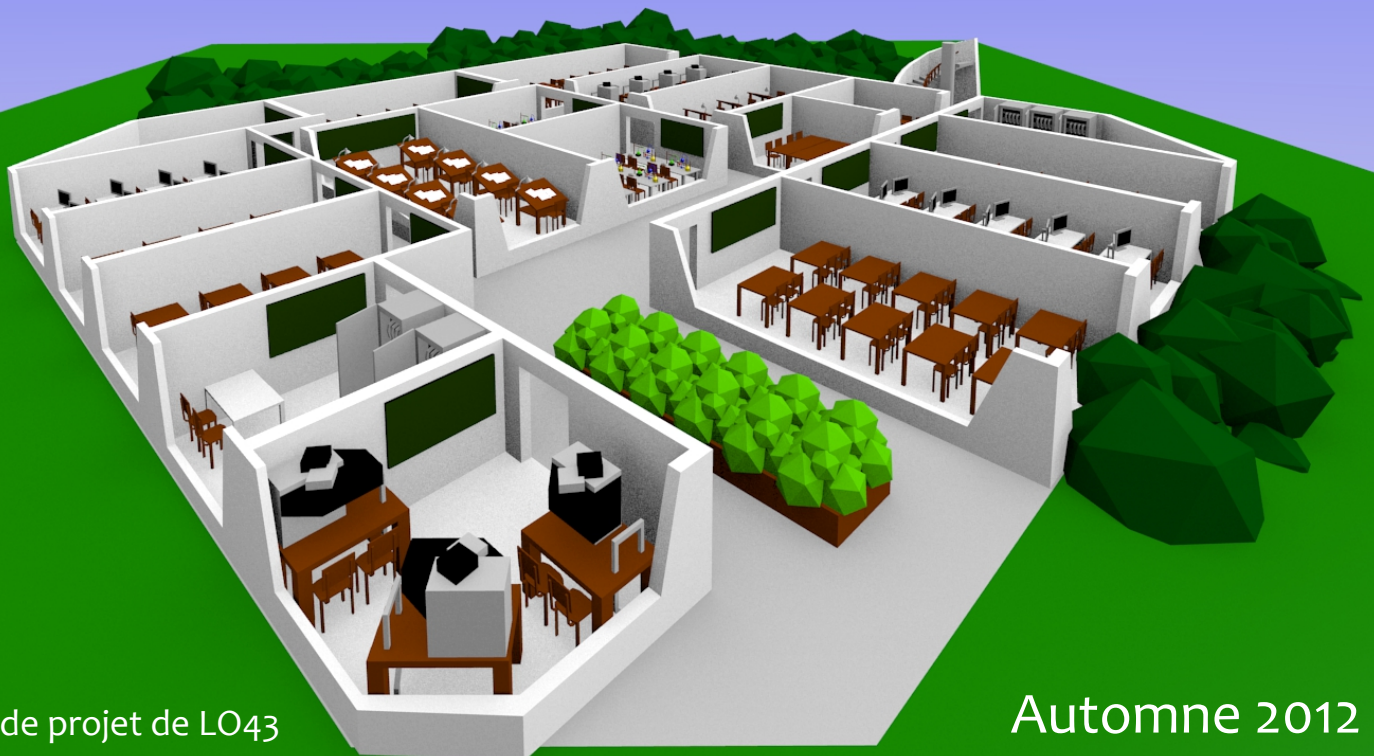


UTBM Total War

Commanditaire : EL ZAHER Madeleine

Etudiants :
LEVEUGLE Antoine
PHETNOUVONG Damien
SEGUIN Baptiste



Remerciements

Nous tenons à remercier dans un premier temps, toute l'équipe pédagogique de Université de technologie Belfort-Montbéliard, et plus particulièrement Monsieur Franck GECHTER, responsable de l'unité de valeur LO43 Base fondamentale de la programmation orientée objet, pour avoir assuré la partie théorique de celle-ci.

Il convient de remercier Monsieur Pascal BROCHET, directeur de l'Université de technologie Belfort-Montbéliard, pour avoir mis à notre disposition tous les moyens nécessaires au bon déroulement de notre projet de fin d'étude.

Nos remerciements vont également à tous ceux et celles qui nous ont aidé de près ou de loin à la réalisation du projet.

