

# The Call Of The Goose

Projet de S2

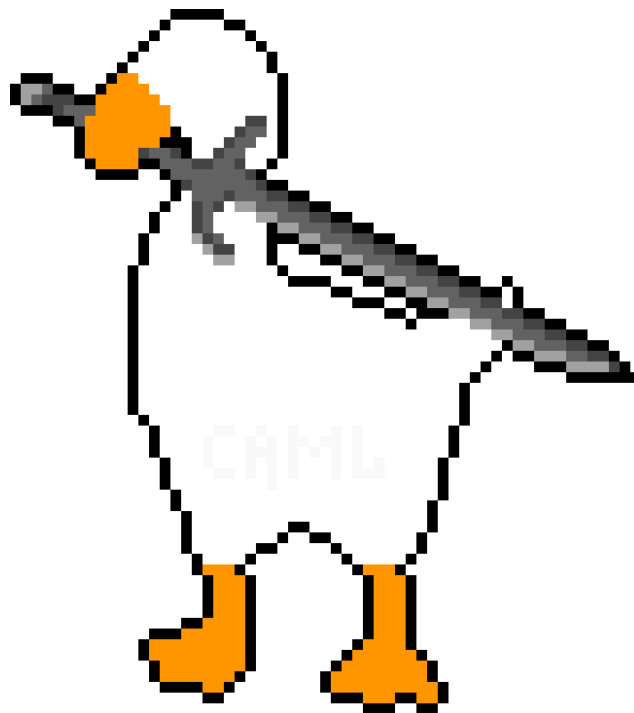
Rapport de soutenance

Yvon Morice

Souleymane Sentici

Amin Salmi

Marie Legay



# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Bibliographie</b>	<b>3</b>
<b>3</b>	<b>Le cahier des charges</b>	<b>4</b>
3.1	Le projet . . . . .	4
3.1.1	Présentation du jeu . . . . .	4
3.1.2	Principe du jeu . . . . .	4
3.2	Le groupe . . . . .	4
3.3	Travail . . . . .	5
3.3.1	Répartition des tâches . . . . .	5
3.3.2	Prévisions . . . . .	6
<b>4</b>	<b>Avancement</b>	<b>7</b>
4.1	Classes (Programmation orientée objet) . . . . .	7
4.2	Niveaux . . . . .	9
4.2.1	Génération de niveaux . . . . .	9
4.2.2	Déplacements des ennemis . . . . .	11
4.3	Interface utilisateur . . . . .	11
4.4	Menus . . . . .	13
4.5	Sprites . . . . .	14
4.6	Musique . . . . .	18
4.7	Combats . . . . .	19
4.8	Multijoueur . . . . .	21
4.9	Installateur . . . . .	22
4.10	Attentes pour la prochaine soutenance . . . . .	22
4.10.1	Design de contenu . . . . .	22
4.10.2	Ennemis . . . . .	22
4.10.3	Système de Combat . . . . .	22
<b>5</b>	<b>Conclusion</b>	<b>23</b>

# 1 Introduction

Notre objectif durant la dernière période de travail a été de passer d'un état de prototype à un état plus avancé, plus proche d'un jeu. Pour cela, nous nous sommes concentrés sur les parties graphiques et sonores du jeu, ainsi que les parties concernant le réseau, la génération de niveau et le comportement des ennemis et du joueur. Cela nous permettra de nous concentrer plus facilement sur la partie concernant le système de combat, une des parties majeures de notre projet.

## 2 Bibliographie

### **Génération de niveaux :**

- <https://www.youtube.com/watch?v=WtDXk6uuZO4>
- <https://www.youtube.com/playlist?list=PLX2vGYjWbI0SKsNH5Rkpxvvr1dPE0Lw8F>
- <https://www.youtube.com/watch?v=hk6cUanSfXQ>
- [https://www.youtube.com/playlist?list=PLBIb\\_auVtBwA-qr2-WnWX0LjZXkqKu5Aj](https://www.youtube.com/playlist?list=PLBIb_auVtBwA-qr2-WnWX0LjZXkqKu5Aj)
- <http://weblog.jamisbuck.org/2011/2/7/maze-generation-algorithm-recap>

### **Menu / Interfaces :**

- [https://www.youtube.com/watch?v=zc8ac\\_qUXQY](https://www.youtube.com/watch?v=zc8ac_qUXQY)

