



EPITA RENNES
RAPPORT DE PROJET

Raiders Tactics



42 SECONDS TO EPITA

Alex DREAU
Arthur BARREAU

Kévin LUBERT
Romain SEZNEC

8 juin 2023

Table des matières

1	Introduction	3
2	Rappel du projet	4
2.1	Rappel du découpage des tâches	4
2.2	Rappel des objectifs	5
2.2.1	Intelligence artificielle	5
2.2.2	Réseaux et Multijoueur	5
2.2.3	Gameplay	5
2.2.4	VFX et SFX	6
2.2.5	Interfaces	6
2.2.6	Gestion des aspects lié au joueur	7
2.2.7	Gestion des niveaux	7
2.2.8	Site Web	7
3	Méthodes de travail	8
3.1	Logiciels et outils utilisés	8
3.1.1	Github	9
3.1.2	GitLab Epita	10
3.2	Méthodes de travail de l'équipe	11
3.3	Communication dans l'équipe	13
4	État final du projet	14
4.1	Intelligence artificielle	14
4.1.1	Recherche automatique de chemin	14
4.1.2	Unités contrôlées par l'ordinateur	15
4.2	Mode multijoueur et réseaux	16
4.3	Mode campagne	24
4.4	VFX et SFX	26
4.5	Interfaces	27
4.6	Gestion des aspects liée au joueur	28
4.7	Gestion des niveaux	28
4.7.1	Éditeur de niveaux	28
4.7.2	Sélection et intégration des modèles	30
4.8	Rendu Physique	31
5	État final du site web	32
5.1	La <i>landing page</i>	32
5.2	Présentation des membres	33
5.3	Téléchargement des ressources	33
5.4	Multilinguisme et accessibilité	34

6 Discussion	35
6.1 Retour sur l'avancement lors de la première soutenance	35
6.2 Retour sur l'avancement lors de la seconde soutenance	36
6.3 Comparaison avec les objectifs	37
6.3.1 Intelligence artificielle	37
6.3.2 Mode multijoueur et réseau	37
6.3.3 Mode campagne	38
6.3.4 VFX et SFX	38
6.3.5 Interfaces	39
6.3.6 Gestion des aspects liée au joueur	39
6.3.7 Gestion des niveaux	40
6.3.8 Site Web	40
6.4 Améliorations possibles sur les méthodes de travail	41
6.5 Améliorations possibles sur le projet	42
7 Synthèse globale	43
8 Synthèses personnelles	44
8.1 Arthur BARREAU	44
8.2 Alex DREAU	45
8.3 Kévin LUBERT	46
8.4 Romain SEZNEC	47
9 Conclusion	48

1 Introduction

Lors de leurs seconds semestres à Epita, les étudiant du groupe *42 seconds to Epita* ont décidé de réaliser un jeu nommé *Raiders Tactics* comme projet informatique. Le groupe est composé d'Alex DREAU, Arthur BARREAU, Kévin LUBERT et Romain SEZNEC. Ce document a pour but de retracer l'histoire de ce projet.

Raiders Tactics est une combinaison du genre *Stratégie en Temps Réel (Real Time Strategy ou RTS)* et *Jeu de Rôle Tactique (Tactical RPG)*. Le RTS est un genre de jeu multijoueur qui permet à tous les joueurs de jouer simultanément en manipulant plusieurs unités avec des ressources limitées. Le Tactical RPG est un autre genre de jeu qui incorpore des éléments de réflexion et de planification stratégique avec la possibilité de faire évoluer ses personnages sur le long-terme par de l'expérience et des attributs.

Le jeu se déroule dans un thème policier sur différents niveaux faisant office d'arènes avec une vue du dessus sur la carte. Le joueur doit combattre l'équipe adverse en dirigeant une escouade d'un à dix agents sur une carte pouvant varier d'un appartement à un petit quartier selon les missions. Les agents sont indirectement contrôlés par la sélection et le clic de la souris. L'objectif du joueur varie d'un mode de jeu à l'autre allant du match à mort par équipe (Team Deathmatch) au désamorçage de bombe.

Dans ce document, nous rappellerons nos objectifs initiaux comme il était défini dans le cahier des charges. Les rappels reprendront en détails les différentes fonctionnalités qui étaient prévues dans le jeu. Ce rappel sera ordonné par groupe de tâches du même type ayant de forts liens entre elles, les référents et contributeurs de chaque groupe seront également rappelés. Nous reviendrons également sur nos méthodes de travail et sur les logiciels et outils utilisés.

Nous présenterons ensuite le résultat final obtenu suite à un semestre de travail sur le projet. Cette présentation gardera la même structure de groupe de tâche qui fut présenté dans le cahier des charges et utilisé pour le rappel des objectifs.

Dans le reste du document, nous discuterons et comparerons les résultats obtenus avec nos attentes initiales au début du projet. Cette discussion se fera tout d'abord chronologiquement en revenant sur le travail effectuer pour les précédentes soutenances puis par groupe de tâches. Nous terminerons la discussion par les améliorations possibles sur le projet et sur nos méthodes de travail. Une synthèse globale sur le projet et des synthèses de chaque membre du groupe sur leurs ressentis personnelles concluront ce document.

2 Rappel du projet

2.1 Rappel du découpage des tâches

Le découpage et la répartition des tâches se sont faits lors de la rédaction du cahier des charges en janvier 2023. Nous avons attribué les tâches en fonction des qualités et compétences de chacun. Le tableau suivant indique pour chaque groupe de tâche, la personne responsable et les contributeurs.

	Alex Dreau	Arthur Barreau	Kévin Lubert	Romain Seznec
Intelligence Artificielle				
Réseaux et Multijoueur				
Gameplay				
VFX et SFX				
Interfaces				
Gestion des aspects lié au joueur				
Gestion des Cartes				
Site Web				

Chef de pôle
Contributeur

Alex a été désigné chef de projet, il est le plus âgé du groupe et a déjà effectué des travaux d'équipe dans sa précédente formation. Il a notamment travaillé sur l'intelligence artificielle puisque c'est une discipline qui se rapproche des sujets qu'il a déjà abordés auparavant (Théorie des jeux et systèmes dynamiques). Ainsi que les VFX et SFX, il s'est aussi intéressé aux aspects sonores du jeu.

Arthur est assigné au Gameplay, et à la gestion des aspects liés au joueur. Il est intéressé par le game design depuis un moment, c'est pour cela qu'il s'occupe d'implémenter le Gameplay, mix entre RTS et Tactical. Quant à la gestion des aspects liés au joueur, il a une certaine expérience dans le limit testing, et aimerait la mettre à profit pour créer une interaction fluide entre le joueur et le jeu.

Kévin est bénévole en tant que développeur java senior dans l'association *Ekalia* sous le pseudonyme de Popop12. Il a déjà participé à l'élaboration d'événements rassemblant plusieurs centaines de personnes. Il est assigné à la création des niveaux et la gestion des cartes. Il a également participé à la création des interfaces.

Romain s'est occupé de l'implémentation du multijoueur et de la création du site Web, il est intéressé par le fonctionnement du réseau et à ses implémentations. Il a une bonne expérience dans le Web et ses différentes technologies pour faire un site de promotion attractif.

2.2 Rappel des objectifs

2.2.1 Intelligence artificielle

L'intelligence artificielle du jeu doit s'opposer ou s'allier au joueur dans les différents modes de jeu en utilisant des heuristiques simples et des algorithmes plus complexes.

L'IA doit donc pouvoir prendre des décisions entre attaquer, se protéger, se déplacer. Elle doit aussi être capable de se déplacer de manière logique à l'aide d'un algorithme de recherche de chemin couplé à un système d'heuristiques.

2.2.2 Réseaux et Multijoueur

Le cahier des charges prévoyait une implémentation d'un mode libre, jouable en multijoueur jusque 4. Un joueur devait avoir la possibilité d'héberger une partie de jeu et de permettre aux autres joueurs de le rejoindre sous forme de salon avant le chargement d'une carte. Les joueurs devaient également pouvoir rejoindre la partie lorsqu'elle est en cours dans la limite de 4 joueurs maximum. Les joueurs seraient remplacés par l'intelligence Artificielle dans le cas où la partie comporterait moins de 4 joueurs.

Lors de la rédaction du cahier des charges, nous avons prévu une implémentation du multijoueur avec *Mirror* en raison notamment de sa concordance avec notre modèle de joueur hébergeur, de la popularité de l'interface de programmation et de l'accessibilité et de l'extensivité de la documentation.

2.2.3 Gameplay

D'après le cahier des charges, chaque joueur disposera de un à dix agents, en fonction de la mission et/ou de la taille de la carte. Chaque unité sera indépendante et contrôlée à l'aide de la souris en temps réel, certains raccourcis clavier seront aussi disponible pour faciliter les actions du jeu.

La caméra du jeu est en vue du dessus, avec l'implémentation d'un brouillard de guerre empêchant le joueur de connaître tous les aspects du terrain non-visible par agents de son escouade, générant un élément de tension supplémentaire. Les ennemis ne seront visibles que s'ils sont dans le champ de vision d'un agent. La carte de jeu sera quant à elle affichée tout le temps dès qu'un agent aura vu le détail du terrain.

Chaque agent dispose sa propre barre de vie et de son inventaire. Les agents disposeront de deux armes une arme d'épaule et une arme de poing, celles-ci auront une limite de balle par chargeurs, mais pas de limite dans le nombre de chargeurs, car les

agents tireront de manière automatique. Chaque agent, en fonction de son niveau, aura une limite de poids pour son inventaire, ils pourront porter sur eux des grenades et toute sortes d'objet utile à la progression achetable avant les missions.

En Campagne, le joueur suit un scénario et doit effectuer différentes interventions tactiques, au fur et à mesure de son avancement, il gagne des points d'expériences et de l'argent, à investir pour optimiser l'évolution de son personnage vis-à-vis de l'évolution de la difficulté.

En mode multijoueur, plusieurs équipes d'interventions s'affrontent que ce soit dans un match à mort par équipe, sécurisation de « VIP » ou encore le désamorçage d'une bombe. Avant chaque partie, tous les joueurs disposent d'un même nombre d'agents à un niveau similaire et d'un nombre d'argent prédéfini afin de ne laisser aucune inégalité entre les joueurs.

2.2.4 VFX et SFX

Au début du projet nous avons choisi de respecter une charte graphique cartoonesque. Nous avons eu l'intention d'utiliser des éléments disponibles sur Internet et éventuellement des nôtres pour créer un ensemble cohérent.

Pour ce qui est de l'environnement sonore, nous souhaitions ajouter des effets sonores liés aux actions du joueur ainsi qu'un fond musical destiné à créer un élément de tension supplémentaire durant la partie.

2.2.5 Interfaces

Dans les menus, une interface concise et simple permettra au joueur de naviguer de façon fluide entre les différentes missions et différents modes de jeu.

Dans le jeu, nous distinguons plusieurs zones d'interfaces qui communiquent des informations au joueur. La figure 1 est un exemple d'interface en partie, on note dans l'ordre : ① tous les membres de l'escouade qui sont contrôlés par le joueur avec leur statut (vie restante et munitions), les personnages sans vie restante sont marqués *morts* ; ② le/les agents sélectionnés avec l'affichage de leurs statuts et également l'option d'activer certaines actions par le joueur ; ③ le temps restant dans la manche, les scores des deux équipes et l'objectif principal à atteindre ; ④ option d'action/pause en mode campagne pour permettre au joueur de penser stratégique et bouton menu ; ⑤ objectifs secondaires.

