

# Rapport de la première soutenance



par  
**TAAG TEAM**

composé de :

Thomas Solatges  
Arnaud Corcione  
Axelle Destombes  
Gwennan Jarno

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Avancement du projet</b>	<b>4</b>
2.1	Le Level Design . . . . .	4
2.2	L'IA . . . . .	6
2.2.1	Le Patrouilleur . . . . .	6
2.2.2	Le Tourneur . . . . .	7
2.2.3	Le Follower . . . . .	7
2.2.4	Le Spawner . . . . .	8
2.3	Le Gameplay . . . . .	9
2.3.1	Création de la classe personnage . . . . .	9
2.3.2	Création de la classe ennemi . . . . .	9
2.3.3	Déplacement du joueur . . . . .	10
2.3.4	Attaque du joueur . . . . .	10
2.4	L'Histoire . . . . .	11
2.5	Les Graphismes . . . . .	12
2.5.1	Les personnages . . . . .	12
2.5.2	Les ennemis . . . . .	13
2.5.3	Les items . . . . .	15
2.5.4	Les textures . . . . .	15
2.6	L'Immersion . . . . .	16
2.6.1	Menu Principal . . . . .	16
2.6.2	Menu des options . . . . .	17
2.6.3	Menu Pause . . . . .	17
2.7	Communication . . . . .	18
2.7.1	Le site web : . . . . .	18
<b>3</b>	<b>Ressentis des membres</b>	<b>19</b>
3.1	Arnaud Corcione . . . . .	19
3.2	Axelle Destombes . . . . .	19
3.3	Gwennan Jarno . . . . .	19
3.4	Thomas Solatges . . . . .	19
<b>4</b>	<b>Conclusion</b>	<b>20</b>
<b>5</b>	<b>Bibliographie</b>	<b>21</b>

# 1 Introduction

Depuis deux mois, TAAG Team travaille sur le jeu *Toddler are afraid of ghost*. Nous avons réussi à atteindre les objectifs que nous nous étions fixés durant la rédaction du cahier des charges, même si les dernières semaines ont eu un rythme plus soutenu que les premières.

Pour rappel *Toddler are afraid of ghost* est un roguelike 2D, à un ou deux players, où le joueur affronte des monstres et parcourt des niveaux générés de façon procédurale.

Avant de commencer à développer le jeu, la première étape a été de mettre en place les outils de travail et comprendre leur fonctionnement à savoir *Unity*, *Git-Hub* et *Discord*. De plus, il a fallu s'organiser pour travailler en équipe, une chose nouvelle pour la plupart des membres. Il a aussi été difficile d'apprendre et de travailler efficacement pour certains.

Lorsque nous avons fini de résoudre les divers problèmes nous empêchant de commencer, nous sommes partis sur la création du projet et avons suivi l'avancée prévue dans le cahier des charges. Nous avons donc avancé sur les principaux points suivants :

- Level Design
- Intelligence Artificielle
- Gameplay
- Histoire
- Graphismes
- Immersion
- Communication

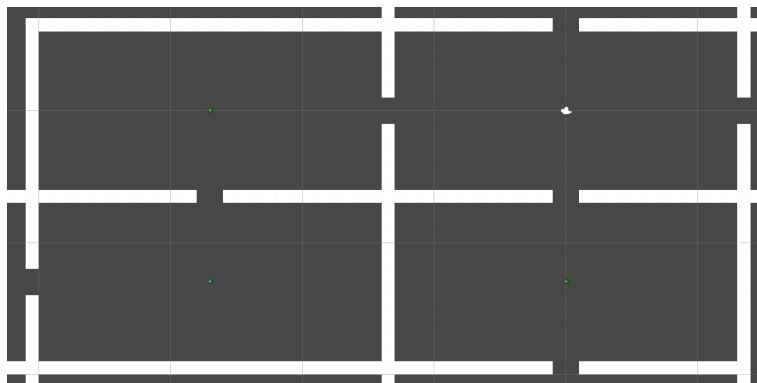
Nous avons réussi à atteindre les objectifs fixés. et nous sommes bien lancés pour la suite. Le projet avance réellement bien.

## 2 Avancement du projet

### 2.1 Le Level Design

A ce jour, le niveau se génère, en remplissant presque toutes les conditions que je me suis fixé pour cette soutenance. J’ai donc ce que nous appelons un étage qui se crée, avec des salles qui partent d’un point de départ et qui placent, au terme de la génération, une salle de boss dans la salle la plus éloignée du point de départ. La caméra, placée au centre de chaque salle, change de position lorsque le joueur change de salle, se repositionnant ainsi dans chaque nouvelle salle. Si un étage est trop petit ou trop grand (paramètres qui changeront ensuite quand les niveaux s’enchaîneront) alors il est régénéré afin de ne pas commencer le jeu sur un niveau interminable et de le finir sur un niveau à cinq salles.

