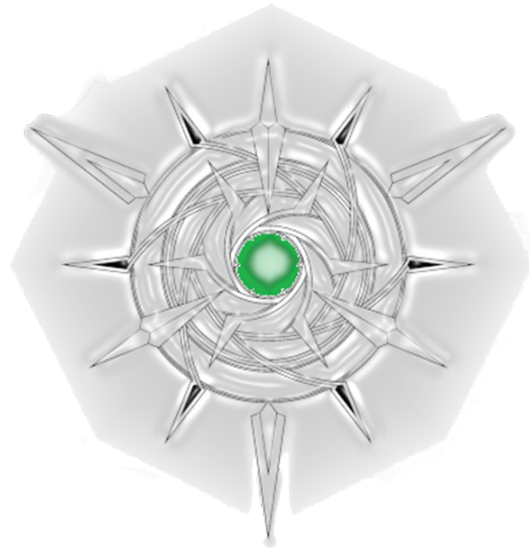

Rapport de projet - Dawn of Gates

0_Dev Team

LARIAU Steven (lariau.s)

BRENET Timothée (brenet.t)

LAFARGUE Victor (lafargue.b)



Epita S2 - XX/06/15

Remerciements

De nombreuses personnes nous ont apportés de l'aide dans la réalisation de notre projet, sans la communauté du forum d'unity pour le jeu et de stack overflow pour le c# en général et le site nous n'aurions pu faire face à tous les problèmes rencontrés lors du développement.

Nous tenons aussi à remercier HTML5Up, le site web sur lequel nous avons été cherché le template css pour notre site web.



Sommaire

~~CODEV~~

1 Introduction

1.1 Origine

Notre projet, intitulé Dawn of Gates, est un jeu en réalité virtuelle, développé avec et pour l'Oculus Rift.

Nous avons choisi cette plateforme afin de ne pas se contenter d'un jeu 3D banal. Le développement d'un jeu 3D oculus demande certes plus de travail, et se révèle plus difficile, mais au final nous sommes bénéficiaires, car les difficultés rencontrés nous forment et nous rendent meilleurs.

1.2 L'histoire

Le synopsis de notre histoire commence par le personnage principal qui se ballade avec sa petite amie, qui se fait alors volée son collier par un méchant. Notre personnage, qui se révèle être un ninja, poursuit alors le renégat afin de récupérer le collier. Il arrive alors dans un étrange bâtiment, où il découvre alors des capacités spéciales. Il devra alors les utiliser afin de combattre les différents ennemis et passer les obstacles qu'il rencontrera au fil des niveaux, jusqu'à atteindre le dernier niveau, afin de combattre le boss final, le voleur, afin d'espérer pouvoir récupérer le collier.

1.3 Gameplay

1.3.1 Le personnage

Le ninja possède de nombreuses facultés, il peut notamment lancer des shurikens. Il peut ainsi combattre ses ennemis. Mais les shurikens ont aussi une autre utilité. Lorsque ceux-ci touchent un mur, ils ouvrent un "portail", c'est à dire une porte qui mène vers un autre portail ouvert précédemment. Le ninja peut alors se téléporter, et utiliser cette capacité afin de venir à bout des différentes énigmes du jeu.

Il a aussi une capacité de "slow motion", c'est à dire qu'il peut ralentir le temps, pendant un laps de temps limité. Cette faculté, utilisé à bon escient, aide aussi notre héros à venir à bout de tous les dangers présents sur sa route.



1.3.2 Environnement

Le héros se déplace ainsi de niveaux à niveaux. Le jeu se situe en lieu clos, le personnage reste dans le bâtiment et parcourt les salles une à une. Ces salles ont un style graphique mélangeant à la fois un style asiatique et un style toon. Une musique asiatique accompagne aussi le personnage tout au long de ses péripéties, afin d'améliorer l'ambiance de jeu.

1.3.3 Les ennemis

Notre ninja rencontre plusieurs types d'ennemis. Tous d'abord il y a des tourelles, elles sont fixes, et peuvent uniquement tirer des flèches blessant le joueur. Certaines visent le joueur, d'autres peuvent uniquement tirer en ligne droite.

En plus, il existe aussi des ennemis mobiles, ils effectuent des patrouilles dans les différents niveaux, et lorsqu'ils détectent le joueur, ils lui courent après et l'attaquent avec une épée.

