

Projet Génie Logiciel

Rapport du Projet

Bastien Lepesant, Arya Jemo et Lucas Nicosia.

25 Mai 2015

Table des matières

1	Introduction	2
1.1	Contexte	2
1.1.1	Objet	2
1.1.2	Équipe	2
2	Spécification	3
2.1	Cahier des Charges	3
2.1.1	Calendrier	3
2.1.2	Environnement de travail	3
2.2	Objectif du projet	3
2.3	Livraison attendues	6
3	Réalisation	7
3.1	Les données	7
3.1.1	Les joueurs	7
3.1.2	Les batiments	7
3.1.3	Territoires	7
3.1.4	Troupes	8
3.2	Le moteur	8
3.2.1	La Map	8
3.2.2	Les phases de jeux	10
3.2.3	Les joueurs	10
3.2.4	La boucle de jeu	11
3.2.5	Fin du jeu	11
3.3	L'interface graphique	11
3.3.1	Les éléments graphiques	12
3.3.2	Les classes graphiques	12
3.3.3	Les extensions	13
4	Manuel Utilisateur	14
4.1	Configuration requise	14
4.2	Configuration le jeu	14
4.3	Jouer à Age of Swag : Conquest CMI	15
4.4	Autres informations	15
5	Déroulement du projet	19
5.1	La conception	19
5.2	La concrétisation	19
5.3	Finalisations et améliorations	19
6	Conclusion	20
7	Remerciements	21

Introduction

1.1 Contexte

Dans le cadre de notre deuxième année de licence en Informatique, nous avons l'opportunité de participer au projet Génie Logiciel encadré par M. Tianxiao LIU. Celui-ci consiste en la réalisation et la conception d'un logiciel java beaucoup plus complexe et de plus grande ambition que ceux réalisés jusqu'à maintenant. Nous avons eu la chance d'obtenir notre premier choix lors de la sélection des projets, soit le sujet **Conquête**.

1.1.1 Objet

Notre objectif est de réaliser un jeu de stratégie multijoueurs. Dans celui-ci, chaque joueur se voit attribuer une capitale puis tour à tour, chacun des joueurs va conquérir des territoires neutres ou ennemis à l'aide de soldats. Le gagnant est celui qui a conquiert toutes les capitales. Toutefois notre projet implique des mécanismes plus complexes afin de répondre au cahier des charges, tels que des alliances entre joueurs, le jeu en réseau ainsi que d'autres fonctionnalités que nous allons décrire dans ce rapport.

1.1.2 Équipe

Notre équipe est composé de trois CMI SIC¹ :

- Arya Jemo
- Bastien Lepsant
- Lucas Nicosia

1. Cursus Master en Ingénierie, systèmes intelligents et communicants

Spécification

2.1 Cahier des Charges

2.1.1 Calendrier

Durant toute la durée du projet, nous avons un calendrier précis à respecter :

Semaine du :	Objectif
19 Janvier	Cahier des charges
26 Janvier	UML modèle
02 Février	UML moteur
09 Février	Prototype graphique
02 Mars	Point d'avancement 1 : démo du prototype du projet
09 Mars	Mise en place Junit + Log4j
23 Mars	Mise en place rédaction LaTeX
06 Avril	Squelette des slides
13 Avril	Point d'avancement 2 : pré-soutenance du projet
18 Mai	Finalisation du projet + Remise de projet (au plus tard le 24 mai)
25 Mai	Soutenances de projet : Mardi 26 et Mercredi 27 Mai

2.1.2 Environnement de travail

Durant le projet, nous allons utiliser les outils suivants (sur nos ordinateurs personnels) :

- L'IDE Java **Eclipse**
- Le logiciel de gestion de versions **SVN**.

2.2 Objectif du projet

Solution envisagée

Dans toute cette partie, nous ajouterons de la couleur suivante **les objectifs bonus que nous aimerions réaliser si le temps nous le permet**. Comme résultat final, nous voulons obtenir un jeu jouable sur Linux, Windows et Mac. Ce dernier doit comporter un moteur de jeu, un moteur graphique et permettre de jouer contre d'autres joueurs et/ou contre l'ordinateur. **Le jeu est jouable en réseau, à plusieurs.**

Méthodes envisagées

L'idée du jeu que nous voulons réaliser peut se diviser en 3 parties :

- **Menu de jeu** : Lors du lancement du jeu, le logiciel affichera en premier temps un Menu permettant au joueur de réaliser les actions suivantes :
 - Lancer une nouvelle partie.
 - Charger une partie existante (ou supprimer une sauvegarde existante).
 - Quitter le jeu.
- **Génération de carte** : Notre logiciel devra permettre la génération d'une carte de jeu, composée d'un ensemble de territoires ayant chacun 2 caractéristiques liées :

