



EPITA RENNES

RAPPORT DE SOUTENANCE 1

Raiders Tactics



42 SECONDS TO EPITA

Alex DREAU
Arthur BARREAU

Kévin LUBERT
Romain SEZNEC

10 Mars 2023

Table des matières

1	Introduction	2
2	Conception du jeu	3
2.1	Découpage de projet	3
2.2	Multijoueur et réseaux	3
2.3	Intelligence artificielle	5
2.4	VFX et SFX	6
2.5	Gameplay	7
2.6	Interfaces	8
2.7	Gestion des aspects liés au joueur	8
3	Editeur de niveaux	10
3.1	Fonctionnement général du système	10
3.2	Sélection et intégration des modèles	11
3.3	Pistes d'améliorations à court terme	11
3.4	Objectifs pour la prochaine soutenance	12
4	Site Web	13
4.1	La <i>landing page</i>	13
4.2	Présentation des membres	14
4.3	Téléchargement des ressources	14
4.4	Multilinguisme et accessibilité	15
4.5	Points d'améliorations	16
5	Retour sur les méthodes de travail et améliorations à prévoir	17
6	Conclusion	18

1 Introduction

Raiders Tactics est un projet de jeu vidéo multijoueur versus et coopératif. Il est réalisé par l'équipe *42 Seconds To EPITA* composé d'Alex DREAU, Arthur BARREAU, Kévin LUBERT et Romain SEZNEC. Le projet est effectué dans un cadre pédagogique au sein de notre second semestre à EPITA.

Ce rapport nous permet de rendre compte de tout le travail que nous avons pu produire à ce jour, présentant l'avancement de chaque membre de *42 Seconds To EPITA* sur ce projet. Il présentera l'évolution de notre projet et les décisions du groupe vis-à-vis des annonces produites dans le cahier des charges pour veiller au bon avancement de sa production.

Le jeu 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 avec le match à mort par équipe (*Team Deathmatch*), la sécurisation de « VIP » et le désamorçage de bombe.

Ces modes de jeux se distinguent en deux modèles, l'un libre implémentant du multijoueur en coopératif ou versus, et l'autre en campagne seule avec différentes missions à réaliser. L'intelligence artificielle joue le rôle d'ennemie ou d'alliée en contrôlant l'escouade adverse selon les parties.

La campagne solo se caractérise notamment sur une marge de progression du joueur au fur et à mesure des missions proposées et cadencées par une boutique d'équipements et d'atouts. Les cartes de jeu et les objectifs y sont prédéfinis.

Quant au mode libre multijoueur, l'équipement de l'escouade du ou des joueurs est défini par un budget fixé et égal pour chacun d'entre eux ainsi que de l'IA le cas échéant. Un système de niveau et de statistiques reste présent à titre indicatif pour le joueur. Les cartes de jeu sont ici celles de la campagne avec l'apport d'un facteur aléatoire remplaçant certains éléments de la carte et leur emplacement pour améliorer la rejouabilité.

La direction artistique du projet repose sur une charte graphique cartoonesque et une ambiance sonore prenante et dynamique.

2 Conception du jeu

2.1 Découpage de projet

Nous avons réparti la création du projet de façon à ce que ce soit équitable, tout en prenant en compte les compétences de chacun. Chaque personne a cependant la possibilité d'aider qui que ce soit. Toutes les tâches à faire figurent dans le tableau suivant.

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

Chef de pôle Contributeur

Les différents éléments mentionnés dans la suite de cette partie ont été développé par Alex avec l'assistance d'Arthur. Ces tâches sont les bases qui nous permettront d'ajouter des fonctionnalités plus complexe pour la seconde soutenance.

2.2 Multijoueur et réseaux

Pour le multijoueur nous avons utilisés le framework Mirror, en modifiant les scripts de Network Management pour que cela corresponde avec nos choix en terme de Gameplay.

Ce fut assez long de comprendre comment fonctionne la librairie aux premiers abords, mais cela nous permet de contrôler exactement ce que le client (joueurs) peuvent envoyer et recevoir au serveur en terme de données. Cette distinction est invisible au joueur qui n'aura qu'à lancer une partie en tant qu'« hôte » avec son ordinateur faisant office, à la fois de client et de serveur comme le montre la figure 1.

