

Rapport de projet

RageQuit



Groupe Souka :
Arthur Pauron
Fabien Lehoux
Alexandre Svilarich
Donovan Niepceron

8 JANVIER 2019

Table des matières

1	Introduction	4
2	Reprise du cahier des charges	5
2.1	Origine et nature du projet	5
2.1.1	Origine du concept	5
2.1.2	Description du jeu	5
2.2	Etat de l'art	6
2.2.1	Les précurseurs du genre	6
2.2.2	Les incontournables	7
2.3	Le projet en détail	8
2.3.1	Objectifs et fonctionnalités	8
2.3.2	Apport du projet	9
2.4	Organisation du projet	10
2.4.1	Outils et matériels	10
2.4.2	Répartition des tâches	11
2.4.3	Estimation des délais et des coûts	12
3	Avancement	14
3.1	Soutenance I	14
3.1.1	Site (Donovan)	14
3.1.2	Carte (Fabien)	16
3.1.3	Intelligence Artificielle (Fabien)	17
3.1.4	Scénario (Alexandre & Arthur)	18
3.1.5	Joueur (Arthur & Alexandre)	18
3.1.6	Menu (Alexandre)	23
3.1.7	Réseau (Donovan)	24
3.2	Soutenance II	25
3.2.1	Site (Donovan)	25
3.2.2	Carte (Fabien)	26

3.2.3	Intelligence Artificielle et Créatures (Fabien & Arthur)	27
3.2.4	Menu (Alexandre)	30
3.2.5	Réseau (Arthur & Donovan)	35
3.3	Soutenance III	35
3.3.1	Sample - Partie (auteur)	35
3.3.2	Menu (Alexandre)	35
3.3.3	Sénaire et animation (Arthur)	36
3.3.4	Carte (Fabien)	39
3.3.5	Site (Donovan)	39
3.3.6	Réseaux (Donovan)	39
4	Récit de Réalisation	41
4.1	Fabien	41
4.2	Alexandre	41
4.3	Arthure	42
4.4	Donovan	42

1 Introduction

Vous trouverez dans ces pages le récapitulatif du travail accompli par notre groupe sur le projet. Pour rappel, notre jeu est un jeu de plates-formes créé avec Unity, possédant un mode multijoueur et un scénario basique. Le projet est né de la rencontre de quatre étudiants de l'EPITA (dont trois en A2) qui partagent une même passion pour l'informatique et pour le jeu vidéo. Nous nous sommes donc naturellement regroupés pour former le groupe Souka.

2 Reprise du cahier des charges

2.1 Origine et nature du projet

2.1.1 Origine du concept

Après concertation, nous avons décidé de partir sur le principe d'un jeu de plates-formes, le but étant de créer des mécaniques de jeux intéressantes pour finir chaque niveau. Pour ajouter du rythme nous avons décidé d'inclure des éléments de jeux procurant une pression sur le joueur pour l'obliger à terminer le niveau le plus vite possible. Nous rajouterons également des ennemis pour rendre le jeu plus dynamique.

Le titre Rage Quit a été choisi pour marquer la difficulté de notre jeu, mais aussi pour illustrer l'objectif de notre personnage, qui est de s'échapper ("rage quitter") de la planète. Notre jeu se déroulera sur une planète inconnue colorée dont la végétation est abondante.

2.1.2 Description du jeu

Il s'agit d'un jeu de "plates-formes", il aura un mode solo, avec un scénario, et un mode multijoueur (allant de 2 à 4 joueurs).

Nous avons pour ambition de rendre le mode solo difficile et immergent au travers d'un univers coloré et plein de végétation extraterrestre. Le joueur devra traverser plusieurs niveaux pour atteindre la fin du jeu. Chaque niveau contiendra plusieurs énigmes et/ou ennemis à contourner et/ou à éliminer. Ainsi le joueur devra être concentré, trouver divers objets et chemins pour sortir de chaque niveau. Plusieurs sorties étant possibles, le joueur ne sera pas contraint d'en suivre une unique. Ainsi, un même objectif sera atteignable de différentes manières.

Le joueur a la possibilité d'incarner deux personnages principaux : un homme ou une femme, afin que celui-ci (ou celle-ci) puisse s'identifier à son personnage.

De nouveaux niveaux seront construits spécialement pour le mode multi-joueur, où deux à quatre joueurs pourront s'allier ou s'affronter pour sortir du niveau. Le premier sorti ou le seul survivant gagnera la partie.

Nous avons limité le nombre de joueurs à quatre, en effet le but est de tester les capacités des joueurs, le fait d'avoir trop de joueurs limiterait cet aspect. La pression ici ne vient pas du niveau, mais des joueurs eux-mêmes.

En effet les joueurs pourront s'allier pour sortir plus vite des niveaux ou trouver des passages qui ne s'ouvrent qu'à cette condition. Mais le but ne change pas : le premier sorti gagne, les joueurs devront se trahir, poser des pièges, se bloquer ou parfois même causer la mort des autres participants pour espérer gagner.

Les joueurs ne sauront pas où les autres se trouvent sur la carte, l'exploration et le suspens de croiser un autre joueur ainsi que la manière dont il vont réagir seront les éléments accrocheurs du mode multijoueur.

Ainsi plusieurs stratégies seront possibles, comme se cacher pour profiter des trouvailles des autres joueurs, piéger un joueur sur une plate-forme, s'allier pour tenter de se débarrasser des autres participants ou encore faire une alliance.

Synopsis : Alors que vous vous dirigez vers une planète pour l'analyser, votre vaisseau s'est écrasé sur cette dernière suite à une panne. Vous vous réveillez sur une île flottante, votre éjection vous a conduit dans les entrailles de la planète. Heureusement, vous n'avez pas les jambes brisées. Vous devez maintenant remonter à la surface de celle-ci en récupérant des échantillons et des morceaux de votre vaisseau. Vous n'êtes pas seul, des animaux et aliens vous empêcheront de remonter.

2.2 Etat de l'art

2.2.1 Les précurseurs du genre

Le jeu de plate-forme est inspiré des bornes d'arcade. En effet le jeu de plate-forme peut se rapprocher des premiers jeux d'arcade par plusieurs points : le personnage principal a un objectif clair sans trame narrative approfondie il y a des dangers à esquiver ou détruire un score à toujours battre

Malgré ses ressemblances, le jeu de plate-forme se distingue tout de même des jeux d'arcade. Le combat peut être présent dans le jeu mais le joueur est souvent limité ou du moins plus faible que ses ennemis. Le sentiment de puissance

que l'on a dans les jeux d'action est absent du jeu de plate-forme ce qui crée une certaine tension chez le joueur, celui-ci cherchera à éviter le plus possible la confrontation. Le joueur doit aussi parfois résoudre des énigmes, trouver des objets pour progresser dans le jeu. Les jeux utilisent des thèmes doux et fantaisistes, et le joueur est souvent immergé dans des environnements joyeux et colorés. Il y a peu de personnages non joueurs, en effet le but étant d'isoler le joueur qui doit se sentir seul face aux menaces inconnues.

Le précurseur du genre est Donkey Kong, un jeu édité par Nintendo en 1981. Dans ce jeu, le joueur incarne un charpentier ayant pour but de sauver une princesse. Ce dernier doit éviter tous les tonneaux, c'est-à-dire : sauter par-dessus ou les détruire. Pour ce faire, il doit récupérer un marteau en avançant dans le niveau. La particularité de ce jeu vient du fait que le joueur se doit absolument de survivre face à un singe géant qu'il ne peut pas tuer.

2.2.2 Les incontournables

Comme le nom de ce point l'indique, nous présenterons ici quelques-uns des jeux incontournables du genre "plates-formes".

Le premier est Mario 64, développé par Nintendo en 1995. On y incarne Mario, un plombier moustachu enquêtant sur la disparition de peach, la princesse. Il entre dans le château de peach pour la délivrer de bowser le roi des tortues. Le jeu étant très coloré, son utilisation est très reposante et permet une bonne immersion. Cependant, le joueur doit utiliser des objets (trouvées dans le jeu) pour l'aider à trouver des étoiles et ainsi pouvoir enquêter en s'enfonçant dans le château. Le joueur fait face à de nombreux ennemis . Dépourvu d'arme, il est dans l'obligation de sauter sur la tête de ceux-ci pour les tuer. Mario 64 se démarque des autres jeux de plate-forme car le joueur peut choisir quel objectif il réalise en premier pour récupérer les étoiles qui lui permettent d'avancer dans les niveaux.