- Une réserve des trois ressources différentes du jeu : nourriture, or et bois.
 - Une production différente pour chacune des trois ressources
- De plus, chaque joueur disposera d'une capitale, l'unique premier territoire lui appartenant (la capitale dispose des mêmes caractéristiques que les autres territoires). La génération évitera que deux capitales soient directement voisines au début du jeu. Le reste des territoires sera neutre. Lors du lancement d'une partie, le joueur aura la possibilité de générer la map avec 3 modes différents :
- **Mode normal** : Il peut choisir 3 paramètres, et le logiciel se charge du reste :
 - Le nombre de joueurs.
 - Le taux de production durant tout le jeu : faible, moyenne, importante, etc.
 - L'emplacement des capitales : Soit il les pose lui même, soit elles le sont aléatoirement.
 - **Mode aléatoire** : Tous les paramètres du jeu sont fixés de manière totalement aléatoire, sans aucune contrainte : le nombre de joueurs, l'ordre des tours de chaque joueur, l'emplacement des capitales, la réserve et la production de chaque territoire, etc.
 - **Mode éditeur** : Ouverture d'un mode édition, permettant la modification des tous les paramètres de la partie : nombre de joueurs (un nombre maximum sera tout de même fixé), emplacement des capitales, réserve et production de chaque territoire, etc.
-
- **Moteur de jeu** : Lorsqu'une partie démarre, chaque joueur a donc une capitale. Le jeu se déroule au tour par tour, sur la même machine. Suivant le choix de création de la carte, un ordre de jeu entre les joueurs est défini. Durant le tour d'un joueur, celui-ci peut réaliser différentes actions, réparties en 2 phases :
 - **Phase de Gestion** : Le joueur peut :
 - Créer un bâtiment (unique) sur chacun de ses territoires. Il existe 3 types de bâtiments, chacun ayant des caractéristiques et un temps de construction différents :
 - Le Bâtiment de production : celui-ci augmente la production de toutes les ressources du territoire en question de X % (X reste à définir pour des questions d'équilibrage).
 - La Caserne : ce bâtiment sert à la production de soldats. **Plusieurs types de bâtiments militaires, associés à différents groupes de soldats : Caserne, Ecurie, etc.**
 - Le Bastion : ce bâtiment octroie un bonus de défense aux troupes présentes sur le territoire en question.
 - N.B : Les capitales sont considérées comme possédant chacun des 3 bâtiments. Elles ont donc une production accélérée et une défense augmentée par rapport aux autres territoires. Elles permettent aussi la production de soldats.
 - Produire des soldats : Sur chaque territoire possédant une caserne, le joueur peut lancer la production d'un bataillon de soldats. Chaque soldat dispose d'une valeur d'attaque et d'une valeur de défense. Il existe plusieurs types d'unités différents (ex : guerrier, chevalier, roi, etc). Par conséquent, chaque unité dispose d'un temps de construction propre, d'un nombre de soldats par bataillon propre, et d'une valeur d'attaque et de défense propres. Toutes ces informations ne sont pas encore fixées pour des raisons d'équilibrage.
 - Déplacer ses soldats : Chaque bataillon peut se déplacer d'un territoire par tour. Les déplacements sur les territoires alliés (les territoires alliés comprennent ceux du joueur ET ceux de ses alliés) sont sans conséquences. Les déplacements sur un territoire ennemi ou neutre vide entraînent la capture immédiate de celui-ci.

Les déplacements sur un territoire occupé par des unités ennemies entraîneront un combat lors de la phase en question. Le vainqueur capturera le territoire.

- Gérer ses alliances : Le joueur peut demander à n'importe quel autre la création d'un pacte. Dès lors, le jeu demande au potentiel allié sa réponse, directement pendant le tour du premier (ceci permet de n'avantager personne lors de la phase de gestion). A l'inverse, il peut choisir de rompre un pacte déjà existant. Dans ce cas, l'autre joueur concerné n'est pas informé jusqu'à la phase de combat et devra anticiper ou deviner en fonction des déplacements du traître. Toutes les informations de relations entre les différents pays et leurs frontières seront traitées à l'aide de graphes.
 - **Phase de Combat** : La deuxième phase consiste à la réalisation des combats. Celle-ci se déroule en même temps pour tous les joueurs. Pour commencer, les ruptures d'alliance sont annoncées. Le programme analyse la carte, et pour chaque territoire où au moins deux unités non alliées sont présentes simultanément, un combat a lieu. Ceci comprend le cas où deux unités alliées se trouvent sur le même territoire, mais l'un des deux joueurs a rompu le pacte pendant sa phase de gestion. Le déroulement d'un combat est divisé en plusieurs étapes :
 - Calcul des valeurs de combat des joueurs : pour chaque joueur différent présent dans le combat, une valeur de combat est calculée. Celle-ci est obtenue en multipliant le nombre de soldats de chaque type par la valeur d'attaque de ce type. Dans le cas où le territoire où se déroule le combat appartient à un joueur, ce sont les valeurs de défense de ses soldats qui sont prises en compte. **Intégration d'unités maritimes, combats navales et amélioration du gameplay.**
 - Une fois les valeurs de combat calculées, le jeu affiche le pourcentage de chances de chaque joueur de remporter le combat. Dans le cas où deux joueurs sont alliés, on additionne leurs valeurs.
 - Enfin, l'issue du combat est calculée à partir des chances de chacun des joueurs de gagner. Les unités de tous les perdants sont détruites, le gagnant capture le territoire. Dans le cas où les attaquants sont des alliés, celui qui avait la plus grande valeur de combat obtient le territoire, l'autre obtient X % (X reste à définir pour des questions d'équilibrage) des productions en ressources du territoire.Une fois tous les combats effectués, le tour suivant peut commencer avec la phase de gestion du premier joueur.
- Pendant une partie, le joueur doit avoir accès à un menu lui permettant de sauvegarder sa partie, d'en charger une autre ou enfin de quitter.

