégration des VFX

Une de mes tâches fut d'apprendre à gérer les effets visuels autour du personnage. Par exemple, l'ajout d'effets de poussière soufflés par le saut ou l'atterrissage d'un personnage. Cette tâche n'a pas été facile, mais après de nombreux essais et l'aide du discord officiel de Unity, j'ai réussi à trouver des méthodes pour faire fonctionner les effets. Je travaillerai ensuite par conséquent sur tous les autres effets visuels qui pourront roucher au joueur, son environnement, et ses ennemis, tel que les effets de particules, tremblement d'écran, changement de lumière, etc. L'atmosphère graphique du jeu est un point très important, et qui se doit de marcher avec le design des personnages et des sprites. Il sera donc de mon devoir de perfectionner l'ambiance du jeu au second plan.



FIGURE 13 – Effets de saut et de poussière à l'atterrissage

### 3.5 Git

L'utilisation du git fut, pour le moins, houleuse lors de ce projet. Tôt dans l'année, nous fûmes face à de nombreuses erreurs pour cloner le projet depuis le repo fourni. Une fois l'accès enfin rétabli, des erreurs de manipulation m'ont causé de push bien trop de fichiers sur le repo, mon .gitignore n'étant pas correctement fait pour gérer les projets Unity. Cela a mené à de nombreux problèmes pour le partage des fichiers, le repo étant plein de fichiers trop volumineux pour qu'on puisse en rajouter. De fil en aiguille, d'autres manipulations ont menées à la création d'un fichier corrompu qui fut push sur le repo au travers d'un problème de connexion, ce qui causa une corruption sur le git qui nous empêchait de l'utiliser correctement. Après de nombreuses tentatives et l'utilisation du gitlab host sur le PIE, nous avons enfin fini par nettoyer nos branches et repartir sur un master propre. L'utilisation du git reste un sujet sensible, et les manipulations faites avec précaution pour ne surtout pas risquer un nouveau désastre et perdre notre travail en le rendant inaccessible.

### 3.6 Objectifs pour la prochaine soutenance

J'ai été malheureusement peu satisfait de ma contribution lors de cette première partie de projet. Je compte mettre les bouchées doubles pour la suite, et travailler tout à tour sur le perfectionnement de l'ambiance graphique, les travaux de lumières, la finalisation du design son, mais il faudra aussi avancer sur le design des items potentiels que le personnage pourra obtenir, ses capacités, et les façons de modifier son comportement. Il faudra aussi se concentrer sur l'implémentation en profondeur du multijoueur, de son fonctionnement, et enfin, il faudra terminer, ou progresser largement sur le HUD d'ici à la prochaine soutenance.

## 4 Ruben

### 4.1 Début du projet

Motivé et travaillant sur le projet depuis plusieurs mois déjà, l'ajout de la plupart des fonctionnalités suivantes s'est fait rapidement et sans trop de soucis ou de bugs particuliers. Sans pour autant avoir utilisé le logiciel Unity auparavant, sa prise en main s'est faite naturellement et malgré certains bugs et comportements non désirés qui reclamaient toute ma concentration et ma patience, je suis donc plutôt confiant quant à l'avancée du projet et de ma capacité à utiliser au mieux les ressources données.

### 4.2 Implémentations

Afin d'avancer correctement dans le projet, il était nécessaire de commencer par les bases et donc d'implémenter en premier lieu le joueur et ses déplacements. Grâce aux nombreux tutoriels présents sur internet, à la documentation ou tout simplement à l'aide de tests effectués de mon côté je suis plutôt satisfait du résultat. Je présenterais donc dans cette partie mon travail implémenté, les problèmes rencontrés et la résolution de ces derniers.

#### 4.2.1 Les déplacements latéraux

Sans trop de soucis particuliers, il suffit de multiplier le `RawInput` (une valeur correspondante à 1 ou -1 si la touche de déplacement vers la droite ou vers la gauche est pressée) par la vitesse du joueur donnée. L'animation de course se joue si la vitesse de déplacement est supérieure à 0.3.

