



Rapport de projet

Theo Rolland, Arsène Hiriart, Sacha Villepreux-Petit,
Alexis Boussyguine, Alessandro Platz

Juin 2024

Table des matières

1 Alexis Boussyguine	3
1.1 Mon rapport du site web avant mars	3
1.1.1 Leçons après avoir fait cela	5
1.2 Créer un grand ennemi "boss" après mars	6
2 Théo Rolland	13
2.1 Création du site web	13
2.2 Mise en ligne du site web sur internet	13
2.3 Organisation du projet	15
2.4 Musique	17
2.5 Identité visuelle	18
2.5.1 Le choix des couleurs	18
2.5.2 Le choix des éléments	18
2.6 Les réseaux sociaux	18
2.7 Dossier d'exploitation	19
2.8 Rapport de méthodologie	19
3 Sacha Villepreux-Petit	21
3.1 Mon rôle dans le projet	21
3.2 Le déroulement de ma partie	22
3.3 Mes attentes pour la suite du projet et vision globale	24
3.4 Quelques problèmes	24
3.5 Compte rendu après la première soutenance et mise au point de mi-projet	25
3.6 Ce qui a été fait	26
3.7 Un ajout trop ambitieux	28
3.8 Quelques problèmes 2	29

4 Arsène Hiriart	31
4.1 Les déplacements	31
4.2 Les ennemis	32
4.3 Les niveaux	32
5 Alessandro Platz	34
5.1 Mon rôle	34
5.2 Ce que j'avais fait pour la dernière soutenance	34
5.2.1 L'identité graphique au jeu	34
5.2.2 Les "tile sheets"	36
5.2.3 Les Animations	37
5.2.4 Ce qu'il reste à faire	37
5.3 Le parallaxe	38
5.4 L'amélioration du tile sheet	39
5.5 Décoration des niveau	41
5.6 Le Boss	41
5.7 Les items du jeu	42
5.8 "Oeuf de Pâque"	47
6 Annexe	48

Alexis Boussyguine

Je suis responsable du site web. Je ne m'occupe pas du jeu vidéo sur Godot avant mars dans le but de mieux nous répartir les tâches.

Ainsi, comme on est 5, on peut plus facilement avancer en nous concentrant sur une tâche précise et ainsi éviter à ce que tout le monde se pose des questions sur trop de sujets différents et ainsi mieux avancer sur leur tâche à eux.

Mais malgré cela, il a fallu que je fasse un peu le jeu qui est sur Godot pour ne pas totalement être à l'écart du groupe et donc si je devais changer de rôle, il fallait donc que j'apprenne totalement de nouvelles compétences, et cela a un impact non négligeable sur notre gestion du temps.

1.1 Mon rapport du site web avant mars

Le site web que je suis en train de construire est un élément central dans notre projet de jeu vidéo car il permet d'avoir une vision d'ensemble et imagée sur l'ensemble très conséquent de toutes nos ressources utiles pour créer ce jeu.

En effet les ressources dont on a à disposition sont très nombreuses et on se perd très facilement, peu importe la manière dont on veut les structurer dans toute notre architecture de fichiers et de dossiers qui structure tout ce que l'on a construit pour ce jeu vidéo. Et en plus si c'est la première fois qu'on touche à Github, cette plateforme en ligne est certes facilement accessible mais ne permet pas de bien se repérer et rien que télécharger des documents qu'on a placés dans cet espace n'est pas visible au premier regard en comparaison avec l'explorateur de fichiers de tout ordinateur.

Mais créer un site web, en particulier sur Notepad ++ aussi est très loin

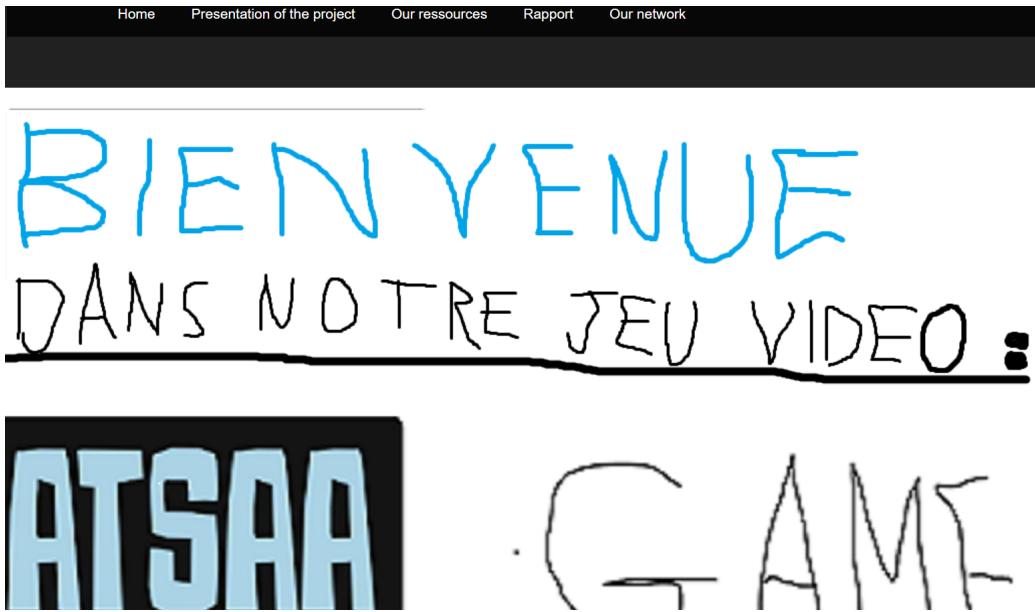


FIGURE 1.1 – Prototype du site web

d'être une tâche aisée, et ce pour plusieurs raisons :

si on doit créer un site web , ça veut dire qu'on doit le coder pour y imposer son propre style afin de s'assurer que c'est bien nous en personne qui avons créé ce site. Et personne sur Internet ne peut nous apprendre à être original , c'est-à dire justement de concevoir de la meilleure des manières un site web adapté à un jeu vidéo , et ce sans être trop simple ni trop difficile.

trop simple dans le sens où ne fait que générer un peu de texte ainsi qu' utiliser 10 balises HTML et CSS de base

et trop difficile si on doit créer quelque chose le plus abouti possible avec plusieurs milliers de ligne de code

Sans outils pour analyser tout ce code, on se trompe très facilement car le nombre d'erreurs qu'on a s'accumule. ça ne semble pas être des erreurs en regardant son code de loin mais en réalité si. Et même utiliser ce genre d'outils spécialisés dans l'analyse ne sont pas évidents à utiliser car il faut les utiliser dans des cas bien précis et il faut avoir déjà beaucoup d'expérience pour comprendre comment faire cela.

si on doit trouver de l'inspiration quelque part mais que l'on ne change pas grand chose car on pense que cette ressource est très adaptée à notre

projet, on ne fait que le copier et donc on fait ce projet pour rien.

Une autre difficulté que j'ai dû affronter était de devoir sans cesse modifier plusieurs fichiers tout le temps car si on veut faire une modification sur un fichier et qu'on veut des liens avec tous ces fichiers représentant une page web, il faut toutes les modifier et également le faire dans un premier temps sans connaître les raccourcis claviers permettant de modifier plusieurs caractères identiques.

Et malheureusement je n'ai pas commencé le projet assez tôt, je n'ai jamais vraiment su comment réaliser un travail tel que ce projet à long terme, je pensais qu'à chaque fois que j'avais des tâches tels que la préparation de QCM et d'examens me rendait prioritaire sur tout.

Aussi le fait qu'on soit à l'école surtout pour les cours et qu'on se voit que très peu en réalité me rend assez mal à l'aise car cela renforce notre autonomie et le fait de ne pas vraiment se connaître et au risque comme moi d'être trop autonome et de ressentir de la faiblesse si on veut m'aider mais pour en fait tout faire à ma place.

Il est vrai que la meilleure manière d'apprendre quelque chose est en le pratiquant de manière concrète, alors que pour moi regarder des documentations s'éloignent souvent de cet aspect pour seulement comprendre des paramètres qui s'utilisent partout dans tous les sens.

1.1.1 Leçons après avoir fait cela

C'est comme ça que j'ai compris que je dois absolument me chronométrier sur des tâches plus petites et mieux devoir réfléchir à comment éviter des erreurs causées par le fait de vouloir tester plein de choses, même si elles ont l'air d'apporter du style à mon travail.

Il faut absolument aller droit au but. On a l'impression qu'en multipliant les détails, par exemple en voulant à tout prix réaliser un site web esthétique, on veut ainsi montrer qu'on a une réelle compétence mais qui n'est pas relié à un contexte réel, sans tenir compte des attendus les plus importants, ce qui amène à les éviter.

Et de même pour ce genre de travail de groupe je viens de comprendre qu'il faut plutôt travailler pour ce genre de tâches mais en favorisant le travail de 2 personnes pour une tâche, d'être absolument guidé.

