

BIBLE WHITE CORP.

CAHIER DES CHARGES

Elves Must Live

Développeurs :

Thomas LUPIN

Antoine NICOLAÏ

Clément GONNAUD

Louis HOLLEVILLE

E-mail :

`contact@elves-must-live.fr`

Site web :

`http://elves-must-live.fr/`

Janvier 2017



Table des matières

	Page
1 Introduction	2
1.1 Notre groupe	3
1.1.1 Thomas 'Thetoto' LUPIN	3
1.1.2 Antoine 'Django' NICOLAÏ	3
1.1.3 Clément 'Nefraxor' GONNAUD	3
1.1.4 Louis 'Nat' HOLLEVILLE	4
2 Présentation du projet	4
2.1 D'où nous vient l'idée ?	4
2.2 Quel sera le type de jeu ?	5
2.3 Quel est l'histoire du Tower Defense ?	5
2.4 Quel sera le scénario ?	5
2.5 Quel est le but/objectif de ce projet ?	6
3 Description du projet	7
3.1 Moteur de jeu	7
3.2 Gameplay	7
3.3 Interface	8
3.4 Intelligence Artificielle	8
3.5 Audio	8
3.6 Multi-Joueur	8
3.7 Site Web	9
4 Coût général	9
4.1 Software	9
4.2 Hardware	10
5 Répartition des tâches	11
6 Avancement du projet	11
7 Conclusion	13

1 Introduction

Voici le cahier des charges du projet “Elves Must Live” du groupe Bible White Corp.

Nous sommes initialement quatre élèves à l’EPITA motivés par la volonté de présenter un jeu complet et abouti durant une soutenance de projet de fin d’année.

Parmi tous les choix possibles, c’est unanimement que nous nous sommes regroupés autour du jeu vidéo. Chacun d’entre nous baigne depuis son enfance dans cette culture, et le développement d’un tel projet, au delà de l’aspect challenge de la chose, est la résultante de notre développement personnel, notre fierté.

Parmi tous les types de jeux possibles, nous avons choisi un type qui semble de plus en plus délaissé par les joueurs malgré le côté nostalgique de l’époque Warcraft, le Tower Defense. Bible White Corp. va donc développer un jeu vidéo de ce type mais en y ajoutant des nouveautés : il faudra vaincre des vagues d’ennemis en PvM (Player vs Mobs) ou à l’aide de tours de défense, le tout en vue TPS. Un mode multijoueur va même voir le jour, avec la possibilité de partager cette expérience en coopération et peut être, par la suite, un mode versus ! De nombreuses autres fonctionnalités seront disponibles et vous seront expliqués dans la partie gameplay.

Néanmoins, la création d’un tel jeu, pratiquement de A à Z, nécessite, comme pour tout autre, une rigueur implacable et un fil conducteur que nul ne saurait perdre, d’où la nécessité d’un tel cahier des charges. Vous trouverez ici tous les éléments utiles à la compréhension de l’élaboration d’un tel jeu, de sa genèse à son aboutissement en passant exhaustivement par l’évaluation des enjeux. Tout le processus créatif et le déroulement du projet vous y seront présentés.



1.1 Notre groupe

Bible White Corp. est un groupe formé par des étudiants en première année à l'EPITA dans le cadre d'un projet de semestre. Thomas 'Thetoto' LUPIN, Antoine 'Django' NICOLAÏ, Clément 'Nefraxor' GONNAUD et Louis 'Nat' HOLLEVILLE le forment. Notre groupe est composé par des membres venant de deux classes de l'EPITA, la B1 et la D2, mais avec la même motivation, le même esprit et surtout le même niveau en programmation. De la sorte, le groupe ainsi formé est très homogène et, le résultat sera un mélange unique de nos univers. Ce groupe sera donc la Bible White Corp., d'où le nom est inspiré de The Binding of Isaac.