Table des matières

I.Introduction.....	3
II.Cahier des charges.....	4
III.Adaptation des règles du jeu UTBM TOTAL WAR:.....	6
IV.Avancement et décisions.....	11
1Étapes du projet.....	11
2Réalisation du jeu.....	11
Partie graphique.....	11
Répartition des tâches.....	12
3Technologies utilisées.....	13
Swing.....	13
Git.....	13
Architecture MVC.....	13
V.Résultats.....	15
1Explications.....	15
2Structure.....	24
3Fonctionnement.....	25
4Diagramme MVC.....	26
5Contrôleur.....	26
6Modèle.....	27
7Vue.....	28
8Diagramme de cas d'utilisation.....	28
VI.Améliorations possibles.....	30
VII.CONCLUSION.....	31

I. Introduction

Le présent projet concerne la modélisation d'une adaptation numérique d'un jeu de société nommé Small World.

Notre adaptation du jeu se fait dans le contexte d'une unité de valeur de l'UTBM : LO43. Afin, de rendre le jeu plus intéressant, il a été demandé de changer le monde et la charte graphique du jeu en ayant pour thème l'UTBM. Notre application s'appelle donc UTBM Total War.

Ce rapport va se dérouler de la manière suivante : pour commencer, un léger rappel sur le jeu et le cahier des charges. Ensuite, nous verrons comment s'est déroulé la réalisation du projet avec ses étapes clés, en terme des répartitions des tâches, du travail effectué, des problèmes rencontrés et des solutions apportées.

II. Cahier des charges

Sujet : Small UTBM World.

Ce projet s'inspire directement du jeu Small World, un jeu de plateau dans lequel des joueurs s'affrontent pour conquérir le plus de territoires sur une carte, accumulant ainsi un maximum de points de victoire à la fin de la partie. Le principal problème étant que ce monde est trop petit pour plusieurs peuples, ce qui très vite, assure aux joueurs de s'affronter un maximum.

L'objectif de ce projet est, par conséquent, de créer une adaptation UTBMienne d'un tel jeu de plateau, en restant fidèle au fonctionnement de Small World.

Le principe de Small World est le suivant, plusieurs joueurs s'affrontent dans un monde médiéval fantastique. La carte est divisée en toute sorte de petits territoires (montagnes, forêts, etc.). Chaque peuple doit sans cesse lutter pour sa survie. Chaque peuple étant associé à un pouvoir spécial, qui lui permet d'avoir un avantage s'il se trouve sur certains territoires. Le couple peuple/territoire est créé aléatoirement au début de chaque partie. Le jeu s'organise autour de pions peuple, qui permettent aux différents peuples de pouvoir conquérir de nouveaux territoires.

Les conquêtes de territoires vierges et les attaques de territoires occupés se font par l'engagement de pions peuple. Il n'y a pas de hasard dans cette manche, pour l'attaque d'un nouveau territoire, il faut engager au minimum deux pions peuple, et pour attaquer un territoire sur lequel un ennemi est déjà en place, il faut tout simplement engager un nombre de pions peuple plus important que le nombre d'occupants ennemis. A la fin du tour, le joueur marque autant de points de victoire qu'il a de territoires.

Une autre règle du jeu tout aussi importante est la possibilité pour chaque joueur de mettre, à tout moment du jeu, son peuple en déclin. En effet, dans le cas où le joueur n'a plus de pions peuples ou souhaite changer de peuple, il a la possibilité de le faire avec cette méthode. Ainsi, son peuple devient en déclin, et continue à lui rapporter quelques points pour chaque territoire qu'il occupe. Une fois que le joueur demande le déclin pour son peuple, il peut immédiatement choisir un nouveau peuple et continuer sa partie avec une nouvelle base plus solide.

Ce projet n'est pas une copie de ce jeu de plateau mais bien une adaptation UTBMienne, nous ne parlerons donc pas de peuples ni de pouvoirs, mais d'étudiants et de spécialités. Les territoires du monde fantastique de Small World ne sont plus des montagnes, plaines ou forêts mais des salles de cours associées à une spécialité. Le choix des noms de types d'étudiants, des salles, des spécialités sont laissés au bon soin de l'équipe de projet. L'équipe de projet possède un maximum de liberté quand à la réalisation de ce projet, la seule limite est l'imagination.