C'est malheureusement ce qu'il faudrait mettre en place, sinon on part dans tous les sens, même si on est autonome.

```

<div class="head-bou1">
  <div class="main">
    <nav>
      <ul class="sf-menu">
        <li><a href="#">Home</a>
          <ul>
            <li><a href="resources_morrrffff.html">About us</a></li>
            <li><a href="resources_morrrffff.html">Archive</a></li>
            <li><a href="resources_morrrffff.html">FAQs</a></li>
          </ul>
        </li>
        <li><a href="#">Presentation_du_projet</a></li>
        <li><a href="#">ressources_morrrffff.html>Our network</a></li>
        <li><a href="blog_ligh(fehjfbhj.html">Report</a></li>
        <li><a href="#">our contact-modifvv.html>Our network</a></li>
      </ul>
    </nav>
    <div class="clear"></div>
  </div>
</div>
<div class="main">
  <h1><a href="resources_morrrffff.html">Logo</a></h1>
  <a href="resource_morrrffff.html"></a>
  <div class="clear"></div>
</div>
<!--*****content*****-->
<section id="content">
  <div class="main">
    <div class="contenu_12">
      <div class="expose">
        <div class="grid_9">
          <article class="p1">
            <h2>Blog</h2>
            <div id="page4">

              <div class="page5-hed">
                <figure class="page5-img2"></figure>
                <div class="extra-wrap">
                  <span class="text1 p17">Post-apocalyptic</span>
                  <span class="date prev-indent-bot">March 17, 2024</span>
                  <span class="comment">>2</span>
                </div class="extra-wrap">
                <div class="link3 p17">
                  <a href="https://github.com/ATSAAG/TesteGameCore">Vous pouvez trouver mon projet sur GitHub :</a>
                  <a href="https://github.com/ATSAAG/TesteGameCore">Lien vers le dépôt GitHub:</a>
                  <span>Read more</span>
                </div>
              </div>
            </div>
          </article>
        </div>
      </div>
    </div>
  </div>
</section>

```

FIGURE 1.2 – Exemple du code

Donc travailler tout seul est bien plus difficile que ce qu'on voulait croire même si on ressentait une liberté totale de la forme de notre travail.

Et malheureusement je n'avais pas assez de bases en HTML et CSS, ce qui pouvait facilement pouvoir avancer plus vite et de comprendre bien plus facilement les erreurs que j'aurais pu ne pas faire si bien que je suis tombé dans beaucoup de ces erreurs , de même en réfléchissant plutôt par rapport à une vue d'ensemble.

Et c'est également en pratiquant qu'on trouve son style, c'est à dire généralement quel genre de site web créé, quelles actions mettre en place et lesquelles éviter.

1.2 Crée un grand ennemi "boss" après mars

Comme dans tout jeu vidéo, les joueurs ne doivent pas trop facilement le finir, sinon cela ne servirait à rien d'y jouer à part le finir plus rapidement que d'autres personnes. Mais ce n'est pas ce qui va permettre d'attirer beaucoup de joueurs Et les joueurs sont souvent plus attirés par des jeux avec beaucoup de défis. Vous voulez en essayer un, celui que beaucoup de joueurs ont hâte d'essayer, eh bien le voici !

En plus c'est nous qui l'avons crée, et il n'est heureusement ni trop dif-



FIGURE 1.3 – Exemple du boss

ficle à vaincre, déjà qu'on a plein de bug, notamment quand on rajoute de plus en plus de parties de code et de synchronisation multijoueur dans notre projet, donc il fallait qu'on se limite. Et heureusement ce jeu n'est non plus trop facile, sauf si on a trouvé un hack dans le jeu que vous pourrez découvrir si vous remarquer des détails dans le gameplay, mais je ne vais pas vous le dévoiler

Ce défi est en effet le combat entre 1, non même 2 joueurs contre un monstre et ce monstre n'est pas n'importe lequel. Il ne s'agit pas simplement d'un petit monstre qui ne fait que se balader en se déplaçant de gauche à droite d'une plateforme.

Non, d'abord sa taille imposante fait comprendre au joueur qu'il est là, prêt à attaquer les joueurs lorsque qu'au moins l'un de ces derniers le voit dans leur caméra.

En plus de cela le boss se trouve entre 2 plateformes, 1 au-dessus de lui et une autre en dessous de lui, en laissant juste assez de place pour que le joueur puisse passer entre le boss et la plateforme au-dessus du boss. Et ce chemin possible est d'ailleurs le seul possible pour que le joueur puisse accéder à la



FIGURE 1.4 – avant juin : animation d'attaque du boss

suite du parcours, après le niveau 4.

Comme ça il n'y a aucune excuse pour que le seul ne soit pas confronté à un défi. Et en plus en étant 2 joueurs, c'est encore mieux car il existe plus de bonnes combinaisons de jeu pour avancer malgré cet obstacle pour au moins survivre et également en se prenant le moins de dégâts possible, et donc accomplir un défi de plus.

Le boss attaque les joueurs par une rangée de 3 projectiles superposés à l'horizontale tirés en même temps vers sa droite ou sa gauche et qui les forcent à réagir en se déplaçant à droite ou à gauche et également en sautant car sinon ils vont rapidement mourir.

En effet, en se faisant toucher par 3 projectiles du boss un joueur meurt automatiquement, ou même moins de 3 projectiles si il a déjà perdu des points de vie. Et cette confirmation de missiles lancés permet même de toucher le joueur lorsque celui ci effectué n'importe quel type de saut ou de déplacement, comme ça même un double saut ne permet pas toujours aux joueurs d'échapper au boss, et ils ne peuvent en plus pas passer en dessous de ce dernier.

Mais le boss peut mourir s'il se prend 8 attaques à distance de joueur. Mais

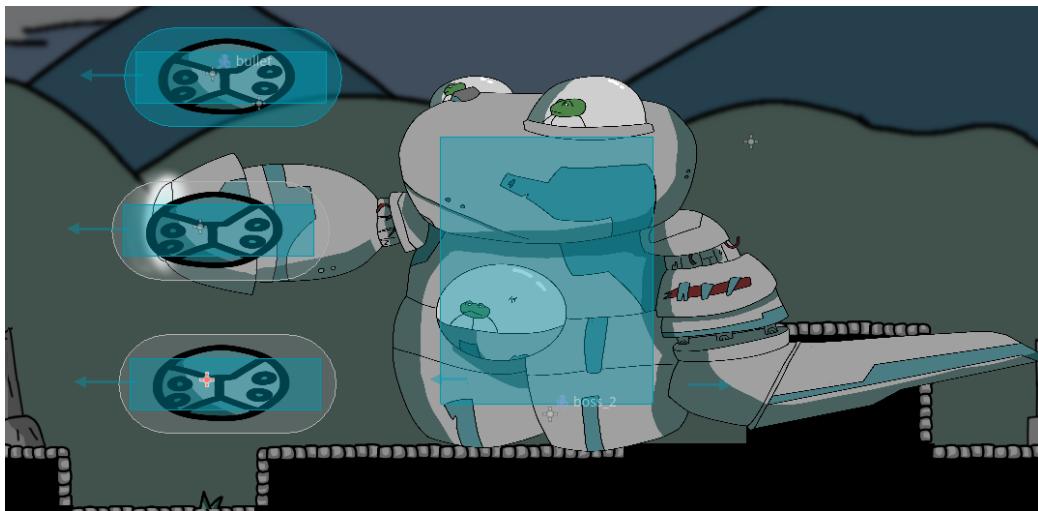


FIGURE 1.5 – Voilà comment j'ai édité cela dans le jeu

même cela n'est pas facile car si le boss prend des dégâts, le joueur doit être suffisamment proche du boss tout en esquivant ses attaques de projectiles, et ces projectiles ont aussi un effet repousoir qui permet d'éloigner les joueurs de lui. Et plus les joueurs sont éloignés du boss et si le boss n'est pas loin d'eux, alors les projectiles iront plus vite qu'en étant à côté du boss et ces projectiles se détruisent s'ils touchent un mur.

Cependant c'était loin d'être facile et évident, il fallait trouver les bonnes ressources, mais le gros problème est qu'il y en a aucune en Godot avec C# et la documentation officielle de Godot est très brève et en autre langage, Godot script , beaucoup ne sont pas à jour donc c'est très dur de changer le script tout en ayant le résultat attendu. J'ai voulu aller plus loin et créer un projectile à tête chercheuse qui se déplace sans arrêt vers la position d'un joueur mais après de nombreuses heures de recherche et d'implémentation dans mon script, même en essayant d'utiliser un timer pour avoir des actions cohérentes et millimétrées, je n'ai pas pu créer ce déplacement malgré le fait d'avoir un script cohérent. J'ai donc abandonné cela pour me concentrer sur quelque chose qui est sûr de fonctionner de plus facile, c'est-à-dire le lancement de 3 projectiles superposés et qui est fun à réaliser pour un débutant comme moi. Et en plus ça s'inspire de la mécanique de déplacement du joueur.

```

public partial class bullet : CharacterBody2D
{
    private RayCast2D[] _rayCasts;
    private float speed = 100.0f;
    public int Direction = -1; // aller à gauche
    //public int AutreDirection = 1; // aller à droite
    private bool Launched = false;
    public player player;
    ...

    // player
    // public Camera2D camera = new Camera2D();
    // AddChild(camera);
    ...

    // multijoueur
    // var scene = ResourceLoader.Load<PackedScene>("res://world.tscn").Instantiate<Node2D>();
    // GetTree().Root.AddChild(scene);
    ...

    public override void _Ready()
    {
        GD.Print("Missile creeee");
        _rayCasts = new RayCast2D[1];
        _rayCasts[0] = GetNode<RayCast2D>("RayCast2D");

        boss_2 boss_testee = GetParent().GetNode("World").GetNode("boss_2");
        GlobalPosition = new Vector2(boss_testee.GlobalPosition.X, boss_testee.GlobalPosition.Y - 65);
    }

    public override void _PhysicsProcess(double delta)
    {
        //base._PhysicsProcess(delta);
        Vector2 velocity = Velocity;
        velocity.X += speed * Direction * (float)delta;
        Velocity = velocity;
        //GD.Print("missile move" + velocity);
        MoveAndSlide();
    }
}