### **Graphismes :**

- <https://blog.studiominiboss.com/pixelart>
- <https://www.piskelapp.com>

### **Site web :**

- [bootstrapstudio.io/](http://bootstrapstudio.io/)

### **Installateur :**

- <https://jrsoftware.org/isdl.php>

## 3 Le cahier des charges

### 3.1 Le projet

#### 3.1.1 Présentation du jeu

- **Genre** : Rogue-like
- **Style** : Action, RPG
- **Plateforme** : PC (Windows)
- **Moteur** : Unity 2019.3.0f5

#### 3.1.2 Principe du jeu

##### **Au lancement de la partie :**

- Choix du type de partie (Solo ou Multijoueur)
- Sélection de la classe du personnage
- Lancement du jeu

##### **Lors de la partie :**

- Génération du niveau en cours de manière aléatoire
- Le jeu commence à l'étage 0 (sur 5 étages)
- Il y a un boss par étage ainsi qu'un changement d'ambiance aussi bien visuelle que sonore
- Objectif du joueur : atteindre le 5ème étage et vaincre le boss

### 3.2 Le groupe

Legay Marie, passionnée de jeux vidéos et de musique.

Morice Yvon, passionné de jeux vidéos qui a pour vocation de devenir développeur, si possible dans le domaine du jeu vidéo.

Salmi Amin, passionné de littérature Lovecraftienne, de cuisine et de jeux vidéos, qui a pour vocation de devenir développeur.

Sentici Souleymane est un passionné de jeux vidéos qui a pour vocation de devenir développeur.

### 3.3 Travail

#### 3.3.1 Répartition des tâches

Partage des tâches		
Tâche	Responsable	Suppléant
Histoire	Amin	Marie
Ambiance	Amin	Marie
Bande son	Marie	Amin
Character design	Amin	Marie
Animations	Marie	Amin
Dessin 2D	Marie	Amin
Level design	Souleymane	Yvon
Gameplay	Yvon	Souleymane
Tutoriel	Souleymane	Yvon
Système de combat	Marie	Amin
Interactions avec le marchand	Marie	Souleymane
Génération des niveaux	Yvon	Souleymane
Items	Souleymane	Amin
Ennemis	Marie	Yvon
IA des ennemis	Marie	Yvon
Personnages	Souleymane	Amin
Pathfinding	Amin	Souleymane
Menu/Menu de pause	Amin	Yvon
Multijoueur	Marie	Yvon
Site web	Yvon	Souleymane

### 3.3.2 Prévisions

Tâche	Soutenance		
	1ère	2nde	3ème
Bande son	0	25	100
Character design	50	70	100
Animations	10	50	100
Dessin 2D	10	50	100
Level design	70	90	100
Gameplay	40	80	100
Tutoriel	10	60	100
Système de combat	30	60	100
Interactions avec le marchand	0	50	100
Génération des niveaux	40	70	100
Items	10	50	100
Ennemis	30	50	100
IA des ennemis	10	60	100
Personnages	20	60	100
Pathfinding	0	50	100
Menu/Menu de pause	50	75	100
Multijoueur	20	60	100
Site web	30	50	100

## 4 Avancement

### 4.1 Classes (Programmation orientée objet)

Réalisation (Souleymane)

Celles-ci sont primordiales pour le bon fonctionnement du système de combat que nous prévoyons de concevoir.

1. La classe Player sert à attribuer les différentes statistiques que possédera le joueur en combat (telles que les points de vie, la mana ou l'endurance par exemple). Elle sert également à créer les méthodes qui permettront au joueur d'utiliser les différents éléments des autres classes présentées ici.

```
public class Player : MonoBehaviour
{
    protected int maxHp;
    protected int maxMana;
    protected int maxEndurance;
    protected int hp;
    protected int armor;
    protected int lvl = 1;
    protected int exp;
    protected int mana;
    protected int endurance;
    protected int gold;
    protected int attack;
    public string name; // Unchanged
    public HealthPoint HPPlayer; // Unchanged
    public HealthPoint ManaPlayer; // Unchanged
    public HealthPoint EndurancePlayer; // Unchanged
    public int floor; // Unchanged
    protected int[] expTreshold = {100, 164, 268, 441, 723, 1186, 1945, 3190, 5233};
    public Consumables[] consumablesInventory; // Serializable
    public List<Relic> relicInventory; // Serializable
}
```

FIGURE 1 – Les mentions en commentaires correspondent aux éléments des classes pas encore implémentées

2. La classe Relic sert à créer les objets que trouvera le joueur et qui serviront à augmenter ses statistiques de manières définitives pour la partie en cours. Celles-ci sont permanentes (contrairement aux consommables) et seront trouvables sur les ennemis tués ou chez le marchand.