- **Moteur Graphique** : Notre logiciel sera capable de gérer un moteur graphique. Celui-ci aura pour but d'afficher à l'écran toutes les informations nécessaires à l'utilisateur pour jouer. Cela comprend les options suivantes :
 - Affichage des menus, avec boutons.
 - Affichage de la carte vue de dessus, des différents territoires avec surlignage des frontières, ainsi que des troupes (de même pour le mode édition). Chaque joueur sera représenté d'une couleur différente.
 - Permettre à l'utilisateur de sélectionner ses territoires et ses troupes afin d'y effectuer les actions de son choix. Affichage d'un menu diplomatique afin d'y ajouter ou de rompre des alliances.
 - Lors de la phase de combat, affichage des informations et des résultats de chaque combat.

- Ambiance sonore : **Ajout de musiques d’ambiances différentes selon les niveaux.**
- **IA** : Le jeu devra permettre, lors de la sélection du nombre de joueurs, de choisir de jouer contre l’ordinateur. Il proposera alors plusieurs niveaux de difficultés correspondants à des IAs de plus en plus fortes, ayant des stratégies de plus en plus élaborées.

2.3 Livraison attendues

Programmes

Nous rendrons la totalité du code source de notre programme, ainsi qu’un exécutable du résultat final.

Documents à remettre

Nous rendrons également les documents suivants :

- Une notice d’utilisation de notre logiciel.
- La Javadoc de notre code source.
- Un rapport sur la réalisation du projet .

Autres supports

Nous rendrons aussi le jeu dans une version exécutable et jouable. De plus, nous réaliserons une soutenance de projet (avec diaporama) le Mardi 26 ou le Mercredi 27 Mai 2015, afin de présenter le résultat obtenu.

Réalisation

3.1 Les données

Dans un projet complexe comme celui de Conquêtes, le nombre de données à traiter est considérable c'est pourquoi nous avons du commencer le projet par la schématisation des données puis leur réalisation. Nous comptons au total, 28 classes de données répartis autour de plusieurs catégories :

- Le joueur
- Bâtiments
- Territoires
- Troupes

3.1.1 Les joueurs

Pour les joueurs nous avons besoin de différentes données :

- Son stock de ressources
Par simplification nous avons implémenté un seul type de ressource. Mais la gestion de plusieurs ressource est déjà effective (utilisation d'un HashMap).
- Alliés
Les alliés sont représentés sous forme de graphes. Un joueur peut-être alliés à différents joueurs sans que ceux ci soit forcément alliés.
- Informations pratiques
Le joueur possède aussi un nom et un numéro afin de pouvoir rapidement le reconnaître.

3.1.2 Les batiments

Nous avons besoin de plusieurs type de bâtiment ayant chacun une fonctionnalité (Militaire, Production de ressources, Défenses) ainsi que d'un bâtiment général pour la capitale qui implémente toutes les fonctionnalités. Nous avons donc une classe mère qui nous permettra de gérer chaque bâtiment de la même manière quel que soit sa spécification. Ensuite chacune de ses classes filles implémente une ou plusieurs interface correspondant à leur spécification. La capitale implémente donc les trois interfaces. Chaque bâtiment possède :

- Un nombre de tours qu'il faudra au bâtiment pour être opérationnel après que le joueur ai lancé sa construction
- Un coût en ressource pour pouvoir construire le bâtiment
- Un ou plusieurs attributs spécifique à sa ou ses spécifications
Chaque spécifications rajoute un ou plusieurs attributs au batiment. Des unités en construction pour les batiments militaires, des pourcentages d'augmentation de ressources pour les batiments de ressources ou des statistiques de défenses améliorées pour les batiments de défenses

3.1.3 Territoires

Les territoires sont les données qui composent notre map. Nous avons choisi de représenter cette map sous forme de graphe. C'est à dire que nous ne voyons pas la carte comme un tableau de territoire mais comme un ensemble de territoires voisins.

Nous avons défini aussi plusieurs types de territoires (Plaine, Montagne etc...) gérés par une classe mère comme les batiments.

Chaque territoire peut être représenté comme un hexagone (il a six voisins).

Un territoire peut-être définie comme non traversable ou non.