Ce système requiert cependant une certaine attention lors de l'implémentation des éléments de Gameplay et des fonctionnalités surtout par rapport au fait que tout doit coïncider avec ce dernier.

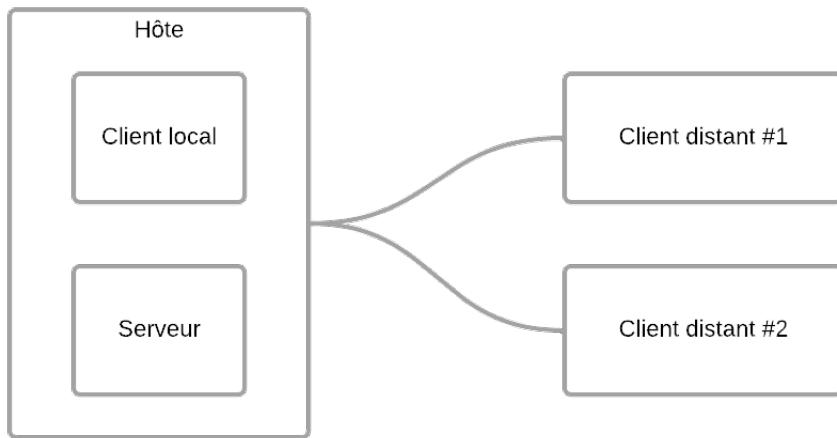


FIGURE 1 – Le joueur peut faire office de client et serveur



FIGURE 2 – Interface de menu multijoueur

Le but de notre implémentation du multijoueur est que deux joueurs se connectent rapidement l'un à l'autre, et que sans délais, ils puissent instantanément générer les unités avec lesquelles ils auront choisis de jouer. Et que chacun puissent avoir les bonnes informations et possibilités vis-à-vis de ses unités et pas de celles des autres.

De ce fait, il est possible pour les joueurs de se connecter sur une même partie avec l'un d'eux connecté en tant qu'hôte de session et les autres en inscrivant l'adresse IP du joueur hôte au travers d'une interface de menu tel que décrite dans la figure 2.

2.3 Intelligence artificielle

Pour le moment, nous n'avons pas pu implémenter beaucoup d'algorithmes pour notre IA.

L'algorithme qui permet au joueur de déplacer ses unités est pour le moment uniquement basé sur le clic. Il vérifie ensuite simplement que la zone autour du clic est accessible par l'unité et dans la zone de jeu.

Nous avons aussi un algorithme de détection, et de suivi (tracking) des unités adverses, chaque ennemi entrant dans le champ de vision défini d'un allié est ainsi traqué, et si il est à porté, l'allié tire sur le dit ennemi automatiquement.



FIGURE 3 – Les unités vont automatiquement viser les ennemis

La figure 4 présente un algorithme plus avancé du suivi de l'ennemi que nous implémenterons pour la seconde soutenance. Cet algorithme permet de savoir si un ennemi doit être visible par le joueur.

L'avancement limité sur l'implémentation des algorithmes liés aux déplacements automatique des unités ennemis s'explique par le fait que nous attendons d'obtenir un système fonctionnel complet pour pouvoir faire des choix beaucoup plus pertinent quand à nos implémentations de ces derniers sur nos phases de jeux.

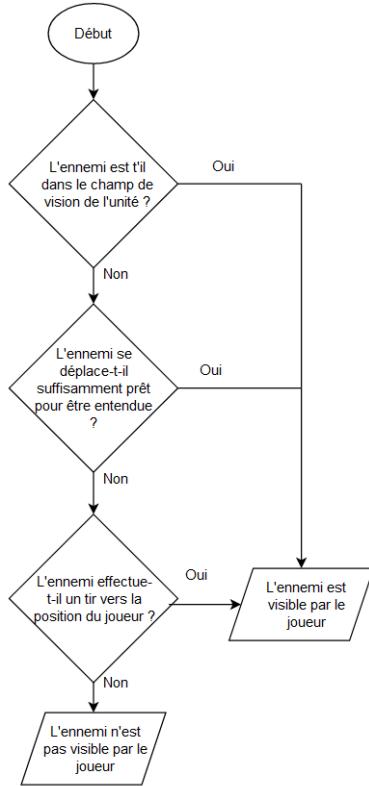


FIGURE 4 – Algorigramme plus avancé sur la détection des ennemis

Il faut déjà voir comment va s'organiser les parties entre les joueurs dans le détail, si les espérances et les attentes que l'on avait vis-à-vis de cela correspondra à la réalité du gameplay.