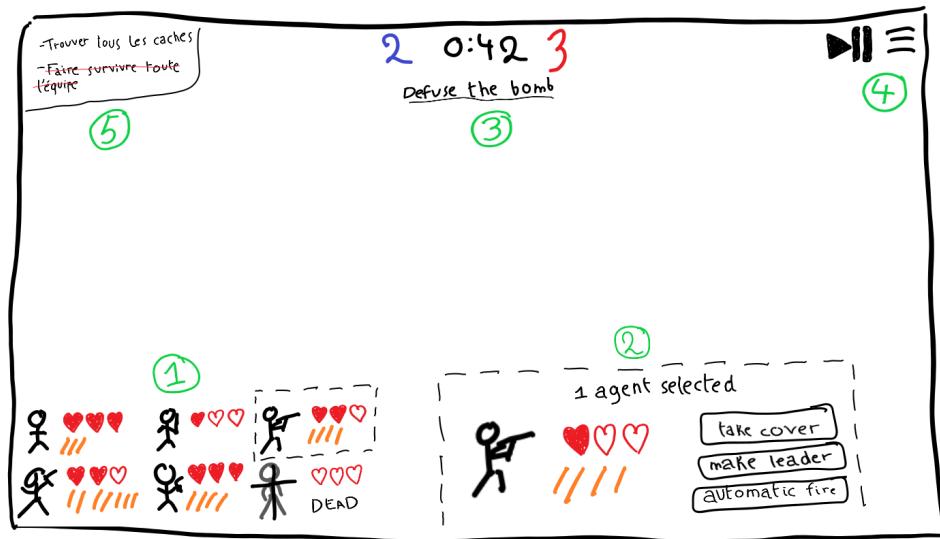


FIGURE 1 – Concept Art de l'interface de jeu

2.2.6 Gestion des aspects lié au joueur

Pour les aspects liés au joueur, il s'agit de permettre au joueur de posséder un niveau et une équipe d'agents. Les agents auront également des niveaux et des statistiques (atouts et contraintes). Le joueur aura accès à une boutique qui lui permettra d'acheter des équipements pour équiper ses agents grâce à l'argent obtenu lors de la réussite de missions.

2.2.7 Gestion des niveaux

Lors de la rédaction du cahier des charges, nous souhaitions implémenter des cartes de jeux de différentes tailles. Elles devront permettre différentes situations donnant intérêt au niveau tactique comme la présence de couloir ou de situations pouvant menacer le joueur. Elles devront aussi permettre la réalisation de certains objectifs liés aux conditions de victoire. En mode campagne les cartes seront conçues par avance. En mode libre certaines maps seront celles de la campagne. Nous souhaitons également ajouter de l'aléatoire pour modifier certains éléments et augmenter la rejouabilité.

2.2.8 Site Web

Un site Web doit présenter l'équipe de développement *42 Seconds to Epita* et permettra la promotion du jeu Raiders Tactics avec un lien de téléchargement et aides d'installations.

3 Méthodes de travail

3.1 Logiciels et outils utilisés

Afin de réaliser les différentes tâches nécessaire à la création et au développement du projet, nous avons dû utiliser de nombreux outils. Nous avons fait le choix de privilégiés des outils gratuit ou dont l'abonnement est déjà compris dans notre inscription à l'Epita.

Unity Ce moteur de jeu multi-plateforme développé par Unity Technologies est l'un des plus répandus dans l'industrie du jeu vidéo. C'est le logiciel que nous avons le plus utilisé. Il nous a permis de créer l'ensemble du jeu.

Mirror Networking Mirror Networking est un paquet Unity pour les jeux en multijoueurs qui permet une reproduction et une synchronisation en temps réel des données entre les appareils.

L^AT_EX L^AT_EX est un langage et un système de composition de documents. Les différents documents que nous devons rendre pour les soutenances ont été intégralement rédigés en L^AT_EX.

Overleaf Overleaf est un éditeur de L^AT_EX en ligne et collaboratif. Il nous a permis de rédiger en groupe tous les rapports en constatant en direct les phrases écrites pas les autres membres du groupe.

Photoshop Ce logiciel de traitement d'image conçu par Adobe nous a permis de retoucher des images, de créer des illustrations et d'effectuer bien d'autres choses concernant le traitement des images.

GIMP *GNU Image Manipulation Program* est un logiciel gratuit d'édition et de retouche d'image. Nous l'avons utilisé en complément de Photoshop, notamment pour travailler sur les interfaces et sur les autres images contenues dans le jeu.

CSharp C# ou CSharp est un langage de programmation orientée objet commercialisé par Microsoft. C'est le langage utilisé par Unity, nous avons donc été obligés de l'utiliser pour le développement.

Rider C'est un environnement de développement (aussi appelé IDE). Il nous a permis d'écrire le code C# qui est utilisé dans Unity. C'est un logiciel payant, mais la licence Epita nous permet de l'utiliser de façon gratuite.

Discord C'est un logiciel gratuit de messagerie et d'appels instantanés. Nous l'avons utilisé pour la communication au sein du groupe.

Inno Setup Ce logiciel permet de créer rapidement un installateur pour un logiciel. Bien que ne permettant pas la personnalisation, c'est un logiciel efficace et bien documenté.

3.1.1 Github

Git est un logiciel de gestion de versions décentralisé permettant de travailler en collaboration sur des projets. Nous avons héberger nos répertoires de travail sur GitHub, un service web d'hébergement basé sur Git.

Nous avons crée une organisation GitHub regroupant les membres du groupe et séparé notre travail en 4 répertoires de travail. Le premire répertoire contient le code et les éléments du jeu en lui même, c'est le repértoire principale. Le second répertoire est utilisé pour le site web. Les autres répertoires permette de stocker de la documentation et les éléments utilisé pour la création de l'installateur du jeu.

En quelques chiffres, le projet GitHub principale représente

- 4 contributeurs
- 3 981 942 784 octets
- 257 commits
- 98 merge requests

En plus des statistiques qui font état du projet à un instant précis, GitHub permet de récupérer des données chronologiques qui font état du projet sur une période de temps.

Le premier type de données disponibles est le nombre de ligne ajouté ou retiré par semaine. Cette première statistique permet de voir les périodes d'activités du projet. La figure 2 présente les périodes d'activités ou d'inactivité du projet d'après ce premier type de données.

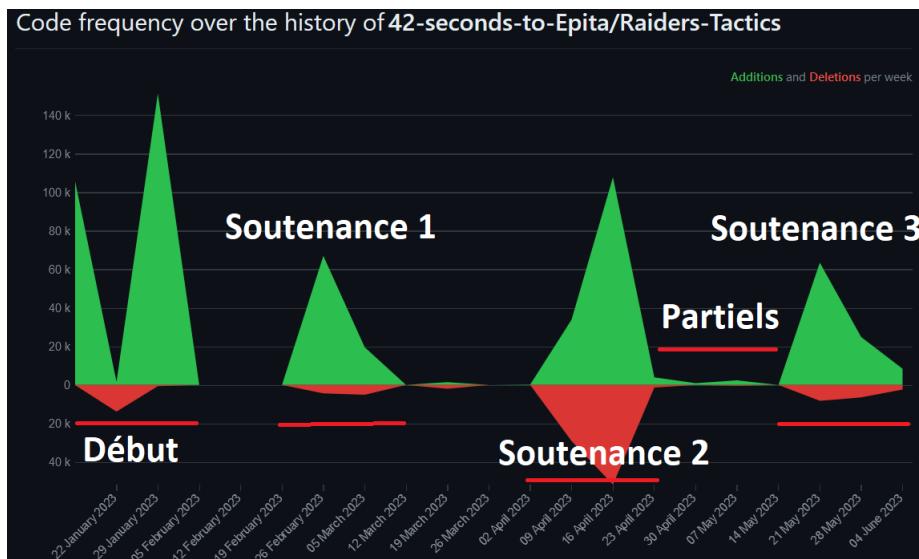


FIGURE 2 – Nombre de ligne ajouté ou supprimé en fonctions des semaines

Ces statistiques ne sont toutefois pas forcément totalement représentative. En effet, certaines fonctionnalités demandent plus de ligne sans avoir de réels impacts sur le jeu. La présence de fichier n'ayant aucun rapport avec du code sur le projet amplifie ce phénomène, notamment avec les objets qui sont des fichiers de plusieurs milliers de ligne, mais ne représente qu'une table ou une chaise dans un monde vide une fois charger dans Unity.

Le second type de statistique qui montre le nombre de modifications effectué permet d'éliminer ce phénomène. On voit notamment que le début du projet représente moins d'une dizaine de modifications alors que le nombre de ligne modifiée est supérieur à tout ce qui a été fait dans la suite du projet.

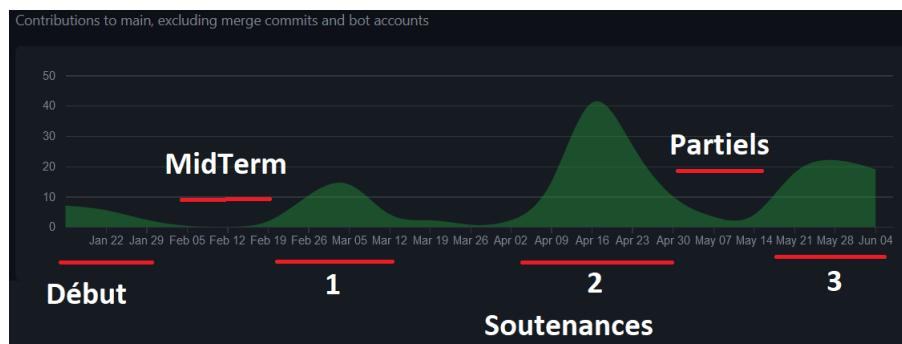


FIGURE 3 – Nombre de modifications par semaines

Nous avons également utilisé le service GitHub Pages pour héberger notre site internet. Ce service permet d'héberger un site web directement depuis un répertoire GitHub.

Certaines fonctionnalités de GitHub sont toutefois payantes comme Git LFS (*Large File Storage*) qui permet de stocker les fichiers de taille importante. La licence professionnelle obtenue gratuitement grâce au programme étudiant de GitHub ne permet qu'une quantité limitée de stockage qui n'était pas suffisante pour notre utilisation. Nous avons donc dû chercher d'autres solutions.

3.1.2 GitLab Epita

GitLab est une plateforme de gestion de développement de logiciels qui permet aux développeurs de collaborer, de suivre et de gérer efficacement leurs projets. C'est un outil de gestion de versions distribué basé sur Git. Nous utilisons la version hébergée par Epita.

Le Gitlab d'Epita nous permet de stocker nos modèles 3D et les différents fichiers volumineux nécessaires à notre projet sous la forme d'objets Git LFS ("Large file storage"). Ceci sans les frais supplémentaires qui aurait été facturé par GitHub si nous les

avions hébergés chez eux. Cela nous permet également de rester en conformité avec les licences d'assets qui ne nous permettent pas la diffusion publique de ceux-ci.

Les modèles sont stockés dans un projet qui est ensuite intégré comme un sous-module du projet principal hébergé sur GitHub.

3.2 Méthodes de travail de l'équipe

La façon dont l'équipe doit travailler a été établie dans le cahier du charge au début du projet. Cette organisation a toutefois été adaptée au fil du projet pour s'adapter au mieux aux exigences liées aux dates de rendu.

L'utilisation de git a été très normée dans le groupe pour éviter tout conflit lors de la rédaction du code. Quand une personne souhaite travailler sur une fonctionnalité, elle doit s'isoler dans ce qu'on appelle une branche. Lorsque la fonctionnalité est terminée, le code doit être relu par un autre membre du groupe, si le code est satisfaisant, il peut être fusionné et rejoindre le code principal du projet.

Le travail sous forme de branche et avec une relecture systématique du code a permis d'éviter des problèmes de conflits et a permis de mieux compartimenter le travail sur chaque fonctionnalité. Chaque personne a également pu se familiariser à l'ensemble du code lors de relectures, ce qui évite les problèmes dus au fait qu'une seule personne connaisse certains morceaux du code.

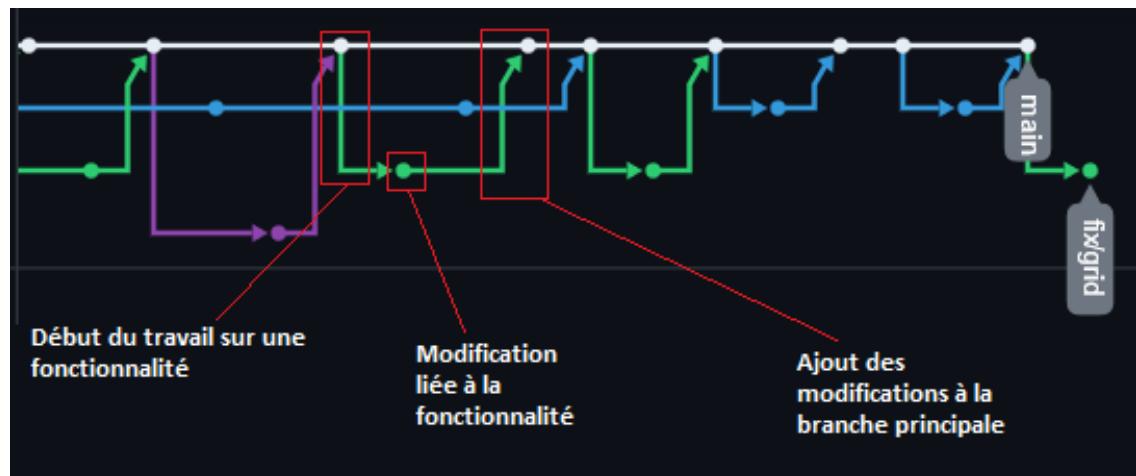


FIGURE 4 – Illustration de la méthode de travail Git

Des conflits dans les conventions d'usages de Git sont rapidement apparus dans le groupe. Ces conflits sont notamment dus à l'expérience de certaines personnes dans le groupe qui avait donc des conventions différentes au niveau du nommage. La figure 6 présente un tableau d'exemples de différents noms ayant été utilisé pour nommer des modifications. La figure 5 présente de façon plus normé les différentes façons qui ont été utilisées pour nommer les modifications sur Git.