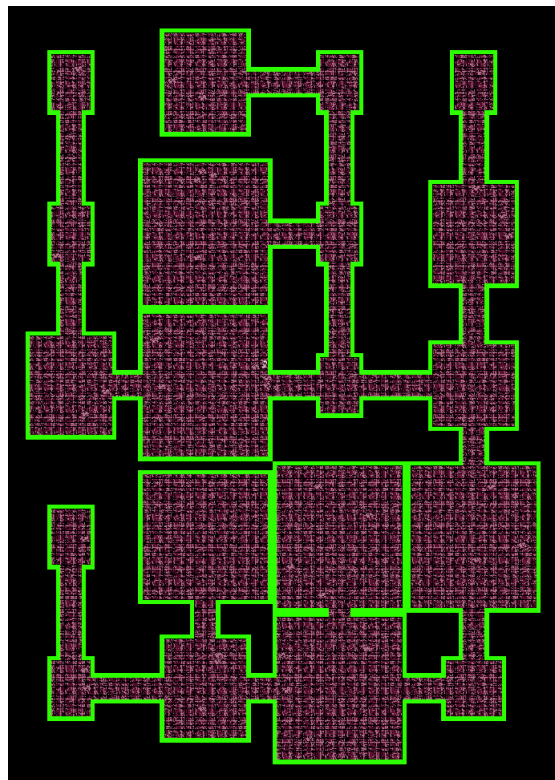
3. La classe Consumables sert à implémenter les différents objets à utilisation unique que le joueur pourra rencontrer et utiliser durant ou entre les combats. Les méthodes les concernant sont déjà implémentées dans la classe Player, celle-ci sert donc à créer les objets en eux-même.
4. La classe Attacks permet de créer les différentes attaques que possèdent les entités du jeu. Celle-ci permet de leur attribuer un montant de dégats ainsi qu'un coût en mana et/ou en endurance pour être lancée. Ceci permet de créer plus de diversité au sein du système de combat, en offrant aux classes ainsi qu'aux ennemis une plus grande diversité de coups.

## 4.2 Niveaux

### 4.2.1 Génération de niveaux

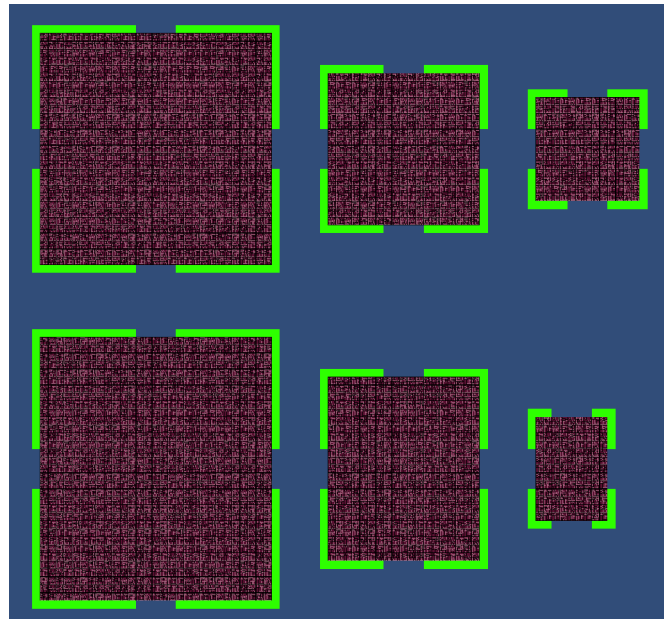
Réalisation (Yvon)

Cette section parle de la génération des différents niveaux que le joueur va avoir a traverser, ceux-ci étant au nombre de cinq. Chaque niveau est généré de manière aléatoire de façon à former un labyrinthe ayant toujours la même taille (4 salles de larges pour 5 de hauteur).