1.1.1 Thomas 'Thetoto' LUPIN

Venant de Terminale S 'SVT', je n'ai pas fait d'informatique au lycée. En cours, du moins, car je m'intéresse depuis quelques temps à la programmation, notamment le Web, comme beaucoup de débutants, ou sur ma calculette en Ti-Basic, lors des cours de philosophie peu amusants. J'ai donc notamment appris le HTML5/CSS3 ainsi que le PHP, mais aussi le JavaScript et le Node.JS que j'affectionne beaucoup. Cependant, je n'ai que peu d'expérience dans les langages objets tel que le C# et ce projet sera parfait pour combler ce manque, ainsi que pour m'ouvrir à d'autres thèmes tout au long du projet. De plus, coder un jeu vidéo de A à Z est comme un rêve d'enfant, je suis donc très motivé, et, en tant que chef de projet, je vais tenter de motiver l'ensemble des membres de la Bible White Corp. dans ce projet ambitieux.

1.1.2 Antoine 'Django' NICOLAÏ

Également issu d'une Terminale S 'SVT', l'informatique au lycée occupait une part plutôt très restreinte de notre programme. Pourtant, j'étais fasciné par les possibilités qu'offrait cet outil sans malheureusement ne jamais m'être plongé dedans, d'où mon envie de rattraper le temps perdu et faire mes études supérieures dans ce domaine. Ma maigre expérience de programmation n'a été que la réalisation de petits programmes en Ti-Basic sur ma fameuse Ti-82 afin de grappiller quelques points et secondes pendant mes épreuves de maths. Ma réelle expérience avec la programmation n'a commencé qu'en intégrant l'EPITA cette année et je n'ai pas à regretter mon choix. Je suis très motivé par ce projet qui représente un vrai challenge pour ma part et je suis sûr que la Bible White Corp. mènera à bien toutes ses ambitions.

1.1.3 Clément 'Nefraxor' GONNAUD

En fin de première, j'ai commencé à sérieusement me demander ce que je voulais faire. Je jouais beaucoup aux jeux vidéos et me disais que faire de l'informatique pourrait me plaire. J'ai

donc commencé à chercher des écoles d'ingénieur en informatique en demandant, notamment à un membre de ma famille travaillant dans ce secteur, quelles étaient les écoles les plus réputées. Il m'a principalement cité EPITA. En début de terminale j'ai donc fait une JPO EPITA, et ça m'a vraiment donné envie de l'intégrer. C'est pourquoi je suis cette année à EPITA, et bien que je n'ai jamais codé avant cette année, je suis investi et motivé car le domaine de l'informatique est quelque chose qui me plaît. A travers ce projet, je vais pouvoir développer de nombreuses compétences en informatique mais aussi acquérir des méthodes de travail en groupe permettant la réalisation d'un projet dans de bonnes conditions.

1.1.4 Louis 'Nat' HOLLEVILLE

Étant initialement destiné à la biologie, c'est donc normalement que mes études s'orientent vers la SVT, la médecine puis la biologie. Mais c'est en deuxième année de licence de sciences de la vie que j'ai vraiment découvert le monde de l'informatique. Bien que cela paraisse étrange, l'informatique est présent de nos jours partout, et les algorithmes de recherches appliquées en Python sur le séquençage ADN ont piqué ma curiosité à tel point que quelques mois plus tard, j'intégrais l'EPITA. Baigné depuis tout jeune dans la culture geek, de part mon père informaticien et mes affinités ; ce monde ne m'était pas étranger et c'est donc tout naturellement qu'au commencement de ce projet que je me suis rapproché de gens partageant mes convictions et mes idées, fondant ainsi Bible White Corp. Pour moi, ce projet pourrait m'apporter beaucoup dans le sens où il m'apprend à rechercher par moi-même des explications, des connaissances dans un univers qui m'est alors inconnu afin de pouvoir travailler avec.

2 Présentation du projet

2.1 D'où nous vient l'idée ?

Une méthode pour trouver un consensus sur un tel projet est de parler de tout et de rien. De nombreuses idées furent lancées : RPG ? Peu original, STR ? Trop compliqué, Rogue-Like ? Pourquoi pas... mais ce fut incontestablement devant le souvenir commun des Tower Defense que le projet se forma vraiment. Le projet est donc un hybride entre notre imagination et de nombreuses références. Le titre n'est d'ailleurs pas sans rappeler la série de chez Robot Entertainment. L'univers est évidemment fantaisiste et emprunté aux nombreux récits classiques dits "tolkieniens".