```

FIGURE 1.6 – voilà comment le missile se déplace dans Physics Process

```

>| base._PhysicsProcess(delta);
>| if (isAttacking)
>| {
>| >| if (!isMissileLaunched)
>| >| {
>| >| >| _sprite.Play("shoot");
>| >| >| //GD.Print("Boss tire à gauche");|| ->
>| >| >| var bulletScene = (PackedScene)ResourceLoader.Load("res://Enemies/bullet.tscn");
>| >| >| var bullet2Scene = (PackedScene)ResourceLoader.Load("res://Enemies/bullet_2.tscn");
>| >| >| var bullet3Scene = (PackedScene)ResourceLoader.Load("res://Enemies/bullet_3.tscn");

>| >| >| if (bulletScene != null && bullet2Scene != null && bullet3Scene != null)
>| >| >{
>| >| >| >| bullet Bullet1 = (bullet)bulletScene.Instantiate();
>| >| >| >| GD.Print("missile 1 init");
>| >| >| >| bullet_2 Bullet2 = (bullet_2)bullet2Scene.Instantiate();
>| >| >| >| GD.Print("missile 2 init");
>| >| >| >| bullet_3 Bullet3 = (bullet_3)bullet3Scene.Instantiate();
>| >| >| >| GD.Print("missile 3 init");

>| >| >| >| Vector2 basePosition = this.GlobalPosition;
>| >| >| >| float spacing = 20.0f; // Space between missiles
>| >| >| >| Bullet1.GlobalPosition = new Vector2(basePosition.X - spacing, basePosition.Y);
>| >| >| >| Bullet2.GlobalPosition = basePosition;
>| >| >| >| Bullet3.GlobalPosition = new Vector2(basePosition.X + spacing, basePosition.Y);
>| >| >| >| GetTree().Root.AddChild(Bullet1);
>| >| >| >| GetTree().Root.AddChild(Bullet2);
>| >| >| >| GetTree().Root.AddChild(Bullet3);

>| >| >| >| int direction = GlobalPosition.X < PlayerPos.X ? 1 : -1;
>| >| >| >| Bullet1.Direction = direction;
>| >| >| >| Bullet2.Direction = direction;
>| >| >| >| Bullet3.Direction = direction;
>| >| >| >| isMissileLaunched = true;
>| >| >| >| player.Health -= 5;

```

FIGURE 1.7 – voilà comment les missiles sont instanciés par le boss ainsi que leur mécanisme d'apparition



FIGURE 1.8 – avant juin : image d'animation quand le boss devait mourir

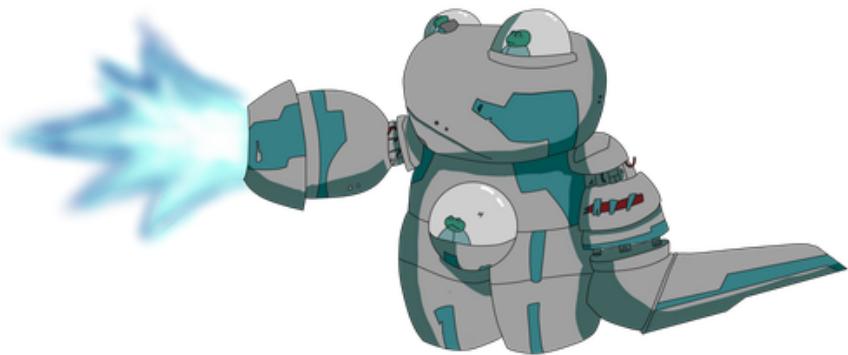


FIGURE 1.9 – En juin : le tir du boss

Théo Rolland

2.1 Création du site web

Le site web est notre vitrine, et dois donner envie. C'est pourquoi j'ai ajouté des icônes, mais surtout des images dès que nous avions les visuels finals de notre jeu.

Pour que notre site web soit plus léger, j'ai implémenter le code de manière à ce que les boutons de téléchargement télécharge le jeu directement depuis Github.

Avoir des boutons vers nos réseaux sociaux sur le site web est un gros point fort : il permet de donner du crédit à notre projet pour les utilisateurs qui se rendent sur le site.

J'ai également ajouté des boutons vers le dossier d'exploitation ou le rapport de projet

Et pour finir, j'ai créé deux versions du site web, une à héberger en ligne qui doit être la plus légère possible, et une contenant tous les différents fichiers en local pour l'envoyer sur l'intranet.

2.2 Mise en ligne du site web sur internet

Je tenais à apprendre de nouvelles compétences lors de la création de ce site web, c'est pourquoi j'ai décidé d'apprendre à utiliser un hébergeur me mettant notre site web en ligne.

J'ai découvert que les hébergeurs gratuits existent mais sont souvent très

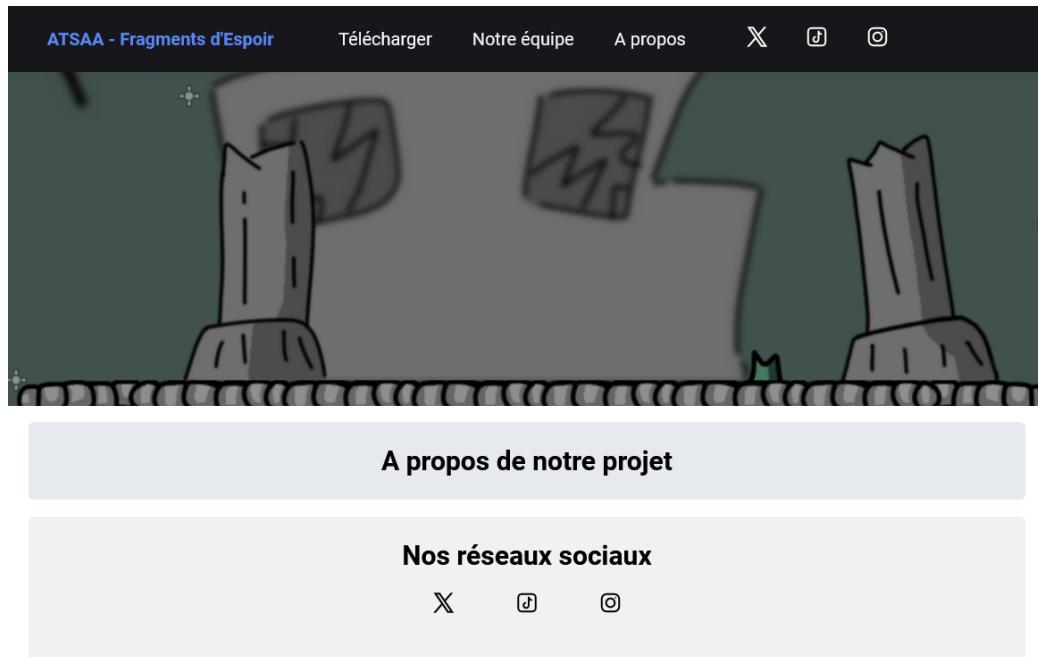


FIGURE 2.1 – Notre site web sur lequel est indiqué notre présence sur les réseaux sociaux

The screenshot shows a Notion workspace with two main pages visible:

- Tâches**: A table page with columns for Nom, Étiquettes, and actions. It contains rows for UI, Site web (which is selected), A fournir, and Tout.
- Site web**: A page with a sidebar containing a checklist of items related to website management. Some items are checked (e.g., "Voir comment importer les fichiers sur le site", "Voir comment ne pas avoir besoin d'héberger le jeu sur le serveur du site web", "push un pdf vide sur le repo github pour pouvoir le télécharger"). Other items are unchecked.

FIGURE 2.2 – Mon espace Notion, simple et concis

peu intutifs voire instable, j'ai donc du regarder de nombreux tutoriels avant de réussir à enfin avoir notre site sur internet.

Notre site web est maintenant hébergé à l'adresse suivante : atsaa.go.yo.fr, et est accessible à tous gratuitement.

Nous avons ajouté un lien vers notre site sur tous nos réseaux, nous permettant d'attirer n'importe quels utilisateurs curieux passant dessus.

2.3 Organistion du projet

Pour être efficace dans la création de notre projet, nous devons avoir des outils d'organisation. C'est pourquoi j'ai selectionné les meilleurs : Notion et Obsidian.

Notion permet de gérer des tâches, des bases de données, et est extrêmement flexible, ce qui permet de parfaitement organiser et planifier l'entiereté du projet. Je l'ai utilisé pour noter et cocher les différentes tâches à effectuer.

Obsidian lui est parfait pour créer des bases des connaissances et garder une trace facilement accessible de son apprentissage. Je l'ai utilisé pour noter

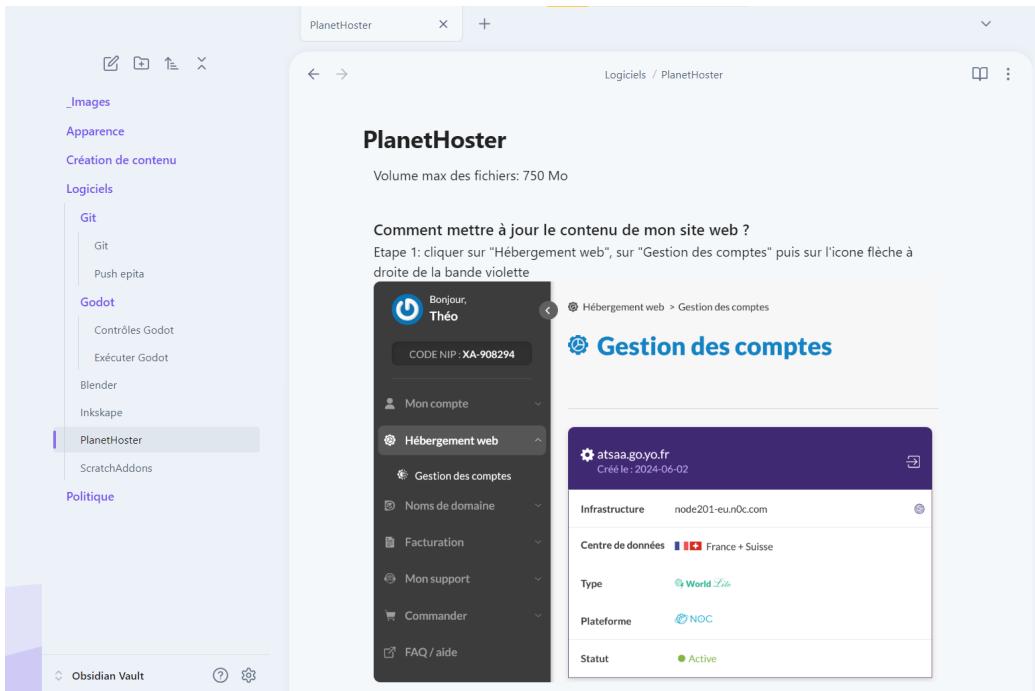


FIGURE 2.3 – Ma base de connaissances sur Obsidian

les tutoriels, commandes, et autres ressources dont j'avais besoin pour réaliser le projet, et il m'a été vraiment utile pour retrouver en quelques clics les éléments dont j'avais souvent besoin.