Je suis actuellement face à un problème, qui est celui-ci :



Deux salles sont côte-à-côte, mais elles ne possèdent pas d’ouverture. Cela rend le niveau peu cohérent. De plus, on a quatre salles qui forment un “carré”, ce qui ne doit pas arriver.

Voilà pour les problèmes actuels. Maintenant, en ce qui concerne les problèmes passés, j’ai plusieurs fois échoué la génération procédurale à cause d’un manque de compréhension du système de coordonnées et de prefab d’Unity. Mes premières tentatives de faire des templates de salles n’ont pas été concluantes car je créais une salle directement dans une scène au lieu de faire une prefab, ce qui faisait des salles non centrées avec des coordonnées flottantes imprécises.

De plus, puisque le ratio de la caméra n’aidait pas, j’ai donc abandonné le traditionnel 16 :9 pour un élégant 2 :1. Ce procédé est plus simple pour une continuité dans la largeur des murs à la caméra, et laisse ainsi de la place en haut et en bas de l’écran pour y mettre le futur HUD.

Pour la suite, je souhaite mettre en place des salles spéciales telles que des salles contenant les futurs items. Je compte aussi faire en sorte que le spawn ne soit pas toujours une salle avec quatre ouvertures, mais plutôt une salle en forme de T ou en forme de +.

## 2.2 L'IA

Nous nous sommes mis d'accord et avons conclu que l'une des priorités était d'avoir des ennemis capables de se déplacer seuls, possédant déjà leurs statistiques et leur comportement. En plus de la map, les ennemis constituent une partie importante de notre jeu. Il fallait donc impérativement créer une première version d'ennemis, capables de se déplacer et de réagir en fonction de la position du joueur. Pour se faire, Arnaud, chargé de l'IA, a divisé ce thème en 2 parties : le déplacement et le comportement des ennemis.

Avant de s'occuper du comportement des ennemis, il a d'abord fallu créer un système de déplacement fonctionnel. De même que plusieurs types d'ennemis capables de se déplacer de manière différente avec donc un comportement différent. Pour se faire, Arnaud a réfléchi à comment déplacer un objet de manière autonome. Plusieurs choix possibles se sont offerts à lui, mais le plus simple fut de créer un point où l'ennemi se dirigeait.

Grâce à ce système, quatre types de déplacement sont nés :

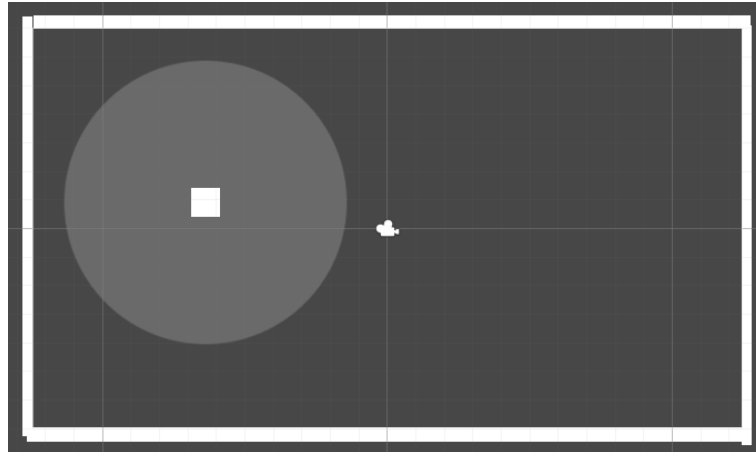
### 2.2.1 Le Patrouilleur

Cet ennemi possède deux types de déplacement :

- Le mode patrouille : il va se déplacer vers un point, attendre quelques secondes puis repartir vers un autre point.
- Le mode attaque : le patrouilleur possède une zone autour de lui. Dès que le joueur passe dans cette zone, il va se faire poursuivre par le patrouilleur tant qu'il est dedans.

A partir du moment où le joueur sort de cette zone (zone transparente sur l'image), l'ennemi se rend vers la dernière position connue (l'endroit où le joueur est sorti) et attend quelques secondes avant de reprendre son mode patrouille.