- (feat ou fix ou chore) : Description de la modification en anglais
- [Domaine de la modification] | Description de la modification en anglais
- Description de la modification en anglais ou en français

FIGURE 5 – Exemples de noms donnés à des modifications

Les différences de conventions n'ont néanmoins pas réellement impacté le dérouler du développement même si elles sont facilement remarquables lors de la lecture de l'historique du projet Git.

Romain	Kévin	Alex et Arthur
”feat : git submodule for asset pack”	”[MapEditor] Add floor and walls”	”Sprites ratés, c'est terrible”
”chore : did not save scene”	”[Solo] all the system and first mission”	”adding props/floors/walls + wall system changed”
”fix : Unit list”	”[AI] First working AI”	”Réctification du script Projectile”

FIGURE 6 – Exemples de noms donnés à des modifications

Concernant la répartition des tâches, celle-ci avait été créée lors de la rédaction du cahier des charges. La période entre le moment où nous avons eu les informations relatives au projet S2 et la date limite de rendu étant d'uniquement une semaine, nous n'avons pas eu le temps de faire des essais sur Unity et avons donc assigner les tâches d'une façon non-optimale.

Entre le rendu du cahier des charges et la seconde soutenance, nous avons respecté strictement la répartition des tâches prévue dans le cahier des charges. Une plus grande souplesse a été observée suite à la seconde soutenance, cela pour que le projet puisse avancer. Un plus grand mélange dans les contributeurs aux différentes tâches a permis un regard nouveau sur le travail effectué et celui encore à effectuer.

3.3 Communication dans l'équipe

Comme pour les méthodes de travail, la communication dans l'équipe a été modifiée entre ce qui était prévu dans le cahier des charges et ce qu'il s'est réellement passé.

Nous voulions originalement nous réunir au moins une fois par semaine pour faire le point sur les avancées, les difficultés rencontrées et ce qu'il faut faire la semaine suivante. Dans la réalité, les emplois du temps des membres du groupe sont extrêmement différents ; nous réunir aurait donc obligé certains membres du groupe à attendre plusieurs heures en salle machine, un environnement non-propice au travail la majorité du temps. Certaines semaines nous demandaient également plus de travail dans les cours et les travaux pratiques de programmation ; nous ne travaillions alors pas sur le projet, ce qui impliquait une inutilité de la réunion de la semaine. Finalement, les réunions hebdomadaires étaient plus profitables à Okawa, une société de distributeurs automatiques, plutôt qu'au travail du groupe. Nous avons donc décidé d'arrêter les réunions hebdomadaires.

Les communications sur le projet se sont majoritairement effectuées par Discord. Ces discussions concernaient notamment l'attribution des tâches et l'assistance aux personnes bloquées sur un problème. Nous avons également effectué quelques réunions en présentielles, généralement avant et après les grands points d'étapes liées au développement du projet.

Certains problèmes de communication ont été rencontrés, il était par moment difficile de savoir qui travaillait sur quoi suite à des oubliés de communication. Un manque de communication a également été rencontré suite à des incompréhensions de code. Certaines réécritures intégrales de codes auraient donc pu être évitées par un simple message ou par une plus grande communication.

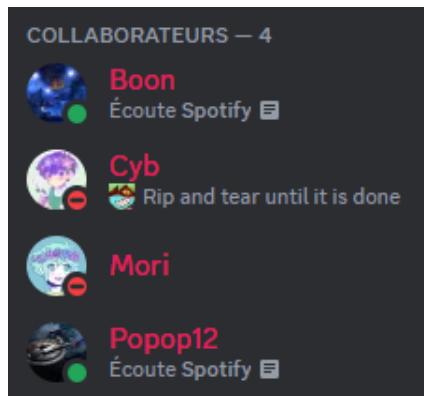


FIGURE 7 – L'équipe au travail

4 État final du projet

Pour la dernière soutenance, nous avons réalisé de nombreux éléments. Cette section présente la version finale du projet en précisant qui a réalisé chaque tâche.

4.1 Intelligence artificielle

Les implémentations de l'intelligence artificielle sont séparées en deux parties dans notre projet. La première forme est liée à la recherche automatique de chemin que nous avons dû implémenter. Les unités contrôlées par l'ordinateur sont le second type. La partie sur l'intelligence artificielle a été développée par Kévin.

4.1.1 Recherche automatique de chemin

Nous ne voulions initialement pas implémenter notre propre mécanique de recherche de chemin. En effet, celle-ci est nativement présente dans Unity et est utilisable par tous les développeurs. Néanmoins, nous nous sommes rendu compte que cette fonctionnalité était principalement conçue pour fonctionner sur des cartes conçues lors de la création du jeu et qui ne sont pas dynamiques. Nos cartes permettent aux joueurs de créer leur propre monde et changent donc en fonction de la partie. Ces conditions amènent la recherche de chemin de Unity à effectuer des actions illogiques comme foncer dans un mur ou essayer de traverser une table pour rejoindre un autre point de la carte. Malgré les nombreuses tentatives effectuées pour corriger ce problème, les unités ont continué d'essayer de traverser des murs. Nous avons donc dû implémenter notre propre recherche de chemin.



FIGURE 8 – Illustration du chemin choisi par une unité

L'implémentation de la recherche de chemin s'est faite en se basant sur ce qui a été fait pour l'éditeur de niveaux. C'est-à-dire une grille de la taille de la carte, chaque case

peut accueillir un joueur ou non en fonction de ce qui a été placée dessus. La grille est ensuite utilisée avec un algorithme A*¹ pour trouver un chemin optimal pour déplacer l'unité de sa position jusqu'à une autre position sur la grille.

4.1.2 Unités contrôlées par l'ordinateur

Des unités contrôlées par l'ordinateur sont présentes dans le jeu. Elles sont principalement visibles dans le mode campagne où elles sont les seules ennemis du joueur. Ses unités ont deux comportements possibles qui sont la recherche d'ennemie et l'attaque d'une cible.

Dans un premier temps, les unités sont en patrouille. Elles vont alors se déplacer de façon aléatoire autour de leurs points d'apparition. Dans cette phase, elles sont à la recherche d'ennemis et n'auront pas d'autres actions que de se déplacer. Dans le cas particulier des missions de protection de VIP dans le mode campagne, les unités iront patrouiller autour de la position de la personne qu'elles doivent éliminer et non pas proche de leurs points d'apparition.

Après avoir détecté un ennemi, les unités l'attaqueront si elles sont à portée. Dans le cas où l'ennemi ne serait pas à portée d'attaque, l'unité se déplacera pour se rapprocher en utilisant le chemin optimale. La poursuite prend fin si l'ennemi est trop loin de l'unité contrôlé par l'ordinateur.



FIGURE 9 – Unités contrôlées par l'ordinateur

1. Introduction à A* : <https://www.redblobgames.com/pathfinding/a-star/introduction.html>

4.2 Mode multijoueur et réseaux

Concernant le mode multijoueur de Raiders Tactics, nous avons poursuivis nos avancées lors des précédentes soutenances. L'objectif de cette implémentation étant que deux joueurs se connectent rapidement l'un à l'autre. Romain s'occupe de l'ensemble de cette partie en s'assurant de la structure cohérente et du bon fonctionnement de celui-ci.

Afin de convenablement gérer les communications entre ordinateurs, nous choisissons un modèle de communication client/serveur basée sur une implémentation de la bibliothèque Mirror. Le choix de Mirror se base sur le fait de sa stabilité, fiabilité et du fait que la bibliothèque ne dépende pas d'un intermédiaire tiers, permettant au jeu de fonctionner en tous temps et sans restriction de contenu, ceci contrairement à l'usage éventuel de la bibliothèque Photon que nous écartons pour cette dernière raison. Cette implémentation dépend du module de transport KCPTransport que nous sélectionnons et qui nous permet de relier les différents clients à l'aide du protocole UDP/IP sur un port donné (ici 7777) et qui nous permet de ne pas avoir à gérer un traitement bas-niveau des communications de paquets du jeu et qui nous permet également de ne pas avoir à diviser notre projet en différents langages de programmation, les autres implémentations requérant l'usage de langages comme le WebGL ou C.

Notre usage de Mirror en combinaison avec KCPTransport nous permet également de créer une structure client/hôte pour lequel l'hôte joue le rôle de client et de serveur simultanément, les clients se connectant ainsi au joueur hôte sans intermédiaire. Une description de ce modèle de fonctionnement est visible en figure 10.

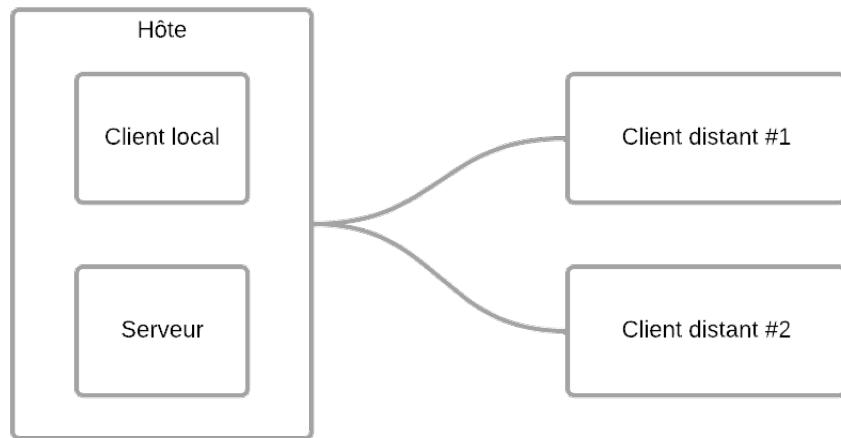


FIGURE 10 – Modèle de fonctionnement client/hôte Mirror

D'un aspect implémentation, nous nous servons premièrement des templates fournies par Mirror afin de redéfinir et surcharger (override) certains aspects de la bi-

bibliothèque Mirror pour mieux les adapter à notre jeu et à nos interfaces. Un exemple de ceux-ci est le Script C# `NewNetworkManager.cs` qui surcharge l'ensemble du module `NetworkManager.cs` de Mirror par défaut. Parmi les éléments les plus notables lors de cette surcharge, nous retrouvons certains appels de fonctions comme le démarrage du serveur qui doit, pour notre jeu, prendre en compte le chargement de carte, la création des joueurs en tant qu'objet en jeu et la création des unités. L'objet lié à la gestion du mode multijoueur joue donc un rôle majeur dans notre jeu.

Pour en revenir aux interfaces jouables, les différents joueurs ont le choix lors de leur lancement du jeu d'accéder à différents éléments par le menu visible en figure 11. L'un de ceux-ci est donc le multijoueur et possède tous les aspects et paramètres requis pour permettre la connexion des différents joueurs et le début de partie. Ce menu se décompose ainsi en deux parties distinctes, un côté contenant une liste de serveurs et un autre pour l'hébergement d'une partie de jeu.