1.3.4 Le boss final

Le boss final est différent des autres ennemis. Il est bien plus difficile à tuer, et possède une intelligence artificielle dédiée. Il est impossible à vaincre sans une technique spéciale.

2 La team

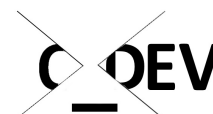
2.1 Modifications du groupe

Au départ il y avait un quatrième membre dans notre équipe, Idris Sequeira, mais il a décidé d'arrêter Epita une fois la première année terminée, et ne s'impliquait plus du tout dans le projet. Nous avons donc décidé de l'exclure du groupe.

2.2 Présentation des membres

2.2.1 Steven Lariau

Ce projet sera sans aucun doute une grande aventure, vouloir réaliser un jeu en réalité virtuelle, c'est un choix ambitieux, mais qui me permettra de



découvrir de nouveaux domaines de la programmation. Bien que je connaisse déjà un peu le développement web, notamment javascript avec node, mais aussi la programmation en Java, C#, C++, tout ce qui relève de la 3D et de l'IA par exemple sont des notions que j'ai hate de découvrir afin de réaliser ce projet. Etant un expert dans l'art de "commencer des projets colossaux, d'y passer des journées entières dessus en partant dans tous les sens puis d'arrêter complètement", j'espère ainsi apprendre à mener un véritable projet à son terme, à travailler en équipe, et à faire face aux différents problèmes sans se décourager.

2.2.2 Timothée Brenet

Passionné d'informatique depuis la quatrième, j'ai appris la programmation grâce au site du zero. J'ai d'abord suivi les cours sur le C puis je me suis tourné vers les langages web comme l'html, php, javascript et drupal. Je me suis finalement majoritairement concentré sur le développement en Java. J'ai participé à quelques projets en groupe de programmation en Java, la plupart centrés sur la creation d'un jeu en 2D grâce a la librairie lwjgl. Ce projet de sup, m'intéresse donc énormément et je pense qu'il va me permettre de me faire progresser en C# notamment et de créer mon premier jeux 3D avec Unity. Je suis donc très motivée pour ce projet, je travaillerais surtout sur la programmation en C# et le portage sur l'Oculus Rift.

2.2.3 Victor Lafargue

Je m'appelle Victor Lafargue, je suis originaire de Seine-Et-Marne près de Fontainebleau. J'ai toujours été fasciné par la technologie, l'électronique et l'informatique. Ce sont les domaines d'expertises d'Epita et c'est pourquoi j'ai choisi cette école. J'ai déjà un peu d'expérience en informatique surtout en C# et en python que je connaissais avant de venir ici. J'ai aussi déjà essayé de créer le début d'un jeu-vidéo avec OpenGL ce qui pourra peut-être aider. Ce projet informatique en équipe et sur le long terme est une superbe occasion d'avoir un avant-gout de ce qu'est le développement informatique. Il me permettra, j'en suis sûr d'énormément progresser de démontrer ma détermination à réussir et à terminer un grand projet.



2.3 Organisation

Pour travailler en groupe, nous utilisons tout d'abord Github. Cette plateforme de versionning nous permettait ainsi de pouvoir partager et travailler à plusieurs sur le projet sans problème.

Nous utilisons aussi Skype afin de communiquer, de prendre des décisions sur le projet, de demander son avis à d'autres personnes du groupe, etc. Pour les trois soutenances nous nous sommes plus ou moins organisées de la même façon. La première étape était de se répartir les tâches, de décider qui allait faire quoi. Ensuite, nous réalisons chacun nos tâches respectives, avant de se réunir et de mettre notre en commun pour régler les incohérences et les différents bugs.