Nous aimerions pour la prochaine soutenance avoir un bon algorithme pour que les unités suivent des chemins optimisés (Pathfinding) ; et surtout de la "Local Avoidance" pour éviter que plusieurs unités du joueur ne se rentrent dedans ou ne s'empile de manière gênante pour pouvoir les sélectionner.

2.4 VFX et SFX

Comme dit dans notre cahier des charges, nous n'avons pas encore introduit ni de VFX ni de SFX dans notre jeu pour cet première soutenance.

Pour l'ambiance sonore, nous commencerons d'abord par tout ce qui est bruit de pas, de balles, de recharge. Cela fera partie intégrante du gameplay final.

Pour tout ce qui est VFX, nous chercherons les effets de tir en priorité.

Pour la seconde soutenance nous chercherons à implémenter une première version de ses différents effets.

2.5 Gameplay

Notre première approche du gameplay a été de placer la caméra en vue de dessus, avec un joueur qui n'a pas d'existence physique dans le jeu. Le joueur, comme prévu dans le cahier des charges, a la possibilité de déplacer sa caméra avec les flèches de son clavier ou à l'aide des touches W, A, S et D. Il a également la possibilité de zoomer ou de dézoomer sur la carte et de faire effectuer une rotation sur la droite ou la gauche à sa caméra. Le joueur ne peut cependant pas modifier l'angle que sa caméra a avec le sol.

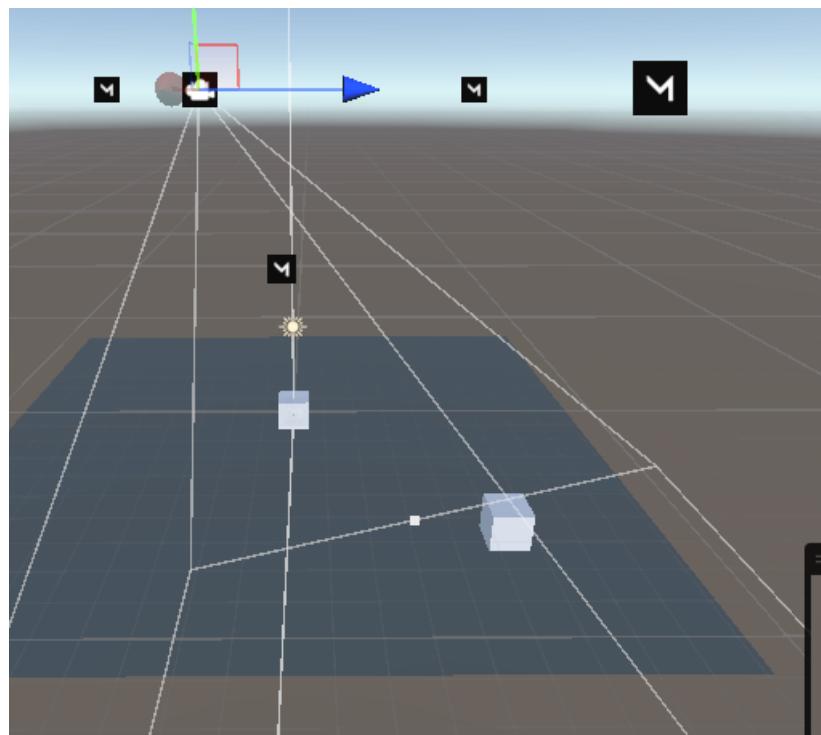


FIGURE 5 – Angle de la caméra lors du lancement de la partie

Quand il se connecte à un serveur, un point de génération d'unités apparaît sur le projeté orthogonale du joueur par rapport au plan, le joueur a ensuite une autorité sur les unités qui sortiront de ce point de génération. Pour la seconde soutenance nous remplacerons le générateur d'unité par un nombre prédéfini d'unités au début de la partie.