J'ai également testé whimsical pour créer des cartes mentales et des croquis, mais ais rapidement remarqué que j'étais plus efficace avec un crayon et une feuille de papier.

Le point le plus important est de comprendre qu'un espace "beau" est bien moins efficace qu'un espace simple : passer du temps à ajouter plein de détails, différentes pages et icônes, c'est perdre du temps! Alors oubliez les personnes vous promettant qu'un logiciel peut révolutionner une vie, car se contenter du plus simple est certes moins impressionnant mais bien plus efficace.

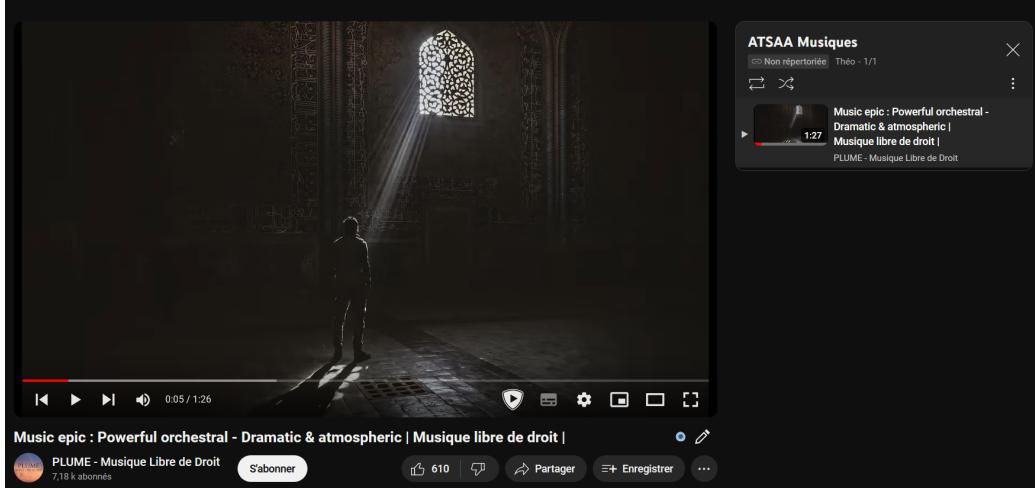


FIGURE 2.4 – La musique de notre projet

2.4 Musique

La musique est un élément qui doit immerger le joueur dans l’ambiance voulue. Je voulais ici un environnement qui transmet de l’énergie, ainsi qu’une touche de tension.

La quantité de musique libre de droits qui existent m’a agréablement surpris, j’ai donc simplement parcouru une playlist, écouté chaque musique, et sélectionné celle qui correspondait le mieux à l’ambiance de notre projet.

La deuxième étape à été d’implémenter cette musique dans le jeu. Cette étape à été bien plus laborieuse, notamment car ça a été le premier élément que j’ai ajouté au projet et que je devais tout apprendre.

J’ai ajouté différentes musiques d’une première manière, avant de me rendre compte que je devais créer un fichier séparé pour pouvoir push le projet sur GitHub sans écraser le travail de mes camarades. J’ai donc du chercher le bon tutoriel - la communauté Godot est relativement petite - et réimplémenter les musiques dans des fichiers séparés.

Cette étape crée vraiment une différence dans le jeu, avoir une musique rend notre projet très agréable à explorer.

2.5 Identité visuelle

2.5.1 Le choix des couleurs

Notre projet étant une aventure dans un monde post-apocalyptique, j'ai choisi une palette de couleur qui reflète ce monde.

J'ai fait le choix d'ajouter de la verdure pour que ce monde post-apocalyptique en soit un dans lequel la nature reprend ses droits. C'est donc tout naturellement qu'a été ajouté un vert sapin à la palette de couleur.

A cela j'ai ajouté de bleu clair, ce qui évoque l'eau pure et claire ainsi que le diamant, permettant d'ajouter de la variété et d'attirer l'attention lorsque cette couleur est placée.

Les couleurs neutres, le noir, le gris, et le blanc ont été beaucoup utilisées pour donner un sentiment de vide et créer une atmosphère de tension. Cela permet également de faire ressortir les éléments bleu clairs.

2.5.2 Le choix des éléments

Chaque élément a été soigneusement réfléchi pour enrichir l'expérience de jeu et correspondre à notre univers post-apocalyptique

Les chats sont souvent considérés comme indépendants, et n'ayant de compte à rendre à personne, je pense donc qu'ils sont parfait dans un monde post-apocalyptique. Les chats sont également populaires auprès du grand public et leur inclusion peut attirer certaines personnes. C'est pourquoi les personnages sont tous des chats.

En contraste avec cela, nous avons ajouté des armes à feu. J'ai choisi cela car je pense que cela permet de rappeler la réalité et la dureté d'un monde après une apocalypses. Je souhaite également que le public ciblé soit relativement agé, et ne voulais pas que notre jeu ait l'air trop enfantin.

2.6 Les réseaux sociaux

J'ai fait la promotion de notre jeu en postant sur les réseaux sociaux, et en reprenant les codes de chacun d'entre eux, c'est-à-dire :

- Du contenu spontané et en grande quantité sur TikTok et Instagram
- Du partage d'opinions tranchées qui fait réagir sur Twitter

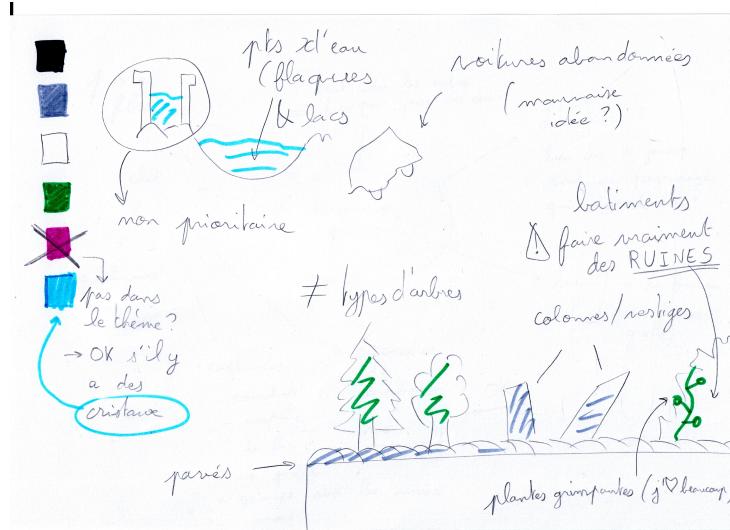


FIGURE 2.5 – Croquis autour de l'identité artistique du projet

— Partage de photos de notre projet sur tous les réseaux sociaux cités ci-dessus.

Le contenu est orienté de manière à rediriger les utilisateurs vers notre site web, par exemple en citant notre site web dans nos posts.

2.7 Dossier d'exploitation

J'ai entièrement écrit le dossier d'exploitation pour guider l'utilisateur dans l'installation. J'ai ajouté une section d'aide qui anticipe certains problèmes que l'utilisateur pourrait rencontrer lors de l'installation.

Ce document contient également une brève section qui explique comment utiliser le projet.

2.8 Rapport de méthodologie

J'ai écrit le rapport de méthodologie. J'y ai détaillé le parcours effectué dans la création de la pochette du jeu, ainsi que notre identité artistique. J'ai également expliqué tous les éléments de communication que nous avons

mis en place, à savoir héberger notre site web en ligne, et être présent sur les réseaux sociaux.

J'ai fait attention à la formulation de chaque phrase de manière à mettre en valeur notre travail.

Sacha Villepreux-Petit

3.1 Mon rôle dans le projet

Pour commencer, je vais vous parler de la partie la plus importante, mon rôle dans le projet. Après la première répartition des rôles, je devais me charger du level design. Le level design consiste à la création de niveaux en assemblant les éléments tel que les animations (ou sprites) et les caractéristiques mécaniques. A la rédaction du cahier des charges nous nous sommes rendu compte que personne ne s'occupait de la mise en place du multijoueur et je me suis porté volontaire pour m'en charger pour plusieurs raisons. Tout d'abord la mise en place du multijoueur est prioritaire car chaque élément mis en place et configuré pour un jeu solo pourrait ne plus fonctionné une fois implémenter en multijoueur donc le multijoueur est la première chose à mettre en place dans un jeu vidéo. De plus le level design est la dernière partie de la création d'un jeu car cette étape nécessite tous les autres éléments, la musique, les effets visuels et sonores, les graphismes, les mécaniques de jeu et les entités non jouables. Bien évidemment, le level design peut se faire en parallèle du développement du jeu si celui-ci se réalise niveau par niveau. Le niveau un peut être crée une fois que le nécessaire à sa création a été crée en amont et pendant que le nécessaire pour le niveau deux est en cours de création, et ainsi de suite.