2.4 Plannings de répartition des tâches

2.4.1 Première période

	Timothée	Steven	Victor	Total
Code (Base)	20	20	20	60
Graphismes	0	0	40	40
Animations	10	0	30	40
Niveaux	6	6	8	20
IA	15	20	15	40
Audio	0	0	0	0
Site web	0	50	0	50
Réseau	0	0	0	0

2.4.2 Deuxième période

	Timothée	Steven	Victor	Total
Code (Base)	30	25	25	80
Graphismes	0	0	60	60
Animations	20	0	40	60
Niveaux	25	25	5	55
IA	25	30	15	70
Audio	25	0	0	25
Site web	0	60	0	60
Réseau	0	10	10	20



2.4.3 Troisième période

	Timothée	Steven	Victor	Total
Code (Base)	40	30	30	100
Graphismes	0	0	100	100
Animations	20	0	80	100
Niveaux	35	35	30	100
IA	45	35	20	100
Audio	33	34	33	100
Site web	10	90	0	100
Réseau	0	50	50	100

3 Capacités du personnage principal

3.1 Déplacements

3.2 Les caméras

3.3 Le lancer de shurikens

3.4 Les portails

Les portails sont une des features les plus importantes du jeu, mais aussi l'une des plus complexes à réaliser. J'ai fait de nombreuses tentatives et de nombreux échecs avant d'arriver à une solution fonctionnelle et viable.

3.4.1 Ouverture et placement

Un portail est sensé s'ouvrir lorsqu'un shuriken lancé par le portail touche un mur. De plus, si il n'y a qu'un seul portail placé, celui-ci reste fermé. Ce n'est que lorsque les deux portails sont placés qu'ils s'ouvrent. Le click de la souris détermine quel portail est lancé. Le click gauche correspond à un portail gauche, et le click droit à un portail droit.

Pour cela, lorsque le shuriken est lancé, un raycast est réalisé afin de savoir quel objet le shuriken va rencontrer. Le script de placement ne commencera que si la cible est un mur. Pour détecter si c'est un mur ou non, j'ai mis un Tag nommé "room" sur toute la pièce. Je n'ai plus qu'à vérifier si le tag de l'objet trouvé par le raycast est bien "room". Si c'est le cas, je calcule la



distance entre le shuriken et le mur, et lorsque le shuriken a parcouru cette distance, le portail est placé à l'endroit où se trouve le shuriken, et le shuriken est détruit.

Le raycast me donne aussi le vecteur de la face touchée. J'utilise ce vecteur afin que le portail soit bien posée à plat, quelle que soit la face du mur, du sol ou du plafond.

Une fois le portail placé, je vérifie si l'autre portail est ouvert. Si c'est le cas, l'animation d'ouverture est lancée. Sinon, elle est lancée sur les deux portails, uniquement lorsque les deux sont placés.

3.4.2 L'effet portal

L'effet portal, c'est que lorsque le joueur regarde un portail, il voit à travers celui-ci, ce qu'il y a en face de l'autre portail. Cela donne un effet de téléportation bien plus réaliste et impressionnant. Pour réussir cela, j'ai beaucoup réfléchi, fait de nombreuses tentatives, regardés sur les forums, des vidéos.

Ma première idée était tout d'abord de réussir à faire un simple miroir. J'espérais ensuite pouvoir l'adapter aux portails, le premier portail serait un miroir du second, et inversement. J'ai trouvé sur internet un tutoriel expliquant comment faire un miroir avec un script et un shader, mais celui-ci était très complexe, et je n'ai pas réussi à bien le comprendre pour pouvoir l'adapter à mon problème.

Finalement, en cherchant sur les forums, j'ai trouvé une solution beaucoup plus simple : les texture renderer. Ils permettent de sauvegarder tout ce qu'enregistre une caméra. Je n'ai alors plus qu'à placer une caméra sur chaque portail. Le portail de gauche à un matériel possédant la texture renderer sauvegardé par la caméra du portail de droite, et inversement. Ainsi j'obtiens l'effet désiré.

Bien que le résultat final est très simple et court à implémenter, il m'a fallu beaucoup de temps de recherches et de réflexions, ainsi que de faux pas, avant d'arriver à cette solution.



3.4.3 La téléportation

La téléportation décuple les possibilités de déplacement du personnage. Elle permet d'ajouter un gameplay intéressant pour les différents niveaux du jeu. Le principe de la téléportation est que le joueur rentre dans un portail, il ressort par un autre portail. La aussi j'ai pas mal réfléchi avant d'arriver à l'implémentation finale. Tout d'abord, j'ai pensé à dédoubler le personnage lorsqu'il commençait à rentrer dans un portail. Son double étant en train de sortir de l'autre portail. Mais cette solution était bien trop compliquée à implémenter et aurait complexifié tous les autres scripts du personnage. Finalement j'ai opté pour une solution plus simple, lorsque le ninja rentre dans le portail, il ressort immédiatement par l'autre portail, avec l'orientation adéquate.

