



Cahier des charges du projet - « The ByteCoder »

BLOC Software

Epita

Pierre « BLUESCREEN »BOULAY boulay_p

Serge « Kh4L »PANEV panev_s

Paul-Henry « Dr_c0w »PERRISSEL perris_p

Alessandro « d0t_iKs »PISU pisu_a

Table des matières

1	Origine et nature du projet	2
2	Présentation des membres du groupe	3
2.1	Pierre « BLUESCREEN »BOULAY	3
2.2	Serge « Kh4L »PANEV	4
2.3	Paul-Henry « Dr_c0w »PERRISSEL	5
2.4	Alessandro « d0t_iKs »PISU	6
3	Techniques de développement	7
4	Mécaniques de jeu	9
5	Site web	10
6	Graphismes	11
7	Scénario	12
8	Répartition des tâches	13
9	Coûts de Production	14

1 Origine et nature du projet

Notre groupe est constitué de quatre membres ayant un intérêt prononcé pour la programmation. En effet, nous avons tous déjà pris part à des projets qui nous ont apportés des connaissances dans ce domaine. Forts de ces expériences, nous avons décidé de travailler ensemble sur ce projet. Pour nous, le développement d'un jeu vidéo en équipe est un excellent moyen d'apprendre, d'une part à programmer un jeu vidéo, mais surtout à travailler en groupe sur un projet, ce qui implique savoir diviser les tâches, rester soudé mais avant tout résoudre les conflits, prendre les bonnes décisions et travailler intelligemment.

Le nom de notre équipe, « BLOC » est l'acronyme de Best Lines of Code. Ce nom montre à quel point nous prenons du plaisir à programmer un jeu vidéo. Nous étions tous d'accord pour travailler sur un jeu vidéo en deux dimensions. Etant tous les quatre fans de jeu de plateformes, nous avons décidé de concevoir un jeu de ce type. Notre objectif est de faire vivre une aventure épique au joueur à travers notre univers, qu'il prenne plaisir à y jouer. Nous souhaitons élaborer un jeu qui soit aussi agréable et amusant pour le joueur à jouer que pour nous à développer.

Notre jeu prendra la forme d'un jeu de plateformes classique, le joueur contrôlera un personnage qui pourra se déplacer, sauter et se battre contre des ennemis. Le personnage évoluera dans des environnements dynamiques et captivants. L'action se déroule dans un univers où règnent la mémoire et les registres. Dans un monde où un processus¹ corrompt le noyau du système d'exploitation. Un seul programme peut tout sauver, le ByteCoder. Sa mission, même s'il ne l'accepte pas, est de localiser l'adresse du processus maléfique et de terminer tous ses threads² afin de stabiliser le noyau. Le joueur va vivre une aventure unique à travers les mécaniques qui entrent en jeu dans le fonctionnement d'un microordinateur.

Le nom de notre projet provient du « bytocode » (« Code octal » en Français) que l'on obtient lorsque l'on compile un langage de haut niveau tel que le java ou le C#. C'est pour cette raison que notre jeu porte fièrement son nom : « The ByteCoder ». Nous voilà partis pour une formidable aventure dans l'univers du développement vidéoludique, du travail d'équipe et du café. Taïaut, taïaut, taïaut !

1. Programme en cours d'exécution, plusieurs processus peuvent s'exécuter simultanément.
2. Listes d'instructions qui composent les processus.

2 Présentation des membres du groupe

2.1 Pierre « BLUESCREEN » BOULAY

Hello World! Comme mon pseudonyme l'indique, je suis très attaché aux logiciels Microsoft. Je vous laisse imaginer le plaisir que me procure l'idée de développer un programme pour Windows en C# .net. Depuis très longtemps j'ai toujours été attiré par tout ce qui touche à l'informatique. Je me suis mis au C puis au C++ en début de seconde, j'ai utilisé des bibliothèques telles que Qt, SDL, SFML et bien sur l'API WIN32. Je me suis plus tard initié à des langages de plus haut niveaux tels que le java ou encore le Visual Basic .net mais je suis tombé amoureux du C++. C'est cet amour qui m'a permis de travailler sur quelques petits projets tels que des programmes de recherches de fichiers en fonction de certains critères, des programmes de messagerie instantanée, etc.