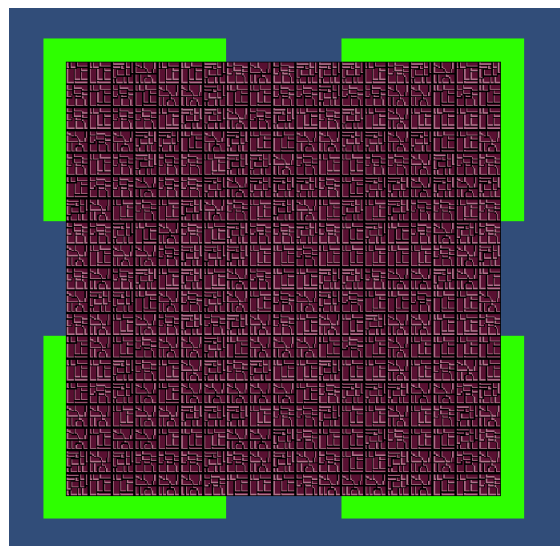


Un exemple de niveau pouvant être généré dans l'état actuel du projet

Il y a six modèles de salles avec chacune une taille différente. Chacune de ces salles va prendre des tiles à afficher et les répartir de manière aléatoires.



Les 6 modèles de salles, de la gauche vers la droite sur la première ligne : boss, spawn, shop. Sur la seconde ligne : grande salle, salle moyenne, petite salle



La salle de spawn vue de près

Afin de générer un labyrinthe, on utilise l'algorithme des arbres binaires. Pour cela, on va itérer au travers de chaque salle. Dans chacune de ces salles, on va décider de créer un couloir, soit vers la droite, soit vers le bas. Si l'une des deux directions n'est pas disponible, on prendra l'autre. Si aucune des deux directions n'est disponible, on ne crée pas de couloirs.

#### **4.2.2 Déplacements des ennemis**

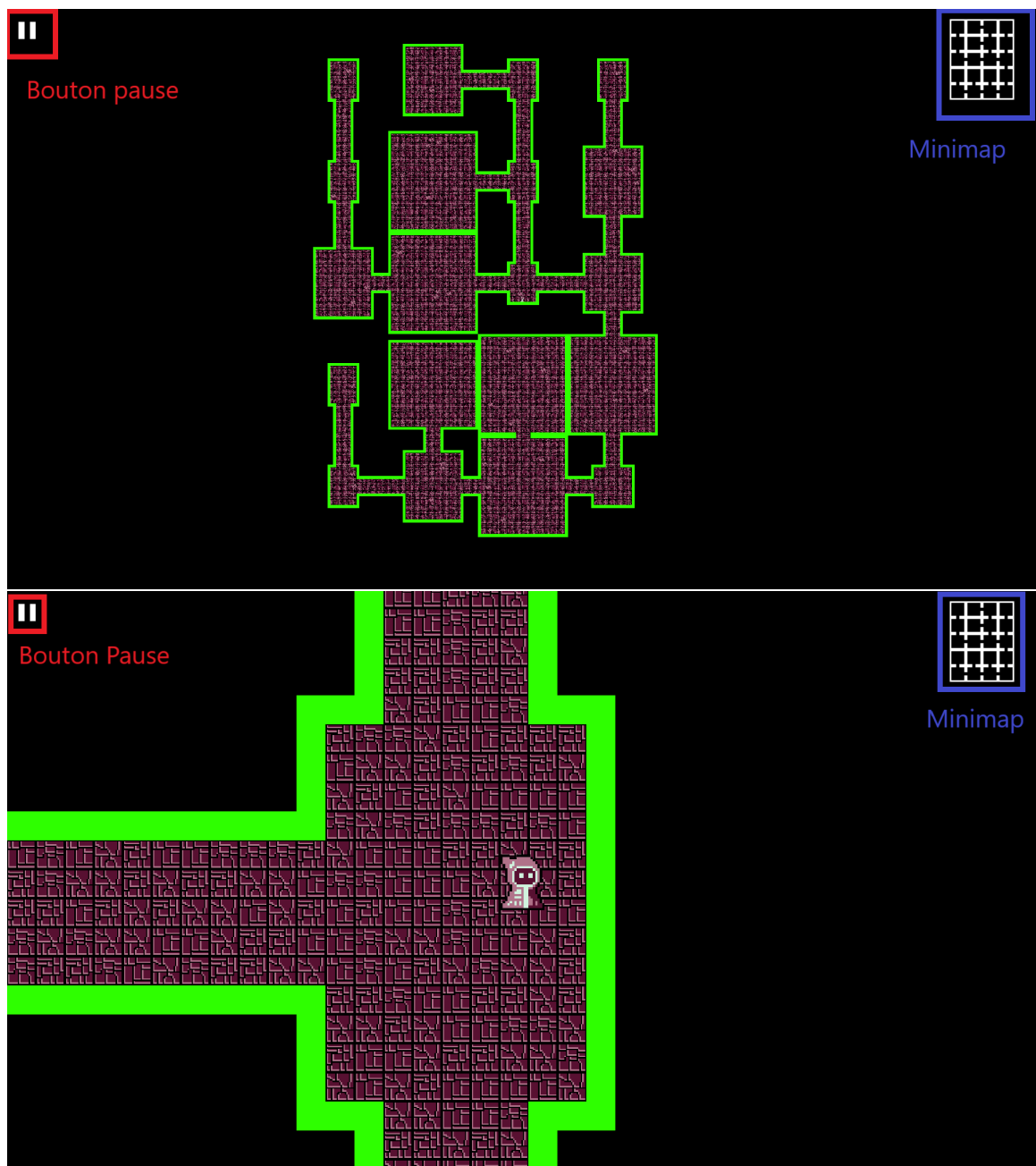
Réalisation (Souleymane)