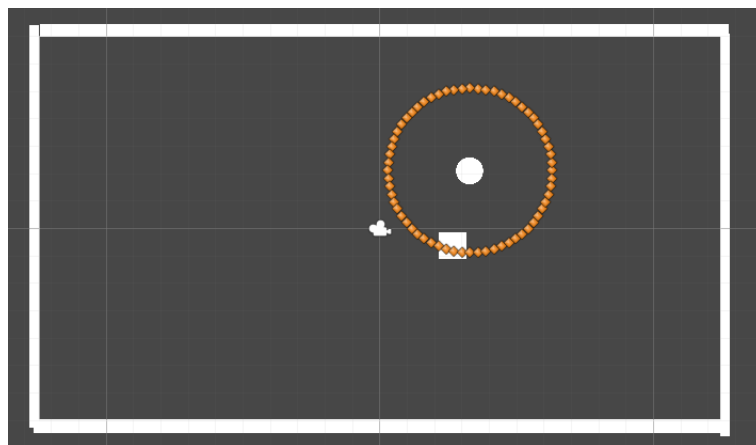
La création du script fut assez compliquée, étant le premier script, il a fallu concevoir la méthode de déplacement pour cet ennemi et les suivants. Cela se traduit par la création du mode patrouille, puis du mode attaque, qui doit avoir la priorité sur le mode patrouille.



### 2.2.2 Le Tourneur

Cet ennemi ne possède qu'un seul type de déplacement. Il tourne en cercle autour du joueur tant qu'il est en vie, sans jamais s'arrêter. Bien que le déplacement soit simple à comprendre, le créer a été un peu plus compliqué. Il a fallu concevoir un algorithme qui produit un cercle puis l'adapter pour que ce dernier se génère autour d'un point et qu'il reste quoiqu'il arrive autour de ce point.

*Attention cet ennemi peut vous donner le tournis*



### 2.2.3 Le Follower

En tant qu'influenceur, vous voulez que votre influence attise la curiosité de tous. C'est donc pour cela que le Follower, dès son apparition, va vous suivre tant qu'il est vivant. Il veut votre signature ou plutôt votre mort. Pour le script, c'est juste une reprise de celui du patrouilleur, avec quelques adaptations.

#### **2.2.4 Le Spawner**

Le Spawner est un ennemi assez fourbe qui fait apparaître des ennemis toutes les  $x$  secondes. Concernant son mode de déplacement, le Spawner reste à un point précis tant que le joueur ne rentre pas dans sa zone. Dès que le joueur est dedans, le Spawner va se rendre vers un autre point et continuer à faire apparaître des ennemis.



## 2.3 Le Gameplay

### 2.3.1 Création de la classe personnage

Afin de permettre au joueur d'incarner un personnage dans le jeu, il faut d'abord créer ce personnage puis implémenter les différentes actions qu'il pourra effectuer : se déplacer, attaquer, ramasser un objet ou bien encore interagir avec un autre personnage non joueur. Axelle s'est donc chargée de créer un script player qui contient plusieurs variables.

La première est le nom du personnage. Elle nous permettra d'identifier les différents joueurs lorsque nous créerons le mode multijoueur. Viennent ensuite les variables *health*, *bonusHealth* et *maxHealth*, représentant respectivement la vie actuelle du personnage, son bonus temporaire de vie et sa vie maximum sans bonus. Ces trois valeurs nous permettront de connaître l'état du personnage : vivant ou mort, afin d'effectuer les actions complémentaires. Une autre variable essentielle de ce script est *speed*, la vitesse du personnage. Les quatre variables restantes nous seront utiles pour les attaques. Ces dernières sont *attack*, le nombre de dégâts qu'inflige une attaque du personnage, *attackRange*, la distance maximale à laquelle le joueur peut attaquer, *fireRate*, la fréquence à laquelle le joueur peut attaquer et *shotSpeed*, la vitesse des projectiles avec lesquels le joueur attaque.

Nous pourrions ainsi modifier la valeur de ces variables notamment lors de la récupération d'objets, modifiant les statistiques du joueur, ou lorsque ce dernier est atteint par une attaque ennemie.

L'étape suivante est de créer les différentes méthodes de ce script telles que le déplacement et l'attaque.

### 2.3.2 Création de la classe ennemi

Les ennemis sont tout aussi importants que les personnages. Sans ennemis, le jeu n'aurait plus de sens. Il faut donc, comme le personnage, créer l'ennemi et ensuite lui attribuer ses actions : se déplacer de façon autonome et attaquer lorsque le personnage rentre dans son champ d'attaque. Gwennan s'est chargée du script enemy. Ce script contient plusieurs variables tels que le nom du monstre, permettant d'identifier son type afin de lui associer plus facilement son type de déplacement et d'attaque.