Pour cela, chaque portail possède un box collider. Lorsque le ninja rentre en collision avec le box collider d'un portail, il est téléporté à la même position et la même rotation que l'autre portail. Malheureusement, cela ne fonctionnait pas, car une fois le ninja téléporté, il déclenchait alors le box collider de l'autre portail, et se retéléportait, et ceci indéfiniment. Pour régler le problème, le personnage avance automatiquement tout droit afin de sortir du collider.

3.5 Le slow motion

Le slow motion, c'est la capacité du ninja à ralentir le temps. Lorsque cela le produit, la vitesse de tous les objets et de tous les ennemis est considérablement ralenti. Seule la vitesse du joueur reste la même.

Pour cela j'utilise une variable de "temps", un float, qui par défaut est à 1. Tous les objets influencés par le slow motion ont leur vitesse multiplié par ce nombre. Lorsque le slow motion est activé, la valeur de cette variable change. Pour ralentir le temps par n , il suffit de mettre la variable à $1/n$. Ainsi, la vitesse de tous les objets affectés par le slow motion est divisée par n . Une fois le slow motion désactivée, la variable revient à 1.

Pour activer et désactiver le slow motion, il suffit d'appuyer sur la touche espace. Afin d'ajouter du gameplay, nous avons préféré l'imiter l'utilisation du slow motion. Le ninja a une certaine quantités d'énergie, qui diminue lorsque le slow motion est activé, et qui augmente lorsque le slow motion est désactivé. L'activation du slow motion si le personnage n'a plus d'énergie. Cela permet au joueur d'éviter certains ennemis ou projectiles en activant



le slow motion, et grâce à la limite d'utilisation, il ne peut pas l'utiliser en permanence, et doit donc s'en servir à bon escient.

3.6 La vie

Le joueur possède une certaine quantités de vie, qui diminue lorsqu'il ennemi l'attaque. Il meurt lorsqu'il n'a plus de vie.

4 Fonctionnement des niveaux

4.1 Transition entre les différents niveaux

Afin de passer entre les niveaux différents, qui correspondent à différents étages du bâtiment, le personnage utilise des ascenseurs. Il en existent de deux types : les ascenseurs de début de niveau, et les ascenseurs de fin de niveau. Chaque niveau possède les deux types d'ascenseurs, excepté le premier et le dernier niveau.

Le joueur commence dans l'ascenseur de début de niveau. Celui est fermé, puis s'ouvre automatiquement après quelques secondes, permettant au joueur de découvrir le niveau.

L'ascenseur de fin de niveau est ouvert et possède un box collider. Lorsque le joueur le déclenche, les portes se referment après quelques secondes. Le niveau suivant est alors chargé.

5 Le Site Web

5.1 Buts

La Site web présente notre site web. Il doit être beau, simple et épuré, afin de donner envie aux joueurs de jouer notre jeu. Il sert aussi à présenter l'équipe le jeu, et permet de télécharger le cahier des charges, le rapport, ainsi que notre jeu bien entendu.

Il possède aussi des fonctionnalités dédiés aux joueurs, il est possible de s'inscrire pour se connecter sur le site web et en jeu. Il est aussi possible de voir les performances des autres joueurs.



5.2 Base du site

Tout d'abord, je suis parti sur un site web statique, c'est à dire ne comportant que des fichiers HTML, CSS et javascript. Afin d'avoir un beau design, j'ai décidé d'utiliser un templates CSS. Un autre avantage des templates est qu'ils s'adaptent automatiquement à toutes les résolutions, tous les navigateurs et tous les appareils. J'en ai trouvé un qui me convenait sur le site HTML5 Up. Toutes les pages web du site ont le même menu, la même en-tête et le même bas de page, seul le corps du document varie.

Ensuite, j'ai décidé de partir sur un site dynamique, indispensable pour l'interaction avec les visiteurs du site. Pour cela, j'ai choisi d'utiliser PHP, et j'ai développé le site web avec XAMP, un suite logistiques regroupant tous les programmes nécessaires pour faire tourner un serveur local.