2.2 Quel sera le type de jeu ?

Comme dit précédemment, notre jeu sera donc un jeu dans un univers 3D de type "Tower Defense". La caméra sera dans un setup TPS et le gameplay se basera sur un hybride entre du Tower Defense et du combat direct. Cela signifie que notre héros sera dans une configuration de PvM (Player Vs Mobs) avec la possibilité de les combattre par un système de piège et de défense autonome, ici donc du Tower Defense pur, mais aussi la possibilité de combattre directement ces mobs, avec des capacités, éventuellement un inventaire et un arbre de compétence, ainsi que de prendre des dégâts, vulgarisable en beat them up.

2.3 Quel est l'histoire du Tower Defense ?

Apparu en 1990, le Tower Defense consiste à placer des tours sur une carte afin d'empêcher les vagues successives d'ennemis d'atteindre un certain point de la carte. Les ennemis ne se déplacent pas librement et suivent une tracé bien précis, d'où l'importance de l'emplacement des tourelles défensives. Ce type de jeu s'est popularisé en 2003 par Warcraft III. Une fonctionnalité typique et inédite de Warcraft III qui a en partie fait son succès était la possibilité d'éditer soi-même les cartes et ainsi personnaliser entièrement ses parties. Le Tower Defense s'est ensuite largement répandu notamment grâce à l'émergence des jeux flash dans les années 2000 qui permettaient à des joueurs occasionnels de pouvoir jouer gratuitement sur navigateur. Ce type de jeu a encore du succès aujourd'hui, on peut par exemple citer Plants vs Zombies, dont la première apparition date de 2009, où un jardinier doit protéger sa maison d'une invasion zombie à l'aide de plantes. Les zombies ne se déplacent que ligne par ligne et non pas en mêlée, et les plantes attaquent également en ligne droite. Le jeu, initialement joué sur PC et Mac, s'est depuis exporté sur smartphones et consoles depuis. Le TD (comprenez Tower Defense) ne cesse donc d'évoluer dans son gameplay, son design, ses supports, etc., et ne semble pas être tombé dans l'oubli.

2.4 Quel sera le scénario ?

La guerre, la famine, les pillages, que des mots devenus étrangers au monde paisible des elfes. Voilà bien quelques décennies que les derniers guerriers se sont retirés et que le monde ne court plus aucun danger. La menace des orcs est pour beaucoup oubliée. Mais dans cette insouciance, une menace se profile à l'est. La horde orque s'est reformée et aujourd'hui, elle est avide de vengeance contre le peuple immortel. Vous jouerez donc un elfe, fraîchement rejeté de l'académie d'ingénierie de la grande cité d'Argedhen, et qui, du haut de ses 70 ans vise les plus hautes sphères de l'armée. Hélas, enrôlé dans une guerre qui le dépasse, saura-t-il surmonter ses doutes et s'illustrer aux yeux de tous comme le fier descendant du Maréchal Thirquen et guider son peuple vers la victoire ? Ses

connaissances en ingénierie vont être d'une aide précieuse pour construire les bâtiments défensifs elfiques, mais cela sera-t-il suffisant ? Ce sera à vous d'en décider dans...



2.5 Quel est le but/objectif de ce projet ?

Ce projet a de multiples buts dont la majorité réside dans l'application interdisciplinaire de nos connaissances acquises à travers nos études. Bien évidemment, les mathématiques et la physique seront de rigueur, et notamment pour la gestion des collisions, des déplacements, des chutes et de tout ce qu'un jeu nécessite pour que sa physique soit agréable à l'utilisateur. Bien évidemment, nos modules d'anglais ont pour but de nous rendre capable de produire un jeu, non seulement codé en anglais mais également avec l'univers entier disponible en 2 langues (anglais et français). La partie codage et création de scripts fera appel à nos connaissances en matière de programmation, mais plus encore ! En effet, dans un jeu régi par des dizaines de scripts exécutés à chaque frame, il est nécessaire d'avoir une solide connaissance de l'algorithmique afin de pouvoir proposer une optimisation telle que le jeu puisse tourner de manière fluide sur des configurations modestes. De plus, il nous permettra de découvrir de nouveaux logiciels comme Unity, Blender et d'approfondir nos connaissances en C#, \LaTeX , PHP, HTML, et tout autres langages que nous rencontrerons. Enfin, et de loin la partie la plus importante, le travail en équipe. Au delà de l'aspect applicatif de nos connaissances, ce projet a pour but de nous initier au travail d'équipe, tels les futurs ingénieurs que nous espérons devenir. Nous n'avons que très peu de connaissances initiales en terme de développement de logiciels et nous nous retrouvons à en créer un dans un délai imparti. Ce délai est bien évidemment trop court pour le réaliser seul. Nous devons donc chercher des connaissances de notre côté, partager nos découvertes, équilibrer les tâches et surtout, compter sur le travail d'équipe pour réunir chaque pièce du puzzle. De la sorte, le projet réalisé nous aura non seulement formé à l'utilisation de nombreux outils, mais en plus, il nous aura appris la gestion du temps, des ressources et le travail en équipe.