Cette section concerne les déplacements des ennemis présents sur la carte du jeu. Grâce à un script Unity, nous avons fait en sorte que chaque ennemi sache à tout moment quel est le joueur qui lui est le plus proche, et calcule la distance qui les sépare l'un de l'autre. Si celle-ci descend en dessous d'un certain seuil, l'ennemi se met à suivre le joueur en prenant le chemin le plus court, et s'il réussit à le toucher, un combat est censé avoir lieu.

### **4.3 Interface utilisateur**

Réalisation (Yvon)

Cette section concerne l'interface utilisateur (UI) du joueur lors d'une partie. Celle-ci est pour le moment, composée de deux éléments, une minimap ainsi qu'un bouton permettant de mettre le jeu en pause. La minimap est créée juste après le niveau, pour cela, elle récupère la liste des salles générées et affiche une version simplifiée permettant de connaître de manière simple la disposition du niveau. À terme, cette minimap comprendra également la position du joueur.



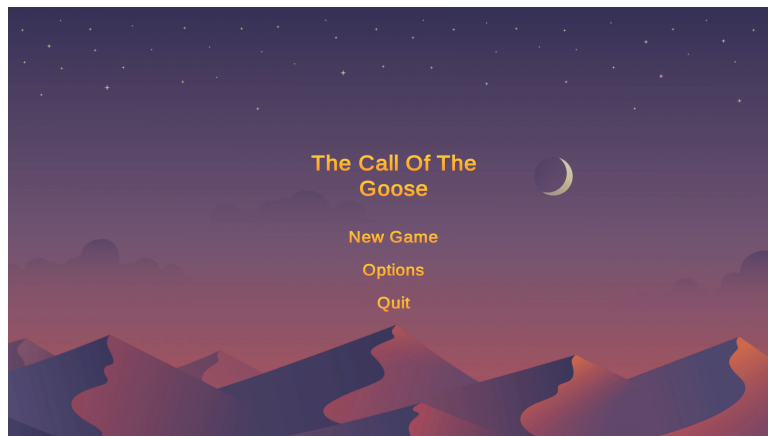
L'état actuel de l'interface utilisateur

## 4.4 Menus

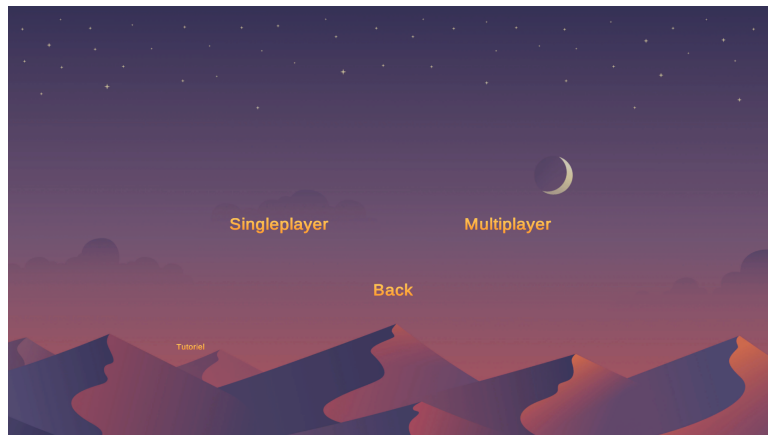
Réalisation (Amin)