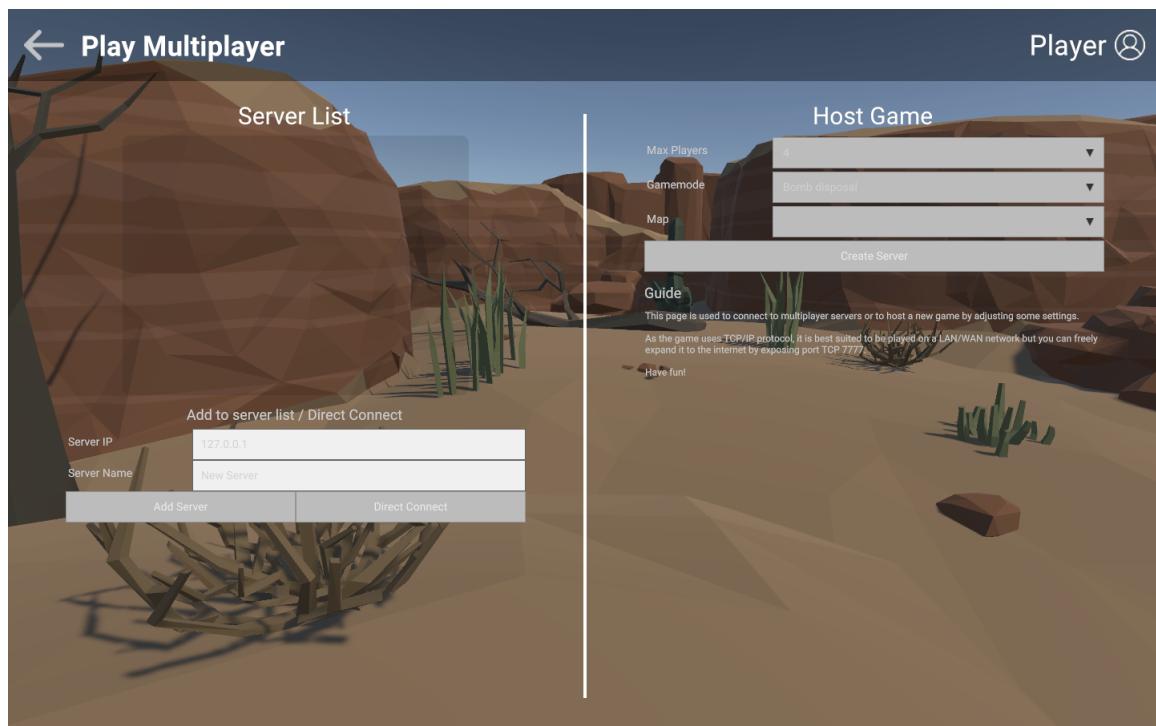


FIGURE 11 – Menu multijoueur

La partie de connexion de jeu se compose ainsi en une liste de serveurs et permet aux joueurs d'ajouter un nombre illimité de serveurs à l'aide de deux paramètres possible à insérer à l'aide d'entrées textuelles composées du nom de serveur qui ne sert qu'à l'affichage du serveur dans la liste et de l'adresse IP du serveur sous un format IPv4 ou d'un nom d'hôte le cas où le joueur souhaite exposer son serveur derrière un nom de domaine. Le port d'écoute du serveur ne pouvant pas être remplacé par le joueur, il n'est

effectivement pas possible de définir le port de connexion pour rejoindre un serveur de jeu. Une démonstration de celui-ci est visible en figure figure 12

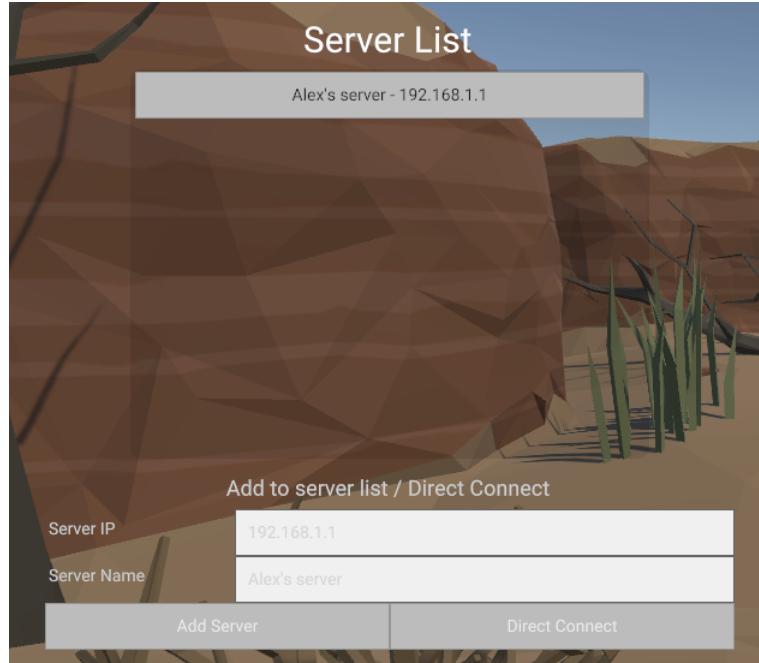


FIGURE 12 – Ajout de serveurs dans la liste

La partie d'hébergement du jeu se compose quant à elle à diverses options que le joueur peut librement définir à sa convenance. Celles-ci se succèdent en une liste pour laquelle figure : le nombre de joueurs maximal, le mode de jeu (Bomb Defusal ou Team Deathmatch) et la carte de jeu (sous l'appellation "Map"). Certaines de ces options sont prédéfinies et possèdent des valeurs par défaut mais nécessitent une attention particulière des joueurs. Il a été décidé d'omettre certains éléments jugés trop technique comme la redéfinition de port qui n'était pas vital pour permettre la connexion de joueurs, d'autres éléments tels que l'adressage, type de communication pourtant requises pour le fonctionnement de Mirror ont été simplifié pour le joueur et rendre le jeu le plus accessible possible sur le modèle client/hôte. Il est enfin possible pour le joueur de démarrer le serveur en cliquant sur le bouton adéquat.

Parmi les cartes jouables qu'il est possible de sélectionner, deux catégories se distinguent avec pour l'une, les cartes officielles et pour l'autre les cartes personnalisés et créées dans l'éditeur de carte (sous l'appellation "Custom"). Il est ainsi possible de jouer sur toute les cartes du jeu dans le mode multijoueur. Dans le cas où le joueur ne sélectionne pas de carte de jeu ou une carte invalide, la partie ne se lance pas et un message d'erreur apparaît pour que le joueur puisse corriger le manque. Le menu de sélection est visible en figure figure 13.

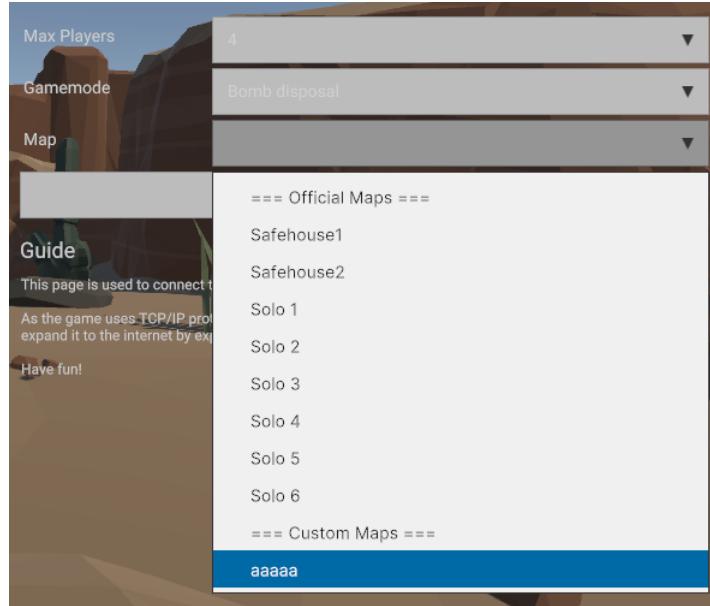


FIGURE 13 – Modèle de fonctionnement client/hôte Mirror

À des fins d'accessibilité, un guide d'usage des différentes options du menu apparaît également et donne diverses indications telles que le mode de fonctionnement du mode multijoueur et d'autres aspects pouvant aider des joueurs plus avancées pour la connexion des différents joueurs avec le port du jeu et des indications pour la redirection de celui-ci sur routeur afin de l'exposer sur internet. Il contient ainsi les phrases suivantes : "This page is used to connect to multiplayer servers or to host a new game by adjusting some settings. As the game uses TCP/IP protocol, it is best suited to be played on a LAN/WAN network but you can freely expand it to the internet by exposing port TCP 7777".

Un soucis d'exploitation du jeu cependant réside au fait que les membres de l'équipe 42 Seconds To Epita n'est nullement experte en sécurité et réseau, il va ainsi de fait qu'il est recommandé aux joueurs de sécuriser convenablement leur sessions de jeu et de ne pas permettre à n'importe qui de rejoindre leurs parties.

Le joueur peut enfin définir son nom d'affichage en haut à droite de l'écran et y indiquer le nom souhaité. Il est à noté que celui-ci doit être compris entre 6 et 12 caractères et ne peut pas contenir de caractères spéciaux. Ce nom d'affichage est sauvegardé entre les parties et au redémarrage du jeu. Au premier lancement, un nom prédéfini sous le format "Player" suivi d'un nombre aléatoire de 4 chiffres est fourni.

En créant une partie de jeu en tant que joueur hôte, le jeu assemble différents éléments qui créent le rendu final de la carte, les unités et la partie. Nous séparons ces éléments sous la forme d'abstraction qui prend en compte divers éléments et classes abstraites. Parmi elles, il est ainsi possible de regrouper le fonctionnement du jeu et des propriétés

associées au sein d'un unique objet synchronisé entre les joueurs mais y étant invisible. Cet objet s'occupe ainsi de maintenir une liste de joueurs qui décrit la position de la caméra et de l'équipe jouée par ceux-ci (pour rappel, soit "Police", soit "Raiders"). Il maintient également le bon déroulement des parties de jeu en générant les parties par "Round" ce qui permet de contenir l'ensemble des unités jouables et de la gestion d'évènement en un seul bloc. C'est également ici que les scores sont enregistrés et gérés.

Lorsque les joueurs arrivent en jeu, il leur est possible de sélectionner leur équipe, affectant ainsi directement leur objet joueur, voir figure figure 14. Le déplacement de caméra est bien synchronisé entre le serveur et les joueurs afin de prévenir les mouvements illicites et fait l'objet d'une communication client vers serveur, le serveur s'occupant de corriger les imprécisions et éventuelles erreurs de déplacement client. Un certain nombre des propriétés liées aux joueurs fait aussi l'objet d'une synchronisation entre l'ensemble des clients comme le nom du joueur ou son équipe.

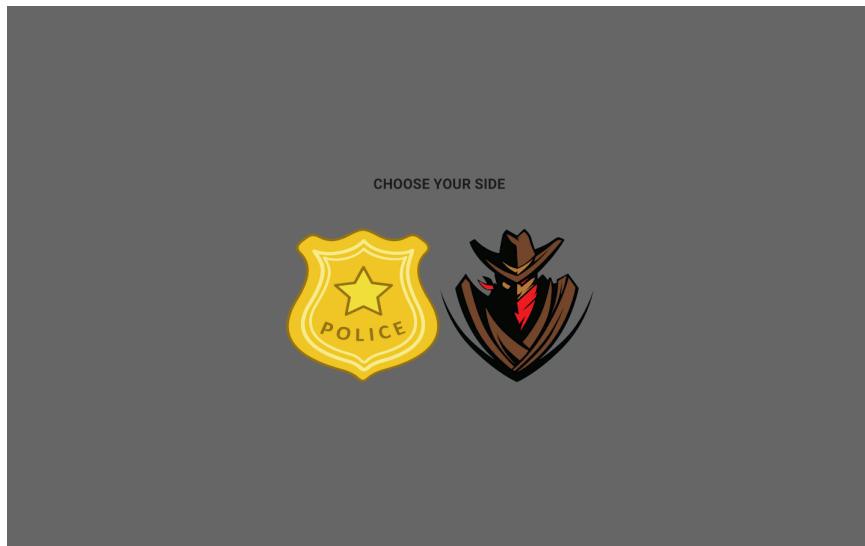


FIGURE 14 – Ajout de serveurs dans la liste

La partie de jeu en elle-même se décompose d'éléments supplémentaires tels que les unités et le temps restant en jeu si applicable. Un évènement lors de la fin de la partie a également été implémenté afin de permettre aux autres sections du programme de connaître l'équipe vainqueur pour la mise à jour des scores par exemple.

Quant aux unités, celles-ci se décomposent en de nombreuses propriétés synchronisées et notamment utilisés afin de distinguer celles jouées par l'intelligence artificielle de celles jouées par le joueur, les positions, leur points de vie (appellation "HP" ou "Health Point"). D'autres éléments locaux aux clients permettent de définir le fait qu'une unité soit sélectionnée ou non ou de gérer l'ensemble des animations, VFX et SFX liés.