Tout comme le script player, le script enemy contient aussi la variable *health*, représentant la vie actuelle de l'ennemi et donc permettant de connaître son état : vivant ou mort. Cet état rendra alors possible d'effectuer les actions attendues. Une autre variable utile de ce script est *speed*, la vitesse de l'ennemi. Les quatre autres variables (*attack*, *attackRange*, *fireRate* et *shotSpeed*), reprenant le même principe que le script player, seront utiles pour les attaques.

Pour la prochaine soutenance, l'objectif serait donc de réussir à associer pour chaque type d'ennemi son type de déplacement et aussi se charger des attaques.

### 2.3.3 Déplacement du joueur

Parmi les actions de base que doit effectuer le personnage incarné par le joueur, le déplacement est la première action que nous avons implémentée. Conformément à la répartition des tâches, cette mission a été confiée à Axelle. Unity met à disposition divers outils afin de permettre le déplacement d'un personnage. Nous avons choisi d'utiliser l'input manager fourni afin de récupérer les entrées du joueur nécessaires au déplacement. Dans un premier temps, nous utiliserons les touches Z, Q, S et D pour le déplacement. Cela correspond à une configuration classique pour les jeux vidéo sur un clavier AZERTY.

Notre algorithme récupère à chaque nouvelle image générée la valeur de chacun des deux axes X et Y (-1, 0 ou 1) en fonction des touches pressées par le joueur. Il crée ensuite un vecteur ayant pour coordonnées la valeur de chaque axe récupérée multipliée par la vitesse du personnage ainsi que le temps écoulé depuis la dernière image générée. Ce dernier facteur permet d'assurer un déplacement constant, indépendamment du nombre d'image par seconde. Ainsi, le personnage se déplacera à la même vitesse quelle que soit la fréquence de rafraîchissement de l'image. La vitesse du personnage quant à elle est un attribut de celui-ci, éventuellement modifiée par des items récupérés au cours de la partie.

### 2.3.4 Attaque du joueur

La seconde action à implémenter pour le personnage du joueur est l'attaque. Dans notre jeu, l'attaque de base du joueur sera une attaque à distance. Il disposera de quatre touches pour définir la direction de ses attaques. Il pourra ainsi attaquer dans une direction différente de celle où il se déplace. Pour parvenir à créer cette mécanique, Axelle a choisi d'utiliser une nouvelle fois l'input manager fourni par Unity, en créant deux nouveaux axes. Le joueur pourra ainsi définir la direction de ses tirs avec les flèches directionnelles.

La prochaine étape est de créer une instance de projectile avec les caractéristiques suivantes : les dégâts qu'il infligera lorsqu'il atteindra une cible, la distance maximale à laquelle il peut toucher une cible, ses coordonnées de départ et sa direction.

## 2.4 L'Histoire

Même si le plot de l'histoire peut vous sembler assez court, la TAAG TEAM estime que les jeux avec une histoire simple et facile à comprendre sont les plus attrayants pour le public.

Ainsi, voici un résumé de l'histoire du jeu *Toddlers Are Afraids of Ghosts* :

Dans un orphelinat sombre et humide, des enfants se retrouvent chaque nuit prisonniers du même cauchemar. Ils sont enfermés dans un donjon dont le seul moyen d'en sortir est d'affronter le dernier boss, leur surveillant.

Ainsi, l'histoire remet en contexte les événements (la nuit dans un orphelinat), annonce le statut des joueurs (orphelins), le boss final (le surveillant) et comment gagner le jeu.

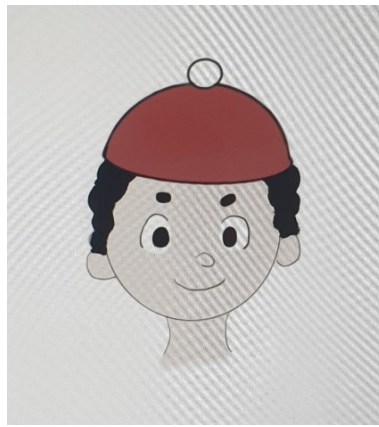
## 2.5 Les Graphismes

Pour cette première soutenance, Gwennan et Thomas ont du réfléchir comment faire pour que les graphismes se coordonnent et restent dans un thème précis.

### 2.5.1 Les personnages

Commençons par l'élément le plus important pour le joueur : son personnage. Non seulement il doit être quelque peu esthétique, mais il faut aussi que le joueur dispose d'une variété de choix, avec des capacités différentes.