Cette section concerne, comme son nom l'indique, le menu de notre jeu. Réalisé sur Unity et avec Rider pour les scripts, le menu principale permet de :

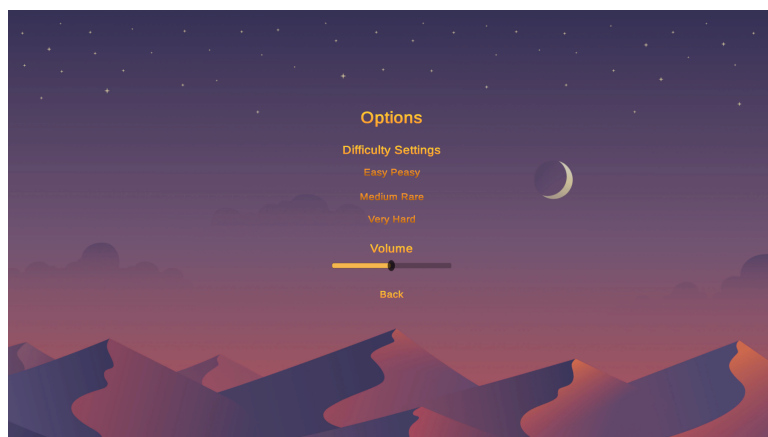
1. Quitter le jeu à l'aide du bouton "Quit".
2. Accéder au sous-menu des options à l'aide du bouton "options", ce sous menu contient un slider qui modifie le volume du son, un bouton "Back" pour revenir au menu précédent, et 3 boutons de difficultés "easy", "medium", et "very hard", chaque bouton assignera à une variable le chiffre 0, 1 ou 2 qui permettront de modifier certaines valeurs du jeu, tel que l'attaque, les point de vie et le nombre de points d'expérience que donneront les ennemies.
3. Accéder au sous-menu de lancement de jeux, avec les trois boutons "Singleplayer", "Multiplayer" et "Tutorial" qui n'ont pas besoin d'être expliqués, et le bouton "Back" pour revenir au dernier menu.



Le menu principal



L'écran de sélection entre les modes de jeu solo, multijoueur et le tutoriel



Le menu d'options, pour changer la difficulté du jeu et modifier le volume du son

## 4.5 Sprites

Réalisation (Amin et Marie)

Cette section regroupe les Sprites réalisés jusqu'alors grâce au logiciel

Piskel, certains ne sont pas encore inclus dans le jeu. Nous ne montrons que la Sprite Sheet car il est compliqué d'imprimer un gif sur une feuille en papier. La totalité des sprites sont réalisés avec la même palette constituée de seulement 3 couleurs, nous nous sommes nous-même imposé cette contrainte afin de donner une certaine patte artistique à notre jeu.

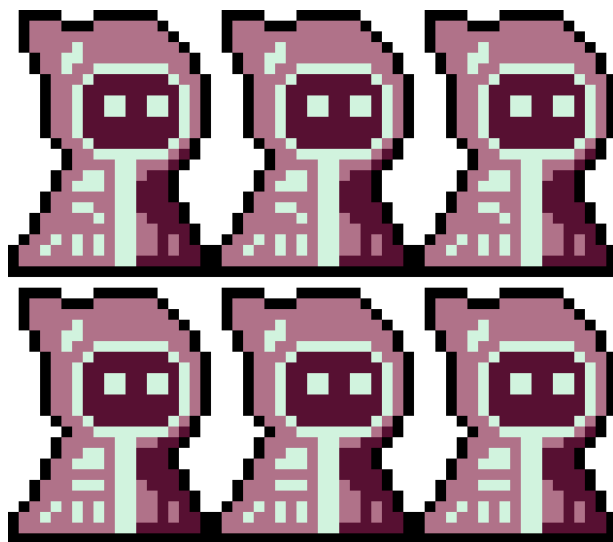


Le personnage jouable dans le monde.