Cet angle de caméra permet de centrer le jeu autour de la gestion des unités et de leur déplacement, il met l'emphase sur le côté tactique du jeu. En posant cela, nous

avons décidé que les unités tireront automatiquement, avec la possibilité à l'avenir de donner des ordres du type "Fire at will" (Feu à volonté), "Hold your fire" (Ne pas tirer), pour renforcer les possibilités d'interactions tactiques.

Comme mentionné précédemment la difficulté ici est de bien implémenter les différences Serveur / Client. Pour éviter, par exemple, que des informations cruciales menant à la victoire ou la défaite d'un des joueurs soient perdues, non envoyées ou puisse être manipulé de façon illégitime pour avantager un joueur.

Pour l'instant notre système fonctionne sur la partie de la gestion d'autorité des joueurs, des déplacements des unités, le fait qu'ils suivent les ordres du joueur, le "tracking" (Suivi.) des unités adverses, et finalement le tir sur un ennemi.

Pour la seconde soutenance nous devrons ajouter un système de vie aux unités. Il faudra également améliorer les systèmes déjà en place.

2.6 Interfaces

Le menu du jeu donné en figure 6 en est encore à sa phase préliminaire mais possède déjà les modes de jeu les plus importants. Celui-ci se décompose ainsi du logo du jeu, du mode campagne, du mode multijoueur et de l'éditeur de carte. Il est possible de quitter le jeu depuis cette fenêtre. Un fond composé de personnages du jeu permet d'y ajouter une certaine vie.

Quant à une interface en jeu type "HUD" (Head up display), nous n'en avons pas encore implémenté et commencerons celle-ci comme prévue dans le cahier des charges. Les premiers éléments qui pourront être ajouté seront les informations sur les unités du joueur et les premières information sur la partie.

La figure 7 est un exemple d'interface en partie que nous allons commencer à implémenter dans les semaines à venir, 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.

2.7 Gestion des aspects liés au joueur

Le travail sur les aspects liés au joueur n'a pas encore commencé. Toutefois, le travail sur la création d'une unité est une base qui permettra de commencer une première ébauche de cette partie pour la prochaine soutenance.