Le suivant de cette liste Ori and the Blind Forest développé par Moon Studios sorti le 11 mars 2015. Le joueur incarne un petit esprit sylvestre nommé Ori, dont la forêt dans laquelle il vit s'est considérablement dégradée à la suite d'évènements mystérieux. Il lui faudra affronter moult et moult énigmes et épreuves pour redonner vie à l'arbre qui protège la forêt. Ori and the Blind Forest fut extrêmement bien reçu par le public car ses graphismes émerveillent les joueurs.

En dernier, On peut citer le jeu Cuphead, créé par le studio MDHR Entertainment Inc. Il s'agit d'un jeu de plate-forme/action, très dynamique, avec de nombreux affrontements épiques contre des boss à la fin des parcours. Sa patte graphique est très fortement inspirée des cartoons américains des années 30.

2.3 Le projet en détail

2.3.1 Objectifs et fonctionnalités

Installation et désinstallation	
Menu du jeu	<ul style="list-style-type: none">— Bouton Jouer— Bouton Quitter— Options (voir la partie sur les paramètres de jeu)— Bouton accès au multijoueur— Arrière-plan
Paramètres	<ul style="list-style-type: none">— Résolution (nombre de pixels * pixels)— Pleine écran, fenêtré— Son du jeu, des dialogues, des éléments extérieurs— Commandes— Sensibilité horizontale, verticale
Jeu Général	<ul style="list-style-type: none">— Mouvement du personnage— Deux personnages à modéliser (homme ou femme ou autres)— Ennemis (IA, plusieurs personnages)— Création de chaque objet à utiliser pour les maps— Endurance du personnage— Interface (Menu echap : retour au menu, quitter le jeu,)— Interaction avec certains objets— Musiques— Différents niveaux (étages)— Mode invincibilité (pour la présentation)

Solo	<ul style="list-style-type: none"> — Scénario (Pendant les niveaux) — Sauvegardes — Eléments scriptés
Multijoueur	<ul style="list-style-type: none"> — 2 à 4 joueurs — Gestion des caméras — Gestion réseau

2.3.2 Apport du projet

Pour le groupe :

Nous étions tous impatients de démarrer ce projet. Nous allons mettre en place des méthodes afin d'assurer un travail d'équipe efficace, tel qu'un test organisé toutes les semaines afin de faire le point sur les avancées, les difficultés et les découvertes de chacun. Cela permettra aussi de régulièrement vérifier que nos travaux sont compatibles. Tout au long de ce projet nous espérons nous améliorer au niveau du travail d'équipe et de l'organisation.

Individuellement :

Alexandre : Comme tous ici, j'ai toujours voulu passer du côté de joueur à celui de développeur. De plus la réalisation de ce projet me permet d'apprendre à travailler et s'organiser en équipe, ainsi qu'à maîtriser les outils qui le permettent. Cela me permettra aussi d'acquérir de l'expérience dans les domaines que nous allons explorer lors de la réalisation du projet. Enfin, je pense que les soutenances me forceront à m'améliorer à l'oral, ce qui n'est pas un mal.

Arthur : J'ai toujours souhaité créer un jeu vidéo, mais je n'avais pas encore les outils nécessaires. C'est pour ça que j'attends beaucoup de ce projet j'aimerais qu'il m'apporte la passion de création ainsi que de la fierté. Il me permettra de m'habituer au travail de groupe et ce qu'il représente. Le travail de groupe me semble très sympathique même si j'appréhende d'être obligé de restreindre mes idées de gameplay car elles ne font pas l'unanimité.

Donovan : Étant joueur régulier, créer mon propre jeux vidéo est un rêve de longue date. Le fait de réaliser ce projet en groupe nous permettra de développer des fonctionnalités plus poussées et plus travaillées, ainsi que de confronter nos idées afin d'aboutir à un projet original. La réalisation de ce projet apporte en plus une meilleure compréhension du fonctionnement des jeux vidéo.

Fabien : Depuis longtemps j'ai voulu créer mes propres jeux avancés, loin des jeux rétro en 2D. Cependant je n'avais ni les compétences, ni l'équipe pour cela. De plus le travail d'équipe permettra de réaliser des éléments plus élaborés pour pouvoir faire un jeu au plus proche de notre projet.

2.4 Organisation du projet

2.4.1 Outils et matériels

Latex	Souvent utilisé par des personnes nécessitant une mise en forme particulière. Pour cette raison, il est particulièrement utilisé par des étudiants, éditeurs, professeurs de mathématiques et de physique, scientifiques et principalement en informatique, en ingénierie, en mathématiques et en physique.
Unity	Il est l'un des moteurs de jeu les plus répandus du fait de sa rapidité et de sa portabilité. Il a la particularité de proposer une licence gratuite sans limitation au niveau du moteur.
Autodesk	Site de création d'objets modifiables grâce à un moteur graphique, permettant un choix diversifié et personnalisé, compatible avec de multiples plates-formes.
Microsoft visual studio	Environnement de développement pour les langages C développés par Microsoft, permet de coder en C# les scripts reliés directement à la plate-forme d'Unity.
Google Docs	Partage en continu un dossier entre plusieurs utilisateurs de google, disponible via Google Drive, (inclus Google Docs).

GitHub	Hébergeurs de projets. Le site offre de nombreuses fonctionnalités habituellement retrouvées sur les réseaux sociaux comme les flux, la possibilité de suivre des personnes ou des projets. GitHub offre aussi la possibilité de créer un wiki et une page web. Le site offre aussi un logiciel de suivi de problèmes. GitHub propose aussi l'intégration d'un grand nombre de services externes, tels que l'intégration continue, la gestion de versions, badges, chats basés sur les projets.
--------	---

2.4.2 Répartition des tâches

Tâches :	Alexandre	Arthur	Fabien	Donovan
Site internet			S	T
Scénario :	T	S		
Installation et Désinstallation :		S		T
Paramètres du jeu	T		S	
Bruitage et sons			T	S
Animations et Cinématiques		T	S	
Intelligence artificielle	S		T	
Réseau		S		T
Modélisation Personnages et Monstres	S	T		
Trailer du jeu et vidéo de gameplay		T		S
Création d'objets et composants de jeu	S		T	
GitHub	T			S
Création des niveaux	S		T	

2.4.3 Estimation des délais et des coûts

Semaines du	Tâche à réaliser	Avancement
1/10/2018	Lancement du site internet Mise en place du serveur GitHub	20% 100%
8/10/2018	Utilisation d'Unity, Premiers objets et scripts. Suivi de tutorielle pour se familiariser avec le logiciel.	N/A
8/10/2018	Initialisation du Réseau	65%
15/10/2018	Conception d'un premier décor	50%
15/10/2018	Premiers ennemis	20%
22/10/2018	Premier menu Personnage jouable Décor Scénario L'objectif est de présenter quelques petits éléments de jeu afin d'avoir une idée du rendu final.	100% 80% 20% 60%
29/10/2018	Niveau solo et premières scènes de scénario.	20%
5/11/2018	Mise en place du réseau Premier niveau en réseau	30%
12/11/2018	IA mise en place	30%
19/11/2018	Sons et ambiance Objets utilisables	20% 30%
19/11/2018	Début du jeu au point et jouable. Montage d'une première version. IA mise en place.	100% 100% 50%
26/11/2018	Semaine de tests, correction des bugs et entraînement pour la soutenance.	N/A

3/12/2018	Menu du jeu Personnages principaux Niveaux jouables Scénario implémenté dans le jeu Niveaux en réseau jouables L'objectif est de montrer une démonstration de notre jeu, nos objets et décors seront créés. Il ne restera plus qu'à construire le reste des niveaux et terminer l'IA des ennemis et le reste du scénario.	100% 70% 30% 40% 30%
10/12/2018	Construction de niveaux et suite du scénario.	70%
17/12/2018	Dernier Niveau et fin du scénario. Niveaux et histoire. Travail sur la boîte du jeu et le rendu CD.	100% 100% 100%
24/12/2018	Détails, correction de bugs et autres finitions.	100%
31/12/2018	Entrainement pour l'oral.	N/A
14/01/2019	Niveaux complets (Réseau et solo) Scénarios Installation et Désinstallation Disponible sur le site Trailer du jeu	100% 100% 100% 100% 100%

Le rapport du projet sera rédigé tout au long de notre projet, nous tiendrons tous un journal de bord dans lequel nos impressions, nos difficultés et nos réussites seront notées, et qui sera présent dans chaque rapport de soutenance.