Une représentation schématique du modèle d'abstraction est donnée en figure fi-

Figure 15

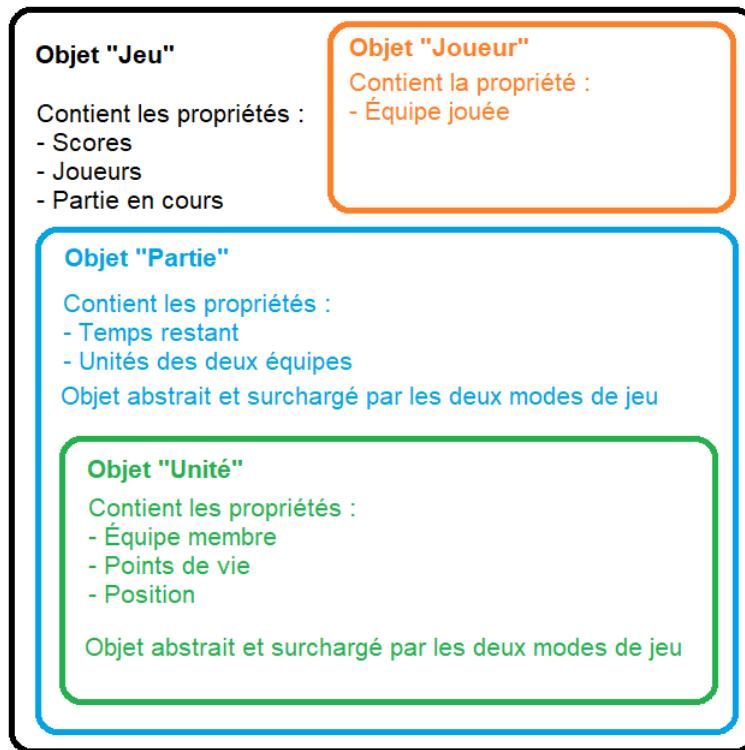


FIGURE 15 – Modèle d'abstraction du multijoueur

Parmi les modes de jeu jouables, nous implémentons le mode "Bomb Disposal" qui est mis en adversité les deux équipes avec l'équipe terroriste "Raiders" qui doit protéger une bombe pendant la durée d'une minute et 55 secondes et de l'équipe des policiers qui doit s'approcher de la bombe afin de la désamorcer. Si l'équipe des policiers ne possède plus d'unités en vie ou qu'il n'y a plus de temps restant, celle-ci perd la partie et le point est accordé à l'équipe adverse.

Nous implémentons également le mode "Team Deathmatch" qui met également ces deux équipes en adversité, mais avec pour objectif l'élimination totale de l'équipe adverse. Une limite de temps d'une minute et 55 secondes force les joueurs à s'investir dans le jeu et crée une tension de Gameplay. Si le temps imparti est écoulé, l'équipe ayant le plus d'unités vivantes emporte le point, dans le cas où un nombre égal d'unité est présent, aucun point n'est accordé par un match nul.

Au niveau du jeu en lui-même, le joueur contrôle ses unités avec la souris. Il peut sélectionner une unité en utilisant le clic gauche au-dessus de sa position dans l'interface ou dans le monde. Il peut ensuite la déplacer en effectuant un clic droit sur la position

de destination et les unités tireront automatiquement sur les ennemis s'il elles en ont la portée. Un exemple de jeu multijoueur est disponible en figure figure 16 et figure 17



FIGURE 16 – Exemple 1 du jeu multijoueur



FIGURE 17 – Exemple 2 du jeu multijoueur

J'enregistre enfin ma voix afin que lorsqu'une équipe emporte la victoire, on puisse entendre "Raiders Wins", "Police Wins", "The bomb has been defused" et un autre SFX concernant l'explosion de la bombe. Cette fonctionnalité vitale est tout droit inspirée des plus grands, voir CS :GO.

Lorsqu'une partie se termine, une autre commence immédiatement après afin de conserver cet aspect de tension, cœur du gameplay.

4.3 Mode campagne

Le mode campagne a été développé par Kévin avec des cartes et une histoire créée par Arthur grâce à l'éditeur de niveau. La campagne intègre 8 missions jouables depuis un menu accessible dans le menu principal et qui est séparé du mode multijoueur. Les missions sont accompagnées d'une histoire et le joueur joue contre des unités contrôlées par l'ordinateur. Différents objectifs sont implémentés pour diversifier le jeu et le rendre plus intéressant pour le joueur. Le mode campagne se base sur le travail qui a été effectué pour le multijoueur tout en implémentant quelques modifications pour le rendre plus intéressant et jouable.



FIGURE 18 – Menu principal du mode campagne

La figure 18 présente le menu principal du mode campagne. Il est composé d'une zone centrale où figure les différentes missions. Les missions non encore débloquées sont en noir, la mission en cours est en rouge et les missions précédentes en bleu. Le joueur a la possibilité d'afficher les informations supplémentaires sur une mission en cliquant sur sa représentation dans la carte. Les détails d'une mission sont affichés à droite de l'écran et comprennent des détails sur la mission, des objectifs et un bouton permettant de lancer la mission s'il s'agit de la mission actuelle.

L'interface de jeu est similaire à celle du mode multijoueur. Les unités du joueur et leurs statistiques sont affichées en bas de l'écran. Les objectifs de la mission et le temps restant ou le temps écoulé depuis le début sont afficher dans la partie supérieure.



FIGURE 19 – Exemple de mission

À la fin d'une partie, le menu de la figure 22 apparaît et indique au joueur s'il a réussi ou échouer sa mission. Il a la possibilité de retourner au menu avec un bouton. La progression du joueur est automatiquement sauvegardée lors de la fin d'une mission.

Trois différents types d'objectifs sont présents dans le mode campagne, certaines missions combinent plusieurs objectifs. Les missions d'élimination sont réussies si le joueur élimine toutes les unités ennemis. Un second type de mission consiste à trouver une bombe et de rester à côté pendant un certain temps pour la désamorcer. Le dernier type de missions est une protection de personne, le joueur doit protéger un personnage pendant une période déterminé pour réussir la mission.

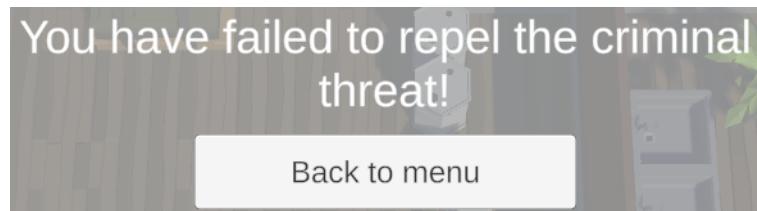


FIGURE 20 – Interface de fin de mission

4.4 VFX et SFX

C'est Alex qui s'est occupé des VFX et des SFX. Pour les VFX tout d'abord, nous avons rajoutés et synchronisés un animator (gestion d'animations) qui transite entre un idle (état statique en mouvement) pour aller vers différentes animations :

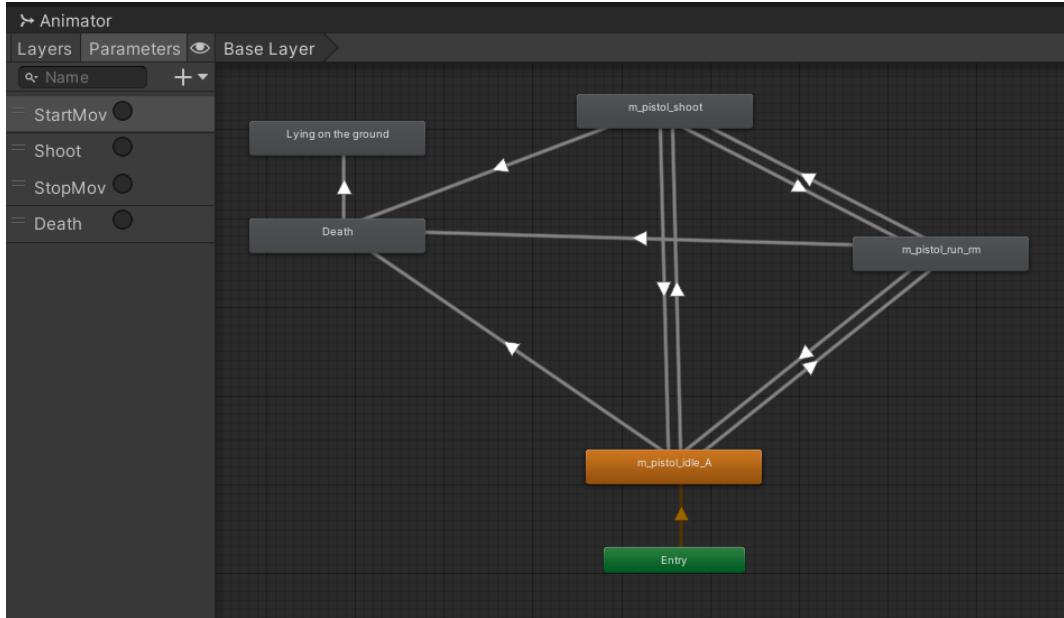


FIGURE 21 – Unity Animator

1. "m_pistol_idle" phase statique avec arme
2. "m_pistol_run" phase de course avec arme
3. "m_pistol_shoot" phase de tir
4. "Death" animation de mort
5. "Lie on the ground" phase statique de mort

De plus nous avons rajoutés des effets de trainé et de fumée derrière les tirs des balles pour intensifier l'effet de tir, et des effets de splash de sang à l'impact des balles sur les corps.

Pour les SFX, tout d'abord nous avons sélectionnés des musiques libres de droit qui génèrent une certaine tension.

Nous avions en tête celles de CSGO et Battlefield et avons recherché quelque chose du genre disponible gratuitement sur youtube.

Pour l'ambiance sonore nous avons rajoutés des bruits de tirs.

4.5 Interfaces



FIGURE 22 – Menu principal

Pour les interfaces, nous avons décidé d'intégrer un menu dynamique, qui charge à chaque fois avec un personnage différent à chaque lancement. De plus, cliquer sur le menu multijoueur effectue une rotation dans la scène pour renforcer le dynamisme de la sélection du mode de jeu.

Lorsqu'une unité est sélectionnée, il y a un cercle de surbrillance vert qui apparaît en dessous uniquement chez la personne qui la sélectionnée, pour que le joueur puisse avoir une information sur l'unité qu'il est en train de contrôler. L'interface est interactive, c'est-à-dire que lorsque l'on clique sur un portrait, cela sélectionne l'unité. Lorsqu'une unité est sélectionnée, il y a un cercle de surbrillance vert qui apparaît en dessous uniquement chez la personne qui la sélectionnée, pour que le joueur puisse avoir une information sur l'unité qu'il est en train de contrôler.

4.6 Gestion des aspects liée au joueur

La partie liée aux joueurs n'a pas un impact important sur le jeu car elle était supposée uniquement permettre au joueur de personnaliser une équipe d'unités et d'associer des statistiques à ses unités. Nous avons donc décidé de ne pas effectuer cette partie et de prioriser les éléments ayant un grand impact sur le rendu final.

4.7 Gestion des niveaux

Dans la version finale du jeu, le joueur à la possibilité de sélectionner la carte avec laquelle il souhaite créer une nouvelle partie multijoueur. Ses cartes sont séparées en deux catégories. La gestion des cartes et l'éditeur de niveaux ont été développé par Kévin.

Les cartes non-modifiables par le joueur sont intégrées directement dans le jeu. Ce sont des cartes originales commune à tous les joueurs. Les cartes du mode campagne font partie de cette catégorie et peuvent donc être utilisées également en multijoueur et ont été créé par Arthur.

Les cartes modifiables doivent être ajoutées dans le fichier sauvegarde présent dans les fichiers du jeu. Le joueur à la possibilité de créer des nouvelles cartes grâce à un éditeur de niveaux présent dans le jeu. Il pourra créer une carte en utilisant les objets présents dans le jeu. Le joueur pourra ensuite créer une partie multijoueur que d'autres personnes pourront rejoindre sans avoir l'obligation de posséder la carte sur leurs ordinateurs.

4.7.1 Éditeur de niveaux

L'éditeur de niveau permet aux joueurs ou aux développeurs de créer des cartes de jeu facilement sans devoir passer par un logiciel spécialisé comme Unity.

Le système de construction est basé sur une grille. Chaque case pouvant accueillir un sol et un objet principal. Une interface permet au joueur de sauvegarder la carte ou de visualiser les différents objets qu'il peut poser.