FIGURE 6 – Menu du jeu en phase préliminaire

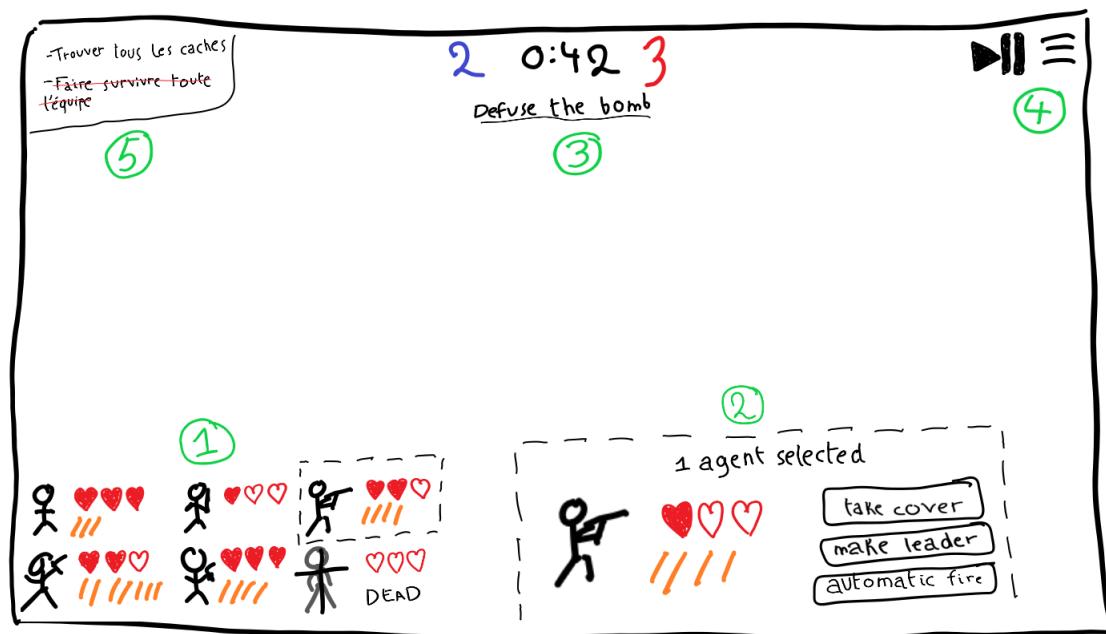


FIGURE 7 – Rappel du Concept Art de l'interface en jeu

3 Editeur de niveaux

Pour la première soutenance, nous avions prévu d'avoir effectué uniquement une carte de test qui ne serait relié à aucun système de modification ou de sauvegarde.

Lors du début du développement nous nous sommes rendu compte qu'il était important de commencer une première base pour le système de niveau dès le début.

Nous sommes donc en avance sur cette partie du projet. Cette avance nous permettra de prendre plus de temps pour développer les fonctionnalités liées à ce système ou à d'autres fonctionnalités d'ici à la prochaine soutenance.

Avoir un éditeur de niveau à notre disposition dès le début nous a permis de pouvoir travailler sur le système qui sera utilisé plus tard et de modifier directement les niveaux pour tester au mieux les fonctionnalités.

Celui-ci a été développé par Kévin avec l'assistance de tous les membres du groupe pour les questions de design.

3.1 Fonctionnement général du système

Le système actuel de construction est extrêmement basique et n'intègre pas d'interface. Il repose sur une grille dont la taille représente la taille du niveau.

Chaque case de la grille ne peut posséder qu'un seul objet à sa surface en excluant la texture du sol et des 4 murs.

Pour construire le joueur devra pour le moment utiliser un système archaïque. Les éléments qui ont la possibilité d'être posé sont mis à la suite, pour passer d'un élément à l'autre le joueur appuiera sur un bouton de son clavier, si le joueur appuie suffisamment longtemps sur la touche de changement pour arriver à la fin de la liste d'objet à placer, il est renvoyé au début. Pour placer, il lui suffit alors de cliquer sur le bouton 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.

Il suffira ensuite au joueur de charger cette sauvegarde pour voir sa création apparaître à l'écran.

3.2 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.

Pour la deuxième soutenance, nous allons faire une sélection des textures et modèles 3D que nous allons utilisés et nous les intégrerons dans l'éditeur de niveau.