Au niveau des coûts, le but est de développer un maximum d'éléments par nous-mêmes et d'éviter les achats d'asset ou de matériel. Le site web sera hébergé sur une plate-forme d'hébergement gratuite.

3 Avancement

3.1 Soutenance I

3.1.1 Site (Donovan)

J'avais déjà réalisé quelques sites, en langages HTML et CSS. Grâce à des tutos en ligne, j'ai appris à maîtriser le HTML et CSS qui sont véritablement les "briques" et la "peinture" de la page internet.

Le site est donc réalisé avec ces deux langages :

- HTML permet de gérer le contenu de la page web (textes, liens, images)
- CSS permet de gérer le design du site (coloration, positionnement, arrière-plan, etc.).

Le site que j'ai créé est basé sur le fond avec du HTML. Avec celui-ci j'ai créé une page d'accueil contenant toutes les nouvelles concernant l'avancement du projet ainsi que sa présentation et les différentes soutenances.

Puis j'ai créé une page nommée "l'équipe" où vous pouvez retrouver une présentation sur chaque membre du groupe, une autre page nommée "le jeu" qui contient des informations de suivi sur le jeu comme des images ainsi qu'une présentation détaillée du jeu vidéo. J'ai aussi implémenté une page qui permet de télécharger RageQuit dans sa version la plus à jour et de récupérer les rapports de soutenance.

Pour terminer j'ai créé la page qui permet de contacter mon équipe pouvant corriger un éventuel dysfonctionnement du jeu. Un formulaire est mis à disposition des utilisateurs pour faciliter la prise de contact avec notre équipe. Les différentes pages du site internet sont "articulées" en HTML qui est son véritable squelette. Il regroupe le texte, les liens vers les autres pages et les images. Le CSS quant à lui, gère l'aspect esthétique : par exemple la forme des boutons,

ou le style de l'écriture.

3.1.2 Carte (Fabien)

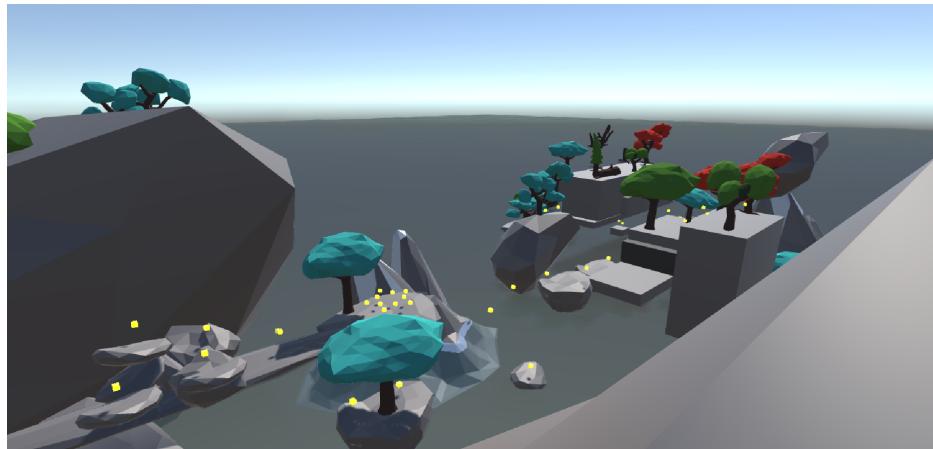


FIGURE 3.1 – Vue sur la majeur partie du premier niveau

La carte est constituée de multiples plates-formes, de rochers et de végétation. Toute la végétation provient d'assets gratuits du pack «LowPoly Forest Pack» disponible sur l'Asset Store de Unity. En revanche les rochers et les plates-formes ont été réalisés sur Blender à partir d'un bloc qu'on sculpte à l'aide de l'outil “Bissector”, puis une fois l'asset transféré sur Unity, il se voit rajouter un box collider. On notera que chaque asset de végétation s'est vu rajouté un box collider qui n'était pas fourni par le set.

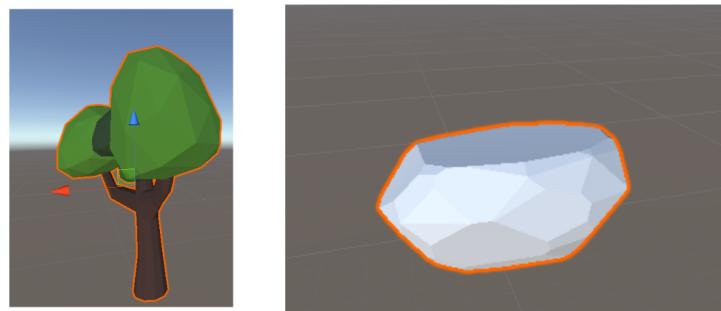


FIGURE 3.2 – Deux assets utilisés sur les niveaux

Grâce à ces assets (figure 3.2), nous pouvons réaliser des niveaux à notre guise. Actuellement, j'ai construit un niveau qui comporte en plus des éléments cités, des pièges qui sont des animations de blocs dans notre scène. Dans notre niveau le joueur est censé sauter sur des plates-formes et rester sur celles-ci lors de ses déplacements, c'est pourquoi nous avons attaché à ces plates-formes un script qui passe le gameobject représentant le joueur en enfant de la plate-forme pour qu'il hérite de son déplacement quand le joueur entre dans la zone de détection de la plate-forme (nous avons utilisé un box collider comme zone de détection).

3.1.3 Intelligence Artificielle (Fabien)

Dans le jeu, le joueur peut se faire attaquer en multijoueur et en solo par les Zoïds, les habitants de Xtra. Les Zoïds ont pour but d'attaquer le joueur à vue et de l'éjecter des plates-formes. Pour cela nous avons créé le script «`EnemyController` » qui a pour but de déplacer l'ennemi à l'approche du joueur. La fonction `Onedrawselecter` dessine en rouge la gizmo (la zone de détection de l'IA) dans la scène mais pas dans la partie. Ensuite le principal but de ce script est de vérifier si le joueur est dans la zone de détection. Pour cela on utilise la boucle `if (distance = look radius)` qui vérifie si la distance entre le joueur et l'IA est égale à la taille de notre zone de détection (si elle est inférieure le joueur est forcément à l'intérieur) si oui on utilise la méthode `Setdestination()` sur agent qui va déplacer notre IA à la position du joueur.

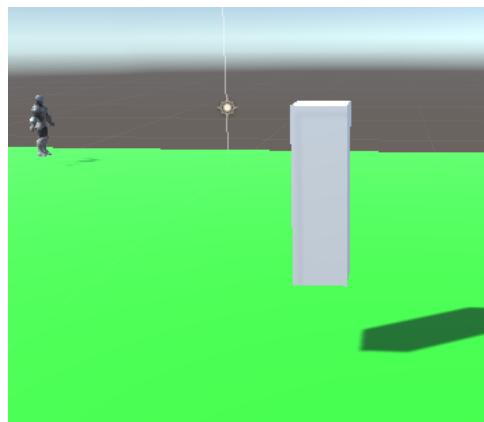


FIGURE 3.3 – Illustration du fonctionnement de l'IA

3.1.4 Scénario (Alexandre & Arthur)

L'humanité ayant découvert des sources de vie extraterrestre sur une planète lointaine, baptisé Xtra, elle décida de protéger ces dernières. C'est lors d'un regroupement des dirigeants de la Terre qu'il fut convenu que la planète en question deviendrait une planète protégée. Cependant, un groupe de chercheurs est parvenu à convaincre les dirigeants des intérêts stratégiques que pourrait représenter la planète, et se donc sont vus confiés la mission Athéna, mission ayant pour objectif de mener des recherches approfondies sur la planète secrètement.

Alors que d'importantes informations étaient en train d'être récoltées pour la mission grâce à un satellite en orbite autour d'Xtra, celui-ci est mystérieusement endommagé. Il est alors décidé que VOUS êtes envoyé pour réparer ce satellite. Mais comme vous l'imaginez, rien ne se passe comme prévu ... Votre vaisseau subit lui aussi une panne en approchant du satellite, et s'écrase sur la planète, vous laissant tout juste le temps de vous éjecter.