Les cartes créent peuvent ensuite être sauvegardé et partager dans la communauté de joueurs comme de simple fichier texte.

Le système de construction intègre une interface de sélection de l'objet à poser et un placement des objets par la souris. Il repose sur une grille dont la taille représente la taille du niveau.

Pour construire, le joueur devra sélectionner l'objet qu'il souhaite poser grâce au menu de sélection à la droite de l'écran. Ce menu de sélection comporte des boutons

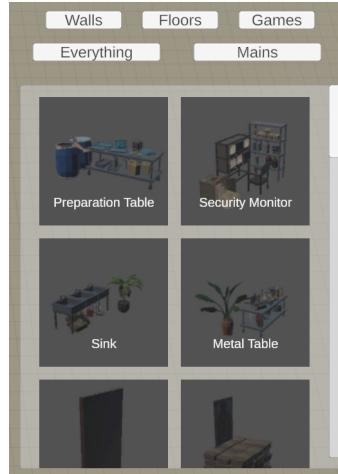


FIGURE 23 – Interface de sélection

permettant de filtrer les objets qui apparaissent à l'écran. Il pourra également passer d'un objet à l'autre avec les touches de son clavier, avec cette méthode, il parcourra une liste d'objets. S'il arrive à la fin des objets disponibles, il sera renvoyé au début. Il pourra ensuite placer l'objet avec le clic gauche de sa souris. Il a également la possibilité de faire effectuer une rotation à cet objet à l'aide d'une touche de son clavier. Finalement, le joueur pourra supprimer tous les éléments d'une case avec le clic droit de sa souris.

Après avoir construit une carte, le joueur à la possibilité de la sauvegarder dans un fichier texte avec un nom qu'il définit. La représentation textuelle de la carte sera sauvegardée dans un dossier sauvegarde présent dans les fichiers du jeu. Le fichier qui est sauvegardé peut ensuite être partagé dans une communauté de joueur. Il suffit à un joueur ayant reçu une carte de l'ajouter dans son dossier sauvegarde pour pouvoir la charger dans son jeu. Il suffira ensuite au joueur de charger cette sauvegarde pour voir sa création apparaître à l'écran. La sauvegarde se fait à l'aide d'une interface comportant un champ de texte permettant de choisir le nom de la carte à sauvegarder ou charger. L'interface comporte également un bouton sauvegardé et un bouton charger. Un bouton de réinitialisation permet de supprimer tous les objets placés sur la carte.

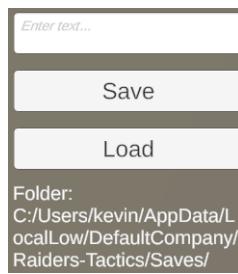


FIGURE 24 – Interface de sauvegarde

4.7.2 Sélection et intégration des modèles

Étant donné qu'aucun membre du groupe ne maîtrise de logiciel de modélisation 3D et que nous préférons investir les nombreuses heures nécessaires pour maîtriser ce genre de logiciel dans le travail des cours ou sur la partie purement technique du projet ; nous avons décidé d'utiliser des assets trouvés sur Internet.

Nous avons pour cela eu la chance de trouver un pack de modèles correspondants exactement à nos besoins sur Humble bundle ; un site proposant des packs de différents types à la vente et dont les bénéfices sont reversés à des associations caritatives.

Nous avons ajouté au système d'éditeur de niveau un procédé permettant d'ajouter facilement de nouveaux objets au jeu. Ce système attribue automatiquement un identifiant unique à l'objet et l'ajoute à l'interface de sélection. Arthur s'est occupé de sélectionner les modèles 3D intéressant et de les ajouter dans le jeu grâce à ce procédé.



FIGURE 25 – Exemple de réalisation possible

4.8 Rendu Physique

Nous rendons le jeu dans une boîte format blu-ray contenant une clé usb de 16Gb avec l'installateur du jeu à l'intérieur. Pour ce qui est de la jaquette du jeu nous avons utilisé photoshop et un peu de notre créativité pour donner à notre boîte un rendus attrayant et amusant qui permet de se donner une idée de notre jeu grâce à son visuel.



FIGURE 26 – Jaquette du jeu

La clé USB étant un rendu de projet nous avons ajouté d'autres éléments qui ne seraient pas contenus dans une hypothétique version commercialiser du jeu. Le contenu supplémentaire inclue nos différents rapports de soutenances déjà rendus au format papier, le projet Unity du jeu ainsi que le code du Site Web. Un dossier d'exploitation et un pack de logo sont également inclus dans le jeu pour permettre au joueur de mieux appréhender le jeu.

5 État final du site web

Le site est à présent accessible par le lien suivant : <https://raiders-tactics.com/>.

Les points clés de son architecture actuelle n'évolue que peu depuis la première soutenance et est constitué de : la *landing page*, page sur laquelle l'utilisateur atterrit sur le site ; la présentation des membres de l'équipe 42 Seconds To EPITA ; et les liens de téléchargement du jeu avec la documentation associée. Ces éléments sont aisément navigable à l'aide d'une barre de navigation dans le haut de page. Un pied de page donne à l'utilisateur un moyen de contact mail vers l'organisation et une redirection vers sa page Github.

Le développement de celui-ci se fait par Romain appuyé par Kévin notamment lors des phases de conceptualisation et de retour sur l'utilisation. Les technologies utilisées se limitent à de l'HTML/CSS pur, l'hébergement ayant lieu sur Github Pages. Le choix graphique du site s'est inspiré de nombreux autres sites vitrines existants ayant un lien avec le domaine du jeu vidéo.

5.1 La *landing page*

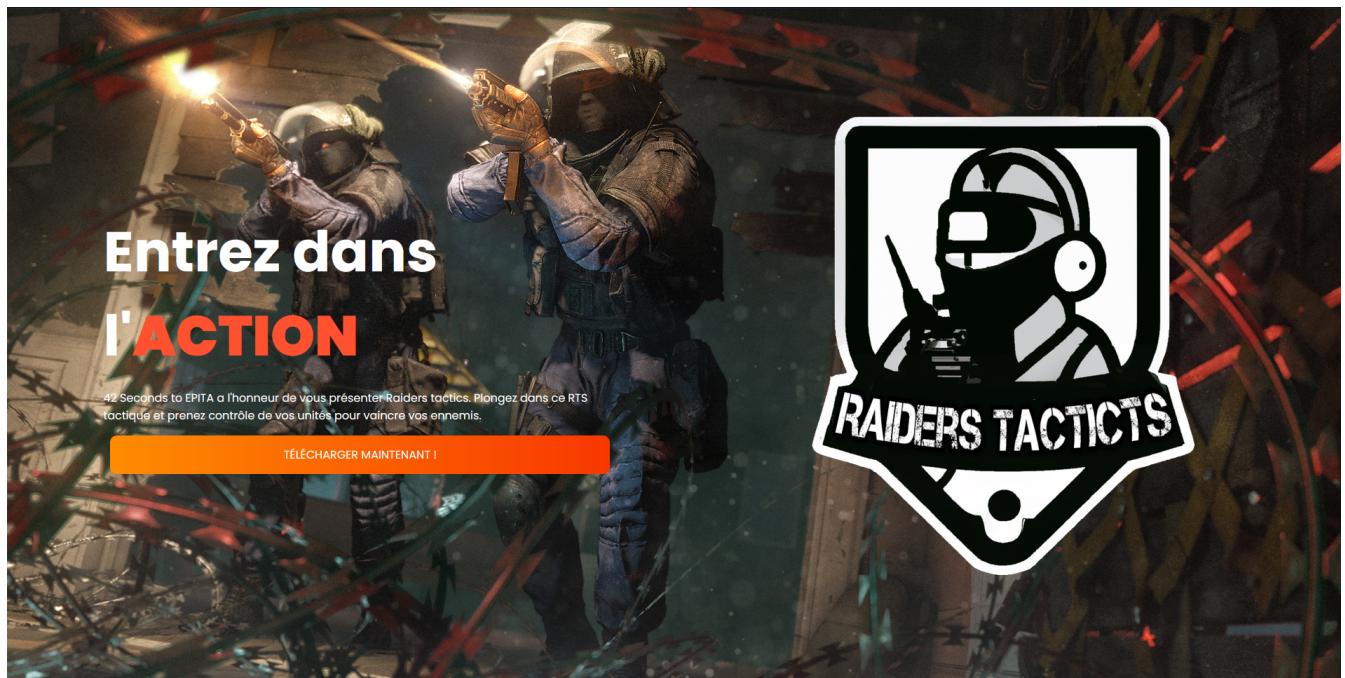


FIGURE 27 – Page d'accueil du site Web

Cette page constitue la page d'accueil des visiteurs du site et est joue un rôle majeur pour cette première impression. Elle contient ainsi : le logo du jeu ; une image de fond qui reflète le thème du jeu ; une phrase marquante suivie d'une description brève du jeu ; et un lien vers la page de téléchargement.

5.2 Présentation des membres

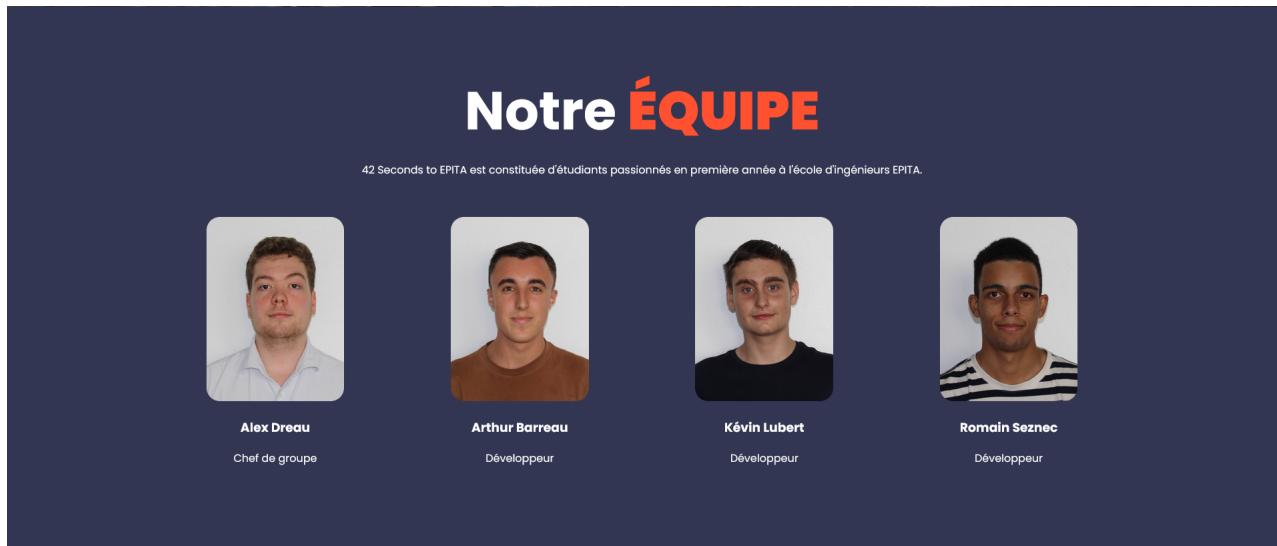


FIGURE 28 – Section de téléchargement du site Web

La section de présentation des membres de l'équipe 42 Seconds To EPITA est constituée d'une contextualisation brève de la nature du groupe et de nos profils individuels avec un bref descriptif des tâches qui incombent à chacun.

5.3 Téléchargement des ressources

La dernière section du site Web donne accès au téléchargement des ressources publiquement disponible avec des liens notamment vers le téléchargement du cahier des charges ; du plan de soutenance ; et enfin de ce même rapport.

Le jeu ayant désormais une version exécutable, il donne également accès au téléchargement de l'exécutable.

TÉLÉCHARGER maintenant !

Toutes les ressources du jeu sont disponibles ici.

Documentation

Cahier des charges

Tout ce qu'il faut savoir sur le développement de Raiders Tactics !

[Cahier des charges](#) [Télécharger en PDF](#)

Première soutenance - 10 mars 2023

Plan et rapport de soutenance qui résume le développement du jeu depuis l'édition du cahier des charges.

[Rapport de soutenance](#) [Télécharger en PDF](#)

[Plan de soutenance](#) [Télécharger en PDF](#)

Téléchargement du jeu

Le téléchargement du jeu n'est pas encore disponible. Tenez vous à jour de la progression du développement avec les rapports de soutenances !