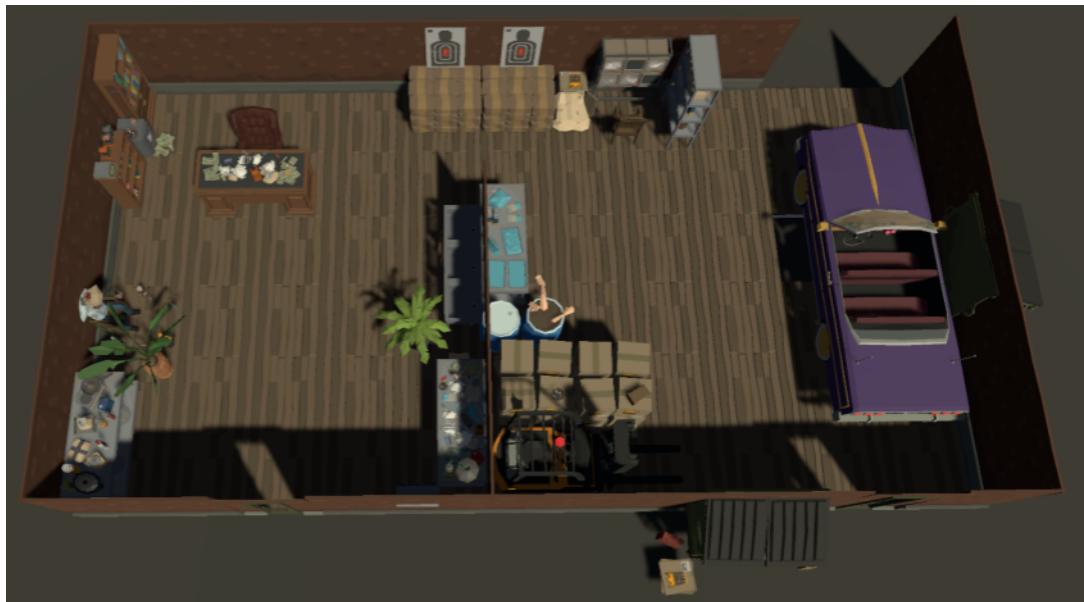


FIGURE 8 – Exemple de réalisation possible actuellement

3.3 Pistes d'améliorations à court terme

Nous allons citer ici des pistes d'améliorations que nous aurions mis en place rapidement si nous avions eu plus de temps et que nous allons effectuer rapidement avant de commencer les fonctionnalités principales prévues pour la prochaine soutenance.

La première nécessité actuelle est de rendre le système de création des cartes plus intuitif. Pour cela, nous intégrerons une interface de sélection des objets que le joueur placera. Il faudra également ajouter des boutons pour les fonctionnalités comme la suppression ou la rotation pour rendre le système plus intuitif et ne pas forcer le joueur

à apprendre une combinaison complexe de touche pour simplement pouvoir créer une carte.

Une seconde piste d'amélioration est possible au niveau du système de sauvegarde des cartes. Il est nécessaire d'ajouter une plus grande vérification sur la validation des fichiers et de rajouter un retour visuel au joueur s'il essaye de charger un fichier invalide.

Une dernière piste d'amélioration serait une réduction de la taille des cases utilisées par le système. Les cases actuelles sont particulièrement grandes et un rétrécissement permettrait au joueur une plus grande précision lors du placement des objets. Cette proposition devra être étudiée après la sélection des modèles définitifs.

3.4 Objectifs pour la prochaine soutenance

Pour la prochaine soutenance, nous souhaitons avoir implémenté les points d'améliorations cités précédemment pour rendre plus ergonomique le système de modification de carte.

Nous allons également ajouter des notions d'aléatoires pour varier les cartes créées par le joueur. Une des pistes que nous étudierons pour cette fonctionnalité est de permettre au joueur de placer un type général d'objets, par exemple un objet générique "lampe", qui sera ensuite remplacée aléatoirement par une des lampes disponibles.

Nous commencerons également à placer les bases d'une génération plus procédurale des cartes.

4 Site Web

Le site Web du projet en est à son point d'avancée anticipé par le cahier des charges, c'est-à-dire rendue à une version fonctionnelle prête à la mise en ligne pour cette soutenance. Le site est dès à présent accessible par le lien suivant : <https://raiders-tactics.com/>.

Les points clés de son architecture actuelle en sont : la *landing page*, page sur laquelle l'utilisateur « atterrit » à son arrivée 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.

4.1 La *landing page*



FIGURE 9 – 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.