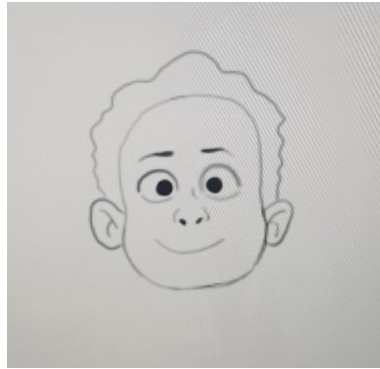
Gwennan a ainsi réalisé six des huit personnages proposés en version visible pour le menu. Le design est passé par plusieurs réflexions afin de créer un personnage enfantin.



**Première proposition :** En dehors du fait que ce soit un simple croquis, la forme de la tête était un peu trop ronde et les éléments rappelaient un peu trop les anciens designs de jeux. Il fallait quelque chose de plus vif et plus moderne pour le joueur.

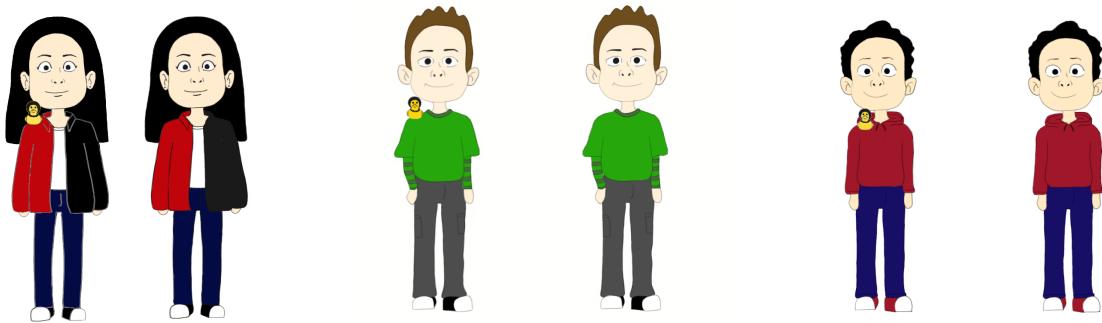


**Deuxième proposition :** la forme de yeux diffère de la première version dans un but de comparaison afin de déterminer quel design semble être le plus adapté. La forme de la tête, un peu moins ronde, du nez, de la bouche et des oreilles diffère aussi. Mais, le résultat n'était pas assez abouti et peut être toujours trop simple.



**Proposition finale :** puisque les précédents designs ne convainquaient pas. Gwen-nan a fait des recherches de design sur internet et a sélectionné des éléments distincts permettant ce rendu final. Les traits sont plus développés, la forme du visage n'est pas ronde et le reste des éléments diffère des précédentes versions mais l'esthétique souhaitée de design est là. Un personnage assez expressif et agréable d'esthétique.

Voici, après coloration, les six personnages définitifs qui seront proposés dans le menu :