#### 4.2.2 Le saut

L'implémentation du saut n'est théoriquement pas compliquée et cela s'est fait effectivement sans complications. Cependant la physique de base de Unity donnait une impression de flotement dans l'air qui ne me plaisait pas et que j'ai du changer. Une mécanique implémentée que j'apprécie particulièrement est le 'Coyote Time'. Cela consiste à laisser un peu moins d'une seconde au joueur pour sauter lorsque celui-ci tombe d'une plateforme et rend donc des reflexes plutôt lents beaucoup moins punitifs. Notons également que la force du saut change en fonction du temps passé à presser la touche de saut.

#### 4.2.3 Les dash

Une des implémentations des plus intéressantes mais également des plus compliquées était celle du dash. Les dash latéraux et verticaux étaient assez simples mais l'idée d'également implémenter des dash diagonaux était presque nécessaire afin de compléter notre gameplay. Il a fallu beaucoup de temps et patience mais grâce à l'utilisation du vecteur normalisé cela s'est vite implémenté. Pourtant, un bug persistait, à chaque utilisation du dash diagonal et du dash vertical, un comportement non désiré s'activait et ajoutait un autre vecteur vertical qui propulsait une seconde fois le joueur vers le haut. Après beaucoup de temps et de patience j'ai décidé de contrer le problème en ajoutant un vecteur contraire à la propulsion ce qui a réglé le problème.



FIGURE 14 – Dash diagonal et lateral

#### 4.2.4 La notion d'énergie

Si le pouvoir de dash est assez puissant, il en convient de limiter son utilisation. Pour cela, j'ai introduit une barre d'énergie, située en bas à gauche de l'écran du joueur. Chaque utilisation de ce pouvoir utilise un peu de cette énergie et à 0, l'utilisation en devient impossible. Pour le moment, la barre d'énergie se remplit grâce aux déplacements du joueur, c'est à dire que lorsque celui-ci ne bouge pas, la barre ne se remplit pas. Lorsqu'ils seront implémentés, attaquer des ennemis sera la manière la plus viable de remplir la barre d'énergie.

### 4.2.5 L'attaque

Je n'avais jamais fait d'attaques auparavant mais avec certains tutoriels disponibles sur internet cela s'est fait très rapidement. Cependant, il a tout de même fallu trouver un moyen de geler la position de joueur lorsque celui-ci attaque sur le sol afin qu'il ne puisse pas attaquer en courant et donc éviter des interactions non désirées.

### 4.2.6 Les animations

Les animations ont personnellement fait partie des fonctionnalités les plus difficiles à implémenter. En effet il m'a fallu du temps pour comprendre comment fonctionnait l'animateur de Unity et certains blocks de commande. Par ailleurs certaines animations se jouer pendant d'autres et après quelques tests des animations non prévues se sont jouées. Mais avec l'aide de Gaspard, nous avons pu correctement régler les problèmes rencontrés.

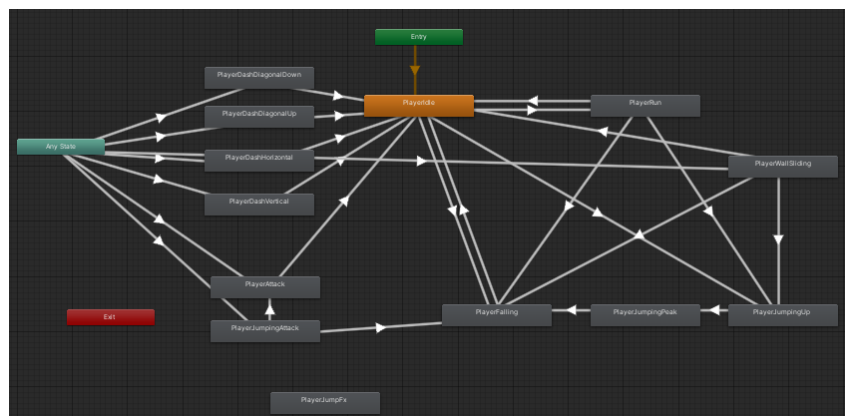


FIGURE 15 – L'animateur de Unity

#### 4.2.7 Le menu principal

Je ne suis pas désigné ni spécialiste du graphic-design du projet mais le menu principal, bien que très simple me plaît beaucoup. Il n'a pas été très difficile de l'implémenter puisque cela est une scène à part entière constituée de textes, d'images et de boutons.