Un territoire traversable peut-être définie comme une capitale. Chaque territoire a :

- Un possesseur (Vaut null si le territoire est neutre)
- Un bâtiment de n'importe quel type (sauf la bâtiment des capitales ne pouvant être que sur une capitale). Ce bâtiment est soit construit soit en construction soit non construit.
- Des troupes d'une même alliance (celle du possesseur)
- Six territoires voisins
- Un stock et une production de ressources

3.1.4 Troupes

Nous avons réalisé plusieurs types de troupes afin de varier les techniques d'assauts et de défenses. Chaque troupes peut-être composée de plusieurs soldats d'un même type.

Chaque troupe a

- Un possesseur
- Une valeur d'attaque (en fonction du nombre/type de soldat)
- Une valeur de défense (en fonction du nombre/type de soldat)
- Un nombre de tours pour être formé (en fonction du type de soldat)
- Le coût nécessaire en ressource (en fonction du type de soldat)

3.2 Le moteur

Le moteur est la partie calculatoire de notre programme. C'est lui qui va construire la partie et la faire avancer en fonction de ce que lui enverra l'interface graphique. Nous avons répartis la conception du moteur selon différentes catégories :

- La génération et le controle de la map
- La gestion des différentes phases de jeux
- La gestion des joueurs et leurs actions possibles
- Enfin la boucle de jeu qui initialise et coordonne tout ces éléments

3.2.1 La Map

Cette partie va permettre de générer le graphe de territoire en fonctions de plusieurs paramètres. Ce graphe est toujours généré de la même manière, seuls le placement des types de territoires sera fait en fonction des différents types de maps.

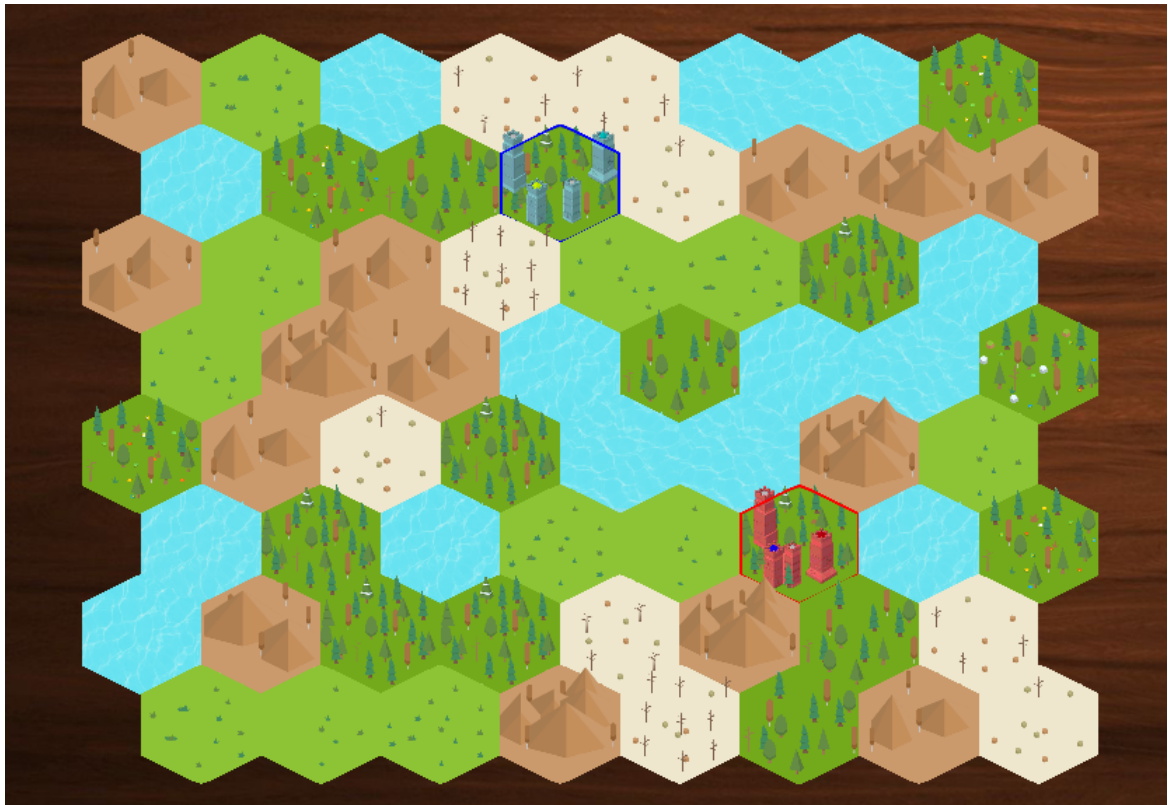
Les map possèdent un itérateur propre pour faciliter leur parcours.