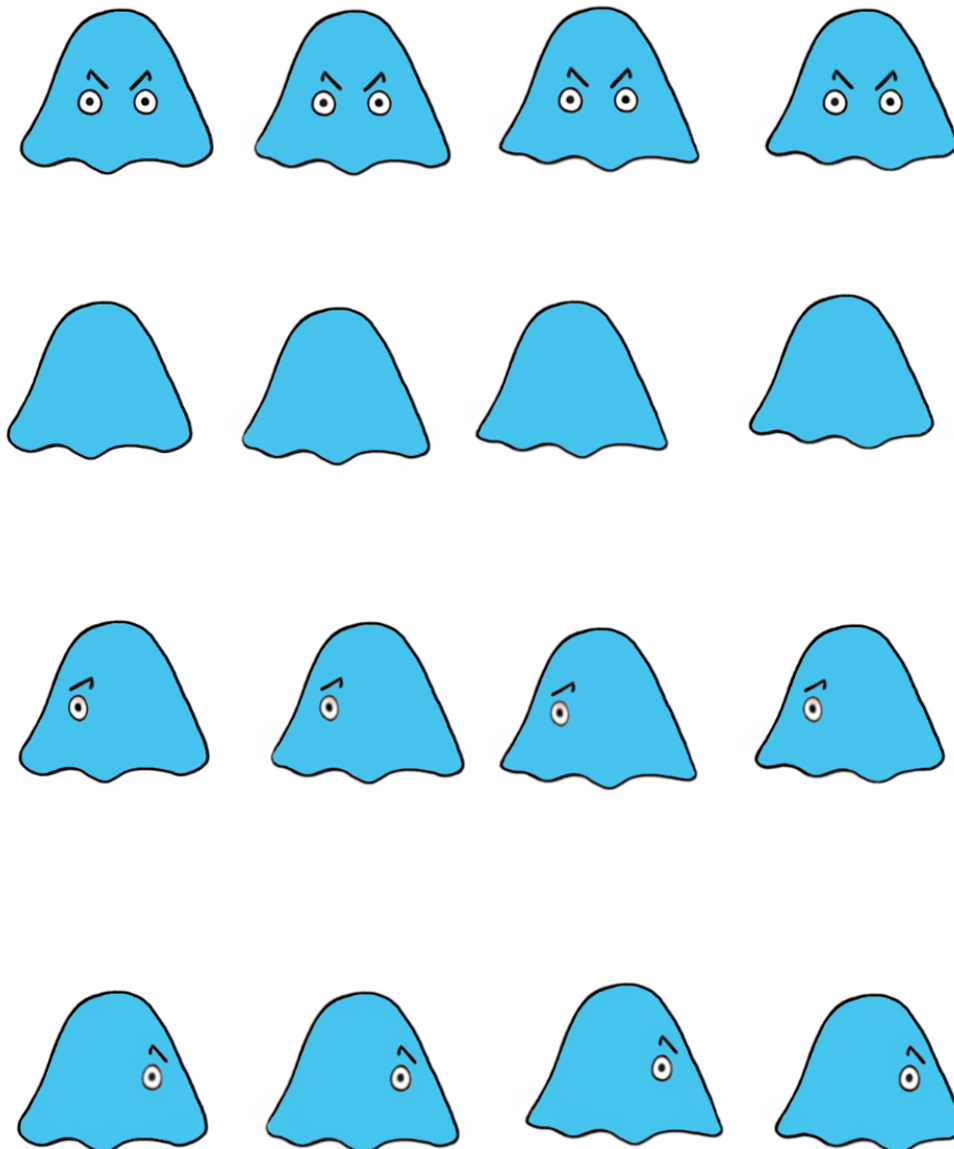


Pour la prochaine soutenance, le but sera donc de finir les 2 derniers visuels des personnages pour le menu et de commencer à dessiner les personnages du point de vue de la caméra, ainsi que leurs mouvements, afin de les animer.

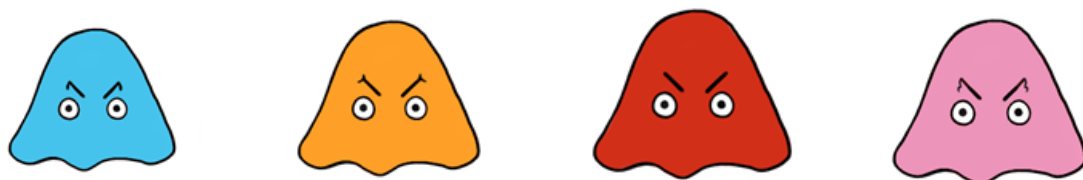
### 2.5.2 Les ennemis

En ce qui concerne les ennemis, le design devait être plus simple à animer mais aussi en concordance avec le thème du jeu : des monstres pour enfants. L'équipe a donc sélectionné des ennemis basés sur les méchants des jeux de leur enfance.

Gwennan a donc tout d'abord repris le design des fantômes du fameux jeu *Pacman* et l'a adapté selon son style pour répondre à ces exigences. Tout d'abord, les formes devaient être simples, faciles à modifier afin de faciliter l'animation. C'est pourquoi le mouvement de vague est le plus simple à dessiner.



Le design général des fantômes ne change pas. Les seuls éléments qui puissent les différencier sont leur couleur ainsi que la forme de leurs sourcils.



En ce qui concerne l'animation des mouvements dans Unity, j'ai cherché sur internet et grâce à des vidéos, j'ai pu la réaliser pour chaque côté de chaque personnage. C'est assez simple à faire, il suffit de mettre l'image en png, de faire des multiples sprites, d'insérer le premier sprite dans le visuel du jeu et enfin d'ajouter les sprites

dans l'ordre et espacés par un court délais dans la partie animation de Unity.

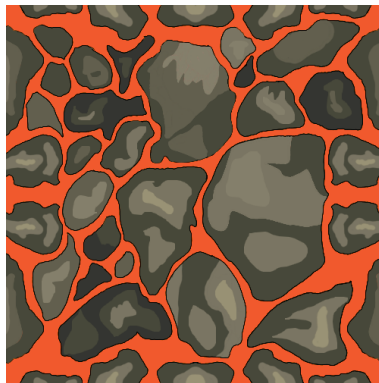
Les buts pour la prochaine soutenance seront donc de développer et d'animer un boss, semblable au design de *Pacman*, avec un peu plus de détails que ses sbires, mais aussi d'intégrer l'animation au code.