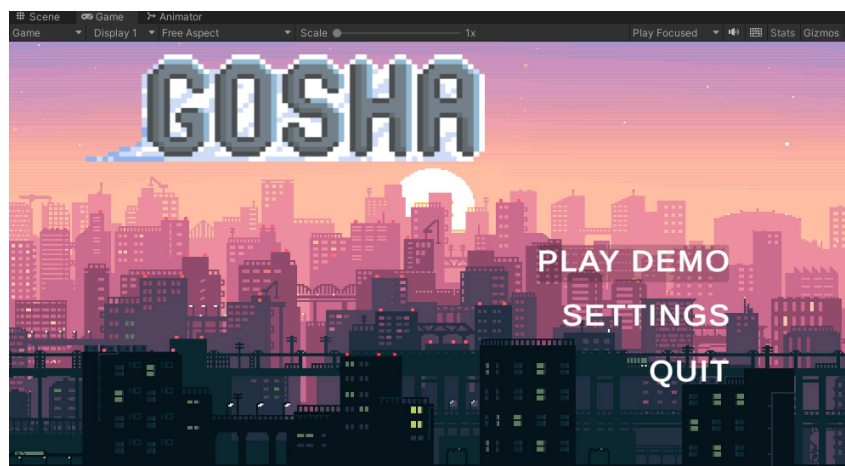


FIGURE 16 – Menu principal

#### 4.2.8 Le menu pause

Le menu pause en jeu était un peu plus compliqué à créer que le menu principal. En effet il a fallu trouver le moyen de cacher l'écran du joueur et fixer la position actuelle de ce dernier tout en ne figeant pas l'entiereté du jeu puisque cela est censé faire partie du multijoueur.

### 4.3 Objectifs pour la prochaine soutenance

Le multijoueur est déjà bien travaillé mais pas encore assez pour être jouable et donc présentable lors de la première soutenance. Par exemple, on peut créer des "Rooms" avec ID et les rejoindre mais seul l'hôte controle tous les joueurs présents. Je dois donc régler cela le plus rapidement possible. Pour la seconde soutenance, j'implémenterai le multijoueur afin qu'il soit jouable. Il sera également nécessaire d'améliorer les menus aussi bien graphiquement que dans leur fonctionnalité. La tâche me tenant à coeur, j'aiderai par ailleurs à la programmation des ennemis.



## 5 Alexandre

### 5.1 Début du projet

J'ai déjà utilisé Unity pour des projets personnels et j'avais aussi appris grâce à des tutoriels Unity3D et Unity2D, en particulier ceux de Brackeys pour la 3D et TUTO UNITY FR pour la 2D. Lorsque j'ai eu besoin d'aide plus spécifique, je me suis renseigné sur des forums (principalement Discord) et sur la documentation officielle de Unity ainsi que de nombreuses autres sources. J'ai donc commencé mon travail avec confiance.

### 5.2 Génération procédurale

Je me suis d'abord penché sur la génération procédurale de terrain, en faisant des recherches sur le bruit de Perlin. Mais je me suis vite rendu compte que le jeu risquait de devenir répétitif et ennuyeux. C'est pourquoi nous avons opté pour un système de salle faites à la main qui seront connectées entre elles de façon procédurale afin de garder une certaine variété dans le jeu et de rendre chaque partie unique. Dans ce cas, on se rapproche plus du genre Rogue-like, j'ai donc cherché comment faire et j'ai trouvé un algorithme de génération procédurale de salle. Cette algorithme crée des salles de tailles différentes puis les relie entre elles avec des couloirs (en violet sur la figure 17).