Map aléatoire

La map aléatoire génère des territoires de manières complètement aléatoires. Elle prend deux paramètres :

- Le nombre de ligne de la map
- Le nombre de colonne de la map

Représentation d'une map aléatoire :

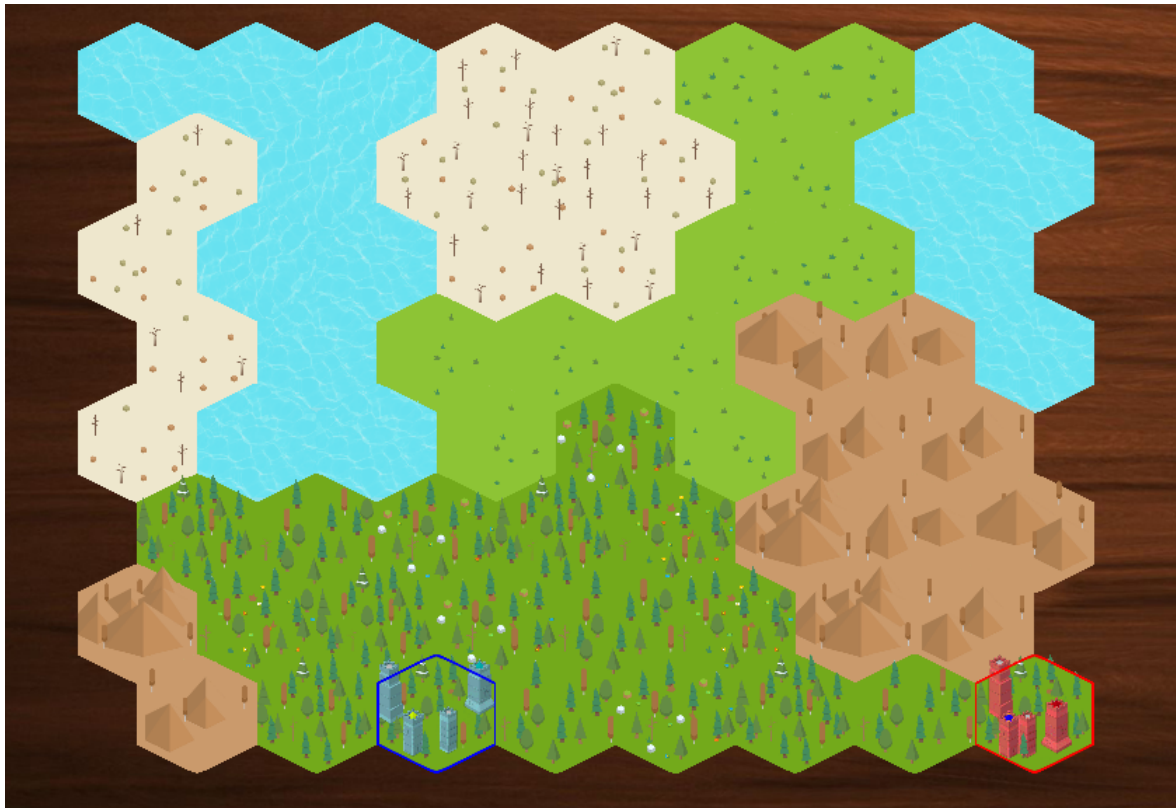


Map procédurale

La map procédurale va générer les différents types de territoires sous forme de biomes. C'est à dire que les territoires ne seront plus placés sans ordre précis, mais les territoires de mêmes types vont avoir tendance à se rassembler. Le placement des biomes est lui aléatoire.

Elle prend quatre paramètres :

- Le nombre de ligne de la map
- Le nombre de colonne de la map
- La taille minimum d'un biome
- La taille maximum d'un biome



Map générée par le joueur

Voir la partie éditeur de la section "graphique"

3.2.2 Les phases de jeux

Le système de jeu se divisent en plusieurs tours. Chaque tour est divisé en deux phases

La phase de gestion

C'est la phase ou les joueurs vont pouvoir faire leurs différentes actions. Il y a une phase de gestion par joueur par tour.

La phase de combat

Elle commence quand toutes les phases de gestions sont terminées.

On va regarder tous les territoires en conflits, c'est à dire ou des unités d'alliances différentes sont présentes.

On fait le rapport des forces et on détermine le vainqueur (celui qui a le plus de troupes). En cas de forces égalitaires, le vainqueur est tiré au hasard.

3.2.3 Les joueurs

On peut séparé deux types de joueurs, les joueurs humains et les IA. Ils seront traités de la même façon dans le jeu, seul la manière dont ils font leurs actions changent.

Les joueurs humains

Ils font leurs actions à partir de l'interface graphique pendant les phases de gestion.

Les IA

Leurs actions est calculés par le moteur pendant leurs phases de gestions. On peut définir deux IA de niveaux différentes :

- L'IA défensive :
C'est une IA basique. Elle va juste chercher à établir des défenses autour de sa position pour empêcher l'ennemi de l'atteindre et s'expandre ainsi.
- L'IA agressive :
Cette IA a un comportement qui s'adapte en fonction des actions de(s) adversaire(s). Pour cela nous avons définis des états dans lequel l'IA peut se trouver. Chaque état mène vers un ou plusieurs états en un nombre de tour défini jusqu'à arriver à l'état final recherché (la destruction de la capitale ennemi).
L'IA va donc appliquer l'algorithme de dijkstra qui est un algorithme très utilisé en pathfinding. Elle va donc pouvoir décider seule des meilleurs actions à faire.