Perdu dans les entrailles de la planète, il ne vous reste plus qu'à ramasser les morceaux de votre vaisseau éparpillés sur cette étrange planète, aidé seulement de votre IA qui vous guidera sur cette planète hostile. Le monde n'est pas vierge de toute civilisation, et des temples anciens vous barreront la route, remplis de pièges et de mystères comme des peintures représentant d'immenses créatures. Les indigènes extraterrestres sont humanoïdes, sociaux et intelligents cependant leur peuple est tribal et dangereux. Armés de lance, la peur de l'inconnu les amènera à vous attaquer instinctivement. Cependant, certains chapitres (niveaux) de notre jeu seront très différents de la thématique de base.

3.1.5 Joueur (Arthur & Alexandre)

Pour ma part (Arthur), les contrôles du personnage ont bien avancé, il est possible de se déplacer sur quatre axes dans les deux directions. Le personnage se déplace lors de l'utilisation des flèches ou des touches Z.Q.S.D du clavier (configurable avec le menu prochainement). Unity récupère les touches enfoncées sous forme de flotants qui augmentent selon le temps d'appui, je les récupère en les affectant à des variables. Celles-ci ont été assignées aux forces appliquées sur mon personnage sous forme d'un vecteur tri-dimensionnel qui permet le déplacement du joueur. Ensuite, j'ai réalisé le saut du personnage en m'aideant de la bibliothèque de Unity qui me permet de savoir si la partie solide (le collider) touche le

sol. J'ai donc vérifié si mon personnage touche le sol et si le joueur appuie sur la barre espace. Si c'est le cas mon personnage saute, sinon il lui est appliquée une force négative sur l'axe Y qui représente la gravité et permet donc au joueur de retomber. La capacité de courrir a aussi été ajoutée permettant au joueur une plus grande maniabilité. Puisqu'il s'agit d'un jeu de plates-formes, un simple saut n'est pas suffisant. C'est pour cela que j'ai ajouté un double saut permettant une sensation de liberté de mouvement au joueur. Lors de tests, j'ai remarqué qu'il était difficile de savoir si le double saut à déjà été réalisé. Par conséquent, lors d'un double saut du personnage j'ai ajouté des particules sous forme d'une queue (Tail) derrière lui, qui disparaît lorsqu'il touche le sol. Il est aussi possible de s'accroupir réduisant ainsi la taille de la partie solide du joueur. Cela permettra au joueur de se faulir dans des espaces étroits. Par contre, lorsque le personnage est accroupi, sa vitesse est diminué pour un soucis de réalisme.

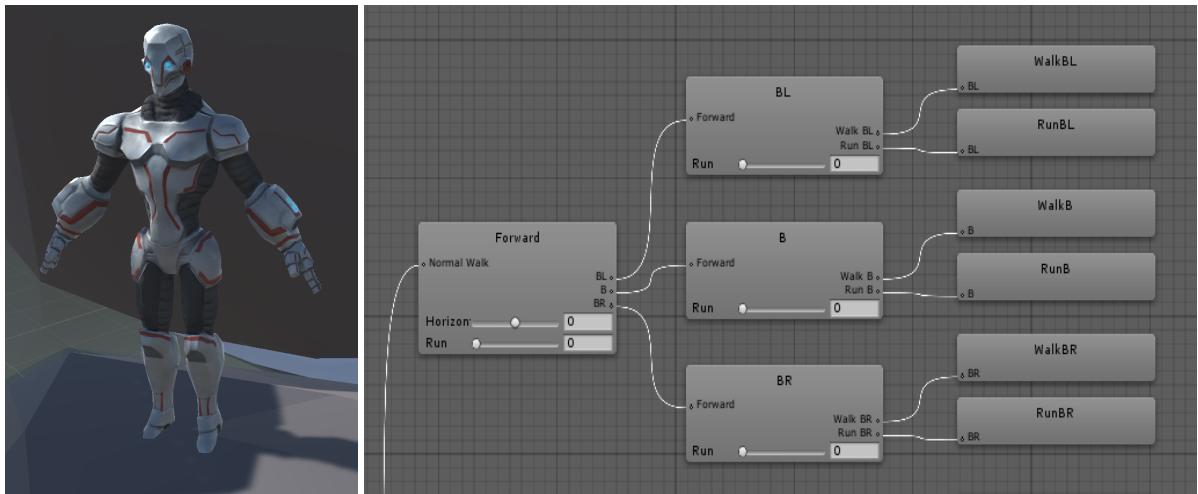


FIGURE 3.4 – Personnage à gauche et exemple d'animations à droite

Les animations du personnage ont aussi été réalisées. Tout d'abord j'ai téléchargé un modèle gratuit représentant le personnage pour l'animer. Pour cela j'ai créé un script en C# qui récupère les valeurs des axes vus précédemment. Ceux-ci servent d'input à un arbre d'animation que j'ai entièrement édité. Cet arbre joue les animations correspondant aux mouvements du personnage sur les quatre axes. Ces animations ont tout d'abord été récupérées sur une asset puis modifiées à l'aide d'un éditeur émergeant sur unity appelé Umotion car les animations ne

correspondaient pas à mes attentes. Grâce à ces animations, le personnage est humanisé et le joueur sait ce qu'il fait. Par exemple lorsque le joueur souhaite que le personnage court, celui-ci accélère et l'animation de course est jouée. Le passage de courrir à accroupi étant trop "brutal" j'ai décidé d'ajouter une animation de passage entre les deux, sous forme d'une glissade ou d'une roulade dépendant de l'axe sur lequel le joueur courait.

Pour terminer, j'ai réalisé des objets récupérables par le joueur pour lui donner envie de se déplacer dans toute la carte. Ceux-ci sont représentés par des petits carrés jaunes tournoyants.

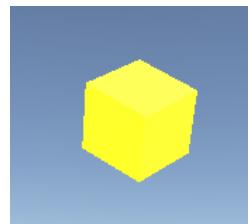


FIGURE 3.5 – Consommable

À l'aide d'un script, lorsque la partie solide du joueur rencontre la partie piégée (Trigger) du consommable, le point disparaît et est comptabilisé. Un affichage (HUD) est donc présent pour indiquer au joueur son score pendant qu'il joue.

De mon coté (Alexandre), je me suis chargé de la caméra du personnage, qui devait d'abord être à la troisième personne (vue de derrière le personnage).



FIGURE 3.6 – À gauche détecteur d'objet de la caméra, à droite rendu en jeu

J'ai d'abord suivi un tutoriel pour la mise en place de la caméra et du script qui gère ses rotations en fonction du mouvement de la souris par le joueur. Lors de l'ajout de la caméra au joueur réalisé par Arthur, j'ai aussi dû rajouter la gestion des rotations du joueur en même temps que la caméra tourne.

La prochaine étape était la gestion des collisions de la caméra avec des objets. En fait, on veut éviter deux choses :

- que la caméra passe à travers des objets de l'environnement ou d'autres entités
- que des objets puissent cacher la vue du personnage en étant entre lui et la caméra

Pour ce faire, j'ai d'abord utilisé un raycast (comme présenté dans le tutoriel), dont on testait les collisions avec un objet entre le joueur et la caméra à chaque

frame. Si un objet était détecté, alors on avançait la caméra devant l'objet. C'était donc une façon simple et optimisée de résoudre le problème, mais insuffisante : la caméra pouvait passer à travers de certains objets au niveau des coins, et le fait de passer à travers les objets lorsque le script ramène la caméra en avant n'était pas très agréable pour le joueur.

J'en ai donc déduit qu'il fallait détecter les objets plus tôt, de façon à modifier la position de la caméra par rapport au joueur un peu avant qu'ils ne deviennent gênants pour la caméra. Après plusieurs essais, je suis parvenu à utiliser un collider, situé entre la caméra et le joueur. Lorsque des objets entrent dans ce dernier, ils sont ajoutés à un dictionnaire (Dictionary en C#, nécessaire pour retrouver l'objet par son ID pour le retirer du Dictionary lorsqu'il sort du collider). Puis à chaque frame, on trouve l'objet le plus proche du joueur et on bouge la caméra devant lui. Comme les objets sont détectés avant qu'ils ne soient entre la caméra et le joueur, la caméra ne passe pas à travers de ces derniers lorsqu'elle se rapproche du joueur. On peut aussi éviter à la caméra de réagir lorsqu'elle détecte certains objets entre elle et le joueur en mettant le layer "Camera Ignore Collision" aux objets concernés.