3 Description du projet

3.1 Moteur de jeu

Le moteur choisi est celui proposé par Unity, en effet ce dernier est intuitif et permet de développer des jeux en 3D. De plus, il est multiplateforme et les scripts utilisés peuvent être en C#, langage étudié cette année. Ce moteur bénéficie également d'un moteur physique par défaut, gérant la plupart des collisions et la gravité. Enfin, la license de ce moteur est gratuite et son utilisation se fait sans publicité dans notre cas, ce qui en fait un incontournable pour des néophytes.

3.2 Gameplay

Comme vu précédemment, *Elves Must Live* est un Tower Defense, en vue 3^e personne où le joueur devra défendre une certaine zone des vagues d'ennemis successives. Pour cela, il aura à sa disposition un arsenal de tourelles. La particularité de notre projet est qu'il comprendra également une partie shooter permettant au joueur d'affronter directement les mobs en plus de la partie Tower Defense classique.

Pour la partie Tower Defense pure, le joueur aura la possibilité de choisir différentes tourelles, de les positionner à des endroits stratégiques, en binôme avec des barricades par exemple. Chaque monstre tué rapportera des points et de l'argent. Cet argent permettra d'améliorer les tourelles et d'acheter de nouveaux modèles. Les points accumulés permettent, eux, de faire progresser un arbre de technologie. En fin de partie, ces points serviront également à enregistrer un score de partie.

Pour la partie en combat direct, un arme simple sera initialement disponible et permettra de compenser les lacunes de certaines tourelles. Une possibilité d'amélioration de l'arme pourra aussi apparaître au fur et à mesure de l'aventure avec une éventuelle progression du joueur en terme d'expérience.

Elves must live aura différents mode de jeu disponible en solo :

- un mode dit histoire où le joueur combattrait sur des cartes définies avec des objectifs variés
- un mode libre où le joueur devra chercher à faire le meilleur score sur des cartes de son choix
- un mode infini où le joueur devra résister le plus longtemps possible face aux vagues successives d'orcs féroces sur une carte de son choix

Les lieux de combats seront des "arènes" où les monstres prendront des chemins différents. Chaque arène aura sa spécificité. Par exemple, une arène pourrait être sur plusieurs niveaux tandis qu'une autre pourrait se baser sur un labyrinthe et les joueurs devront anticiper le chemin pris par les monstres.

3.3 Interface

L'interface correspond ici à tous les indicateurs et les fonctionnalités avec lesquelles le joueur va pouvoir interagir durant sa partie; l'ATH en somme. Il sera ici composé de sa barre de vie, de son score, de son argent disponible du nombre d'ennemis restant et de la vague en cours. En outre, il permettra d'accéder au menu du jeu (options, sauvegardes...) mais également aux arbres de technologies/compétences. L'interface comprends aussi le menu principal du jeu permettant d'accéder aux différentes fonctionnalités (solo, multijoueur, réglages...)

3.4 Intelligence Artificielle

L'intelligence Artificielle (IA) est la clé de ce type de jeu. Ce terme, IA, désigne en fait l'arborescence des choix programmés selon les situations que le joueur rencontre. En d'autres termes, cela désigne l'ensemble des scripts qui vont régir le comportement des ennemis, le pathfinding, leur agressivité, ainsi que celui des tourelles (Tower). Par exemple, dans le cas où un obstacle barre la route d'un monstre, ce dernier va "choisir" de le contourner ou de l'attaquer selon le choix le plus évident.