4.2 Présentation des membres

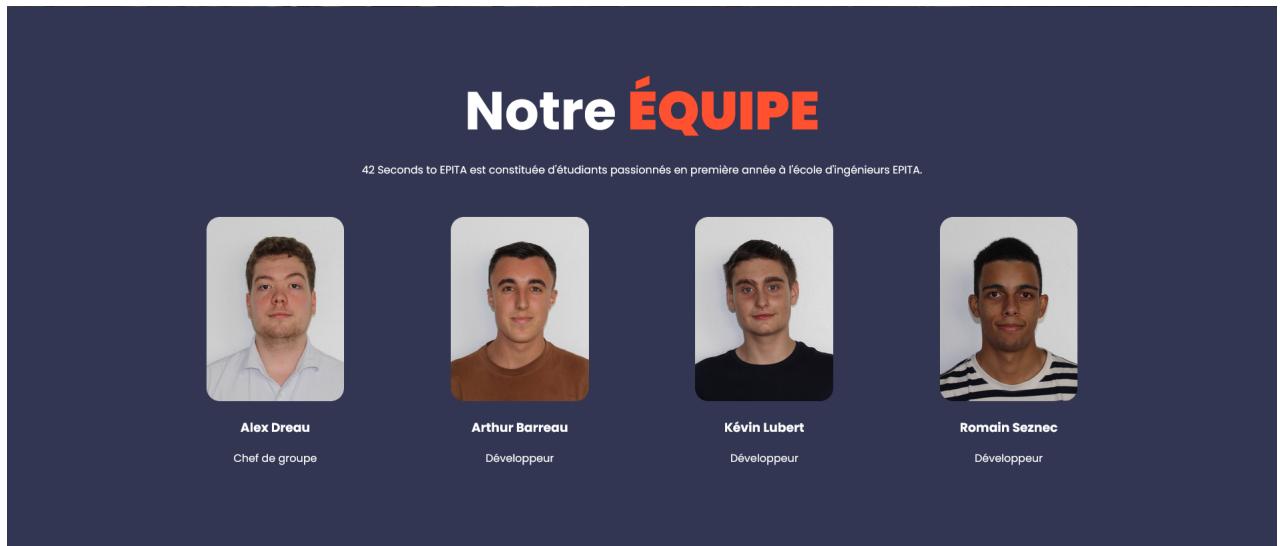


FIGURE 10 – Section de téléchargement du site Web

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.

4.3 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 ne possédant pas pour l'heure de version exécutable publique, une courte phrase incite l'utilisateur à rester informer quant au développement de celui-ci.