Une fois la structure de base de la caméra réalisée, j'ai rajouté plusieurs options :

- une touche pour regarder autour du personnage sans tourner ce dernier
- une touche pour changer de point de vue : la vue à la première était bien plus simple à implémenter une fois celle à la troisième personne réalisée
- la possibilité de zoomer/dézommer avec la molette de la souris (le collider s'adapte à la distance sélectionnée de la caméra en temps réel)

De plus, j'ai fait attention à mettre toutes les variables qui permettent de configurer la caméra en public, de sorte que nous puissions changer tous ses paramètres et son comportement facilement sans toucher au code à nouveau.

3.1.6 Menu (Alexandre)

J'ai créé un menu de démarrage, qui reprend les codes couleurs du logo du jeu, et dont le background est une vue du dessus du premier niveau.



FIGURE 3.7 – Menu de démarrage

La hiérarchie du menu est comme cela :

- Play
- MultiPlayer
- Options
 - Graphics (sous-menu non implementé)
 - Control
 - X axis sensibility
 - Y axis sensibility
 - Forward
 - Backward
 - Left
 - Right
 - Camera View Switch
 - Camera Unlock
 - Sound (sous-menu non implementé)

- Back
- Exit

J'ai utilisé TextMesh Pro (package proposé par Unity) qui offre plus d'options pour la création d'interfaces textuelles que celles disponibles de base sur Unity. Pour les interactions avec le menu, il m'a suffit de mettre des Buttons, Sliders, et Input Fields qui sont assez simple à utiliser et à faire interagir avec les autres objets du menu ou variables de certains scripts.

3.1.7 Réseau (Donovan)

Le réseau est une notion réputée pour être relativement ardue par les étudiants. Pourtant c'est une partie prépondérante de notre jeu et sans lui la qualité du gameplay serait profondément dégradée. J'ai choisi d'utiliser une extension de Unity pour monter le réseau du jeu vidéo. Elle se nomme "Forge networking".

Rapidement, je me suis heurté à une documentation qui me semblait confuse et je n'arrivais pas à construire un réseau digne de ce nom. Puis Arthur m'a proposé d'utiliser les fonctionnalités de base en réseau proposées par Unity, qui étaient bien plus documentées et fournies de tutoriels. Pour cette première soutenance nous avons implémenté un réseau minimaliste, qui transmet les mouvements de chacun en temps réel à tous les autres joueurs connectés.

Au commencement, j'ai été tenté d'utiliser des RPC "Remote Procedure Call" pour traduire le déplacement des personnages, mais le réseau se trouva rapidement saturé de requêtes de toute sorte. Il fallait alors trouver une solution moins gourmande en accès réseau. Ainsi on a construit un réseau en local à l'aide de Unity et de son réseau, en utilisant Network Manager et Network Manager HUD. Ainsi 4 joueurs peuvent rejoindre le serveur en contrôlant leur propre personnage à l'aide des Network Transform et Network Identity pour le réseau.

3.2 Soutenance II

3.2.1 Site (Donovan)

Etant donné que la majorité du site a déjà été codé, et qu'il est fonctionnel nous avons fait des mises à jour pour montrer l'avancement du projet. J'ai donc rajouter quelques lignes explicatives ainsi que des images du jeu.

3.2.2 Carte (Fabien)

Afin de répondre à notre objectif de deuxième soutenance nous avons réalisé 6 cartes distinctes (5 solos, 1 multijoueur) classées en deux catégories : les cartes d'extérieur où le joueur pourra contempler la beauté de la planète Xtra tout en la parcourant et en arpantant les temples où il pourra en apprendre un peu plus sur les Zoids. Les cartes d'extérieur et d'intérieur sont bien différentes : les cartes d'extérieur ont des mécaniques de jeu basées sur du parcours et de l'acrobatie, demandant au joueur d'être attentif et minutieux dans ses déplacements au risque de tomber. Les temples quant à eux mettront à l'épreuve la réflexion et la discréption du joueur, le joueur devra résoudre des énigmes afin de pouvoir continuer son aventure et récupérer des pièces du vaisseau mais attention : les temples sont gardés par les Zoids qui sont à l'affût.



FIGURE 3.8 – Carte du Niveau 2

Nous avons incorporé de nouvelles mécaniques de jeu pour le joueur, dont les plates-formes pivotantes qui bougent sur elles-même pouvant faire tomber le joueur ainsi qu'une nouvelle mécanique de jeu, les portes à bascule : le joueur devra pousser des objets dans la carte afin de les placer sur une zone spécifique, les « bascules ». Une fois activée, elle déclenchera l'ouverture de la porte ou du

piège relié. Par contre, si il n'y a plus rien sur la bascule, la porte se refermera (le piège redeviendra actif quant à lui).

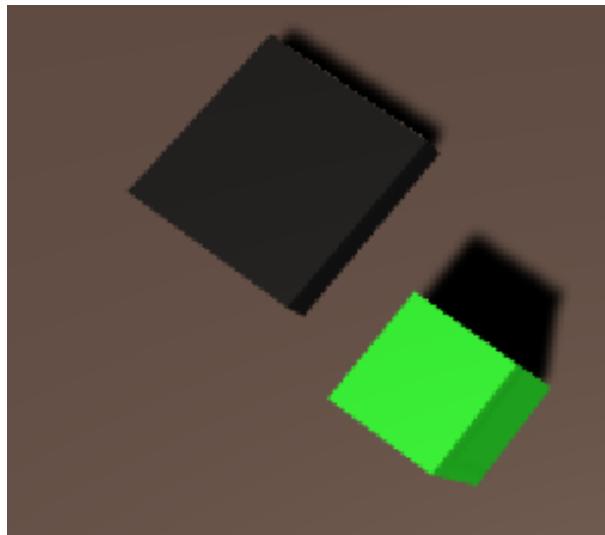


FIGURE 3.9 – Mécanisme d'une bascule

3.2.3 Intelligence Artificielle et Créatures (Fabien & Arthur)

Afin d'augmenter la difficulté et de rendre le monde plus vivant, nous avons développé deux intelligences artificielles. Il s'agit du Zoid et du "Rhamphorhynchus", une espèce d'oiseau alien.

(Fabien) Je me suis chargé de développer le Zoid. Le Zoid a pour but de défendre une zone spécifique. Il n'hésitera pas à attaquer le joueur à vue et à le poursuivre. Si le Zoid arrive à la portée du joueur il l'éjectera de la carte à l'aide de sa lance. Ce coup n'est pas forcément mortel pour le joueur qui peut retomber sur des plates-formes ou bien être projeté contre un mur.

(Arthur) Pour cette deuxième phase de travail je me suis concentré sur la création d'un reptile volant qui défend son territoire. Si le joueur est dans son

champ de vision, celui-ci n'hésitera pas à fondre sur lui. Il a donc fallu créer une IA volante se rapprochant de plus en plus du joueur. Grâce à des calculs de positionnement et de vecteur, le "Rhamphorhynchus" se rapproche de plus en plus, tout en décélérant. Cette décélération est un choix fait par l'équipe pour laisser le temps au joueur de s'enfuir même si la présence de l'oiseau met la pression. Si le joueur est atteint, alors il est propulsé à l'aide d'une symétrie par rapport au joueur pour une sensation de réalisme. Si le joueur réussit à s'enfuir alors l'oiseau le laisse tranquille. Dans les deux cas l'oiseau retrouve le chemin de sa branche et retourne se percher pour défendre son territoire. La partie graphique de l'oiseau n'a pas été réalisé par moi-même, mais les animations oui, grâce à Umotion qui permet de réaliser des animations grâce au squelette de tout personnage, comme l'oiseau. Ainsi les animations de vol et d'attente ont été réalisées en manipulant le squelette de l'animal toutes les dix frames et le logiciel se charge de faire les frames intermédiaires pour avoir une bonne fluidité.