Cependant devant la masse de travail nécessaire pour la mise en place du multijoueur, je ne me suis pas occupé du level design dans un premier temps et pour la première soutenance je me suis concentré sur le multijoueur.

3.2 Le déroulement de ma partie

Il est important de mettre en évidence les caractéristiques de notre multijoueur. Nous avons choisi de faire un jeu qui se joue avec deux joueurs qui doivent coopérer pour avancer dans une aventure épique. La caméra suit le joueur qui héberge le serveur afin de mettre en avant le travail en équipe et que les deux joueurs ne partent pas chacun de leur côté. La méthode de connexion choisie est LAN (pour Local Area Network) car ce qui correspond le mieux à notre projet. D'abord, un LAN est relativement facile à mettre en place l'ordinateur d'un des joueurs sert d'hébergeur local et il n'y a pas besoin de la mise en place d'un serveur. De plus, dans le cas de deux joueurs la connexion peut s'effectuer seulement avec un câble Ethernet et ne nécessite pas de passer par un routeur wifi. Je dois donc m'occuper de la création d'un serveur sur l'ordinateur qui va héberger la partie, la connexion à ce serveur par un autre joueur via une connexion LAN ainsi que s'assurer de la synchronisation entre les deux joueurs. Un autre avantage du LAN est la proximité forcée des joueurs, si les deux joueurs sont dans la même pièce il n'y a pas besoin de créer un canal de discussion.

La première étape est de découvrir notre moteur de jeu, Godot, et pour cela notre équipe a choisi de chacun coder individuellement un petit projet sur Godot en Csharp. A ce moment là j'étais encore chargé du level design donc j'avais suivi des tutoriels pour apprendre la base de Godot et comment créer un niveau, notamment en utilisant la tilemap(grille où l'on place les éléments de décors afin de modéliser l'aspect d'un niveau). Dès les rôles réassignés je me suis lancé dans l'apprentissage du multijoueur. Après un apprentissage théorique, la création de serveur, la connexion à ce serveur, la synchronisation des joueurs à ce serveur et la différenciation des deux joueurs, je me suis lancé dans le codage.

C'est ici que les premiers problèmes apparaissent. Trouvant peu de documentation en Csharp, je commence le développement en GDsharp. Je me rend vite compte que je perds mon temps et je me lance dans la modification du langage pour passer en Csharp. Après un travail laborieux et de très nombreuses erreurs je me rend compte que j'utilisais la mauvaise documentation, celle de Godot 3.2 au lieu de Godot 4.2. Après des recherches supplémentaires et avoir enfin trouvé la bonne documentation, je décide de recommencer le projet, supprimant près de 80 pourcent de mon code (près de 10h de travail au total) et de lire la majorité de la documentation. Ayant

commencé tôt le projet je n'ai pas pris trop de retard dans la réalisation totale mais une erreur de ma part m'a fait perdre un temps conséquent et le moral était un petit peu en baisse.

La suite fût cependant plus simple, et après de nombreux bug(d'abord l'impossibilité de faire apparaître le deuxième joueur, ensuite les deux joueurs bougeaient avec les mêmes contrôles et enfin le bug le plus important était que les joueurs ne pouvaient voir ce que l'autre joueur faisait) il était enfin possible de lancer le jeu avec deux joueurs sur le même ordi en ouvrant deux fenêtres. Cependant cela était loin d'être suffisant pour une version jouable et ne servait que pour le développement des autres éléments et pour les tests. En effet sur le même ordinateur il n'était possible que de jouer sur une fenêtre à la fois. Avant d'implémenter un réseau local (LAN) il était nécessaire de régler un problème majeur, les animations d'un personnage n'était vu que par le joueur qui le contrôlait et non par l'autre. Grâce au MultiplayerSyncronizer déjà intégré dans Godot et à des fonctions utilisant le RPC (Remote Procedure Call) il a été possible de récupérer en temps réel les informations d'un joueur et de les renvoyer à l'autre en passant par le serveur.

J'ai ensuite implémenter le réseau local de façon à ce que dans un même réseau (concrètement deux ordinateur connectés par un câble Ethernet) chaque joueur crée un serveur en appuyant sur le bouton « host » et l'autre joueur puisse le rejoindre via le bouton « join ». Il ne reste plus qu'à appuyer sur « start » et le jeu fonctionne. Les deux personnages ne pouvant être différenciés par leur apparence, j'ai rajouté la possibilité de rentrer un pseudonyme au début de la partie et qui reste visible tout du long. Après un temps total avoisinant les 40 heures, ma partie étant très ralenti par les nombreux bugs et par ma méconnaissance du sujet, je suis capable de soumettre un multijoueur fonctionnel.

L'utilisation de gitlab fût aussi laborieuse lorsque que sur certaine mise en commun certaine partie de code ne fonctionnait plus.

Le travail de rédaction pour ce rapport est aussi une nouveauté et je me suis rendu compte d'une erreur importante dans sa conception. En effet je rédige ce texte une fois mon code terminé et devant les nombreux oubli et imprécisions je me rend compte que j'aurai dû (et je ferai donc pour la prochaine soutenance) un journal faisant état régulièrement de l'avancement du projet. Et ce n'est pas la seule chose qu'il me reste à faire.

3.3 Mes attentes pour la suite du projet et vision globale

Tout d'abord j'aimerai un menu plus esthétique pour notre jeu car je me suis uniquement concentré sur la partie fonctionnelle et non celle esthétique, cependant ce n'est pas moi qui m'en chargerai. Ensuite je souhaiterai faire en sorte que le jeu soit jouable en peer-to-peer, bien que ce ne soit pas la priorité une connexion permettrait de joueur sans la contrainte de proximité des deux joueurs et permet de viser un plus grand public.

Je n'aurai alors en théorie plus besoin de travailler sur le multijoueur et enfin m'atteler à mon premier rôle, le level design. Il est encore difficile de se projeter mais j'ai beaucoup d'espoir pour ce projet. J'ai été depuis le début très intéressé et motivé par ce projet et je suis reconnaissant envers les membres de mon groupe qui me permettent, qui nous permettent d'aller aussi loin dans ce projet qui est loin d'être fini.

3.4 Quelques problèmes

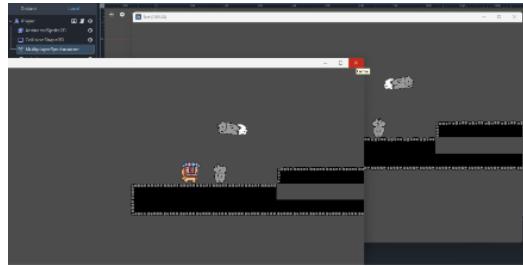


FIGURE 3.1 – Enter Caption

FIGURE 3.2 – Problème de synchronisations des animations

[H] La solution était simple mais pas intuitive, le changement de coté de l'animation(droite ou gauche) dépend du signe de la vitesse. Hors au début seulement les postions et les animations était partagées par le Multiplayer-Syncronizer.

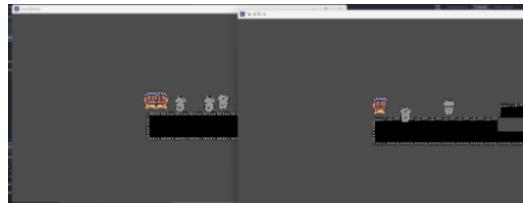


FIGURE 3.3 – au début tout les boutons join menaient au même serveur

Liée à une mauvaise utilisation des delegates, le problème s'est résolu grâce à des forums et tutoriels.

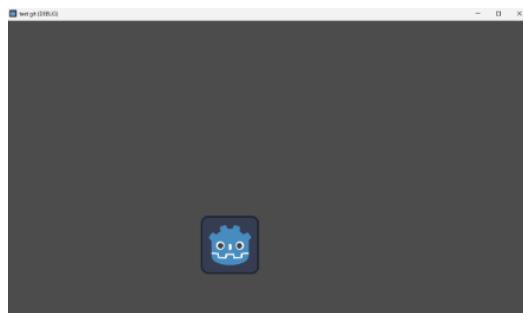


FIGURE 3.4 – Perte du projet

Petite frayeur après une mauvaise update du projet via git. Il m'a fallu supprimer mes fichiers et clone à nouveau le projet.

3.5 Compte rendu après la première soutenance et mise au point de mi-projet

Lors de la première soutenance, suite à un git pull qui effectué le matin même, le fichier qui gérait la connexion LAN entre les deux joueurs a disparu. Nous nous sommes donc retrouvés sans l'un des fichiers les plus importants qui la veille marchait parfaitement. Heureusement ce fichier était sur mon ordi personnel, j'ai donc pu le récupérer et le remettre sur notre repository sans problème mais malheureusement il était trop tard pour la soutenance. Lors de la première nous n'avons pu que montrer très peu de chose mais cette échec nous a préparé pour la suite.

3.6 Ce qui a été fait

Pour rappel, pour l'instant le multijoueur consiste à la connexion entre deux joueur sur le pc de l'un des deux qui héberge la partie. Les actions et les animations des deux joueurs sont synchronisées entre les deux joueurs, qui voit donc la même chose. Mais le niveau en lui-même et les ennemis ne sont pas synchronisés.

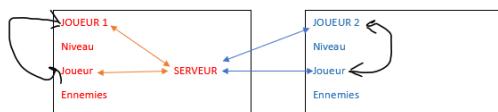


FIGURE 3.5 – Etat du multijoueur au moment de la première soutenance

Alors que l'objectif est le suivant

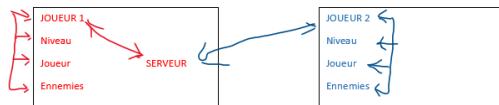


FIGURE 3.6 – Objectif

Mais avant de s'occuper de ça il fallait s'occuper de la connexion LAN entre les deux joueurs. LAN est l'acronyme de local area network, soit réseau local en français. Concrètement c'est un réseau d'appareils se trouvant dans le même espace physique qui communiquent entre eux soit par câble Ethernet soit par connexion Wifi.