III. Adaptation des règles du jeu UTBM TOTAL WAR:

Pour réaliser un jeu personnalisé, nous avons dû adapter les règles. Pour rendre tout ce qui suit plus clair, une présentation des modifications matérielles s'impose :

- Les peuples sont remplacés par les filières de l'UTBM.
- Des spécialités remplacent les pouvoirs.
- Les salles remplacent les territoires à conquérir.
- Les atouts salles remplacent les éléments de terrain du plateau.

L'explication des règles adaptées :

1ère **PHASE** « Choix Étudiant/Spécialité » :

Le joueur choisit un couple Étudiant/Spécialité.

2ème **PHASE** « Conquête d'une salle » :

- **Première Conquête** : Le joueur doit rentrer sur plateau par une salle qui est située sur le bord du plateau.
- **Conquérir une salle** : Pour occuper une salle, le joueur doit déployer 2 pions Étudiant + 1 pion supplémentaire par « objet spécial » que contient la salle + 1 pion par Étudiant ennemi présent dans la salle. Les couloirs ne peuvent pas être conquis. Lorsque le joueur a pris possession d'une salle, il doit laisser les pions Étudiants qui lui ont été nécessaires à la conquête de cette salle.
- **Pertes** : Ses pions doivent rester en place jusqu'à la PHASE de REDEPLOIEMENT de ses troupes à la fin de son tour. En cas de perte dans la lutte pour la possession d'une salle, le perdant récupère dans ses mains les pions qui étaient placés dans la salle.
- **Conquêtes suivantes** : Le joueur peut conquérir des salles tant qu'il a assez de pions pour s'emparer des salles. Les nouvelles conquêtes doivent être adjacentes à celles déjà occupées par le joueur ou doit être sur l'un des bords du plateau.
- **Dernière conquête/renforts** : S'il ne possède plus assez de pions Étudiants, le joueur peut

lancer le dé pour effectuer une dernière conquête. On additionne la valeur du dé au pions Étudiant : si la somme mène à la conquête de la salle, on place le/les pions sur la salle, sinon on redéploie ce/ces pions dans une région déjà occupée.

FIN DE CONQUETE

3ème **PHASE** « Redéploiement » :

Il peut répartir à sa convenance ses pions dans les salles occupées. La condition à respecter est de toujours laisser obligatoirement au moins un pion dans chaque région occupée !

FIN TOUR DU JOUEUR

Un point victoire par salle occupée, des points supplémentaires peuvent être gagnés si la combinaison Étudiant/Spécialité le permet. Idem pour les peuples en déclin sauf que les spécialités ne sont plus prises en compte.

Les joueurs suivants effectuent les mêmes phases. Lorsque tout le monde a déjà joué une fois, le joueur qui a commencé la partie poursuit son tour ainsi :

Expansion du peuple actif

Le joueur a la possibilité de :

- Reprendre ses pions étudiants déployés sur plateau en prenant garde bien laisser UN PION Étudiant par salle occupée. Ces pions récupérés en main vont permettre de conquérir de nouvelles salles.
- Conquête :IDEM que la 2ème phase dans son PREMIER TOUR (sans prendre en compte la règle de première conquête)
- Abandonner une salle : le joueur peut décider d'abandonner une salle en retirant la totalité de ses pions dans la salle occupée. Si plus aucun pion Étudiant n'est présent sur le plateau, c'est à dire s'il abandonne toutes ses salles, le joueur devra appliquer la règle de « la première conquête » de la 2ème phase dans le PREMIER TOUR.

Passer en déclin

Si le joueur estime que ses Étudiants sont trop étendu et qu'il est difficile de continuer l'expansion ou défendre les salles occupées face aux ennemis. Il peut décider de passer ses Pions Étudiants en déclin et de choisir un autre type de pion Étudiants. Le joueur ne peut posséder qu'un seul type d'Étudiant en déclin. Lorsque le type d'Étudiant est en déclin, le joueur doit laisser obligatoirement un et un seul pion dans les salles occupées.

La partie se termine lorsque le nombre de tours définit au début de la partie est écoulé ou lorsque il ne reste plus aucun jeton de points victoires.

Ci-dessous sont présentés les bonus apportés par les spécialités et les atouts-salle :

Atouts Salles :

Toutes salles INFO occupées par les GI rapportent un jeton de victoire supplémentaire en fin de tour.

Toutes salles MACHINE occupées par les GMC rapportent un jeton de victoire supplémentaire en fin de tour.

