

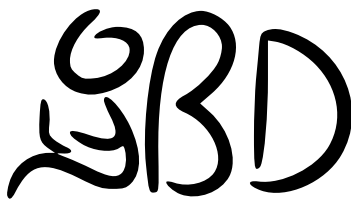
pacification

Cahier des charges

Brainless Devs

Thibault Allanon
Valérien Fayt

Antoine Gonzalez
Cédric Parpet



Dossier Projet Informatique
Info-Sup EPITA
Janvier 2018

Table des matières

1. Introduction

Nous sommes l'équipe *Brainless Devs*. Actuellement, nous travaillons sur notre projet de jeu vidéo nommé *Pacification*. Il s'agit là d'un jeu à vocation stratégique, orienté sur un tour par tour inspiré de la saga *Civilization* développé principalement par Sid Meier et Activision.

Pacification sera un jeu multijoueur dans lequel 2 à 4 joueurs s'affrontent sur un terrain généré procéduralement avec comme objectif de vaincre leurs adversaires. Le mode solo mettra le joueur au défi de survivre aux invasions des tribus barbares environnantes.

Avec une vue plongeante sur leur empire, les joueurs auront pour but d'accroître leur puissance. Sur le terrain, ils leur seront possible de trouver diverses ressources leur permettant de se développer davantage, pour finalement remporter la partie par domination militaire ou scientifique. La complexité du gameplay résidera dans l'approche stratégique de chaque joueur, qui sera accentué par un système d'économie interne à maîtriser.

Ce projet d'une grande ampleur nécessitera une équipe dévouée, soudée et organisée...



Calvin and Hobbes, Bill Watterson

2. Origine du projet

2.1 Présentation de l'équipe

- **Thibault Allançon (Chef de projet)** : Prophète GNU/Linux, venant apporter la bonne nouvelle et convertir les hérétiques vers le Saint OS. Gourou algorithmicien et dictateur durant son temps libre, il saura diriger ce projet **fermement** sous peine de violence verbale et physique.
- **Cédric Parpet** : Amateur de jeu de stratégie tel que la série des *Age of empire*, *Mythologie* et *Warcraft*, il a déjà codé les bases d'un jeu de stratégie au tour par tour pour son projet d'ISN. La légende dit qu'il code avec ses longs cheveux qui sont par ailleurs très soyeux (c'est de la fibre, à seulement 19€99/mois).
- **Valérian Fayt** : Il s'est découvert une passion pour les jeux de stratégie bien jeune et y a sans doutes passé trop de temps. Tout comme son voisin aux longs cheveux, il a lui aussi grandement apprécié la série des *Age of Empire*. Peu expérimenté au codage, ce projet sera pour Valérian l'occasion de développer ses capacités et repousser ses limites. Vers l'infini et l'au-delà !
- **Antoine Gonzalez** : Il vit en France, mais son fuseau horaire est incertain. Parfois il hiberne pendant des jours entiers. Parfois il code non-stop. Parfois il apprend quelque chose de nouveau, because why not. Sinon, il contemple le sens de son existence, entre deux ragequit (on le surnomme le Pape Salé). Mais il s'éloigne rarement de l'ordinateur, sauf pour s'alimenter, éventuellement. Il a plus de respect pour les CGU que pour son fournisseur internet.

2.2 Inspirations

À l’origine, nous hésitions avec un jeu de plateforme-réflexion. Quelques pizzas plus tard, et devant le fabuleux let’s play *Civilization* du Joueur du Grenier, le choix n’était plus. Cela sera un jeu de stratégie tour par tour ou rien. Plusieurs réunions nous ont permis de fixer les prémices du jeu, ainsi que les fonctionnalités voulues.

Pacification prend son inspiration dans divers jeux stratégiques cultes tels que *Age of Empire* mais aussi particulièrement *Civilization*. En effet, notre jeu ressemble aux jeux type 4X (eXplore, eXpand, eXploit, eXterminate) dont le premier représentant a été *Empire* (1977) et dont d’autres exemples sont très connus (la série *Civilization*, *Master of Orion*). Ces jeux ont des codes spécifiques comme la construction d’un empire, un gameplay au tour par tour, la gestion de leur économie, des améliorations technologiques, une vue de haut et globale sur la carte du monde.