La connexion Wifi est plus compliqué à implémenter mais garantit un confort supplémentaire pour les joueurs. Tout le monde n'as pas de câble Ethernet facile d'accès voir même de port Ethernet sur leur PC. Et le point le plus important, la connexion Wifi ne demande aucun effort de mise en place, d'installation ou de configuration. Le choix d'un LAN plutôt qu'une connexion peer-to-peer (P2P) est à la fois pratique et utile.

Du coté des utilisateurs, nous pensons que la proximité physique des joueurs permet d'augmenter la qualité de l'aventure ainsi l'intensité du moment vécu. Ceci est une inspiration des jeux It-Take-Two et A-Way-Out qui, bien que

pouvant se jouer à distance (ce qui brise l'immersion), permet de renforcer ses liens avec un proche en vivant une aventure unique.

Du coté développeur, c'est plus simple à implémenter et cela permet de ne pas soucier de fonctionnalités supplémentaire comme un chat vocal pour pouvoir communiquer directement depuis le jeu(en partant du principes que les joueurs jouant sur un même réseau jouent dans la même pièce.).

La connexion en LAN sur réseau Wifi se fait grâce au protocole UPNP (Universal Plug and Play). Le protocole UPNP permet de gérer l'ouverture et la fermeture des ports qui permettent la connexion entre un appareil et un réseau ou à d'autres appareils. Cependant, une fois un port ouvert grâce au protocole UPNP, il est accessible par tout les appareils présent sur le réseau. Dans le cas d'un réseau LAN c'est exactement ce que l'on souhaite et cela nous facilite beaucoup la vie mais un utilisateur mal attentionné aurait accès sans aucun problème à tout les appareils présents sur le réseau. D'où l'importance de n'utiliser le jeu que sur un réseau de confiance, le plus souvent un réseau domestique.

Une fois ceci réalisé l'étape suivante a été de synchroniser les ennemis vis-à-vis des joueurs. Cela a été beaucoup plus compliqué que de faire communiquer les joueurs entre eux car, pour les joueurs, le « MultiplayerSynchronizer» permet de sélectionner dans Godot les éléments qui doivent être partagé entre les deux joueurs comme les animations ou la vitesse. Or pour les ennemis il faut partager manuellement chaque propriété et chaque fonction appelé, que ce soit par le joueur ou par une entité, doit être réfléchies à l'avance pour savoir si elle nécessite Appel de Procédure à Distance (RPC en anglais) ou non, au risque d'entrainer de très nombreux bug.

J'ai finalement réussi à synchroniser tout les ennemis (car descendant tous de la même Class qui s'occupe de gérer la synchronisation). J'ai ensuite dû œuvrer un petit peu sur tous les plans, car en tant que responsable du multijoueur, je dois m'assurer que chaque nouvelle fonctionnalité fonctionne correctement pour les deux joueurs .Heureusement l'héritage de Class me permet de prévoir en amont de nombreux cas de figure.

Je me suis ensuite lancé dans la création d'un nouveau muni simplifiant l'expérience utilisateur et étant plus intuitif, ainsi que dans la création d'un système de sauvegarde.

3.7 Un ajout trop ambitieux

Ces choix sont plutôt cohérent, le level design, ma tâche initiale avait été relégué à un autre membre du groupe et j'avais quasiment terminé le multijoueur. Je me suis donc penché sur des ajouts ou modifications nécessaires pour notre jeu.

Un menu intuitif est la chose la plus importante dans un jeu mais ce fût aussi le début des complications. La façon dont je voulais arranger le menu permettait pas un multijoueur en LAN car certain appel de procédure de connections doivent être appelé à des moments précis et je n'ai pas réussi à récréer la configuration nécessaire pour le nouveau menu.

Pour le sauvegarde j'ai implémenter une fonction qui sauvegarde les données des joueurs comme leur position ou leur inventaire. Une fonction qui charge n'importe quelle scène et peut charger cette scène avec une sauvegarde déterminée. J'ai aussi modifié certaines propriétés des joueurs et des ennemis afin que les joueurs prennent des dégâts, puissent mourir et réapparaître à des checkpoints. J'ai fait en sorte que lorsque un joueur passe sur un checkpoint, la partie se sauvegarde et le checkpoint devient le nouveau point d'apparition pour les joueurs. Lorsque un joueurs meurt, les deux joueurs réapparaissent au dernier checkpoint, récupèrent entièrement leur vie mais perdent toute progression dans le dernier niveau, les objets récupérés sont remis à leur place et les ennemis réapparaissent.

Mais après de nombreux test, je me rend compte que la fonction la plus importante, celle pour charger les niveaux, ne fonctionne que en solo. En multijoueur, chaque appel à cette fonction déconnecte le deuxième joueurs. Je n'ai pas trouvé de solution et dû retirer de nombreuses fonctionnalités. Sur environ 30h de travail, il ne reste que la mort des joueurs et les checkpoints (qui ne renasalisent plus les niveaux).



FIGURE 3.7 – Lors de l'appel à la fonction Load

De plus la fonction Load devait aussi permettre de charger les différents niveaux, j'ai donc dû regrouper tous les niveaux dans une même scène et m'occuper des téléporteurs invisibles qui permettent de voyager entre les niveaux. Ceci ne fût pas difficile mais très laborieux et a aussi nécessité un temps conséquent.

3.8 Quelques problèmes 2



FIGURE 3.8 – Le checkpoint copiait les joueurs à chaque téléportation



FIGURE 3.9 – Téléportation au mauvais endroit

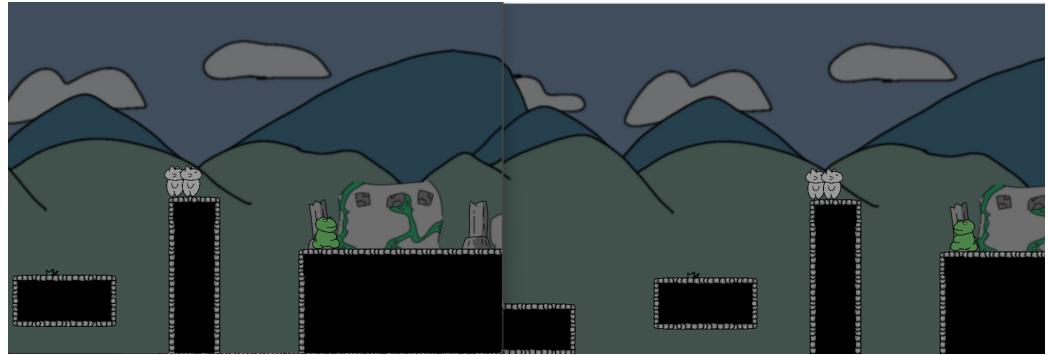


FIGURE 3.10 – Ennemis pas encore synchronisé

Arsène Hiriart

Lors de ce projet, j'ai été "responsable du gameplay". Cela signifie que je me suis occupé de rendre le jeu jouable. Au cours du projet, j'ai réalisé les bases du monde, le joueur avec ses déplacements et son système de combat, les ennemis et leurs comportements, ainsi que la structure des niveaux.

4.1 Les déplacements

Nous avons choisi de réaliser un "metroidvania", un genre de jeu basé sur de l'exploration dans laquelle le joueur se retrouve très libre. Dans un metroidvania, le joueur explore un monde dont il ne sait rien, et se retrouve donc régulièrement perdu et contraint de faire demi-tour. Ce sont ces demi-tours qui forcent un bon metroidvania à être agréable à parcourir, à plusieurs reprises. Le système des déplacements a donc été une priorité. Dans notre jeu, le joueur possède donc une large variété de mouvements, qui lui seront utiles tout au long de son aventure.

Pour commencer, le joueur possède une vitesse de déplacement élevée, au sol comme dans les airs. Lorsque le joueur saute, il peut accélérer sa chute, en tentant de se déplacer vers le bas. Cette accélération peut permettre de gagner simplement de la vitesse, mais aussi d'esquiver des attaques ennemis. Toujours dans les airs, le joueur peut aussi utiliser un saut aérien, lui donnant une seconde impulsion. Ce saut aérien devient ensuite inutilisable jusqu'à ce que le joueur touche le sol, ou qu'il s'agrippe à un mur. Une fois contre un mur, le joueur peut s'y laisser glisser, l'utiliser pour rebondir, en sautant depuis le mur, ou bien l'escalader. Pour finir, le joueur possède aussi un "dash", où le joueur se jette en avant pour gagner de la vitesse et de la distance. Tout comme le saut aérien, le dash ne peut s'utiliser qu'une seule fois dans les

airs. Le joueur devra ensuite attendre de toucher le sol ou un mur pour le récupérer. Si utilisé au sol, le dash aura le même effet, et sera réutilisable après un court délai.

Afin de se défendre, le joueur possède un pistolet, qu'il peut utiliser à sa guise. Il infligera des dégâts aux ennemis situés dans la zone de l'explosion. Mais ce pistolet ne sert pas qu'à se battre, en effet, si le joueur tire dans les airs, l'explosion le poussera légèrement en arrière. Cet effet de recul peut permettre de créer une distance avec l'ennemi. De plus, si le joueur décide de tirer derrière lui, le recul lui permettra donc d'avancer. Cette mécanique peut permettre de gagner de la vitesse, voir de réaliser des sauts jusqu'à lors irréalisables.