Ce changement m'a permis tout d'abord de modifier la structure de mon site, au lieu recopier le même code de base dans chaque page HTML, j'ai fait une seule page PHP regroupant tout ce code. Ensuite toutes les autres pages HTML sont remplacés par des pages PHP ne contenant que le corps du document. Grâce à l'instruction PHP include, ces pages peuvent ainsi être incluses dans la page de base, selon l'url.

Finalement, j'ai aussi fait le choix d'utiliser MySQL, une base de donnée, qui me permet de stocker les informations des différents membres, et de les récupérer, tout cela avec du code PHP.

5.3 Les différentes pages

Note site web contient différentes pages web statiques, accessible sans restriction. Ce sont les toutes premières pages réalisées, avant l'utilisation du PHP :

- Accueil : Présente rapidement qui nous sommes et notre jeu
- Téléchargements : Permet de télécharger le cahier des charges, les rapports de soutenance, le rapport de projet, les sources ainsi que l'exécutable de notre jeu, pour windows et os x
- Présentation : Présente notre projet, ainsi que les différents membres de l'équipe
- Liens : Une liste de liens vers les différents site web, logiciels et ressources externes utilisés pour notre projet



5.4 Les scores

Lors du passage à PHP, j'ai rajouté une nouvelle page, la page des scores. Cette page affiche les 5 meilleurs temps réalisés pour chaque niveau. Ces scores sont en fait stockés dans un fichier sur le serveur, qui est lu par PHP. Ce fichier contient une version textuel d'un tableau PHP. Je peux ainsi "unserialize" ce texte afin de récupérer un tableau PHP contenant les temps pour chaque niveau, que je peux ainsi afficher sur la page.

Comme expliqué précédemment, le serveur reçoit une requête HTTP, envoyée par le jeu, contenant le numéro du niveau terminé, ainsi que le temps réalisé. Un script php va alors "unserialize" le tableau des scores comme pour l'affichage, mais cette fois il va le modifier, en ajoutant le temps au niveau si celui-ci fait partie des 5 meilleurs, puis va mettre à jour le classement. Le tableau est alors "serialize" puis sauvegardé de nouveau dans le fichier texte.

Lorsque j'ai décidé d'utiliser MySQL, j'ai refait le fonctionnement des scores. Maintenant, la page des scores affiche aussi en plus le pseudo des joueurs. Cette fois les scores ne sont plus stockés dans un fichier texte, mais dans la base de donnée. La page PHP fait plusieurs requêtes SQL, une pour chaque niveau, permettant à chaque fois de récupérer les 5 meilleurs temps. La script de sauvegarde a aussi changé, la requête HTTP contient en plus l'identifiant du membre, c'est à dire que maintenant seul les scores des joueurs connectés sont enregistrés. Le script fait une requête sql pour enregistrer le score dans la base de donnée.

5.5 La partie membre

Je me sers aussi de MySQL pour stocker les informations des différents membres. De nombreuses pages ont été ajoutés :

- Inscription : Une simple page web contenant un formulaire d'inscription : Une fois validé, un script vérifie si les informations sont correctes, et renvoie en conséquence sur la page d'inscription avec les erreurs affichées, ou sur une page temporaire confirmant l'inscription, et qui redirige le visiteur sur la page d'accueil après quelques secondes, à l'aide d'un script javascript. Si l'inscription est validée, les informations sont enregistrés dans la base de donnée.
- Connection : Il fonctionne globalement comme l'inscription, sauf que



le script PHP va faire une requête sql avec les identifiants entrés. Si aucune entrée n'est trouvée, alors la connection échoue et le visiteur est renvoyé sur la page de connection. Sinon le visiteur voit une page de confirmation avant d'être redirigé.

- Déconnection : Cette page se contente uniquement de déconnecter le visiteur. Un message de confirmation est affichée avant que l'utilisateur soit redirigé vers la page d'accueil.

L'état du visiteur (connecté ou pas), est affiché en permanence dans le menu. Pour cela, j'utilise des sessions, ce sont des variables PHP qui restent après avoir changé de page. Elles peuvent être effacées manuellement, c'est ce qui arrive lorsque le visiteur va sur la page de déconnection. Ou sinon elles s'effacement automatiquement après une certaine durée d'inactivité.

6 Conclusions Personnelles

7 Conclusion