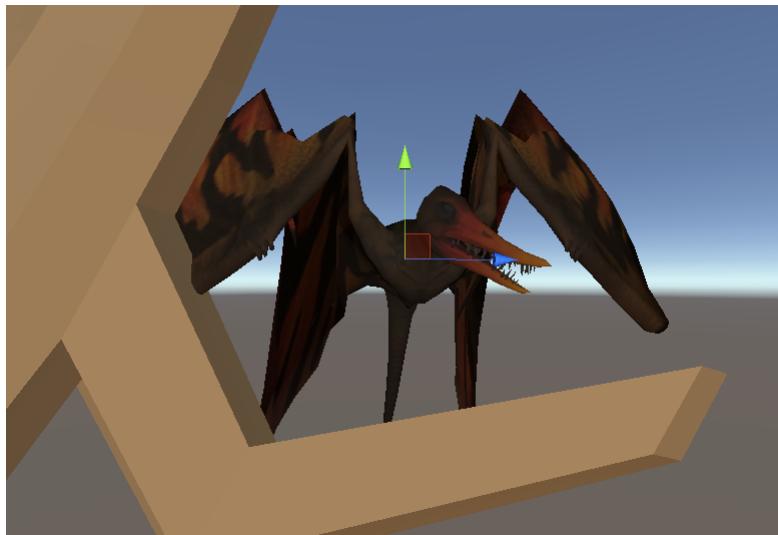


FIGURE 3.10 – Oiseau agressif

D'autre part, j'ai rajouté une nouvelle mécanique de jeu qui restreint le nombre de doubles sauts que peut réaliser le joueur. Le joueur devra alors réguler l'utilisation de son jetpack qui consomme beaucoup d'hydrogène. De l'hydrogène est donc présent sur les cartes pour permettre au joueur d'avancer mais le joueur peut se retrouver coincé. C'est pour cela qu'en cas de chute les consommables représentant l'hydrogène réapparaissent et le jetpack du joueur est de nouveau plein.

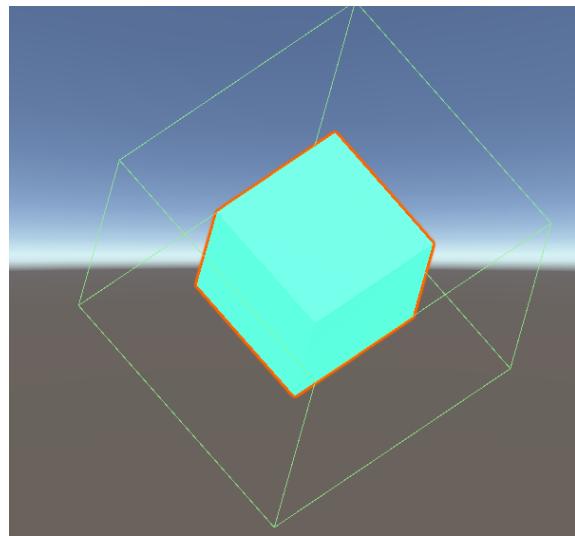


FIGURE 3.11 – Consommable "hydrogène"

Enfin, de nombreux bugs ont été réglés. Des animations ont été retouchées. Et le personnage, après la remarque faite lors de la première soutenance, ne lévite plus au dessus du sol.

3.2.4 Menu (Alexandre)

Le menu a beaucoup évolué depuis la présentation de sa structure de base lors de la dernière soutenance. La totalité des options qui étaient prévues ont été implémentées, et d'autres pourront être ajoutées ultérieurement en quelques secondes maintenant que la structure du menu est fonctionnelle et que tous les types d'entrée utilisateur sont supportés.

La structure a aussi légèrement évolué avec pour préoccupation principale l'expérience utilisateur.



FIGURE 3.12 – Nouveau menu de démarrage

La zone du menu est désormais plus propre et marquée par une zone assombrie. Le bouton Back est accessible sur tous les sous menus, et reste visible dans une zone fixe en bas du menu, y compris dans les menus déroulant, de façon à naviguer facilement entre les menus.

La nouvelle structure du menu de démarrage est comme suit :

- Play
- Level 1
- Level 2
- ...

- Level X
- Back
- MultiPlayer
- Options
 - Graphics
 - Resolution
 - Screen Mode (full / window)
 - Refresh Rate
 - Back
- Control
 - X axis sensibility
 - Y axis sensibility
 - Forward
 - Backward
 - Left
 - Right
 - Jump
 - Crouch
 - Sprint
 - Camera View Switch
 - Camera Unlock
 - Reset controls
 - Back
- Sound
 - Main Volume
 - Music Volume
 - Environment Volume
 - Back
 - Back
- Exit

Les options disponibles sont toutes sauvegardées localement sur la machine du client grâce aux PlayerPrefs.



FIGURE 3.13 – Sous menu des Controles

J'ai rencontré un problème lors de la création des boutons récupérant les différentes touches d'action : Unity n'a pas d'élément d'interface de base permettant de récupérer une touche appuyée. J'ai donc dû utiliser un bouton, qui déclenche lors de l'appui un script, qui "écoute" la prochaine action dans le menu : soit l'appui sur une touche, qui change donc la touche assignée à l'option, soit un clic gauche, qui ne provoque aucun changement et arrête d'écouter (si le joueur clique à coté pour annuler le choix de touche en cours).

Un autre problème que j'ai rencontré est celui de l'utilisation des options de déplacement : On ne peut pas modifier les Inputs de Unity lors de l'exécution du jeu, or le script gérant les mouvements du joueur utilisait les Inputs de Unity pour les déplacements (notamment les axes horizontaux / verticaux). J'ai donc dû créer une fonction remplaçant ces Inputs (notamment pour les axes puisqu'ils peuvent prendre des valeurs de -1 à 1).

Pour le sous menu Play, l'affichage des cartes est dynamique, et le fait de cli-

quer sur le Level de son choix affiche une image du niveau et un bouton Play ! permettant de lancer le niveau. L'emplacement des différents éléments du menu s'adapte automatiquement.



FIGURE 3.14 – Menu Play

Le menu en jeu est très similaire au menu principal, mais la structure principale change légèrement (plus de bouton Multiplayer, bouton Quit remplacé par Main Menu, et un nouveau bouton Resume pour reprendre le menu).



FIGURE 3.15 – Menu en jeu

Lorsque le joueur termine un niveau, un menu apparaît proposant notamment au joueur de continuer vers le niveau suivant.

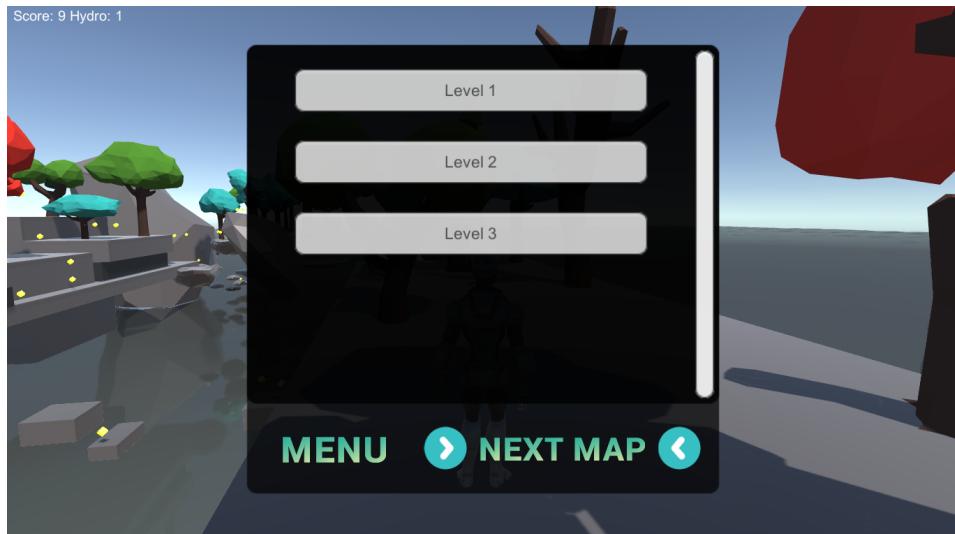


FIGURE 3.16 – Menu en jeu

Enfin, des écrans de chargement servent de transition lors des chargements d'une scène à l'autre. Un écran de chargement est composé d'une image de fond choisie aléatoirement parmi une liste, ainsi que d'un texte d'astuce lui aussi aléatoire.



FIGURE 3.17 – Ecran de chargement

3.2.5 Réseau (Arthur & Donovan)

Certaines fonctions déjà présentes dans le jeu ont été modifiées et séparées pour ne servir qu'à la partie réseau. Les préfabriqués des joueurs en réseau ont été créés et implémentés pour fonctionner dans toutes les cartes. Ceux-ci apparaissent sur leurs points d'apparition respectifs. Un Network Manager permet ainsi de créer une partie en multijoueur en local, pour le moment.