Les actions possibles

Les joueurs peuvent faire plusieurs actions par tour de jeux. En cas d'action impossible, on renvoie une erreur correspondant à l'action.

Les actions possibles sont :

- Créer/Détruire/Accepter/Rejeter une alliance.
- Créer un bâtiment sur un territoire vide.
- Créer une troupe dans un bâtiment militaire
- Déplacer une troupe
- Finir son tour

3.2.4 La boucle de jeu

La boucle de jeu s'exécute de la même façon à chaque tour. Elle va mettre en relation tout les éléments ci-dessus :

- On commence par parcourir la map pour produire les ressources de chaques territoires et les donner à leurs possesseurs, pour faire avancer les unités/bâtiments en constructions etc...
- Ensuite on lance la phase de gestion de tout les joueurs et on attend que chaque joueur ai fini son tour
- Quand tout le monde a fini son tour, on lance la phase de combat.
- Enfin, on regarde si c'est la fin du jeu. Si ce n'est pas le cas on reboucle, sinon on stop et on annonce le vainqueur.

3.2.5 Fin du jeu

La fin du jeu est déterminée quand il ne reste plus qu'un seul joueur en jeu (Quand il possède donc toutes les capitales). Ce joueur est déclaré vainqueur.

3.3 L'interface graphique

Le graphique est un des pôles les plus important dans la réalisation d'un jeu vidéo, dans notre jeux conquêtes, il a fallut dessiner de nombreux éléments graphiques ainsi que de nombreuses classes graphiques.

3.3.1 Les éléments graphiques

Au niveau des éléments graphiques, nous pouvons lister :

- Boutons
- Police d'écritures
- Images de fond
- Textures des territoires
- Menu
- Skin des soldats
- Skins des bâtiments
- Fenêtre de dialogue

La police d'écriture du jeu, est la fameuse Korinna, en libre droit d'utilisation, elle fut désigné en 1974, celle-ci est encore utilisé par Disney, Capcom ainsi que de nombreux journaux. Une partie des éléments graphiques (arbres, boutons, etc) ont été réalisé par Kenney, un game designer hollandais qui publie des pack de sprites pour des développeurs de jeux vidéos, il va de soi que les packs que nous avons utilisés sont libre de droits. Pour la réalisation et l'assemblage des éléments graphique, nous avons utilisé exclusivement GIMP sous Windows, ce qui nous à permis d'adapter les divers éléments graphiques à nos besoin.

3.3.2 Les classes graphiques

Au niveau des classes générant un rendu graphique, nous les avons mis dans un dossier screen, et elles commencent toutes par Gui.

Voici la listes avec leur descriptions respectives :

- GuiEditor GuiEditor nous permet d'accéder au mode éditeur. Le mode éditeur nous permet de créer notre propre map avec les types de territoires, capitales etc.
- GuiGame GuiGame est la classe la plus importante de notre projet, en effet, celle-ci gère l'affichage complet du jeux.
- GuiHelp GuiHelp est une page d'aide pour l'utilisateur accessible via le bouton aide dans GuiGame.
- GuiLoading GuiLoading nous affiche la page de chargement de parties.
- GuiMenu GuiMenu gère l'affichage et les interactions de l'utilisateur avec le menu principal du jeu.
- GuiResult GuiResult s'affiche en fin de partie, celle-ci annonce la fin de la partie et propose au joueur de rejouer.
- GuiSettings GuiSettings comme son nom l'indique permet d'accéder aux menu de réglages du jeux.
- GuiStream GuiStream est l'interface qui permettant de rediriger les messages d'erreurs vers la fenêtre graphique.

3.3.3 Les extensions

Nous avons dans notre projet réussi à réaliser deux extensions du cahier des charges :

Mise en place des langues

Nous avons mise en place un système de fichier de localisations. Il nous est donc possible de changer de langue n'importe quand pendant le jeu. Pour l'instant les langues gérées sont l'anglais et le français mais il nous est très simple d'en rajouter.

Mise en place du réseau

Nous avons aussi mis en place une surcouche réseau "Serveur/Client" dans notre jeu. Le serveur communique avec le moteur et le client avec l'interface graphique. Il est donc possible de jouer jusqu'à 8 joueurs en même temps sur 8 postes différents. Cela accélère la vitesse des parties

Manuel Utilisateur

Age of Swag : Conquest CMI est un jeu amateur libre et gratuit conçu par des étudiants à l'Université de Cergy-Pontoise, développé en Java à l'aide de la librairie graphique LibGdx.

Avant de vous lancer dans l'aventure de Age of Swag, nous vous recommandons de lire ce mode d'emploi pour savoir comment jouer mais aussi connaître les conseils indispensables.

Age of Swag vous propose un support technique gratuit et hors pair, disponible 24/24h 7/7j aux adresses mails suivantes :

- bastien-12@hotmail.fr
- lucas.nicosia@yahoo.fr
- aryus96@hotmail.fr

Attention, cette assistance s'adresse seulement aux bénéficiaires de notre projet Age Of Swag.