FIGURE 29 – Section de téléchargement du site Web

The navigation bar features the project name "Raiders Tactics" on the left. To the right, there are four items: "Le Jeu" (highlighted in orange), "Notre Équipe", "Télécharger", and a small accessibility icon (A with a checkmark).

FIGURE 30 – Barre de navigation du site Web

5.4 Multilinguisme et accessibilité

À des fins d'accessibilité, le site Web du projet est disponible en français et en anglais permettant d'élargir largement ses capacités d'audience. Il est ainsi possible pour l'utilisateur de changer la langue du site au clic de l'icône associée au sein de la barre de navigation.

Le site respecte également les normes et conventions Web pour l'accessibilité décrites par W3C et MDN.

6 Discussion

6.1 Retour sur l'avancement lors de la première soutenance

Pour la première soutenance, nous avions réalisé une première version du multijoueur qui permettait à plusieurs joueurs de se connecter à la même carte. Les joueurs pouvaient déplacer des unités qui leur étaient personnelles et qui ne pouvaient être contrôlées par les autres joueurs.

Au niveau du site web, nous avions une implémentation complète du site comme prévu dans le cahier des charges. Celle-ci n'a que très peu évolué par la suite.

Une première gestion des cartes était implémentée, elle permettait de charger des cartes depuis leurs versions texte. En plus des fonctionnalités prévues, une version initiale de l'éditeur de carte était disponible qui permettait de créer des cartes basiques directement depuis le jeu.

Pour résumé, lors de la première soutenance, nous étions dans les temps au niveau de l'implémentation des fonctionnalités. Certains domaines étaient même en avance sur les dates prévues lors de la rédaction du cahier des charges. Toutefois, nous étions toujours en train de découvrir comment utiliser Unity. Nous aurions dû nous mettre plus tôt à travailler avec Unity, ce qui nous aurait permis d'effectuer un travail plus propre sur le début du jeu. Nous aurions également pu prendre de l'avance sur certains domaines ce qui aurait aidé pour la suite du développement du projet.

Cette première soutenance avec le jury a été très enrichissante pour nous et nous a permis de nous conforter dans nos choix liés à nos implémentations et à nos choix liés au jeu en lui-même. Elle nous a également permis de corriger notre logo qui possédait une erreur.



FIGURE 31 – Ancien logo

6.2 Retour sur l'avancement lors de la seconde soutenance

Les avancer sur le développement du jeu entre la première et la seconde soutenance ne sont que minime. Nous avons toutefois développé quelques fonctionnalités qui formaient des bases plus ou moins solide pour la suite du travail après la seconde soutenance et jusqu'à la fin du projet.

La gestion des cartes a été améliorée et permet de stocker des cartes non-modifiables par le joueur. Des modifications ont également été apportées à l'éditeur de niveau entre la première et la seconde soutenance. Notamment un menu de sélection des objets que l'on va placer, ainsi que différentes fonctionnalités facilitant la création comme un bouton permettant la réinitialisation de tous les objet placé sur la carte.

Un travail à également été effectué sur le multijoueur. Certaines fonctionnalités ont été retravaillées pour être plus facilement utilisé par la suite. Un premier système d'équipe a été implémenté et un système automatique de détection des parties pouvant être rejoindre par le joueur a été ajouté. Les fondations du système purement lié à la connexion des joueurs entre eux ont donc été placé pour la seconde soutenance.

Le menu principal a été complètement retravaillé pour la seconde soutenance. Ceci afin d'accueillir les nouvelles fonctionnalités liées au multijoueur et d'être plus ergonomique pour le joueur. Un menu de communication textuelle a également été ajouté pour permettre aux joueurs jouant sur une même partie de communiquer entre eux.

Toutefois, le Gameplay n'a que très peu avancer. Une sélection des unités a été implémentée ce qui permet au joueur de déplacer spécifiquement une unité contrairement à la première soutenance où le joueur était obligé de déplacer toutes ces unités en même temps. Des modèles de personnage ont également été ajoutés pour remplacer les sphères qui représentaient les joueurs précédemment. Ses personnages ont également reçu des animations, notamment de marche.

En résumé, bien que certains éléments aient avancé, les éléments de jeu en eux même n'ont que très peu avancer. Nous étions donc en retard pour la seconde soutenance sur la majorité des points liée aux éléments de jeu. Nous n'avions donc que très peu d'élément à présenter lors de la soutenance, et ceci même si un travail avait été accompli sur certains points du jeu.

Les retards accumulés entre la première et la seconde soutenance sont liés à la diminution du temps passé à travailler sur le projet dû à l'augmentation du temps passé à travailler sur les travaux pratique de programmation et sur les cours. Le temps passé à travailler sur le projet après la seconde soutenance a donc dû être augmenté pour rattraper le retard.

6.3 Comparaison avec les objectifs

6.3.1 Intelligence artificielle

Le domaine de l'Intelligence artificielle a été complété et les principales tâches qui le composent ont été réalisées.

Au niveau des unités contrôlées par l'ordinateur, le cahier des charges restait très général sur ce qui devait être effectué. Notre implémentation correspond toutefois aux grandes lignes qui étaient énoncées, c'est-à-dire une différenciation entre les états d'attaque et de déplacement. L'intelligence artificielle reste toutefois simpliste et n'implémente pas le fait de pouvoir se protéger par différents moyens, ce qui était défini dans le cahier des charges.

Le développement de notre propre système de recherche de chemin n'était pas prévu dans le cahier des charges, mais a été rendue nécessaire lors du développement. Le domaine de l'intelligence artificielle a donc certains manquements par rapport au cahier des charges, mais certains points sont toutefois plus avancés.

6.3.2 Mode multijoueur et réseau

Le mode multijoueur et réseau fait partie de ceux pour lequel le temps de travail fut le plus important. Nous tenons ainsi comme prévu un mode multijoueur en réseau totalement fonctionnel.

Parmi les modes jouables indiqué dans le cahier des charges, nous prenons la décision après la seconde soutenance de ne pas retenir le mode "VIP Protection" qui consistait en la défense d'une unité complémentaire par l'équipe policière de l'équipe terroriste. Nous implementons ainsi les deux modes de jeu "Team Deathmatch" et "Bomb Disposal", le premier consistant en l'élimination totale de l'équipe adverse et le second en le déminage d'une bombe par l'équipe policière par l'équipe terroriste. Le mode de protection de VIP a toutefois été implémenté dans le mode campagne.

6.3.3 Mode campagne

Le mode campagne comme prévu dans le cahier des charges n'a été que partiellement implémenter.

La partie principale a été créée, c'est-à-dire que le joueur à la possibilité d'afficher le menu de la campagne et l'utiliser. Conformément à ce qui était prévu, le joueur suit un scénario prédéfini et effectue différentes missions.

Néanmoins, nous n'avons pas implémenté les aspects liés au joueur. Dans le mode campagne, nous aurions dû implémenter un système d'expérience pour le joueur et de l'argent qu'il récupérerait pour chaque mission réussie. Nous n'avons également pas implémenté les usages que le joueur aurait dû pouvoir utiliser avec l'argent, notamment la personnalisation d'une équipe qui reste constante dans tous les niveaux de la campagne.

6.3.4 VFX et SFX

Malheureusement suite aux conseils avisés de notre professeur référent, nous nous sommes concentrés sur les aspects du gameplay comme la campagne solo et la jouabilité. Nous n'avons donc pas pu mettre tous les effets visuels et sonores désirés. Cependant, l'amélioration de nos animations, les VFX au niveau des balles, les traces de sang, le bruit des balles ainsi que la musique suffisent pour instaurer un semblant d'ambiance, un ersatz de notre idéal, mais tout de même de quoi accompagné le gameplay nerveux et prenant, compétitif à souhait que nous proposons.

Ses éléments discrets, mais permettant une meilleure immersion sont notamment une animation lorsque qu'une unité est inactive, une traînée de fumée derrière les balles, mais également des sons lorsque les unités effectuent différentes actions.

6.3.5 Interfaces

Les interfaces que nous avons dans le jeu final sont bien aux dessus de ce que nous souhaitions initialement. Notre menu principale est dynamique et les personnages affiché sur le côté droit change de manière aléatoire à chaque lancement du jeu, ce que nous n'avions pas prévues initialement.

L'interface du menu du mode solo est plus que satisfaisant, nous n'avions pas de réel objectif en terme d'interface pour ce mode jeu ce qui fait que le travail que l'on a fourni pour celui ci nous conviens amplement.

Pour ce qui est des interfaces en jeu, nous avons retiré deux éléments initialement prévus. La partie concernant l'agent sélectionné a été supprimée, car celle-ci n'a plus de réel utilité étant donné que la majorité des fonctionnalités que l'on voulait pouvoir utilisé via cet élément de l'interface ne sont pas implémenté, notamment tous les aspects liés aux joueurs comme décrit plus bas. De plus, le menu de paramètre et le bouton permettant de mettre en pause le jeu ne sont pas présents dans le rendu final du jeu, car ces fonctionnalités ne sont pas implémentées.

6.3.6 Gestion des aspects liée au joueur

La gestion des aspects liés aux joueur est un des éléments qui souffre le plus du retard accumulé lors de la seconde soutenance. N'ayant pu atteindre un mode de jeu fonctionnel, nous n'avons pu poursuivre l'aspect gestion micro-management des unités telles que le changement d'armes et de l'implémentation d'un magasin d'achat en début de partie.

Nous permettons cependant au joueur de contrôler l'ensemble des unités de son équipe sur la carte librement à l'aide d'une interface graphique en bas de l'écran en jeu. Cet aspect représentait, pour autant, la majorité de la description fournie par le cahier des charges.

6.3.7 Gestion des niveaux

Notre système de gestion de niveau nous satisfait amplement et correspond totalement à ce que nous souhaitions initialement faire. Celui-ci intègre également un éditeur de niveau ce qui n'était initialement pas prévu.

Nous aurions tout de même pu l'améliorer en rajoutant beaucoup plus d'objets posable via l'éditeur de carte afin de créer un bien plus grand diversité aux cartes sur lesquelles le joueur peut évoluer.

De plus, nous aurions aussi souhaité pouvoir faire des cartes sur plusieurs niveaux en créant par exemple des étages aux maisons et appartements dans lesquels le joueur peut évoluer, bien que nous n'avions pas eu l'idée lors de la rédaction du cahier des charges et n'avions donc pas mentionné l'idée. Malheureusement, ceci est complexe à mettre en place et n'était clairement pas une priorité pour nous à développé. Cependant notre choix de ne pas avoir implémenté ceci n'est pas nécessairement pénalisant et permet même de garder une clarté et une simplicité dans la compréhension de ce qu'il se passe durant la partie.

6.3.8 Site Web

Le site Web ne fait pas l'objet d'un quelconque retard, celui-ci ayant été livré et terminé pour la première soutenance. Des éléments supplémentaire tels que des captures d'écran, les rapports de soutenance et le téléchargement du jeu y ont été ajouté afin de les rendre disponibles à tout le monde en ligne.

6.4 Améliorations possibles sur les méthodes de travail

Parmi les méthodes de travail employées ne figurait pas de réelle gestion de projet. Chacun donnait de lui-même là où il le souhaitait, selon ses affinités et selon ce qu'il fallait faire sans organisation claire et précise, ni délais réels, menant à des approximations et retards.

L'un des majeurs défauts de cette approche menait d'abord à ce que certains d'entre nous ne puisse s'insérer correctement dans le projet ou au moins une de ses parties, que ce soit par crainte de casser un code existant, de créer des conflits Git (notamment lorsque les branches de chacun étaient fusionnées avec la branche principale), ou encore par mauvaise compréhension de la structure du projet.

Nous utilisons pourtant, dans le cadre de ce projet, un serveur de communication Discord qui nous permet à tout instant et de traiter rapidement les problèmes liés au jeu, nous transmettons nos avancées sous la forme de "Pull-Request" (demande de fusion des branches entre les branches personnelles et la branche principale) lorsque nous achevons une fonctionnalité ou la suppression d'un bug pour le jeu.

Cela ne suffit malheureusement pas et mène à de nombreux moments sans que le jeu ne progresse ou que très lentement, les membres du groupe ne sachant pas toujours ce qu'il était possible de faire et n'ayant pas une vision d'ensemble sur le projet. Cette logique de progression n'était par ailleurs que peu soutenable dans un autre cadre que celui d'un projet scolaire avec soutenances régulières forçant l'investissement de chacun dans les tâches sous peine d'abandon définitif du projet par manque de management.