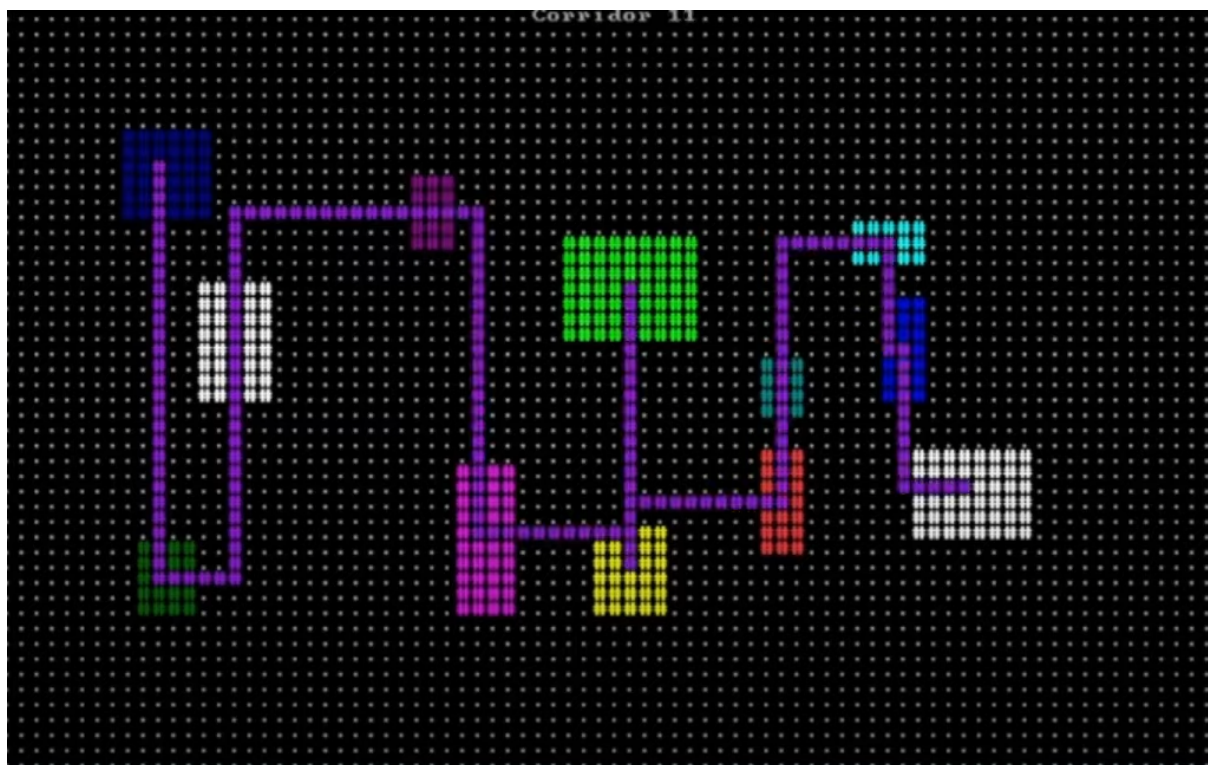


FIGURE 17 – Algorithme de génération procédurale de salle

Mais comme le jeu est un platformer, il ne faut pas oublier de prendre en compte que le joueur ne se déplace pas vers le haut ou le bas de la même façon que sur les côtés. Il n'est pas possible d'implémenter directement l'algorithme de génération procédurale de salle de Rogue-like (figure 11) dans notre jeu. Il va donc falloir adapter cette algorithme



pour qu'il crée des salles plus en longueur qu'en largeur tout en gardant plusieurs choix de direction pour le joueur. Pour le côté implémentation, je me suis déjà un peu renseigné et je pense donc utiliser des prefabs, qui sont des gameobjects sauvegardés dans les assets, pour pouvoir les instancier au bon moment.

### 5.3 Site Web

Je me suis rendu compte que je n'avais rien de concret à faire pour cette première soutenance, j'ai donc décidé de m'occuper du site web du projet. J'ai d'abord commencé à faire des recherches pour apprendre à créer un site web. J'ai donc regardé des tutoriels sur Youtube pour apprendre à utiliser HTML et CSS. Pour avoir une idée de ce que je voulais faire, j'ai regardé des sites web de projets d'anciennes promos, disponibles sur epita.it. J'ai vite remarqué que la plupart des sites web étaient construits de la même façon et j'ai commencé par chercher et ajouter le code d'un header pour le site. Puis j'ai créé la page d'accueil (index.html), la page de téléchargement (download.html), la page d'avancement du jeu (advancement.html) et enfin la page de présentation de l'équipe (team.html). Sur la page principale, je compte mettre un trailer ou tout simplement une vidéo du jeu ainsi qu'une description du jeu. En attendant que la vidéo soit faite, j'ai mis une vidéo de notre chat préféré.