4.1 Configuration requise

Age of Swag : Conquest CMI fonctionne sur divers systèmes tels que Windows, Linux ou Mac OS X. Etant libre, elle peut être facilement portée sur d'autres systèmes (contactez-nous). Nous avons fait de notre mieux pour optimiser la programmation, mais le jeu utilise tout de même un minimum de ressources, donc il risque de ne pas être jouable correctement sur des ordinateurs relativement peu puissants. Les performances diffèrent selon les machines et il est difficile d'établir précisément une configuration minimale. Nous vous conseillons quoi qu'il en soit de fermer toutes les autres applications actives quand vous jouez à Age Of Swag : Conquest CMI.

Age Of Swag à été testé sur divers configuration, et à été jugé parfaitement fonctionnel par notre équipe sur ces configurations :

- i7-3630QM (2.4Ghz)- Nvidia GT740M - 16Go de RAM
- i5-3230M (2.6GHz) - Nvidia NVS 5200M - 8Go de RAM
- i5 650 (3.20Ghz)- AMD Radeon HD 5700 Series - 4Go de RAM

4.2 Configuration le jeu

Installer Age of Swag : Conquest CMI

Pour installer et démarrer Age Of Swag, rien de plus simple, il suffit d'installer la dernière Java Runtime Environment (Pour plus d'informations sur l'obtention de la dernière Java Runtime Environment, connectez-vous à l'adresse suivante <http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>).

Si vous disposez déjà de Java Runtime Environment ou que vous venez de l'installer, il vous suffit juste de double cliquer sur l'exécutable .jar du projet.

Désinstaller Age of Swag : Conquest CMI

Il vous suffit de supprimer l'exécutable .jar .

4.3 Jouer à Age of Swag : Conquest CMI

Modes de jeu

1. Normal Dans ce mode, la carte est générée aléatoirement mais vous pouvez choisir le nombre de joueurs humain, le nombre de joueurs géré par l'ordinateur et si vous voulez qu'ils soient agressifs ou défensifs.
2. Aléatoire.
Dans ce mode, la carte est générée aléatoirement, vous pouvez choisir le nombre de joueurs humain, ainsi que ceux gérés par l'ordinateur mais vous ne pouvez pas choisir leurs stratégies de combats.

Local

1. Démarrez le jeu.
2. Appuyez sur le 1er bouton, New Game/Nouvelle partie.
3. Appuyez sur Local.
4. Dès lors, choisissez votre mode de jeux.

En ligne

1. Démarrez le jeu.
2. Appuyez sur le 1er bouton, New Game/Nouvelle partie.
3. Appuyez sur Online.
4. Choisissez si vous créez la partie ou rejoignez une partie.

Charger une partie

1. Démarrez le jeu.
2. Appuyez sur le 2nd bouton, Load game/Charger une partie.
3. Sélectionnez votre partie précédemment enregistré.

Editeur

1. Démarrez le jeu.
2. Appuyez sur le 3ème bouton, Load game/Charger une partie.
3. Créer votre carte avec le nombre de joueurs voulu.

4.4 Autres informations

Remporter la partie

Pour gagner la partie, vous devrez conquérir l'intégralité des capitales ennemis à l'aide de vos troupes.

Former une unité

Afin de former une unité vous devez sélectionner votre capitale ou bien un territoire possédant une caserne. Vérifiez que vous avez bien les ressources nécessaire à sa formation, puis patienter le nombre de tours exigé pour l'unité. Attention, les casernes et capitales ne peuvent former que une unité à la fois.

Créer un bâtiment

Afin de créer un bâtiment, vous devez sélectionner un territoire autre que votre capital, vous appartenant et appuyer sur le bouton de création du bâtiment souhaité, attention, vérifier bien d'avoir les ressources nécessaires, puis vous devrez patienté un certains nombre de tour.

Taille de la carte La taille de la carte dans le mode éditeur est fixe. La taille dans les autres modes dépends du nombre de joueurs que vous mettez dans la partie.

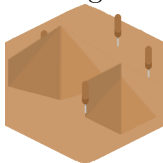
Types de territoires

Il existe 5 types de territoires :

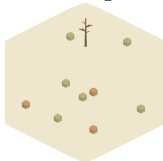
— Forestier



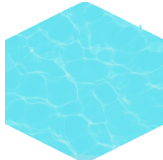
— Montagneux



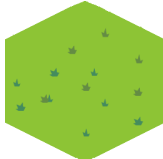
— Désertique



— Aquatique



— Plaine



Les territoires Désertique et aquatiques sont les seuls qui vous seront infranchissable.

Types de troupes

Age of Swag dispose actuellement de 5 types de troupes :



— Archer

L'archer sera intéressant par son faible coût, temps de production et son attaque.

ATK : 15

DEF : 10

2 tours de constructions et un coût de 20 ressources.



— Magicien

Le magicien possède la plus grande puissance d'attaque mais son coût et temps de production font de lui une arme stratégique.

ATK : 25

DEF : 5

3 tours de constructions et un coût de 30 ressources.



— Épéiste

L'épéiste tient sa force dans sa neutralité, l'équilibre de ces caractéristiques font de lui un soldat de choix.