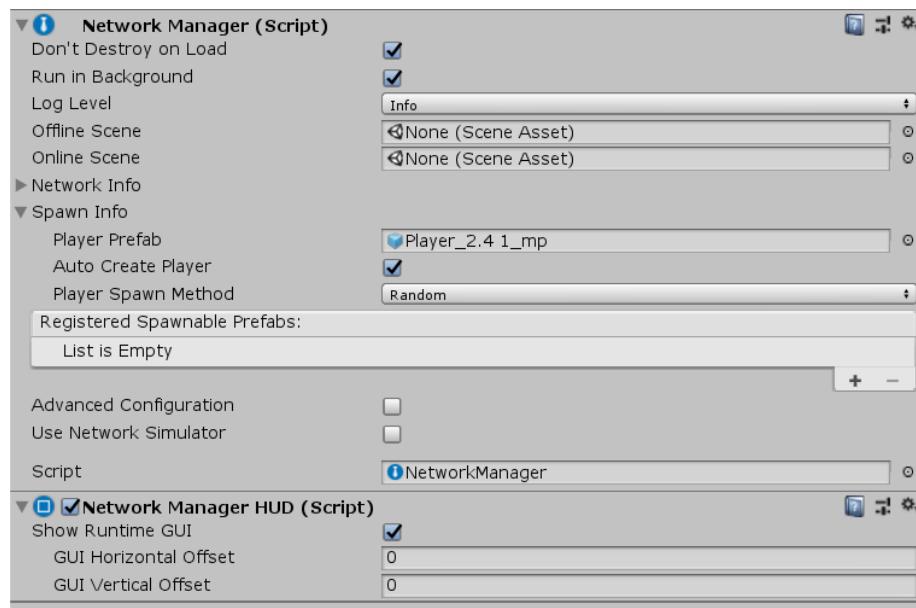


FIGURE 3.18 – Network Manager

3.3 Soutenance III

3.3.1 Sample - Partie (auteur)

3.3.2 Menu (Alexandre)

J'ai rajouté, comme ça me l'avait été demandé lors de la précédente soutenance, une option permettant d'enlever la visibilité du curseur lorsque l'on navigue dans le menu.

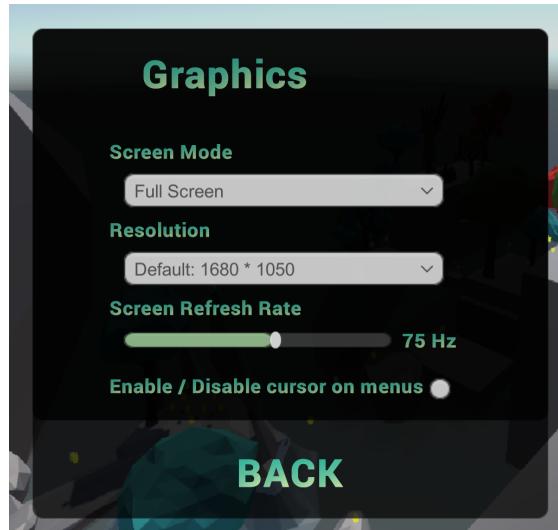


FIGURE 3.19 – Menu sans curseur

De plus, lorsqu'il est visible, le curseur a désormais un meilleur visuel, plus adapté à un curseur de jeu vidéo, avec un petit coté alien.



FIGURE 3.20 – Nouveau curseur

Les images de chargement utilisées pendant les temps de chargement entre les niveaux ont été modifiées, elles sont désormais plus nombreuses et intègrent des images des nouveaux niveaux.

Enfin, une option de choix de langue de l'interface a été rajoutée. Elle a nécessité une refonte du menu, pour que les textes ne soient plus statiques mais que les éléments de l'interface graphique chargent ces derniers en fonction du langage sélectionné.

3.3.3 Sénario et animation (Arthur)

Ayant réalisé l'IA d'un monstre et les mouvements du joueur, il fallait donc rajouter au jeu des finitions sur l'HUD (affichage tête haute), basiquement les

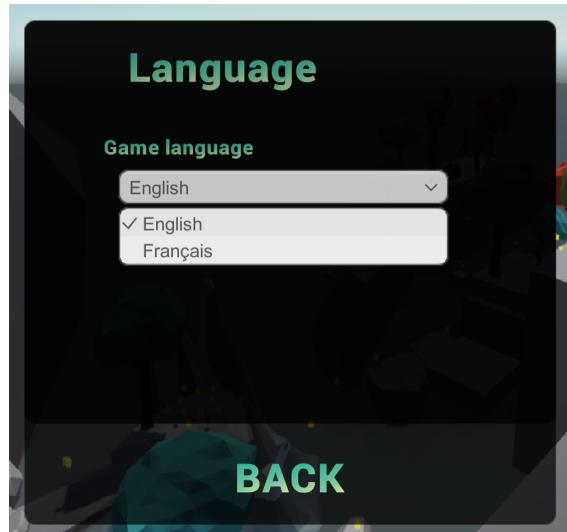


FIGURE 3.21 – Choix de la langue

informations présentent sur l'écran affichant les scores et une jauge symbolisant le carburant d'hydrogène restant. J'ai aussi rajouté les peintures rupestres sur les murs du temple symbolisant les monstres-dieux que vénèrent les Zoids . Pour améliorer la qualité graphique et le réalisme du jeu j'ai modifié l'apparence des points à ramasser qui deviennent dorénavant des cristaux mystérieux et flottent dans les airs

Pour plus d'immersion, j'ai donc réalisé une scène courte au début de la partie. Celle-ci est une cinématique montrant le crash du vaisseau du joueur sur la planète Xtra. Pour cela des animations, des déplacements du joueur et de la caméra, des particules, des sons et un décor ont été réalisé. La scène ne dure que quelques secondes mais a été un réel travail de détails et de finitions pour un rendu correct. On peut y voir le personnage amarré son vaisseau au satellite puis lorsqu'il repart un accident moteur se produit et le joueur sombre lentement vers la planète Xtra.

Pour que le joueur puisse progresser dans l'aventure j'ai rajouté et placé les pièces du vaisseau qu'il doit trouver pour réparer son vaisseau et s'enfuir de la planète. Elles sont présentes à chaque fin de niveaux et récompensent le joueur pour avoir terminé le niveau.

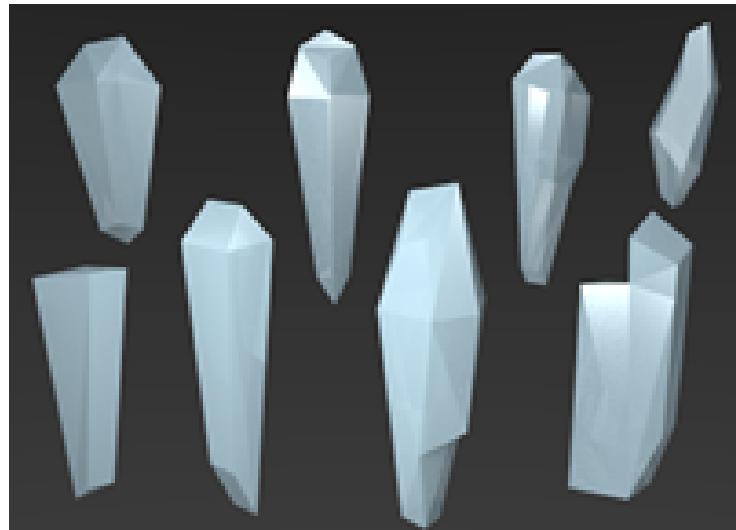


FIGURE 3.22 – cristaux mystérieux



FIGURE 3.23 – exemple illustration

Pour que cette aventure se finisse en beauté il nous fallait un boss de fin de jeu, c'est pour cela que nous avons réalisé ensemble un vaisseau de Zoid qui envoie des boulets de roche que le joueur doit esquiver pour ne pas tomber, le

joueur doit donc regarder autour de lui avec la pression qu'impose les gigantesques rochers projetés vers lui pour s'aider du décor à trois reprises afin de détruire le bateau et récupérer la pièce manquante. Après avoir vaincu le boss nous retrouvons notre personnage dans une nouvelle cinématique près du vaisseau le réparant et ayant terminé, il décolle et ... ? La suite seuls les joueurs ayant terminé le jeu pourront le savoir.

3.3.4 Carte (Fabien)

Suite à la dernière soutenance j'ai redécoré les cartes en incorporant notamment une texture de pierre au mur des temples et en plaçant des presque à certains endroits. J'ai aussi recréé les "bascules" afin de les rendre plus facilement activable avec des objets. J'ai aussi créé une nouvelle carte ou se déroule le combat final.