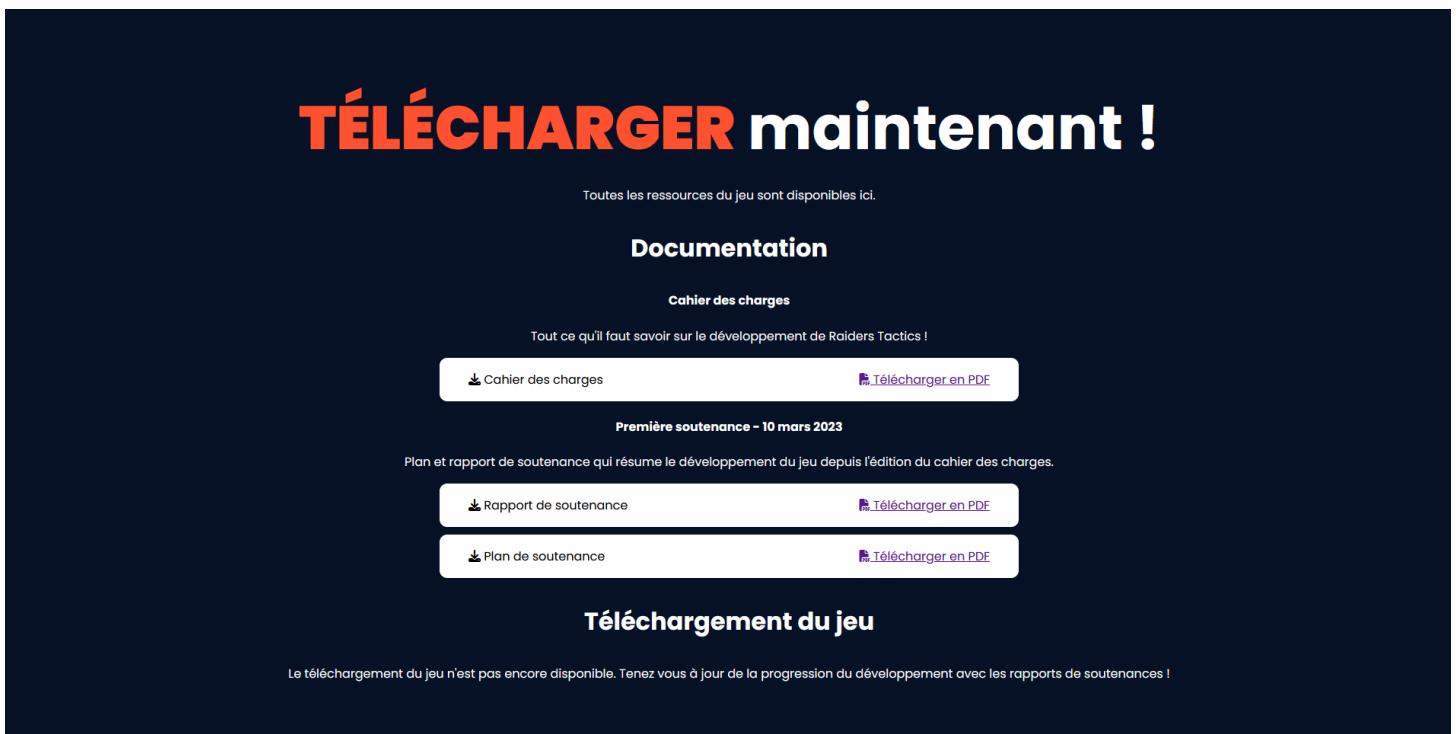


FIGURE 11 – Section de téléchargement du site Web



FIGURE 12 – Barre de navigation du site Web

4.4 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.

4.5 Points d'améliorations

Les points d'améliorations du site découlent directement de l'évolution du jeu vidéo ce qui permettra, à l'avenir, d'y inclure des éléments de jeu telles que captures d'écran, trailers vidéo éventuels et ses fonctionnalités.

Il serait également intéressant d'inclure des précisions quant à la présentation des membres avec d'avantage de détail sur l'impact de chacun sur le projet et de ses ressentis.

5 Retour sur les méthodes de travail et améliorations à prévoir

La réalisation d'un jeu vidéo nous fait prendre violemment conscience de la difficulté de la tâche. Le projet est ambitieux et il faut s'organiser un maximum pour ne pas se laisser prendre de court sur nos prévisions du cahier des charges. Ce n'est pourtant pas sans compter l'ajout de certains imprévus.

En effet, l'une des premières difficultés est inhérente au travail de groupe et repose sur la répartition de nos tâches et de l'architecture du projet. Nous prenons rapidement avantage de l'outil Git avec la création d'une organisation Github et de dépôts associés au jeu pour mutualiser nos travaux individuels mais aussi pour garder une trace des versions précédentes des modifications.

Il aura ainsi fallu, en outre des connaissances acquises à l'école, nous plonger dans le fonctionnement des branches et leur fusion sur Git, celles-ci pouvant s'accompagner de conflits dans certains cas précis qu'il nous faut résoudre correctement pour ne pas perdre nos travaux.

Nous nous sommes également rendu compte, lorsque nous tentions d'ajouter nos assets et modèles, des limites de stockage imposées par Github. Nous avons ainsi dû recourir à la création d'un sous-module Git que l'on héberge sur le Gitlab de l'école, le tout en faisant usage de Git LFS (Large File Storage) pour nous permettre de stocker les fichiers volumineux dans notre dépôt.

Ces points de difficulté ne se limitent pas seulement à la logistique de nos fichiers, il nous aura également fallu nous plonger longuement dans diverses documentation et assister à de nombreux tutoriels pour comprendre certains aspects de développement de jeu vidéo, domaine encore inconnu par les membres du groupe au début du projet.

Nous sommes cependant heureux de ne pas tirer de réels incident quant à la réalisation du projet sur ce premier tiers de développement. De fait, nous pensons que nos méthodes de travail qui consistent à la réalisation d'une mise au point chaque semaine de nos avancées, nos difficultés et nos anticipations est une bonne méthode de travail à conserver bien qu'il ne nous faille pas oublier les cours qui prennent une part bien plus importante que le projet sur notre temps libre.

6 Conclusion

Le développement de Raiders Tactics représente la réalisation d'un travail acharné mais pour autant, ce premier tiers de projet s'est bien déroulé dans l'ensemble. Bien qu'il existe un champ d'amélioration immense, nous avons réussi globalement à tenir nos objectifs donné dans le cahier des charges.

Pour la prochaine soutenance, outre les points mentionnés précédemment, nous anticipons de nombreux points à travailler tels que l'ajout des différents modes de jeu pour le mode multijoueur, puis implémenter une version primitive de la campagne solo avec quelques missions à titre démonstratif.

Ces éléments impliqueront la mise en place et de l'utilisation des assets et modèles mentionnés pour la charte graphique sous forme de cartes de jeu fonctionnelles et jouables. De plus les éléments d'interface devront être implémentés dans l'interface en jeu mais également l'intégralité des menus qui sont actuellement sous une forme non-définitive.

Nous gardons notre enthousiasme et bon espoir de terminer le jeu dans les temps.