ATK : 15

DEF : 15

2 tours de constructions et un coût de 25 ressources.



— Lancier

Le Lancier est une troupe de choix dans le cas d'une défense de territoire.

ATK : 10

DEF : 20

2 tours de constructions et un coût de 25 ressources.



— Barbare

Le barbare est le soldat le plus basique du jeu. Il est néanmoins intéressant par sa rapidité de production et son faible coût.

ATK : 10

DEF : 10

1 tours de constructions et un coût de 20 ressources.

Types de bâtiments

Age Of Swag : Conquest CMI dispose également de 3 types de bâtiments.

— l'Usine



L'usine va permettre d'augmenter la production des ressources sur le territoire sur lesquels vous l'avez construits, attention toutefois, chaque territoires possèdent un stock de ressources, il est judicieux de placer des usines à certains endroits.

— Caserne



La caserne vous permettra de créer des troupes hors de la capitale mais aussi de produire d'avantage de troupes par tours.

— Défense



Le bâtiment de défense va permettre à vos troupes de gagner en défense et donc de mieux résister aux assauts des troupes ennemis.

Interface



Modifier les options de jeu

Les options de Age Of Swag sont relativement basique, vous avez la possibilité de changer la langue du jeu mais aussi de désactiver le son (seulement à partir du menu principal).

Pour afficher l'aide

Si vous souhaitez avoir de l'aide et des informations lors du jeux, vous pouvez cliquer sur le bouton vert aide/help qui vous donneras un panel d'informations sur le fonctionnement du jeu.

Déroulement du projet

5.1 La conception

La conception fut l'étape clé pour notre projet de Génie Logiciel, en effet, impossible de se lancer dans un projet d'une telle envergure sans une étape de conception. Il a fallu tout d'abord concevoir les données que nous allions utiliser, pour cela, nous avons réalisé un diagramme de classe avec l'intégralité des données utilisable et qui vont être traitées par notre programme. Nous avons également réfléchi à l'interface graphique et le style de jeu que nous souhaitions afin de ne pas devoir changer plus tard dans le projet. Mais aussi la manière et les méthodes que nous allions utiliser pour traiter nos données.

5.2 La concrétisation

Une fois l'étape de conception finis, il nous fut facile de le concrétiser car le projet comporte de nombreuses parties que nous nous sommes partager afin d'avoir un travail et une organisation efficiente. Nous avons évidemment commencer par réaliser l'intégralité des données. Par la suite, nous avons commencé à les traiter, dès lors que cela prenais de la forme, il fut crucial d'entamer la création d'une interface graphique afin de pouvoir avoir un meilleur rendu et affichage de notre travail. Par la suite, plus notre projet prenais forme, plus la nécessité d'avoir un menu fut important

5.3 Finalisations et améliorations

Les finalisations d'un jeu vidéos sont innombrables, les jeux d'aujourd'hui, même après leur sortie, continue de

Le projet conquête est débordant d'imagination, en effet, de part son côté jeu vidéo et de notre expérience dans ce domaine, nos idées d'améliorations furent nombreuses. Toutefois par des soucis de temps et non pas de compétences, nous avons préféré nous concentrer sur les plus importantes que je vais vous présenter. Nous avons implémenté le jeux en réseau, vous pouvez dès lors jouer avec plusieurs ordinateurs sur la même partie. Nous avons également rajouter une musique lorsque vous lancez Age Of Swag, en effet, l'absence de musique rendait le jeu moins attrayant. Afin de rendre le gameplay du jeu plus attrayant et agréable, nous avons implémenté différents types de troupes et inclut la possibilité pour le joueur de créer des bâtiments. L'interface graphique à été également peaufiné tout au long du projet afin d'être le plus attractif et intuitif pour l'utilisateur, au delà d'une interface graphique simple, il y a toute une recherche et réflexion derrière la réalisation de l'interface, des choix des couleurs, images, etc.

Conclusion

Pour conclure, à travers le génie logiciel et notre projet conquête, nous avons gagné en professionnalisation en effet, nous avons appris à :

- Réaliser un projet complexe en groupe
 - Pas loin de 8000 lignes de code.
 - 144 classes.
 - 230 images.
 - Des centaines d’heures de travail.
- approfondir les techniques de conception et de programmation
- maîtriser les différents outils de réalisation de logiciels :
 - utilisation avancée d’Eclipse
 - SVN
 - Junit
 - Log4j

Au delà de l’aspect éducatif que nous à apporté le génie logiciel, cela nous a permis de former un groupe de travail productif dans lesquels chacun de nous à pris plaisir à travailler et à s’épanouir au fur et à mesure du projet.

Remerciements

Nous tenons à remercier :

- Notre enseignant tuteur, Monsieur Tianxiao LIU qui nous a été d'une aide précieuse et qui nous a consacré du temps.
- Madame Tuyêt Trâm DANG NGOC, qui a lancé le module de Génie Logiciel.
- La promotion L2 CMI SIC et plus généralement tout le groupe A pour cette chaleureuse ambiance et cet esprit d'entraide.