### 2.5.3 Les items

Toujours en cours de réflexion sur cette partie, Gwennan planifie de les réaliser pour la prochaine soutenance. Ce seraient juste des images inanimées, prises d'un certain point de vue. Rien de trop difficile ne serait à faire si ce n'est après de les ajouter dans la POO afin qu'ils apparaissent et disparaissent et comptent comme bonus.

### 2.5.4 Les textures

La difficulté dans la conception des textures est la continuité. Vu que chaque wall est un carré, il faut que les textures mises les unes à la suite des autres soient liées. Sinon le rendu est saccadé et ne donne pas l'impression d'un mur, mais de plein de carrés mis les uns à la suite des autres. De même, le sol doit donner un sentiment d'unicité.



Pour la suite, Thomas compte faire plus de textures de murs ainsi que les premiers obstacles présents dans les salles, comme des rochers par exemple. Cela se placerait dans la continuité du level design et des layouts de salles.

## 2.6 L'Immersion

En commençant le jeu, la possibilité de faire un menu est vite devenue une évidence afin qu'il contienne la gestion de plusieurs choses telles que la gestion des scènes, du volume, et encore. ...

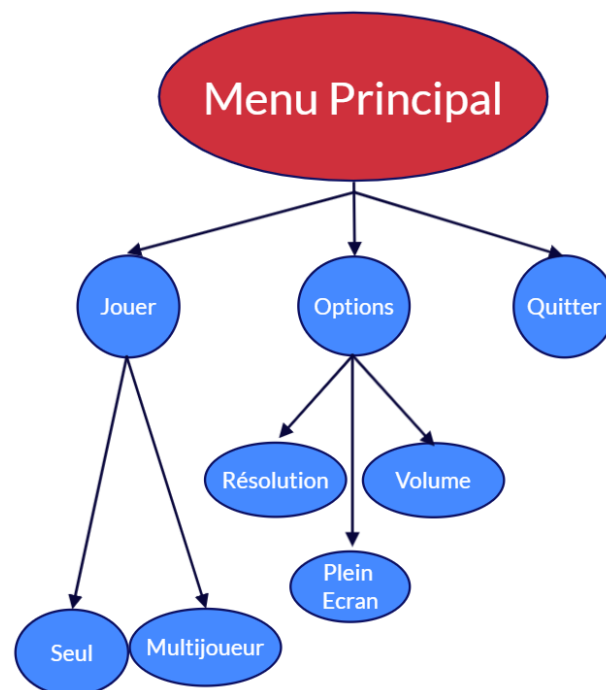
Cette partie est donc divisée en trois sous parties :

- Le menu principal, une fois le jeu *Toddlers Are Afraid of Ghosts* lancé.
- Le menu des réglages, Options.
- Le menu de pause, en jeu, qui s'active à l'appui de la touche échap.

Ces différents menus ne sont pas définitifs et sont susceptibles d'être modifiés dans le futur.

### 2.6.1 Menu Principal

Représentation du menu sous forme d'arbre :



Le menu principal permet d'accueillir le joueur sur le jeu. Il est constitué de 3 boutons :



- Le bouton "Jouer/Play" redirige vers un sous-menu permettant de rejoindre une partie ou bien d'en héberger une.
- Le boutons "Options" : ouvre un sous-menu constitué de différentes possibilités de réglage telles que : Résolution, Plein écran, Volume

### 2.6.2 Menu des options

Le menu Options permet de modifier les réglages du jeu. Il possède 3 possibilités :

- La **Résolution**, permettant de changer la résolution du jeu à l'aide d'un menu déroulant.
- Le **Plein écran**. En cochant une case, le jeu passera en plein écran, et, en la décochant, l'effet inverse se produira.
- Le **Volume**. A l'aide d'un slider, cette possibilité permet de régler le volume général du jeu.

### 2.6.3 Menu Pause

Il est très ressemblant à celui du menu principal. Il est constitué de 3 boutons :

- Le bouton "Jouer" afin de quitter le menu pause pour reprendre la partie là où le joueur en était.
- Le bouton "Options", qui à la même utilité que celui du menu principal.
- Le bouton "Revenir au menu principal", qui permet de quitter la partie et de retourner sur le menu principal.

Pour le moment, plusieurs problèmes sont encore existants tels que l'adaptation des boutons au changement de résolution. Mais ils seront résolus pour la prochaine maintenance. Tout comme le design qui est fortement susceptible d'évoluer.

## 2.7 Communication

### 2.7.1 Le site web :

Element principal de notre communication, le site web a été entièrement réalisé par Gwennan sans utiliser de design déjà existants.