J'ai également travaillé sur un jeu vidéo en deux dimensions lorsque j'étais au lycée. C'est un avantage pour notre groupe. En effet je me souviens très bien de toutes les erreurs que j'ai commises et je sais aujourd'hui comment les éviter. De plus j'avais utilisé une bibliothèque qui ne proposait pas autant de fonctionnalités que XNA, pour cette raison je suis souvent agréablement surpris lorsque je travaille sur quelque chose de complexe et qu'une fonction de XNA ou du C# me simplifie le travail.

Je suis du genre à ne pas avoir peur de passer la nuit sur un projet parce que quelque chose ne fonctionne pas et cela m'empêche de dormir sereinement. Pour moi, coder un jeu n'est pas une corvée, je prends du plaisir à voir un projet se construire petit bout par petit bout.

C'est la première fois que je travaille avec un groupe de 4 personnes mais je suis assez optimiste sur l'avenir de notre projet. Travailler en groupe permet d'avoir les visions de 4 personnes différentes sur le projet, donc quatre fois plus d'idées, de force de travail et d'ambition.

2.2 Serge « Kh4L »PANEV

Bien le bonjour! Moi c'est Kh4L, et voici ma biographie. Né dans le 19ème un beau jour d'été il y a de cela 19 ans, je me suis tout de suite intéressé à l'informatique, au bout d'une semaine, le C n'ayant plus de secret pour moi j'ai commencé l'assembleur x86, j'ai travaillé alors pour Intel. Voilà mon enfance, j'ai oublié pas mal de choses, faute de ne pas avoir pratiqué... Plaisanteries (et rêveries) à part, je me suis mis à coder il y un an et demi, en C, C++ et divers langages de scripts web et UNIX, prenant mon pied plus que jamais. Comme beaucoup d'étudiants, je voue un intérêt certain aux jeux vidéo.

Or ce qui m'intéresse le plus dans les jeux, c'est leur conception, les moyens techniques et artistiques mis en œuvre pour y parvenir. Fan inconditionnel de musiques en tout genre : jazz, rock, funk, électronique, celtique, mes « talents » de guitariste m'ont mené à former un groupe pendant un peu plus de 3 ans avec des amis, ce qui m'a permis de découvrir les joies des concerts mais aussi du travail de groupe... Et me voilà maintenant membre de BLOC Software pour créer, à l'aide de mes camarades, du C# et de XNA, un jeu de plateformes dont seuls les classiques du genre pourront prétendre à se comparer.

La force créatrice, l'imagination artistique ainsi que sophistication technique de notre groupe me motive plus que jamais à créer plus et à me servir de mes années d'expérience dans le domaine de la musique et dans le domaine de l'informatique, afin d'apporter de l'innovation et de l'expertise dans cette épopée numérique qu'est The ByteCoder. Voilà pour les éloges à mon égard, passons à la suite!

2.3 Paul-Henry « Dr_c0w »PERRISSEL

Les vaches c'est bien et c'est gentil ! Ca mange de l'herbe et ça nous permet d'avoir du lait au quotidien ! Donc soyez gentils avec mes amies les vaches !

Sinon, pour être plus sérieux, mais juste un peu, il ne faut pas trop abuser, je suis une jeune personne de 18 ans. Je me passionne pour tout ce qui est informatique et j'espère un jour être le premier bovin à travailler dans une entreprise informatique ou dans la partie informatique d'une entreprise, heureusement que j'avais précisé « juste un peu ».

Sur un plan moins professionnel, je joue de la guitare, plus pour le plaisir j'apprends quelques morceaux de temps en temps, j'aurai bien aimé me mettre à la batterie sauf que dans la vie on ne peut pas tout avoir, et j'ai appris la flûte mais de ne plus en jouer ne me manque absolument pas. En termes de sport, j'ai pratiqué l'escrime pendant quatre ans, et le tennis pendant un an de moins. Bien que ces instruments de musique et ces sports se pratiquent la majorité du temps, pour ma part, en équipe d'une seule personne, je trouve que l'esprit d'équipe est très important.