Lors d'un projet futur, il est primordial d'assigner une personne qui ne s'occupe que, ou pour majorité, de l'orchestration de celui-ci, que les tâches se réalisent dans les temps et qu'il soit possible d'enchaîner entre elles afin d'éviter un manque de motivation de la part des membres de l'équipe. Cette personne doit impérativement posséder une vision d'ensemble du projet, idéalement connaître les bibliothèques utilisés afin de mieux adapter le projet et mener de bout en bout chacune des tâches.

6.5 Améliorations possibles sur le projet

De nombreux points peuvent être améliorés sur notre projet, cette section va les présenter de façon chronologique, dans l'ordre où nous les aurions réalisées si la soutenance avait été repoussée à une date ultérieure.

Nous aurions tout d'abord commencé par effectuer quelques améliorations sur le multijoueur pour rendre le déplacement des unités plus immersive en rajoutant notamment la possibilité de déplacer plusieurs unités en même temps.

La création du mode de jeu concernant la protection d'un VIP est la suite logique. Ce mode de jeu initialement prévue et présent dans la campagne peut être implémentée dans un temps raisonnable. L'ajout d'un mode de jeu augmentant la rejouabilité de notre projet.

Nous aurions ensuite pu ajouter plus de missions au mode campagne. Ceci en ajoutant de nouvelles missions à la campagne actuelles ou en créant un système permettant d'évoluer entre différentes cartes.

Après avoir effectué les modifications les plus simples, nous aurions pu améliorer les unités contrôlées par l'ordinateur pour les rendre plus réalistes. Rendre l'intelligence artificielle plus réaliste aurait augmenter l'intérêt du mode campagne en augmentant le réalisme des missions.

Nous aurions ensuite développé les visuels de notre jeu. Ces améliorations auraient été les plus difficiles pour nous étant donné que nous n'avons aucun graphiste dans le groupe. Nous aurions tout d'abord ajouté des interfaces en jeu avec un design plus développé. Des améliorations sur les animations auraient également effectué, ainsi que l'ajout de nouvelles musiques d'ambiance et de nouveaux sons déclenché lors des actions des joueurs.

Un des derniers points que nous aurions travaillé est l'ajout des aspects lié au joueur. Notamment le fait qu'un joueur puisse constituer son équipe d'agent avec un budget et une boutique. L'attribution d'armes et d'équipements variés sur les unités fait également partie de ces améliorations. Nous aurions également implémenté différentes nouvelles actions pour les unités qui ne se contenteraient donc pas de simplement se déplacer et de tirer.

7 Synthèse globale

Finalement, nous sommes globalement assez satisfaits de notre produit final, et surtout, nous avons tirés beaucoup d'enseignements et d'expériences différentes.

Malgré les améliorations possibles, nous avons quand même une très bonne charte graphique, et une bonne direction de cette dernière, créant une ambiance intéressante et une belle identité à notre jeu.

L'éditeur de carte qui est une feature qui n'était pas prévu du tout à la base se retrouve être une mécanique fondamentale de notre jeu, créant un stimulus créatif intéressant et permettant une grande rejouabilité.

Le mode de jeu Bombe disposal qui n'avait pas non plus été prévu à la base, a été rajouté, pour le bien de la campagne ainsi que des missions type escorte du vip. Ces gameplay différents sont là pour offrir une approche différente du jeu, rendant la campagne encore plus attrayante.

Le jeu n'est peut-être pas aussi complet que l'on avait prévu initialement, mais il présente quand même des bases solides et intéressantes.

Le site est bien fait, ainsi que ses traductions disponibles, pareil pour la jaquette. Le design est satisfaisant.

Malgré des charges de travail inégales, les membres du groupe ont tous contribué du mieux qu'ils peuvent au projet, chacun trouvant des domaines ou des spécificités du jeu vidéo dans lesquelles les autres ne pouvaient intervenir sans leur aide, ce qui a créé une belle complémentarité, cependant, la différence d'importance dans le projet de ces différents domaines aura créé des différences d'engagements, ainsi que les différences de compétences.

8 Synthèses personnelles

8.1 Arthur BARREAU

Ce projet a été une véritable immersion dans le monde de l'informatique, me permettant de développer mes connaissances et compétences de manière significative. Grâce à cette expérience d'envergure, j'ai pu apprendre à gérer des tâches complexes, à collaborer efficacement avec une équipe et à résoudre des problèmes techniques avec créativité. Ces compétences nouvellement acquises sont précieuses pour mes études futures, mais également pour ma carrière professionnelle, où je pourrai appliquer ces aptitudes de résolution de problèmes et de collaboration pour réussir dans des projets d'envergure similaires.

Ce projet m'a offert l'occasion de découvrir Unity et la conception de jeux vidéo, bien que j'aie initialement trouvé cet outil complexe à appréhender, j'ai réussi à surmonter cette difficulté et à le manipuler de manière satisfaisante. J'ai ainsi pu mettre en pratique mes compétences en programmation, vus au long de l'année, en utilisant le langage Csharp pour coder différentes fonctionnalités du jeu. En parallèle, j'ai également développé mes compétences en utilisant Overleaf pour rédiger les rapports et les cahiers des charges.

Ma gestion personnel de ce projet a été une expérience formatrice et essentielle pour moi. En jonglant entre mes cours, mes révisions et les exigences du projet, j'ai dû mettre en place une planification rigoureuse et une gestion efficace du temps pour respecter les échéances et atteindre les objectifs fixés. Travailler en équipe avec mes camarades a également renforcé ma capacité à collaborer, à communiquer efficacement et à coordonner nos efforts pour mener à bien le projet dans les délais impartis.

Ce projet m'a donc permis de développer des compétences humaines essentielles. Travailler en équipe m'a appris à respecter les délais et à collaborer de manière efficace. J'ai amélioré mes compétences en communication écrite en rédigeant des rapports détaillés sur notre travail et j'ai gagné en confiance à l'oral en présentant notre jeu devant les autres.

En résumé, ce projet a été une expérience très enrichissante dans laquelle j'ai découvert Unity, développé des compétences techniques et humaines, et appris à gérer un projet avec des échéances et des objectifs clairs tout en renforçant mes capacités de travail en équipe.

8.2 Alex DREAU

Ayant découvert la programmation cette année, j'avais beaucoup d'appréhension quand à l'arrivée de ce projet, le fait de ne pas avoir de consigne ni de guide step by step et d'être lâché de mon propre chef me faisait très peur.

Ce fut très compliqué de réussir à être indépendant à chercher sois même les ressources dans différents tutoriels et documentations présents sur internet.

Ce projet m'a appris à gérer mon temps, à sortir du cadre scolaire pour élargir mon champ de vision, m'a fait découvrir Unity, Mirror, et aussi le monde des animations 3d.

Mais surtout il m'a appris à me débrouiller, à chercher par moi même, à collecter des informations pertinentes, à utiliser intelligemment git, et à travailler en équipe efficacement.

Ces compétences seront je pense essentielles dans le monde du travail dans le domaine informatique plus tard, et je pense que ces méthodes sont bien plus primordialese et intéressantes qu'une bonne maîtrise du Csharp ou de Unity, donnez un poisson à l'homme qui a faim et vous le nourrirez un jour, apprenez lui à pêchez et il n'aura plus jamais faim.

8.3 Kévin LUBERT

J'appréhendais particulièrement ce projet de second semestre étant donné la probabilité que le choix du projet se termine sur le développement d'un jeu vidéo. Ayant déjà touché à ce domaine pour des petits projets personnels, j'étais conscient qu'il ne m'intéressait que très peu.

Pour ma première expérience dans le domaine du jeu vidéo, j'ai utilisé Unreal Engine un concurrent de Unity. Ce sont deux logiciels très différent dans leurs utilisations et dans les technologies utilisées. Il m'a donc fallut apprendre à utiliser Unity et ses spécificités pour le projet.

J'aurais préféré travailler sur un projet qui ne soit pas un jeu vidéo comme une application. Il est toutefois compliqué d'avoir une idée qui respecte les critères nécessaires pour que le projet soit validé, notamment la nécessité de posséder un élément en réseaux et une intelligence artificielle. L'obligation d'un élément en réseau est intéressante, mais éliminer les idées "classiques" qu'on peut avoir rapidement. En supposant avoir une idée intéressante et valide, il est également plus complexe de réunir un groupe sur un projet d'application plutôt que sur un projet de jeu vidéo.

Je suis toutefois satisfait d'avoir appris à utiliser Unity. Cette compétence me sera certainement utile dans le futur pour prototyper rapidement des idées ou pour réaliser des petits projets personnels.

Ce projet m'a également permis de travailler avec une équipe différente de celle avec laquelle j'ai l'habitude de travailler. Je travaille habituellement avec l'équipe qui m'a formé au développement, j'ai donc eu l'occasion de découvrir de nouvelles façons de travailler.

Finalement, je suis content d'avoir travaillé sur le jeu avec l'équipe et que le projet doit finaliser. Ce projet m'a beaucoup apporté, notamment la confirmation que le développement de jeu vidéo est un des domaines de l'informatique qui ne m'intéresse pas.

8.4 Romain SEZNEC

Ce projet fut fort en rebondissement, et pourtant, j'en tire qu'une expérience incroyable et ai apprécié chacun de ses instants.

Que ce soit car je me passion pour l'expression technique ou artistique en programmation, je trouve que le développement d'un jeu vidéo est un défi plutôt ardu par sa composition de tâches plus complexes les unes que les autres et regroupant tous les domaines liés de près ou de loin à l'informatique. Je trouve que c'est le meilleur moyen de mettre en avant ses compétences et sa progression dans le milieu.

En effet, bien que possédant une expérience antérieur dans la conception de projet, que ce soit avec l'élaboration sites Web et autres type de projet en rapport avec l'usage de divers Framework sous Node.JS, mais également en réseau avec la création d'un Intranet personnel sur serveur Linux configuré intégralement par mes soins, gérant ainsi les tables de routage par le Reverse-Proxy Nginx et pare-feu Iptables, configurations SSH/Wireguard et gestion de services sous Docker Swarm/Cluster Kubernetes, ou bien encore en jeu-vidéo avec la création récréative de jeu sur Roblox ou de la création de modifications pour Garry's Mod, je ne pouvais qu'apprécier l'idée de créer mon propre jeu sous Unity et monter en compétences dans ce domaine que je souhaitait tant découvrir.

Cependant, le fait de travailler en équipe est un point que je considère comme non acquis chez moi après le développement de Raiders Tactics, les projets mentionnés plus tôt n'ayant pas fait l'objet d'un travail d'équipe, je me suis créé une bulle de confort dans un travail seul. Il arrive ainsi à plusieurs reprises que je me donne une surcharge de travail par rapport à d'autres de mes camarades qui me proposaient pourtant leur aide sur certaines tâches. Il m'arrive également de progresser de façon peu saine sur le projet avec de nombreuses soirées et nuits entières passées à développer le jeu alors que d'autres obligations m'incombent et ne souhaitant pas admettre que je puisse diviser les tâches qu'il me reste à effectuer.

Je ne souhaite pourtant pas remettre en cause la cohésion de notre groupe que je ne peux que féliciter, nous sommes parvenu à rester unis du début à la fin du projet, traversant des moments forts comme la première écriture du cahier des charges et le tout début de la création du projet sous Unity qui nous rapproche et créé un réel esprit d'équipe mais aussi des moments durs comme la sortie de la seconde soutenance qui nous porte un coup au moral et un certain sentiment d'amertume pour lequel nous avons su faire face et redoubler d'efforts pour atteindre le résultat tant espéré. Je félicite l'ensemble de mes camarades pour leur investissement dans le projet et leur montée en compétence.

Bravo à tous.

9 Conclusion

Malgré les différents retards qui sont survenus, nous sommes satisfaits de notre rendu final du jeu. Certaines fonctionnalités que nous aurions aimé implémenter ne l'ont pas été, mais la majorité de nos idées sont intégrées au rendu grâce aux nombreuses heures de travail investi par les membres de l'équipe.

Ce projet a apporté certaines nouvelles compétences aux membres du groupe. Ces compétences sont le travail en équipe, l'usage de Git à plusieurs développeurs ou encore l'utilisation d'Unity.

Nous attendons avec enthousiasme les futurs projets de groupe ne concernant pas forcément du jeu vidéo de notre scolarité.