3.5 Audio

Comme l'a dit un grand critique cinéma : On peut pardonner une image pourrie mais pas un son pourri. Le son du jeu est un élément important pour permettre d'immerger au mieux le joueur dans sa partie. L'audio désigne en fait l'ensemble des musiques d'ambiances, des effets sonores et des dialogues. Chaque son doit être adapté à la situation et se déclencher au bon moment. On voit mal une scène triste sur une épique musique de bataille.

3.6 Multi-Joueur

Une fonctionnalité qui est importante à nos yeux et qui est peu commune dans ce type de jeu est le multijoueur. En effet, nous souhaiterions intégrer dans notre jeu deux types de combats multijoueurs : une coopération et un versus avec divers modes de jeu. Le mode coopération se basera sur le solo, mais avec un joueur supplémentaire en renfort et le versus sur une défense de sa propre ligne avec la possibilité d'envoyer des vagues de monstres personnalisées contre son adversaire et sa ligne.

3.7 Site Web

Afin que tout le monde puisse suivre l'avancement du développement du projet, un site web semble être la meilleure solution. Complété par les réseaux sociaux, tout sera en place afin de maintenir le fil d'actualité. De plus, le site bénéficiera d'un système de login pour créer des profils joueurs ou encore télécharger des documents tel que le cahier des charges ou les rapports de soutenance. Dans le futur, un éventuel système de trophées synchronisé au site web pourrait même voir le jour. Nous y intégrerons du PHP pour le rendre dynamique et pour le mettre à jour facilement.

4 Coût général

4.1 Software

Voici une liste des logiciels utilisés tout au long du projet, ainsi que leurs coûts. Des investissements n'ont pas été nécessaire car nous utilisons en majorité des logiciels gratuits et si possible Open Source.

Unity : Unity est notre moteur de jeu. Il est multiplateforme et très répandu dans le domaine. Il propose une licence gratuite, qui nous permet de l'utiliser encore plus aisément. C'est le pilier du projet pour développer notre jeu vidéo dans de bonnes conditions.

Visual studio : Visual Studio est un IDE (Environnement de développement) permettant, entre autres, de coder en langage C#. Couplé à Unity, il permettra d'écrire des scripts pour notre jeu vidéo, et pourra le rendre plus dynamique.

Blender : Blender est un logiciel de modélisation 3D qui va nous permettre de créer nos propres assets 3D pour les intégrer à notre jeu. Il est gratuit et Open Source, c'est pour cela que nous l'avons choisi.

Discord : N'étant pas tous dans la même classe, et n'ayant pas tout le temps les mêmes horaires, un outil de communication est obligatoire. Nous avons choisi un logiciel récent, Discord, qui nous permet d'avoir plusieurs salons textuels et un salon audio pour discuter autour du projet.

GIMP : GIMP est un logiciel de retouche d'images. Pour retoucher quelques images tel que les logos ou d'autres images en 2D, un logiciel de retouche peut être nécessaire, mais Paint va vite montrer ses faiblesses. C'est pourquoi nous avons opté pour le logiciel gratuit GIMP.

L^AT_EX : Nous avons besoin pour rendre ce fichier d'un logiciel de traitement de texte performant. Word nous semble un très mauvais choix. Nous avons donc opté pour L^AT_EX qui nous paraît le plus pratique et le plus adapté. A moins que ce ne soit obligatoire...

Git : Travaillant en groupe, un système de gestion de version qui permet de mettre en commun nos travaux efficacement nous paraît indispensable. Nous avons opté pour le plus connu, git, couplé

à un serveur BitBucket. De plus, pour les plus apeurés des consoles noires, un GUI peut être nécessaire tel que SourceTree par exemple.

Serveur Web : Ce n'est pas un logiciel à proprement parler, mais il est indispensable pour héberger notre site web. De plus, nous avons opté pour un hébergeur épiteen pour notre site. Pour développer notre site, nous aurons sûrement besoin d'un éditeur de texte performant tel que Notepad++ et un gestionnaire FTP tel que FileZilla, tous évidemment gratuits.