Par rapport à la programmation, un peu comme Kh4L, j'ai approfondi les langages qui se rapportent au Web : pour moi, cela se limite à HTML et CSS. Après avoir eu quelques soucis avec des extensions pour le célèbre jeu Minecraft (oui j'apprécie les jeux vidéo, comme tous les membres du groupe), j'ai voulu me lancer dans le Java, puis je me suis vite intéressé au C, sauf que le Bac est arrivé, celui-ci est un langage bien différent qu'il faut bien approfondir pour réussir, cela a nécessité quelques révisions, et m'a obligé à mettre de côté la programmation. Pour faire un peu d'autobiographie, avant de toucher à la partie logicielle d'un ordinateur, je me suis d'abord intéressé à la partie matérielle. Puis, au fur-et-à-mesure, je me suis rendu compte que la partie matérielle était loin d'être suffisante pour faire fonctionner un ordinateur, c'est à partir de là que je me suis intéressé à toute la partie logicielle. Mais ce qui m'intéresse le plus dans l'informatique, c'est la sécurité, j'aimerais bien m'en rapprocher, mais comme on dit : « chaque chose en son temps ».

2.4 Alessandro « d0t_iKs »PISU

Bonjour, je me présente Alessandro, âgé de 18 ans et depuis quelques années déjà passionné d'informatique et depuis très longtemps fan de jeux-vidéos. J'ai eu la chance de pouvoir dès la première avoir accès à une option, Informatique et Sciences Numérique, option que j'ai pu continuer en terminale sous la forme d'une spécialité qui m'a permis d'apprendre les bases de la programmation, notamment en C. Avant mon entrée à l'EPITA je n'avais jamais eu l'occasion ni la curiosité d'utiliser le C#, qui à mon goût est un outil puissant pour développer un projet de type jeu de plateforme en deux dimensions.

Concernant le projet de façon moins technique je trouve que c'est une opportunité de vivre une expérience de groupe dans ce domaine. Par ailleurs ce projet nous permettra de rajouter à notre jeu toutes les mécaniques et autres éléments de gameplays que nous aurions voulu trouver dans les autres jeux. Ce qui par ailleurs rajoute un grand intérêt au projet sachant que la plupart de ces choses ne semblent pas si facile que cela à implémenter. Qui plus est, je pense avoir trouvé des acolytes d'un niveau plus que raisonnable et ayant des expériences différentes ce qui je pense, pourraient permettre au projet de s'étoffer de fonctionnalités qui n'auraient pas émergées sans cette rencontre.

De plus ayant toujours aimé dessiner dans mon temps libre je pourrai faire profiter au jeu d'un univers graphique original dans lequel apparaîtront de nombreuses références liées au monde de l'informatique et aux différents milieux geeks qui nous entourent.

Maintenant je ne peux qu'espérer vivre une expérience enrichissante auprès de mes partenaires et développer comme il se doit notre projet.

3 Techniques de développement

Aucun d'entre nous n'avait de connaissances en programmation en langage C# avant de commencer ce projet. Néanmoins nous savions que sa syntaxe était proche de celle du C++, étant donné que la majorité d'entre nous avons quelques connaissances dans ce langage, nous avons fait le choix de programmer notre jeu en C#.

Le C# fait partie du Microsoft .net Framework. Le .net est un ensemble de bibliothèques permettant d'exécuter des langages de haut niveau tels que le C#, le Visual Basic, le F#, etc. Le fait de travailler avec un langage de haut niveau présente de nombreux avantages. En effet, ce langage comporte beaucoup de fonctions pré-faites, ce qui signifie qu'il ne sera pas nécessaire pour nous de les refaire, en conséquence nous pourrons nous concentrer sur le cœur de notre jeu. La programmation dans un langage .net possède également l'avantage d'utiliser de la mémoire managée, ce qui signifie que la mémoire que notre programme utilisera est gérée quasiment automatiquement par le système d'exploitation.

Néanmoins, le .net Framework présente aussi des inconvénients. Tout d'abord, tout programme écrit avec un langage du .net ne peut être exécuté que sous un système d'exploitation Microsoft (Windows, Xbox ou Windows Phone), même s'il existe des tentatives de portage du .net sous Linux, notre jeu ne pourra être exécuté que sous Windows. De plus, il est facile de retrouver le code source d'un exécutable programmé en C#. Mais le langage ne fait pas tout, il faut également choisir une bibliothèque graphique.