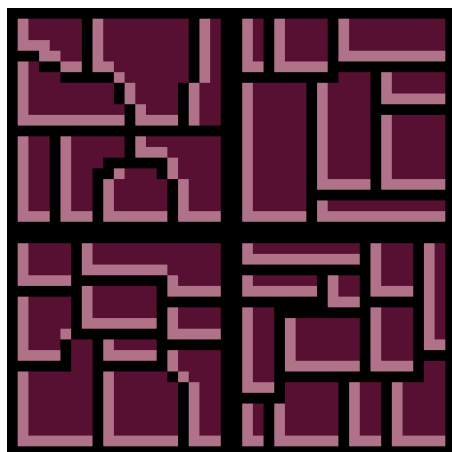


Le personnage jouable se déplaçant à gauche.

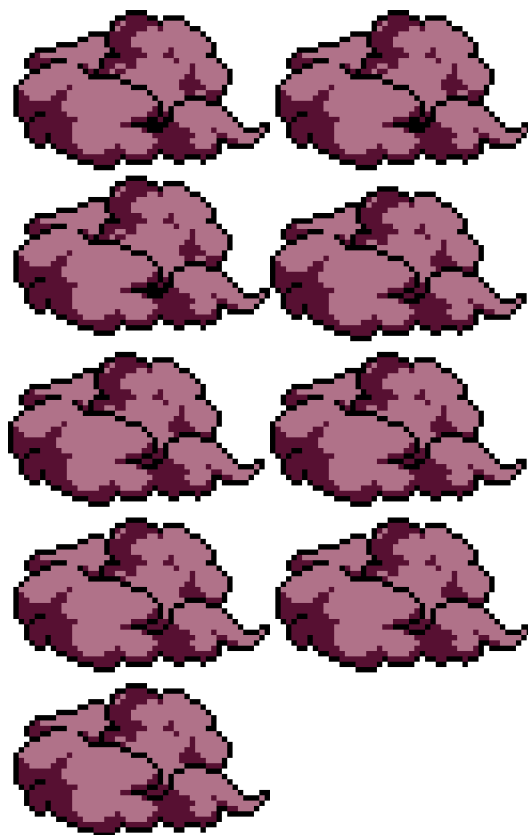




Le personnage jouable se déplaçant à droite.



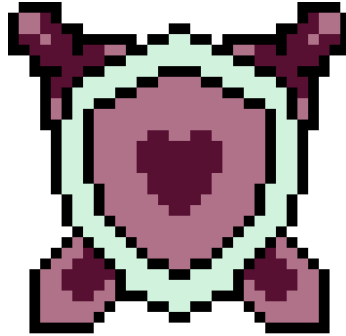
Le set de tiles qui constituera le sol du premier niveau.



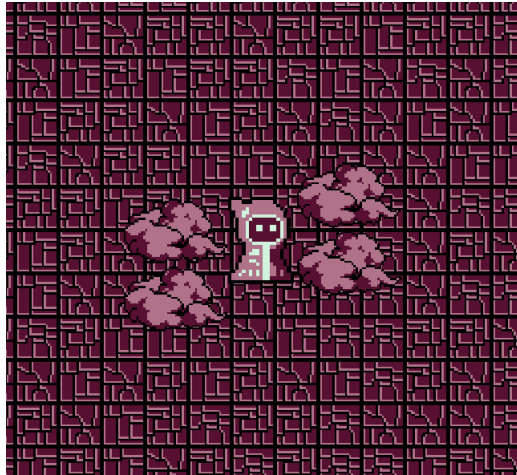
Le nuage qui représente les ennemis dans le monde.



Une épée représentant un item du jeu qui donne un bonus d'attaque au joueur.



Un bouclier représentant un item du jeu qui donne un bonus de défense au joueur.



les sprites du personnage, des nuages et du sol in-game.

## 4.6 Musique

Réalisation Marie

Le but principal de la musique était de donner au joueur un sentiment que quelque chose n'allait pas, à travers de notes graves présentes en arrière plan.

Le résultat final est différent et la mélodie prévu pour être le centre d'attention semble moins important que le violon et la contrebasse qui jouent des notes donnant un aspect inquiétant.