Avec cet objectif en tête, il s’agissait alors de déterminer le nom, premièrement de notre équipe, mais aussi et surtout du jeu auquel nous allions donner vie. Cette partie fut sans conteste la plus longue de notre première réunion. L’objectif étant de combiner une appellation à la fois humoristique et originale, d’où l’ironie de *Pacification* puisque l’une des occupations principales du jeu est de « pacifier les tribus barbares », car comme disait Montaigne, « chacun appelle barbarie ce qui n’est pas de son usage ».

3. Description du gameplay

3.1 Carte et ressources principales

La carte sera découpée en tuiles hexagonales, et sera générée procéduralement. Les cases seront organisées par groupements de mêmes biomes, ayant chacun des caractéristiques particulières. Par exemple le biome forêt réduira la portée de vue des unités ainsi que leurs déplacements, le biome océan empêchera toutes constructions, ou le biome montagne bloquant tous déplacements d'unités.

Comme dans la grande majorité des jeux de stratégie aujourd'hui, nous mettrons en place un brouillard de guerre afin de rajouter un côté exploration dans le jeu masquant les territoires inconnus. Construire un bâtiment ou déplacer une unité à proximité du brouillard révélera une partie de la carte dans un rayon voisin de l'élément.

La carte contiendra également des ressources que le joueur devra exploiter afin de progresser. Ces ressources sont divisées en deux catégories :

- Les **ressources stratégiques** (métaux, chevaux...) serviront à construire et améliorer des bâtiments et unités tout au long de la partie.
- Les **ressources de luxe** (or, diamant...) seront indispensables pour garantir une stabilité économique, sans laquelle le joueur subira des révoltes, traduites en partie par des malus de développement.

3.2 Fonctionnement des unités

Le colon est une unité importante pour la progression du joueur : il sert à fonder une nouvelle ville. Il s'agira d'une unité coûteuse, et extrêmement vulnérable car sans défenses. Il devra être utilisé avec précaution.

Les ouvriers sont les unités sans défense qui construisent les exploitations de ressources sur les cases en contenant. De plus, ils sont nécessaires à la mise en place de routes permettant d'accélérer le déplacement des autres unités. Leur coût est relativement faible, et il est possible de capturer les ouvriers adverses.

Les militaires sont les seules unités capables de piller des cases de ressources, détruire des villes, et tuer ou capturer des unités. Il sera possible de les améliorer au cours de la partie au fur et à mesure des avancées technologiques de l'empire.

3.3 Caractéristiques des villes

Les villes représentent un aspect important de la partie : elles constituent l'empire du joueur, et permettent notamment la création d'unités, de bâtiments, ainsi que la génération de richesses. Ces dernières serviront à construire des bâtiments, ainsi qu'à créer et améliorer des unités.

Elles posséderont une population croissante, dont dépendra la production d'argent, de science, et de productivité. Cette population devra être satisfaite en ressources de luxe, sans quoi elle sera mécontente et moins performante.

3.4 Déroulement d'une partie

Le joueur commence la partie avec un colon et un guerrier, et doit construire sa première ville. Il devra ensuite étendre son empire tout au long de la partie, exploitant les ressources de la carte, en construisant d'autres villes, et en développant une armée.

Le joueur sera amené à se battre contre les adversaires en tour par tour, qui en solo prendront la forme de tribus barbares contrôlées par une IA.

Plusieurs conditions déterminent une fin de partie :

- **Victoire militaire** : en multijoueur lorsqu'il ne reste plus qu'un seul empire
- **Victoire scientifique** : en atteignant le dernier palier scientifique

Un joueur a perdu lorsque toutes ses villes sont détruites.