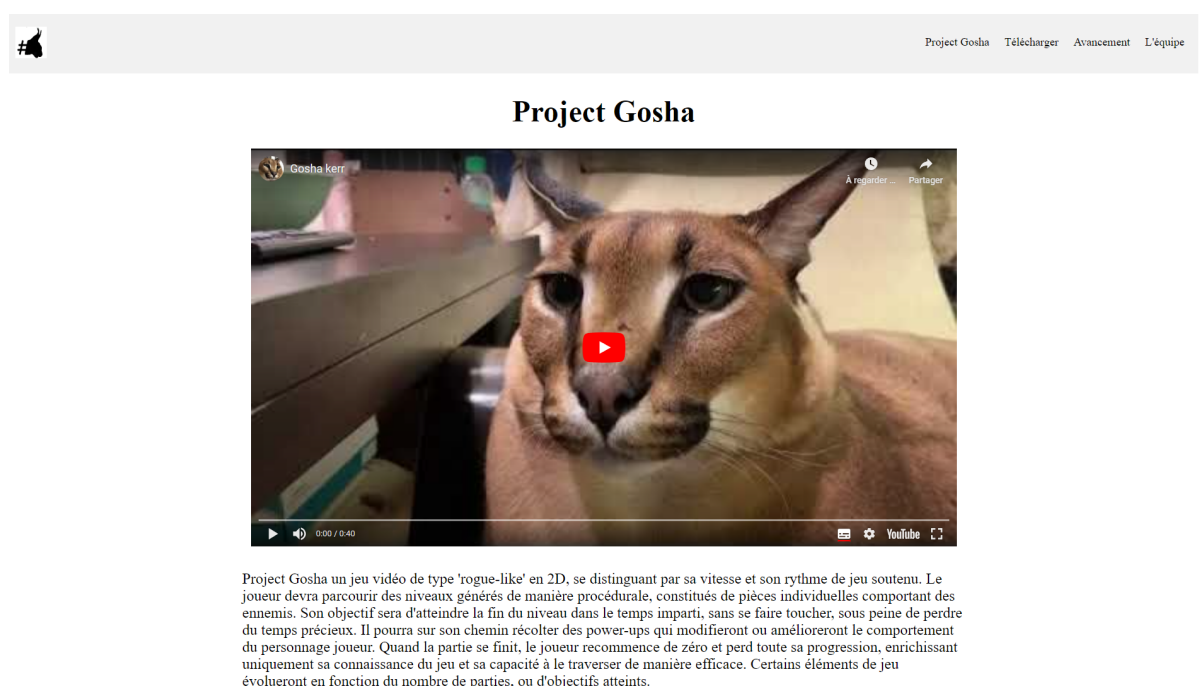


FIGURE 18 – index.html, le menu principal

La page de téléchargement contient un bouton central pour télécharger le jeu mais pourra plus tard contenir différents fichiers en fonction de la compatibilité à différents OS. Par exemple, il pourrait y avoir 3 boutons pour Windows, MacOS et Linux.