Le site web est accessible via ce lien :

[https ://toddlers-are-afraid-of-ghosts.github.io/TAAG-Website/](https://toddlers-are-afraid-of-ghosts.github.io/TAAG-Website/)

Ci-dessous, un aperçu de la page d'accueil avec le menu animé.



Comme vous pouvez le constater, le site web est accessible via internet. Le processus complexe de chercher un hébergeur gratuit a irrémédiablement guidé Gwennan à Github, qui remplissait à la fois les conditions de gratuité d'hébergement mais aussi de facilité d'utilisation. Il y a bien sûr d'autres moyens d'héberger son site web mais très peu qui allient gratuité et facilité d'usage.

De plus, le fou des copyrights des sites en France existe bel et bien. Faut-il acheter le copyright ou bien seulement se fier à la protection du droit d'auteur ? Par défaut de financement mais aussi par soucis de protection, nous avons voté pour un compromis, faire figurer de façon informelle la notion de copyright sans pour autant la posséder. Nous espérons que cela permettra de dissuader et de prévenir le plagiat le plus possible.

Pour la prochaine soutenance, Gwennan compte refaire la page d'accueil, ainsi que les autres, afin de les adapter aux téléphones et ainsi agrandir l'accessibilité.

## 3 Ressentis des membres

### 3.1 Arnaud Corcione

Bien que le projet avance comme il faut, il nous reste encore beaucoup de travail vis-à-vis du jeu dans sa globalité, plein de choses à implémenter, même à retravailler. Au niveau de l'Intelligence Artificielle, bien que je sois satisfait de ce que j'ai fait, je sens que certains problèmes vont apparaître avec le temps et l'amélioration, comme par exemple le système de déplacement qui va devenir obsolète à l'implémentation des obstacles. Le plus complexe reste donc à venir pour moi.

### 3.2 Axelle Destombes

De mon point de vue, notre projet avance bien et nous sommes dans les temps. Nous ne sommes qu'au début et il reste beaucoup de travail à accomplir. Cependant l'état actuel du projet est conforme au planning annoncé dans notre cahier des charges. Au niveau des difficultés rencontrées, j'ai surtout eu du mal à m'habituer aux fonctionnalités de Unity telles que les préfabs ou l'input manager. Une fois accommodée à leur utilisation, ces outils se révèlent néanmoins très utiles. Pour la prochaine soutenance je compte terminer le système d'attaques du joueur, assister Arnaud si besoin sur celui des ennemis et participer à l'amélioration de l'interface. Je vais également ajouter des items collectables et la possibilité pour le joueur de les ramasser, ce qui aura pour effet de changer ses statistiques.

### 3.3 Gwennan Jarno

Je trouve que pour le moment notre travail avance bien. Nous semblons respecter nos objectifs. J'ai juste un peu peur de la quantité de graphisme à faire et je pense donc ne pas en faire beaucoup plus de peur de ne faire que ça dans ce projet. J'ai assez hâte de commencer à associer les animations avec les mouvements et de voir un peu plus comment le jeu évolue. Pour le site web, je suis encore assez mitigée sur mon design et j'aimerais trouver d'autres possibilités et solutions afin de le présenter.

### 3.4 Thomas Solatges

J'ai dû faire un choix dès le début, entre utiliser Windows pour la stabilité d'Unity, ou bien linux pour la facilité de git. J'ai donc choisi la deuxième option. La mise en place du git et la compréhension ont été assez fastidieuses de mon côté. Je trouve que dans un projet, le moment le plus difficile est le départ, car on part de rien et il faut tout mettre en place. Dans mon cas, plus spécifiquement, comprendre le fonctionnement d'Unity. Néanmoins, une fois lancé, le projet avance plutôt bien et j'ai pour l'instant appris beaucoup au cours de mes recherches pour la meilleure implémentation d'un système de génération procédurale.

## 4 Conclusion

En conclusion, chaque membre a réussi à avancer ses tâches, avec plus ou moins de facilité et d'organisation, et nous nous estimons à jour sur le planning par rapport à la première soutenance.

Ainsi, avec toujours la volonté d'avancer et l'esprit d'équipe, TAAG TEAM est assez satisfaite de ses débuts mais elle appréhende les prochaines étapes du projet.

## 5 Bibliographie

Voici les logiciels et sites que nous avons utilisé :

Pour rédiger ce document :

- Overleaf

Pour tout ce qui est création des images et dessins :

- Sketchbook
- Photoshop
- Unity

Pour tout ce qui est collaboratif et communication entre les membre de TAAG TEAM :

- Git
- GitHub
- GitKraken
- Discord
- Google Docs

Pour coder :

- Visual Studio Code
- Rider