Nous avons hésité entre Microsoft XNA (XNA's Not Acronymed) et MonoGame, après plusieurs recherches sur le sujet, nous avons décidé d'utiliser Microsoft XNA car c'est une bibliothèque qui existe depuis plus longtemps que MonoGame, ce qui implique qu'il y a beaucoup plus de cours et de forums sur XNA, et ce malgré le fait que Microsoft ait arrêté de la mettre à jour. Ce contenu sur internet pourra nous aider au cours de notre développement.

Après avoir choisi un langage et une bibliothèque graphique, la programmation peut enfin commencer. Mais avant de se lancer tête baissée dans le cœur de notre code, nous sommes concertés et nous avons dégagé quatre règles que nous suivrons lors du développement de notre jeu, nous avons nommé ces règles SFPC, ce qui signifie :

- **Sécurité** : Le programme doit prévoir l'imprévisible et gérer le maximum d'erreurs possibles.
- **Flexibilité** : Tout le monde change d'avis. Ce que nous faisons aujourd'hui peut nous déplaire la semaine prochaine. Pour cette raison, notre code ne doit pas être rigide, chaque élément du programme doit être paramétrable, et le code une fois écrit ; ne doit pas avoir besoin d'être modifié plus tard.
- **Performance** : Notre code doit s'exécuter le plus rapidement possible en utilisant le moins possible les ressources de la machine.
- **Clarté** : Notre code doit être le plus clair possible, le nom des variables et des fonctions doit être explicite. Lorsque l'on voit une fonction, on doit immédiatement comprendre à quoi elle sert.

Néanmoins, le fait d'éditer simultanément du code informatique n'est pas chose facile. En effet on peut facilement imaginer les problèmes qui peuvent arriver si par exemple deux personnes modifient le même fichier, ou si quelqu'un change le nom d'une fonction, etc. Pour résoudre ces problèmes, nous avons décidé d'utiliser un système de gestion centralisée de code source. Dans la mesure où nous allons travailler dans un avec un environnement .net, il est logique d'utiliser Microsoft Visual Studio. C'est un IDE³ très complet qui fournit de nombreux outils qui nous faciliteront le développement. Microsoft Visual Studio permet la synchronisation ainsi que la résolution automatique des conflits par l'intermédiaire de Microsoft Team Foundation Server. Nous avons donc décidé d'utiliser cette méthode pour gérer la synchronisation de nos travaux. Notre licence Microsoft Visual Studio fournie par Epita nous permet de créer notre propre Team Foundation Server sans frais supplémentaires.

3. Integrated Development Environment ou Environnement De Développement Intégré en français.

4 Mécaniques de jeu

Le jeu reposera sur le principe fondamental du jeu de plateformes : un personnage s'affichant au milieu de l'écran, contrôlé par le joueur et évoluant dans un univers en 2 dimensions.

Le personnage, animé grâce à des touches personnalisables (flèches directionnelles par défaut) avancera dans différents « mondes ». Chaque monde a son propre univers, ses propres personnages et son propre thème. Sur son chemin, le joueur rencontrera également des ennemis l'entravant et ne lui laissant guère le choix que de les affronter. Le système de combat est un élément important du jeu. Le héros aura plusieurs attaques, il pourra attaquer au corps à corps et à distance. Il lui sera également possible d'utiliser des attaques dites « spéciales », il lui sera possible de les enclencher une fois qu'il aura atteint un nombre de points minimum. Ces attaques, extrêmement puissantes pourront être débloquées, et améliorées. Les niveaux seront ponctués par un système d'amélioration des caractéristiques du personnage. Le joueur pourra échanger les points qu'il aura obtenus en parcourant les niveaux du jeu contre des améliorations qui augmenteront la puissance de son avatar.

Parmi les ennemis, en plus de ceux à combat en mêlée et ceux à distance, on retrouvera les « boss » : des ennemis plus résistants, demandant donc plus de concentration et plus de temps au joueur afin de les battre. Ils apparaîtront principalement en fin de niveau et laisseront des bonus importants après leur mort.

Les cinématiques permettront de faire connaître au joueur l'avancement de l'histoire. De même, les dialogues du héros avec les personnages donneront des indications et des conseils pour finir le niveau.