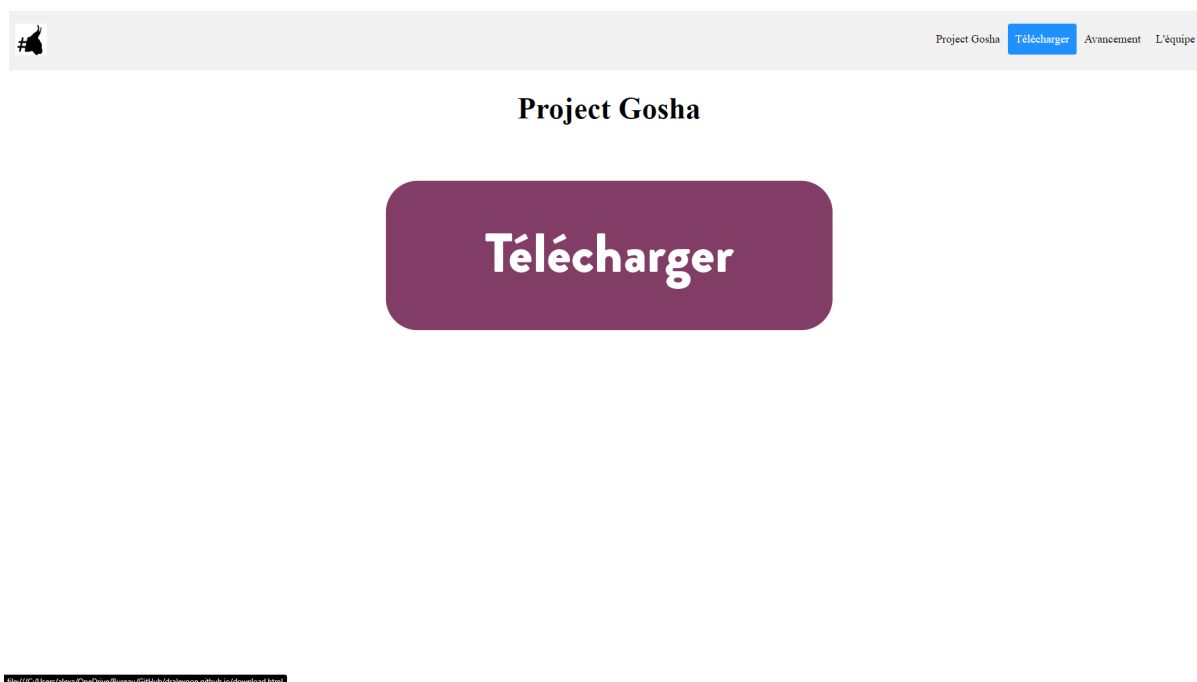


FIGURE 19 – download.html, la page de téléchargement

Pour l'instant, j'ai décidé d'host le site sur GitHub Pages avec mon compte Github personnel, car c'est gratuit et facile à mettre en place, le site est donc disponible sur dralexgon.github.io.

## 5.4 AI et pathfinding

J'ai fait quelques recherches et j'ai trouvé quelques algorithmes de pathfinding qui permettent de trouver le plus court chemin. Mais je vais aussi devoir les adapter car ils sont fait pour de la 2D vue du dessus ou de la 3D. Le plus intéressant serait d'utiliser différents algorithmes en fonction des ennemis. Pour pouvoir commencer à implémenter, je vais devoir attendre que le jeu avance encore pour avoir une meilleure idée de l'environnement et des déplacements.

## 5.5 Objectifs pour la prochaine soutenance

Je compte continuer le site web, notamment la partie avancement du jeu mais aussi celle qui représente l'équipe. Je vais aussi essayer de mettre plus de couleur et de rendre le tout plus esthétique grâce au css. Pour les IA, je compte commencer à implémenter quelques algorithmes de pathfinding comme celui du A\* en rajoutant des ennemis volants puis essayer de créer mes propres algorithmes qui fonctionnent avec le jeu pour faire des ennemis qui marchent. Pour la génération des niveaux, je vais continuer de me renseigner ainsi que commencer à implémenter l'avancée du jeu le permet.

## 6 Conclusion

### 6.1 Cohésion du groupe

L'entente et l'entraide sont essentielles à la réalisation d'un projet en groupe. Fort heureusement pour notre projet et comme dit précédemment, il n'y a eu aucune dispute au sein de notre groupe depuis le début du projet, et tous les membres s'entraident constamment. Personne n'a peur de demander de l'aide, et les problèmes sont résolus en équipe. Le travail commun est fourni, et le futur du projet ne nous inquiète pas.

### 6.2 Avancée du projet

Nous estimons que l'avancée du projet est pour l'instant satisfaisante. Les bases du gameplay sont posées, la conception des niveaux et des ennemis ne devrait pas poser de problèmes. Les défis à venir seront sans aucun doute la génération procédurale, l'IA et le multijoueur.