## 4.7 Combats

Réalisation (Marie)

Cette section abordera la mise en place du système de combat. il se base sur les combat au tour par tour ; plus particulièrement celui de Final Fantasy VII.

```
Frequently called 2 usages
IEnumerator EnemyTurn()
{
    Pl.TakeDamage(Op.Attaque1);

    yield return new WaitForSeconds(1f);

    if (Pl.Hp <= 0)
    {
        State = Battle.Lost;
        End();
    }
    else
    {
        State = Battle.PlayerTurn;
        PlayerTurn();
    }
}
```

Script correspondant au tour de l'ennemi

```

if (Pl.Endurance > 0 && Input.GetKeyDown(KeyCode.Space))
{
    Op.TakeDamage(Pl.Attaque);
    Pl.Endurance -= 5;
    Pl.EndurancePlayer.SetHp(Pl.Endurance);
    yield return new WaitForSeconds(2f);

    if (Op.HP <= 0)
    {
        State = Battle.Won;
        End();
    }
    else
    {
        State = Battle.EnemyTurn;
        StartCoroutine( routine: EnemyTurn());
    }
}

```

Script correspondant au cas où le joueur fait une attaque Physique

A la fin de chaque tour il faut vérifier si le joueur est mort ou si il a tué chaque ennemis sur le terrain. Les dégâts infligés dépendent de la statistique d'attaque du personnage qui inflige les dégâts.

On y trouve une barre de Point de Vie pour chaque entité. Chaque joueur sera en plus munie d'une barre Magie, utilisée lors des attaques à distances, et une barre d'endurance, utile pour les attaques physiques. Ces barres permettront de diversifier la façon de jouer chacun des personnages.

```

4 asset usages 8 usages 4 exposing APIs
public class HealthPoint : MonoBehaviour
{
    public Slider slider;  Changed in 1 asset

    public void MaxHealth(int hp)
    {
        slider.maxValue = hp;
        slider.value = hp;
    }

    Frequently called 8 usages
    public void SetHp(int hp)
    {
        slider.value = hp;
    }
}

```

Code permettant de gérer l'affichage des barres de PV, Mana et Endurance

Le joueur pourra restaurer ces barres grâce à des consommables qui seront ajoutés ultérieurement.

## 4.8 Multijoueur

Réalisation (Marie)

Cette section parle du début de la création du mode multijoueur. Il a été créé grâce à Unity, Photon et Rider pour les scripts.

Un changement majeur s'est opéré depuis le dernier rapport : le multijoueur ne s'effectue plus à l'aide d'un ordinateur "Host". A présent, Photon se charge de connecter les ordinateurs entre eux.

Une fois les deux joueurs connectés, Unity lance le jeu sur les deux machines. Son avancement actuel ne permet cependant pas de finir le mode multijoueur.

## **4.9 Installateur**

Réalisation (Yvon)

Cette section parle de l'installateur qui a été créé pour ce projet. Une procédure de désinstallation a été incluse dans ce dernier afin de pouvoir désinstaller facilement le jeu sans avoir à passer par les paramètres.

## **4.10 Attentes pour la prochaine soutenance**

### **4.10.1 Design de contenu**

Nous devons créer et implémenter au moins une partie des classes jouables que nous avons prévu. Nous devons également créer les différents objets consommables que trouvera le joueur durant les parties, ainsi que certaines des reliques qui augmenteront ses statistiques pour la partie en cours.

### **4.10.2 Ennemis**

De plus, une partie des ennemis trouvables en jeu devraient être terminés. Ces ennemis seront rencontrables sur la carte générée aléatoirement et tenteront de se rapprocher du joueur. Nous allons donc leur créer une IA, accompagnée d'un algorithme de pathfinding. Ceci leur permettra de poursuivre le joueur qui pourra choisir de les éviter pour aller plus vite, ou de les combattre pour obtenir des récompenses.

### **4.10.3 Système de Combat**

Le système de combat de notre jeu est presque terminé, mais il requiert encore quelques modifications. Une fois celles-ci apportées, les combats devraient être fonctionnels.

## 5 Conclusion

Durant cette deuxième période de travail, nous nous sommes concentrés sur certaines parties telles que la génération de niveau, les graphismes ou encore le comportement des ennemis. Nous sommes encore assez loin d'un état fini pour notre projet étant donné que certaines parties majeures telles que le système de combat ne sont pas finies, mais nous espérons pouvoir le terminer d'ici la prochaine soutenance.