Des bonus seront également placés dans les niveaux. Le joueur n'aura qu'à passer dessus afin de voir ses caractéristiques augmenter (vitesse, vitesse d'attaque, probabilité de coup critique, invincibilité) ou de se voir doter de nouvelles compétences et armes inédites.

5 Site web

Disponible sous le domaine blocsoft.net, le site du projet déjà en ligne depuis le mois de janvier aura pour objectifs :

- **Introduire le jeu** : la page d'accueil servant de point d'entrée à l'univers du jeu, visible dès la première visite, notamment via une bande-annonce de la version actuelle du jeu. Un bouton <Télécharger> est directement accessible sur cette page, pour que l'utilisateur puisse immédiatement installer le jeu s'il le souhaite.
- **Présenter le jeu** : sur cette page, l'histoire du jeu sera décrite et l'intrigue sera énoncée. Des images du jeu, accompagnées de descriptions permettront au futur joueur de se familiariser avec le jeu. Les mécaniques de jeu auront un paragraphe dédié.
- **Téléchargement des différentes Ressource** : Le jeu, les médias (images, bandes annonces) ainsi que le rapport du projet. Sur cette même page, les liens vers les sources seront mis en ligne. C'est-à-dire les technologies qui nous permettrons de développer le jeu.
- **Communiquer** : un blog sera régulièrement mis à jour avec l'avancement du projet. Des billets annotés de leur date constitueront l'actualité du jeu. Lorsque de nouvelles mises à jour (nouvelles versions du jeu) seront disponibles, une note détaillée listant toutes les améliorations et les nouveautés sera postée.
- **Présenter le groupe** : les 4 membres du groupe seront présentés sur cette page. Les motivations et objectifs seront aussi exposés dans cette section du site. Les adresses mails des membres seront aussi disponibles, afin de permettre à l'utilisateur de nous contacter.

Le site web du groupe sera mis en ligne sur un serveur Apache installé sur le PC Debian (GNU/Linux) d'un membre du groupe. Le back-end⁴ sera codé en PHP, un langage interprété par le serveur permettant de générer des pages dynamiques, cet aspect sera essentiellement utilisé par le blog. Le front-end⁵ sera codé en HTML/CSS pour la structure et la mise en forme, ainsi qu'en JavaScript pour créer des animations et du contenu dynamique.

4. Désigne la partie non visible par l'utilisateur, en l'occurrence, le back-end est du côté serveur.

5. Langage interprété par le navigateur côté client.

6 Graphismes

Les graphismes seront faits de façon à retranscrire l'univers du jeu, tout en restant originaux. Pour cela toutes les planches d'animations et de décors seront dessinées et éditées par les membres de BLOC Software. En effet, l'ensemble des éléments graphiques du jeu sera tout d'abord créé à l'aide d'outils rudimentaires : le papier et le crayon, puis numérisé pour être ensuite retravaillé à l'aide de logiciels tels que Paint ou Photoshop. De plus cela permet une immersion bien plus importante, pouvoir afficher des animations sur mesures donne la possibilité d'ajouter des d'actions plus précises et plus cohérentes. Rajoutons à cela que les plus observateurs remarqueront des références plus ou moins claires dans les représentations des différents décors et personnages du jeu.

7 Scénario

L'univers prends place dans un monde gouverné par les lois de l'algorithmique et du numérique. En effet, le héros principal (appelé ByteCoder) doit rétablir l'ordre au sein du système corrompu par un processus. Ce dernier ayant eu vent de la rébellion du ByteCoder, il décide d'empêcher le héros d'avancer dans sa quête. Pour cela, il ordonne à des objets contenus dans la mémoire d'arrêter ce ByteCoder. Le ByteCoder est assisté dans sa quête par des PNJ (Personnages Non Joueurs) qui ont été affaiblis par la corruption du système. Ils ne sont donc capables que d'échanger des objets, des bonus ou des armes contre la monnaie récupérée au fur-et-à-mesure de son avancée.