Nom	Description	Coût
Unity	Moteur de jeu	0€
Visual Studio	IDE C#	0€
Blender	Création 3D	0€
Discord	Communication	0€
GIMP	Édition d'image	0€
L ^A T _E X	Traitement de texte	0€
Git	Gestion de version	0€
Serveur Web	Site web	12€
Total		12€

4.2 Hardware

Un ordinateur un minimum performant avec une bonne partie graphique est quasiment obligatoire pour la réalisation d'un tel projet. C'est pourquoi nous avons dû investir dans du matériel informatique :

Nom	Marque	CPU	GPU	Ram	Prix
Thomas L.	HP	i5-6300HQ	GTX 950M	8 GO	650€
Clément G.	ASUS	i5-6600k	GTX 1060	16 GO	1300€
Louis H.	Shinobee	FX 8300	GTX 1060	16 GO	600€
Antoine N.	ASUS	i7-6500U	940MX	8GO	1000€

5 Répartition des tâches

Répartir les tâches pour avancer efficacement est indispensable. Ainsi, nous avons défini plusieurs pôles de travail. Dans chacun de ces pôles, deux personnes (un responsable et un suppléant) ont été désignés selon les compétences de chacun pour être le plus productif possible. Voici ci-dessous le tableau résumant la répartition :

Technologie	Thomas L.	Antoine N.	Clément G.	Louis H.
Environnement/Son	-	●	★	-
Gameplay	★	-	-	●
Multi	●	★	-	-
IA	-	●	-	★
Site Web	●	-	★	-
Interface	-	-	●	★

★ : *Responsable*

● : *Suppléant*

6 Avancement du projet

Il est important de bien gérer son temps pour respecter les deadlines. Ainsi, avec ce tableau, nous nous imposons un certain rythme qui nous permettra de rendre nos travaux en temps et en heures aux différentes soutenances.

Nom	Soutenance 1	Soutenance 2	Soutenance 3
Environnement/Son	20%	50%	100%
Gameplay	50%	80%	
Multi	10%	50%	
IA	20%	60%	
Site Web	75%	90%	
Interface	40%	80%	

Les raisons pour lesquelles le tableau est programmé de la sorte sont les suivantes :

- Pour coder un jeu efficacement, nous aurons besoin de bases solides et c'est pour cela que les mécaniques du gameplay doivent être implémentées le plus rapidement possible. Tout le reste du jeu se base en effet sur le gameplay.
- Quant à eux, l'amélioration des graphismes, l'ajout de textures et de nouveaux éléments d'environnement seront tardifs car ils ne sont pas vitaux dans les premières phases de développement. Il en va de même pour la partie son.
- La fonctionnalité multijoueur se verra balbutiante dans les premières heures du jeu car elle ne pourra être adaptée efficacement que quand le gameplay sera déjà rodé.
- La programmation de l'IA va être linéaire, car elle suivra au fur et à mesure l'ajout de nouveau éléments seront ajoutés au jeu (de nouveaux ennemis, de nouvelles tourelles...) qui se fera tout au long du développement.
- Le site, quant à lui, recevra son architecture presque finale dès le début du projet. Son principal but est de pouvoir se tenir informé sur le développement du jeu à l'aide de publications régulières. Des fonctionnalités supplémentaires sur la synchronisation avec le jeu viendront plus tard.
- Enfin, l'interface joueur évoluera progressivement : simpliste au début, se résumant aux informations essentielles (vie, score...) avant de se doter d'un menu, d'un arbre de technologies et éventuellement de compétences.

7 Conclusion

Ainsi, ce cahier des charges va prendre fin. Nous avons un groupe de travail motivé et qui sera sans aucun doute très efficace. Ce travail de groupe sera un moyen de mettre en commun nos connaissances et de mélanger nos univers afin de nous diversifier dans le domaine de l'informatique, pour apprendre à utiliser de nouveaux langages, de nouveaux systèmes ou encore de nouveaux logiciels tout au long du projet. Ce projet bénéficiera donc des points forts de chacun tout en gommant nos points faibles, pour garder le meilleur de nous tous.

De plus, nous misons sur une bonne communication, notamment à travers les réseaux sociaux (Facebook et Twitter) et notre site web, qui sera bientôt fonctionnel, pour informer les internautes intéressés des avancées du projet.

Nous espérons donc pouvoir rendre un travail complet à la fin de ce semestre, mais comme le disait Ronald Reagan, En travaillant ensemble et réunissant nos richesses nous pouvons accomplir de grandes choses.

Les objectifs sont maintenant fixés, il ne reste plus qu'à les atteindre et peut être même les dépasser.