Sachant que le gameplay en multijoueur est une course de rapidité j'ai réalisé une carte multijoueur jouable à plus de deux joueurs. Il s'agit d'une carte constitué de plusieurs chemins ayant des intersections à plusieurs endroits et allants au même point d'arriver. Les joueurs doivent alors aller d'un point A à un point B sans tomber. Cette carte comporte de nombreux pièges pouvant être activé par chaque joueur afin de ralentir la progression des autres joueurs en plus des pièges habituels tels que les Zoids , les blocs mouvants et autres pièges de RageQuit.

3.3.5 Site (Donovan)

nous avons rajoutés des images et crée plusieurs lien menant à la bande-annonce du jeux ainsi que des images du gameplays du jeu et nous avons mis à jour le cahier des charges . Il y a aussi un lien pour le téléchargement du jeu qui serat fonctionnel à le jour de la sortie du jeu le 16/01/2019.

3.3.6 Réseaux (Donovan)

Lors de la dernière soutenance nous avons remarqué que nous avons peux de carte multijoueur nous avons donc décider d'ajouter des cartes. C'est nouvelles cartes reprennent les mécanismes du mode solo, mais aussi nous avons rajoutés des pièges déclenchable par chaque joueur afin de ralentir leur progression mutuelle. Si un joueur meurs il revient à son point de départ.

Nous avons rajoutés des zones de réapparition distincts et réadapter le débuts des anciennes cartes multijoueurs.

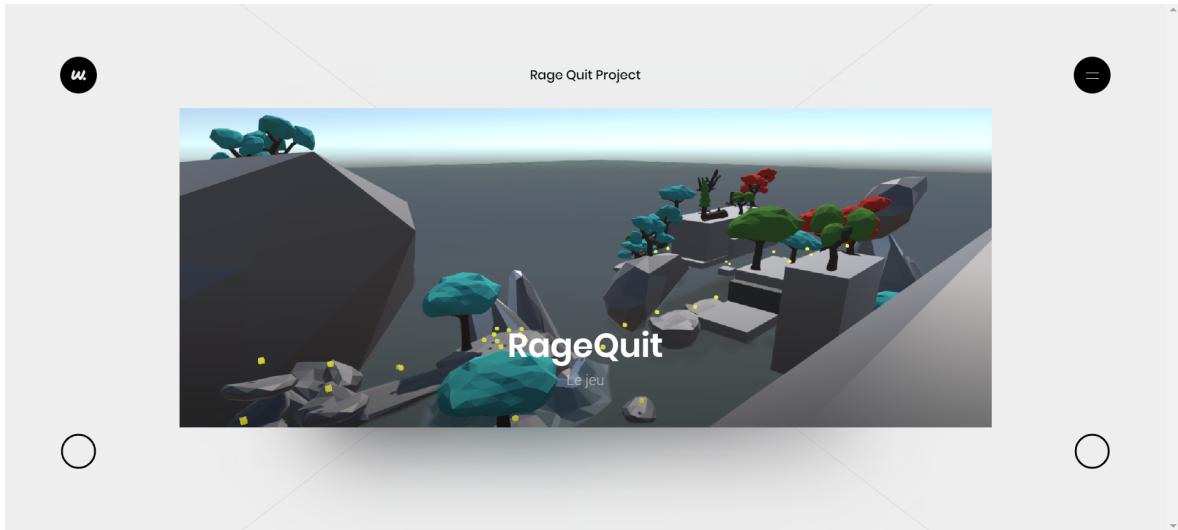


FIGURE 3.24 – Menu du Site

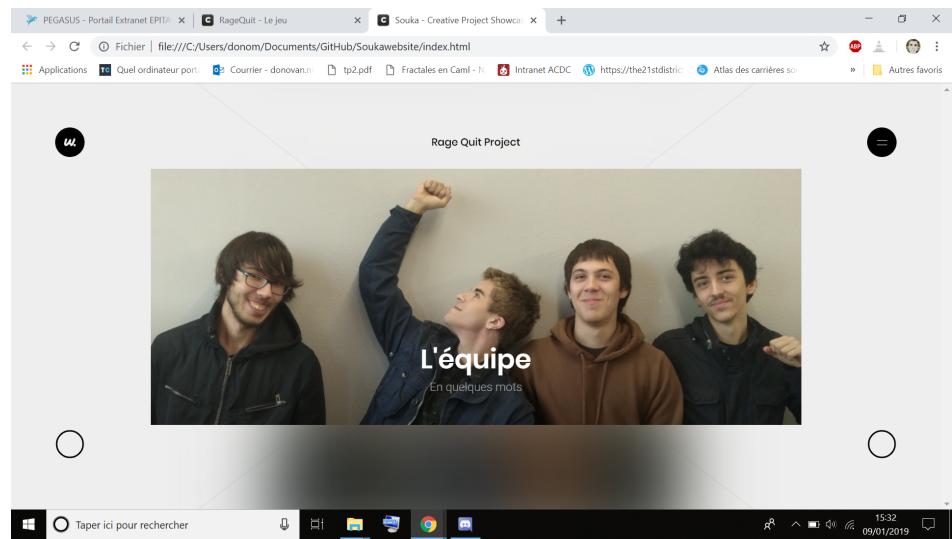


FIGURE 3.25 – Sous-Menu : présentation de l'équipe

4 Récit de Réalisation

4.1 Fabien

Ce projet, a était une excellente expérience de travail d'équipe, de recherche et développement. Il ma permit d'apprendre à utiliser de nombreux logiciel tels que Unity, Blender, GitHub. Mais il m'a surtout donné une nouvelle vision des jeux auxquelles je joue, notamment sur la réalisation des mécaniques de gameplay. Je me suis bien amusé à réaliser les diverses mécaniques présentent sur les cartes tels que les boutons qui permet l'ouverture des portes, ou bien le déplacement des rochers. Malgrès mon implicatition sur unity j'ai mis beaucoup de temps à débuger les erreurs de compilation.

4.2 Alexandre

J'ai apprécié de participer à ce projet. Apprendre par du concret, en créant véritablement quelque chose est à mes yeux plus motivant que les exercices traditionnels. Il s'agit d'une expérience très complète, puisque j'ai pu touché à plusieurs parties du développement du jeu, aider mes camarades à corriger les bugs ou trouver des solutions (et donc comprendre leur code), m'interroger sur la façon de présenter les choses pour le joueur (interface) et les fonctionnalités nécessaires pour rendre le tout plus fluide.

Travailler avec la documentation a aussi été enrichissant, avec la nécessité de chercher dans cette dernière des éléments pouvant apporter une solution à un problème, ou pour approfondir ma compréhension du fonctionnement d'un élément, et pouvoir ainsi l'utiliser proprement sans grande perte de performance.

Bien entendu, il s'agit d'un jeu développé dans un cadre scolaire, ayant un rôle

plus d'apprentissage que fonctionnel. Il y a donc un certain nombre de choses que je ferais différemment si l'on devait repartir sur une base plus solide, en ayant dès la base une vision sur le long terme.

4.3 Arthure

Honnêtement j'ai beaucoup apprécié réaliser ce projet avec mes amis, cela à modifier ma vision de la création de jeux. Maintenant je sais à quel point faire un jeu vidéo par soi-même est difficile et j'ai considérablement réduit les critiques que je faisais aux triples A et ainsi apprécié encore plus jouer aux jeux vidéo.

Ma plus grande peine fut la difficulté à se lancer et revenir sur des codes à modifier, bien qu'ils soient commentés et indentés, les fonctions qu'utilise Unity sont quelquefois ambiguë et difficile à utiliser. Ma joie ma accompagner pendant la création et à chaque fin de session me permettant ainsi de pouvoir jouer avec notre création et trouver les meilleures façons d'avancer dans les niveaux

4.4 Donovan

J'ai trouvé ce projet très intéressant, il permet de mettre en pratique de nombreuses facettes de l'informatique. L'élaboration d'un jeu vidéo exige des connaissances générales en programmation. L'avancée du projet a été quelques peu perturbée par nos obligations

scolaires, mais avec l'expérience, nous nous sommes forgés, au fil des précédentes, soutenances nous avons pu surmonter cette surcharge de travail . Etre en groupe nous a permis de nous entraider dans nos différentes difficultés. Par exemple, il m'est souvent arrivé de passer beaucoup de temps à essayer de résoudre des bugs et qu'un de mes coéquipiers m'aide pour trouver l'origine des problèmes.