4.2 Les ennemis

Je me suis occupé de réaliser plusieurs types d'ennemis, afin que le joueur soit forcé d'utiliser ses différentes compétences pour en venir à bout. L'ennemi basique se contente de se déplacer de droite à gauche, en évitant les murs et les trous. Le joueur apprend donc à se servir de son arme, ainsi qu'à sauter au bons moments pour éviter cet ennemi. Mais lorsque cet ennemi possède lui aussi une arme, le combat devient alors plus complexe. Armé d'un sniper, le joueur ne peut plus lui faire face, il devra alors trouver un moyen d'éviter cet ennemi ou de le prendre à revers.

4.3 Les niveaux

Dans un metroidvania, le terrain de jeu est généralement grand, afin de donner la possibilité au joueur de se déplacer librement, et de parcourir le monde comme il le souhaite. Seulement au vu du temps limité que nous possédions, créer une grande carte n'était pas envisageable. Le joueur aura tout de même accès à plusieurs zones, connectées les unes aux autres de manière logique, et lui permettant tout de même d'explorer. Avec les niveaux que j'ai réalisé, j'ai essayé de présenter au joueur le cheminement classique de ses découvertes dans le jeu. Il débute dans une zone simple et explicative, afin de comprendre les règles de ce monde. Il est ensuite mis face à un choix de direction, il peut choisir d'aller affronter un boss directement, ou bien de d'abord explorer la zone. Un nouveau joueur ira logiquement explorer, ce qui

lui permettra d'apprendre diverses utilisations à ses pouvoirs. Il pourra faire demi-tour à tout instant pour l'affronter, ou bien une fois qu'il aura terminé son exploration, il n'aura plus d'autres choix que de prendre son courage à deux mains.

Alessandro Platz

5.1 Mon rôle

Je suis directeur artistique, je suis donc chargé de gérer la partie artistique du jeu. Ainsi j'ai du créé une identité artistique visuel de qualité, qui me tient a coeur et qui convient a mon équipe et j'ai du ensuite créer l'intégralité des éléments visuels du jeu.

5.2 Ce que j'avais fait pour la dernière soutenance

5.2.1 L'identité graphique au jeu

L'idée d'utiliser des chats comme personnages principaux est venu très tôt dans la conception du projet, en prenant en personnage principal un animal mignon comme un chat, nous pouvons traiter des sujets plus sombre tout en gardant un ton plus léger.

Ensuite pour trouver une identité graphique distincte au jeux j'ai tout d'abord regarder les autres jeux similaire au notre et m'inspirer leur style graphique. Je me suis en particulier inspiré de Hollow-Knight et Crossworm.

J'ai ensuite fait plusieurs prototype visuel pour voir a quoi le jeux pourrait potentiellement ressembler une fois terminer, en m'inspirant des jeux du même style mais en restant original et fidèle a la vision de mes camarades.

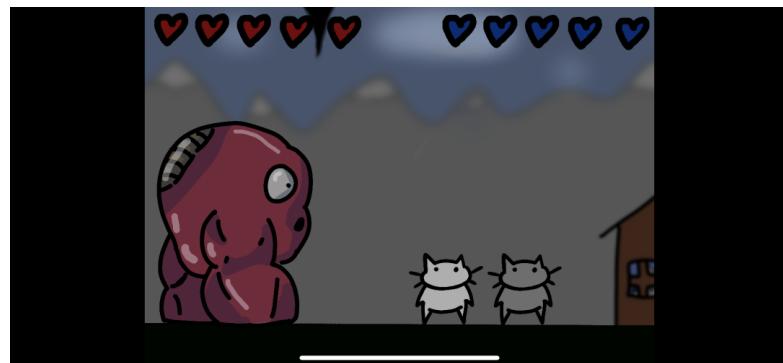


FIGURE 5.1 – 1er prototype visuel



FIGURE 5.2 – 2eme prototype visuel

Aucun élément du premier prototype visuel n'a au final été utilisé dans le projet, mais il m'a permis de me servir de repère quand j'ai créé le reste, et a donné confiance à mes camarades en ma capacité à créer une ambiance visuel, et a instauré une image de ce que le jeu ressemblerait.

Le 2ème prototype visuel a été créé après la conception du début du scénario, et a été créé dans le but d'avoir un appui visuel sur lequel je pouvais me reposer dans la création des éléments visuels restants. J'ai d'ailleurs quasiment recopié la plupart des éléments présents dans ce prototype dans les "tiledsheets".

5.2.2 Les "tiledsheets"

Pour notre jeu-vidéo nous avions fait le choix d'utiliser des "tilemaps" et des "tilesheet" pour la création de nos niveaux. J'ai donc créé des "tilesheet" qui contenait chacun des éléments lié à la création du niveau. Les niveaux se séparent visuellement en 2 catégories : ce que j'appelle le "niveau réel", c'est à dire tout ce qui contribue réellement à la manière dont le jeu se joue, et les décors, c'est à dire tout ce qui contribue uniquement à l'aspect visuel du jeu, mais qui ne doit pas avoir d'impact sur comment on joue. Pour le niveau réel je n'ai que créé une "tilesheet", qui contient les blocs nécessaires à créer une plate-forme. Pour les décors le jeu se sépare en trois plans, j'ai donc créé trois "tiledsheets", une avec de l'herbe qui ne sera donc placé au même niveau que les blocs, des colonnes qui se mettent derrière et des maisons, qui se mettent encore derrière. Rajouté à cela en fond est une image statique, ce qui crée un ensemble complet.

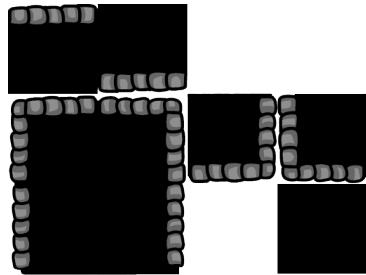


FIGURE 5.3 – la "tilesheet" des blocs

5.2.3 Les Animations

Pour que le jeu prenne vie il est nécessaire que les personnages bougent, j'ai donc été chargé de la création des animations du personnage principal ainsi que celle des ennemis. Pour faire cela j'ai dans mon logiciel de dessin, tout d'abord dessiné la première image d'une animation, puis redessiner par dessus la deuxième. J'importe ensuite mes images dans Godot. Ayant un style plutôt simple le jeu ne contient pas d'animation avec plus de deux images, mais j'ai tout de même fait une quinzaine d'image pour le chat, du au grand nombre de capacités différentes que le personnage principale a.

5.2.4 Ce qu'il reste à faire

Même si je suis plutôt satisfait du résultat esthétique du jeu il reste tout de même du travail. En effet j'aimerais rajouter :

- un effet de parallaxe, c'est un effet de perspective qui fait que plus un objet est loin moins on le voit se déplacer vite
- plus de variété de décors
- plus de variété de blocs
- des ennemis plus complexe visuellement, notamment le boss, qui aura bien plus d'attaque et d'effet visuels que les autres ennemis.

5.3 Le parallaxe

L'effet de parallaxe dans un jeu vidéo est une technique visuelle utilisée pour créer une illusion de profondeur en faisant déplacer les objets à des vitesses différentes en fonction de leur distance par rapport à la caméra ou au joueur. Cette méthode est couramment employée dans les jeux en 2D pour simuler un environnement tridimensionnel et rendre les scènes plus immersives et réalistes. En termes simples, la parallaxe est la différence apparente de position d'un objet lorsqu'il est observé depuis deux points de vue différents. Cet effet est très perceptible lorsqu'on bouge la tête d'un côté à l'autre ; les objets proches semblent se déplacer plus rapidement que les objets éloignés.

Il y a deux façons populaires d'atteindre cet effet. En utilisant une caméra 3D, on a la possibilité d'obtenir un effet de profondeur où je peux placer chaque élément derrière les autres, et le parallaxe est donc réel. C'est ce que fait, par exemple, Hollow Knight, un jeu dont je me suis grandement inspiré. Une autre façon est de simplement déplacer l'élément avec la caméra, à une vitesse variable dépendant de la "distance" entre l'objet et la caméra.

J'ai choisi la deuxième méthode car Godot possède déjà un objet parallax qui facilite grandement la tâche. Grâce à cela, j'ai pu facilement et rapidement implémenter mon parallax, qui consiste en trois images superposées l'une sur l'autre. Les images sont répétées pour donner l'impression que l'arrière-plan est infini. Néanmoins, cette implémentation force la résolution du jeu à être constante, car l'image doit être répétée en fonction de sa taille. Cela peut donc poser problème lorsque l'on souhaite passer en plein écran.

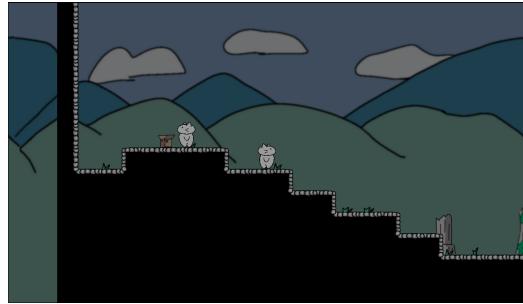


FIGURE 5.4 – pov joueur 1

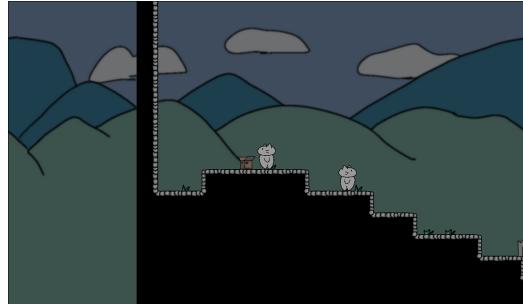


FIGURE 5.5 – pov joueur 2

Nous pouvons donc ici voir l'effet du parallax en action : on constate une différence dans l'arrière-plan entre le joueur 1 et le joueur 2, et l'arrière-plan ne se déplace pas autant que le personnage.