Le héros, originaire des cieux (c'est à dire une page web), se verra dans l'obligation de parcourir les différentes couches de la programmation, des langages de haut niveau vers les plus bas : les limbes du code, le noyau Kernelus, une ancienne puissance oubliée. Pendant cette quête, il devra affronter de nombreux ennemis. Sa tâche sera donc de plus en plus difficile à accomplir, puisque chacun des langages rencontrés permettent la création d'ennemis de plus en plus puissants. Le ByteCoder devra également combattre contre des ennemis finaux, tous uniques. Ces derniers seront emblématiques de chacun des mondes et seront plus puissants que les ennemis rencontrés dans les niveaux précédents..

8 Répartition des tâches

A la suite d'une réunion de groupe de projet, nous avons réparti les tâches de chacun. Pour cela, nous avons pris en compte l'expérience de chacun d'entre nous avec la programmation pour que les objectifs puissent être tenus dans le temps imparti tout en fournissant une dose de travail acceptable.

Les tableaux ci dessous résument les tâches de chacun en fonction de l'avancement des soutenance.

Soutenance 1					
	Pierre	Serge	Paul-Henry	Alessandro	Avancement Global
Gestions des collisions	XX ⁶	X			Achevé
Scripts	XX		X		Achevé
Fichiers de configuration	XX			X	Achevé
Editeur de niveau	X			XX	En cours
Image du jeu			X	XX	En cours
Particules	X	XX			Achevé
Site Internet	X	XX	X	X	En cours
Menus	X		XX	X	En cours
Scénario du jeu			XX	X	En cours
Multijoueur local					Non commencé
Multijoueur Reseau					Non commencé
Ennemis et Intelligence Artificielle		XX	X		En cours
Système de Sauvegarde					Non commencé

Soutenance 2					
	Pierre	Serge	Paul-Henry	Alessandro	Avancement Global
Gestions des collisions	XX	X			Achevé
Scripts	XX		X		Achevé
Fichiers de configuration	XX			X	Achevé
Editeur de niveau	X			XX	Achevé
Image du jeu	XX		X	XX	En cours
Particules	X	XX			Achevé
Site Internet	X	XX	X	X	Achevé
Menus	XX	X	XX	X	En cours
Scénario du jeu			XX	X	En cours
Multijoueur local	XX	XX	X	XX	En cours
Multijoueur Reseau	XX	XX	X	XX	En cours
Ennemis et Intelligence Artificielle		XX	X		Achevé
Système de Sauvegarde	XX	X			En cours

Soutenance 3					
	Pierre	Serge	Paul-Henry	Alessandro	Avancement Global
Gestions des collisions	XX	X			Achevé
Scripts	XX		X		Achevé
Fichiers de configuration	XX			X	Achevé
Editeur de niveau	X			XX	Achevé
Image du jeu			X	XX	Achevé
Particules	X	XX			Achevé
Site Internet	X	XX	X	X	Achevé
Menus	X		XX	X	Achevé
Scénario du jeu			XX	X	Achevé
Multijoueur local	XX	XX	X	XX	Achevé
Multijoueur Réseau	XX	XX	X	XX	Achevé
Ennemis et Intelligence Artificielle		XX	X		Achevé
Système de Sauvegarde	XX	X			Achevé

Lorsque nous travaillons sur notre projet, nous nous occupons chacun de notre partie si celle-ci est individuelle, puis nous archivons sur notre Team Foundation Server. Nous essayons également de nous réunir tous ensemble au même endroit pour se mettre dans des conditions de travail optimales dans lesquelles il est plus facile de s'entraider. Ainsi, si l'un d'entre nous rencontre une difficulté, les trois autres membres peuvent l'aider. De plus, pour les tâches nécessitant d'être en groupe, cela est bien plus pratique. Nous essayons donc d'anticiper sur le travail à réaliser et les éventuels problèmes que nous pourrions rencontrer lorsque nous sommes tous les quatre pour ainsi pouvoir gagner du temps lorsque nous travaillons seuls.

6. XX : Développeurs Majeurs
X : Co-développeurs

9 Coûts de Production

Equipement/Service	Prix
License Adobe Photoshop CS6	1,369.90€
Domaine : blocsoft.net	29.99€
Café,Coca, Pizza, Kinder Bueno	2442.75€
Visual Studio Professional 2013	939.99€

Le développement d'un logiciel engendre des frais. Le tableau ci-dessus présente les dépenses nécessaires à la réalisation du projet « The ByteCoder ».