4. Conception et aspects techniques

Pour réaliser ce projet nous utiliserons le moteur de jeu *Unity*, l'avantage étant qu'il s'agit d'un moteur puissant, complet et grandement documenté. Ceci nous permettra de réaliser un jeu en 3D stable avec un multijoueur en un temps raisonnable. *Unity* supporte le scripting en C#, un langage de programmation orienté objet développé par Microsoft, que nous utiliserons tout au long du projet. Le paradigme orienté objet sera un choix indispensable pour maintenir un projet aussi conséquent, afin de profiter des avantages de l'encapsulation, de l'héritage, ainsi que du polymorphisme.

La partie multijoueur se basera sur les composants réseaux intégrés directement à *Unity*. Il est possible grâce à eux de faire un multijoueur basé soit sur un serveur dédié, soit sur un système *peer-to-peer*.

Les modèles 3D seront principalement tirés de l'Asset Store d'*Unity*, regroupant des milliers de modèles et d'animations quasi-professionnelles. *Blender*, un logiciel de modélisation, d'animation et de rendu 3D très populaire nous sera aussi utile pour créer nos propres modèles uniques au jeu.

Unity possède un module pour gérer les interfaces ainsi que les menus, que nous devrons nécessairement intégrer au sein du projet, car l'interface du jeu sera primordiale pour le joueur. Un *devmode* interne nous sera aussi très utile pour faciliter les tests de nos différentes fonctionnalités.

Afin de travailler en groupe efficacement, nous utiliserons *Git*, un gestionnaire de versions, ainsi que *Github* afin d'héberger les dépôts du projet.

5. Répartition des tâches

Nous nous sommes assignés des tâches spécifiques, toutefois chaque membre reste disponible pour participer aux autres parties du développement. Cela nous permet d'avancer, sans pour autant se verrouiller dans un domaine.

	Thibault	Antoine	Cédric	Valérian
Map	R	S		
IA	R			
Réseau		S	S	R
Graphisme			R	
UI			R	
Site web				R
Gameplay	S	R		S

R = responsable, S = suppléant

	Soutenance 1	Soutenance 2	Soutenance 3
Map	65%	100%	100%
IA	30%	70%	100%
Réseau	30%	75%	100%
Assets	50%	100%	100%
Interface	10%	80%	100%
Site	10%	95%	100%
Gameplay	40%	80%	100%
Budget pizza	100%	60%	null
Jouabilité	25%	60%	100%

6. Conclusion

Individuellement, chacun apprendra énormément via les tâches qui lui sont attribuées, mais aussi grâce à ses coéquipiers. Cependant, travailler en groupe sera pour la plupart d’entre nous une nouveauté, notamment à cette échelle, et constituera le plus gros challenge à surmonter afin de travailler efficacement.

Si le temps nous le permet, nous aimerions ajouter davantage de fonctionnalités à *Pacification*. Certains détails peuvent notamment améliorer grandement l’expérience de jeu, comme l’ajout de musique et de bruitages, mais aussi l’apparition de nouvelles unités et animations de ces dernières. Mettre en place une intelligence artificielle simulant un comportement proche d’un joueur serait d’autant plus intéressant pour le mode solo. Un système d’alliance et d’équipes serait aussi un excellent ajout pour le multijoueur qui se verra alors offrir une nouvelle dimension de gameplay.

Ce projet s’annonce chargé et nécessitera un investissement conséquent de toute l’équipe afin de mener à terme un produit final dont nous sommes fiers et auquel nous aurons envie de jouer. Il faudra également respecter nos plannings, car comme chacun le sait :

*« These violent deadlines have
violent ends »*

— ACDC, 2020

7. Références

Documentation Unity :

- Multijoueur
<https://unity3d.com/learn/tutorials/s/multiplayer-networking>
- Réseau
<https://docs.unity3d.com/560/Documentation/Manual/UNetOverview.html>
- Interface
<https://docs.unity3d.com/Manual/UISystem.html>

Autres éléments de recherche :

- Carte hexagonale
<https://www.redblobgames.com/grids/hexagons/>
- Génération procédurale
<https://forum.unity.com/threads/open-source-procedural-hexagon-terrain.233296/>
- Intelligence Artificielle
<http://forums.conifergames.com/index.php?topic=158.msg739#msg739>
- Liste de tutoriels sous Blender
<https://www.lynda.com/Blender-training-tutorials/1081-0.html>