5.4 L'amélioration du tilesheet

Le tilesheet que j'utilisais auparavant et que j'ai montré lors de la dernière soutenance fonctionnait correctement mais avait quelques soucis : tout d'abord, toutes les tuiles ne se connectaient pas bien entre elles. De plus, les tuiles n'étaient pas de la bonne taille et étaient trop petites, ce qui donnait l'impression que les joueurs flottaient au-dessus des blocs.

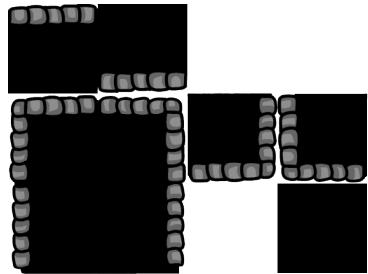


FIGURE 5.6 – L'ancienne tilesheet

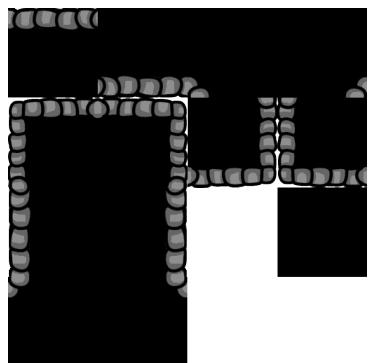


FIGURE 5.7 – La nouvelle tilesheet



FIGURE 5.8 – Exemple de ce qu'Arsenme me fournissait



FIGURE 5.9 – Exemple de resultat decoratif

5.5 Decoration des niveau

En tant que directeur artistique, mon rôle consistait à enrichir chaque niveau du jeu en y ajoutant des décorations soigneusement choisies pour renforcer et amplifier son ambiance unique. À partir des fondations créées par Arsenne pour chaque niveau, j'ai travaillé à superposer des éléments décoratifs qui non seulement complétaient le design existant, mais aussi enrichissaient l'expérience visuelle du joueur.

5.6 Le Boss

Notre jeu se conclut avec un boss final, que j'ai donc dû concevoir, dessiner et animer. Pour ce faire, j'ai d'abord commencé par décider de la forme du boss. En effet, en design, il existe une théorie des formes qui étudie comment les formes influencent la perception et l'émotion. Cette théorie utilise la géométrie pour créer équilibre, contraste et harmonie visuelle dans la composition des objets et des espaces. J'ai donc dessiné en noir une forme que le boss pourrait avoir.

Une fois cela fait, j'ai eu l'idée d'un robot qui serait contrôlé par trois



FIGURE 5.10 – forme premiere du boss

grenouilles : une dans chaque œil et une dans le ventre. Non seulement c'est une idée plutôt cocasse pour un ennemi final, mais cela me permettait aussi de revenir aux thèmes du jeu, qui sont la coopération et les conséquences de la technologie sur le monde. Les grenouilles servent ainsi de miroir à cette apocalypse passée et montrent que l'histoire se répète encore et encore. J'ai donc esquissé ce concept puis je l'ai envoyé à mes camarades pour qu'ils me confirment que l'idée était bonne.

Après avoir reçu le feu vert de la part de mon équipe et avoir enlevé les "ailes" du boss, je me suis mis au travail sur les animations du boss. On peut donc le voir de face, ce qui finalement n'a jamais été utilisé, effectuer une attaque de tir, une attaque de mêlée, et marcher.

Au total, ce processus s'est avéré très long en raison de la taille du boss. Étant dix fois plus grand que les autres ennemis, il m'a pris à peu près dix fois plus de temps.

5.7 Les items du jeu

Lorsque le boss meurt, il est censé faire tomber un item qui donnerait au joueur de nouvelles capacités. J'ai donc dû dessiner cet item. Logiquement, chaque capacité déblocable par le joueur, comme le double saut, le "dash" ou le saut de mur, sera déblocable grâce à un objet comme celui-ci. J'en ai donc créé deux autres : un qui est un crâne d'un être au-delà de notre temps, et un autre qui est une sorte de totem utilisé par les grenouilles, portant donc les mêmes couleurs.

De plus le point de sauvegarde du jeu devait être dans le thème des chats, j'ai donc décidé de faire une boîte en carton.

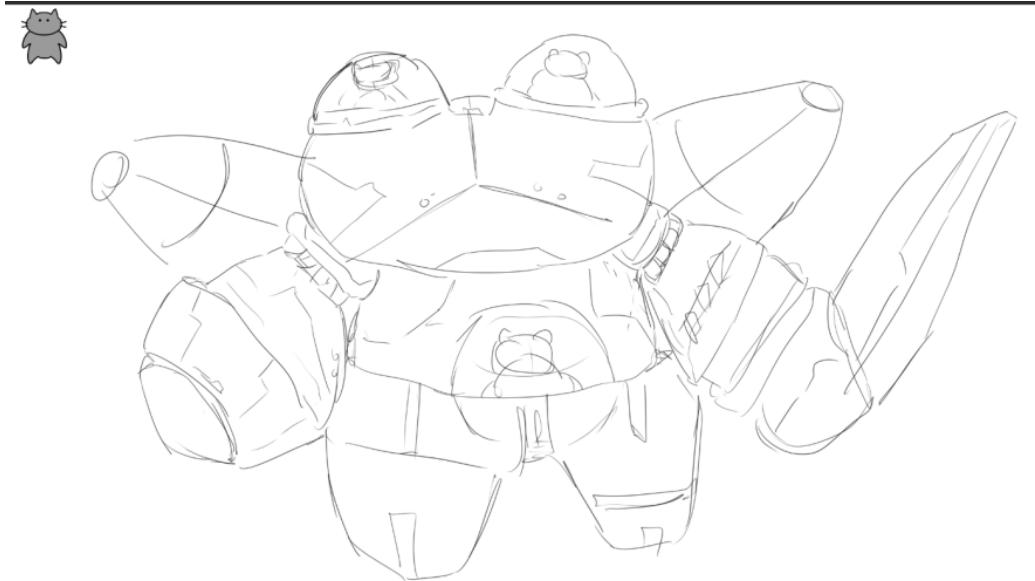


FIGURE 5.11 – premier croquis du bosse

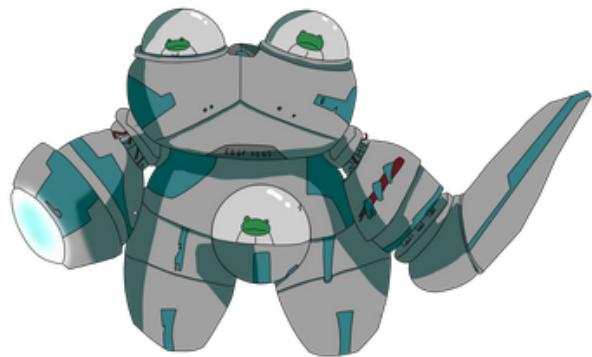


FIGURE 5.12 – le boss de face

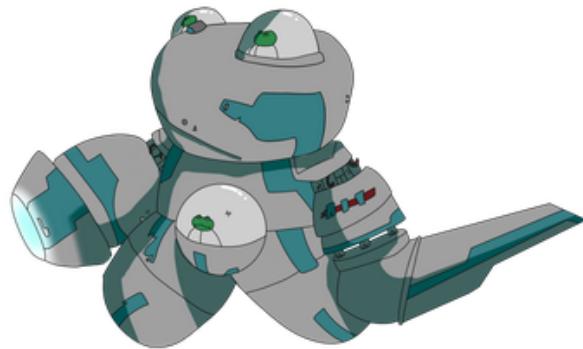


FIGURE 5.13 – une des images du boss qui marche

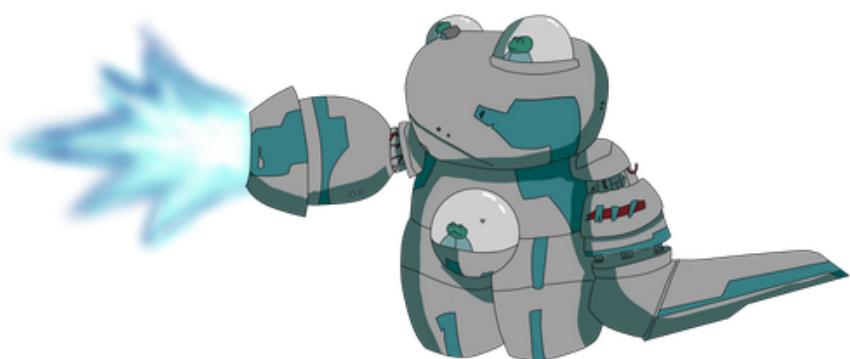


FIGURE 5.14 – Le tir du boss



FIGURE 5.15 – l'objet que fait tomber le boss



FIGURE 5.16 – le crane recuperable



FIGURE 5.17 – le totem des grenouilles

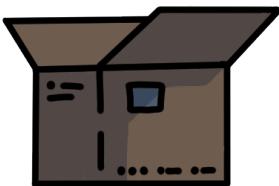


FIGURE 5.18 – le point de sauvegarde

5.8 "Oeuf de Pâque"

Dans un jeu vidéo, un easter egg (Œuf de Pâques en français) est un contenu caché ajouté par les développeurs pour amuser ou surprendre les joueurs. En tant que directeur artistique, j'ai donc pu en cacher quelques-uns. Par exemple, de face, le boss porte un code qui, traduit de l'hexadécimal au code ASCII, donne "Hope", le nom de notre jeu. Un autre exemple est la boîte de sauvegarde. En effet, les traits dessus forment un code morse qui, traduit, donne "ATSAA", le nom de notre entreprise.

Annexe

Les icônes utilisés sur le site web viennent du site <https://icones8.fr/>

Musique de notre jeu : THERE IS NO WAY BACK - Ilya Marfin