Toutes salles ELEC occupées par les EE rapportent un jeton de victoire supplémentaire en fin de tour.

Toutes salles MANAGEMENT occupées par les IMAP rapportent un jeton de victoire supplémentaire en fin de tour.

Toutes salles DESSIN occupées par les EDIM rapportent un jeton de victoire supplémentaire en fin de tour.

Toutes salles COURS occupées par les TC rapportent un jeton de victoire supplémentaire en fin de tour.

Toutes salles BIBLIOTHEQUE occupées par les Doctorant rapportent un jeton de victoire supplémentaire en fin de tour.

Toutes salles AMPHI occupées par n'importe quel étudiant rapportent un jeton de victoire supplémentaire en fin de tour.

Etudiant : (7)

GI

EE

GMC

EDIM

IMAP

TC

Doctorant

Les anciennes Spécialités présentes dans le jeu qui avait été mises en place pour pouvoir déboguer : (10)

Geek : +3 si associé aux GI sinon 0

Mécano : +3 si associé aux GMC sinon 0

Elec : +3 si associé aux EE sinon 0

Gribouilleur : +3 si associé aux EDIM sinon 0

Management : +3 si associé aux IMAP sinon 0

Matheux

Langues

Intello : +3 si associé aux Doctorants sinon 0

CultureGénéral

PhysiqueChimie : +3 si associé aux TC sinon 0

Les nouvelles Spécialités qui ont été élaborés à la fin du projet mais qui n'ont pas pu être implémentées : (17)

Révision Nocturne:+1

Travail Régulier:+2

Au Talent : +1

Acharné : +1

Majoré : +2

Minoré : 0

Assiduité : +1

Matheux : +1

Langues : +1

CultureGénérale : +1

Fou d'informatique: +3 si associé aux GI sinon 0

Beau-parleur: +3 si associé aux IMAF sinon 0

Para-tonnerre : +3 si associé aux EE sinon 0

Machine: +3 si associé aux GMC sinon 0

Artiste: +3 si associé aux EDIM sinon 0

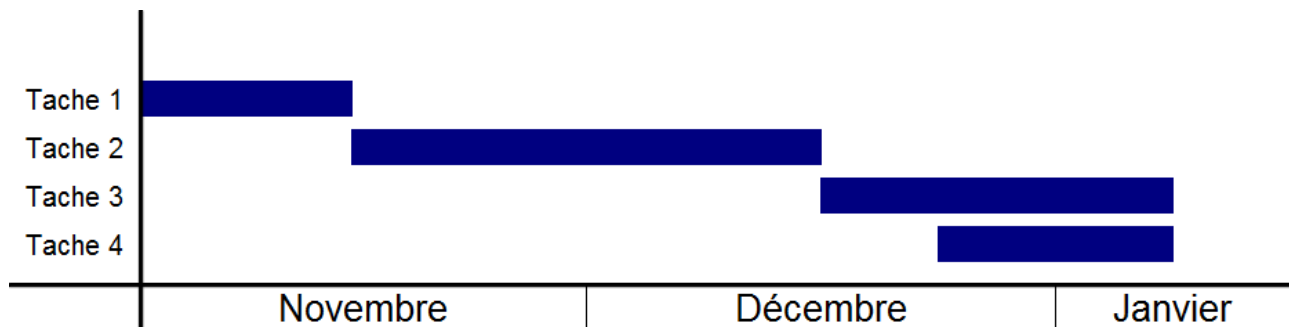
Bachelier Frais: +3 si associé aux TC sinon 0

Bûcheur : +3 si associé aux Doctorants sinon 0

IV. Avancement et décisions

1 Étapes du projet

Afin de mieux comprendre comment s'est articulé le travail, voici un diagramme de Gantt simplifié permettant de visualiser la chronologie du projet et de ses étapes clés :



Tache 1 : Étude de la demande du client afin de bien comprendre ce que ce dernier attendait de nous.

Tache 2 : Cette tâche avait pour but de construire les bases de l'application de façon théorique et schématique (UML) afin de pouvoir programmer rapidement et facilement le jeu par la suite.

Tache 2-3 : Pour les tâches 2 et 3, une répartition des tâches a dû être faite afin d'avancer rapidement et efficacement.

Tache 3 : Enfin, à la mi-décembre, le jeu a commencé à être programmé.

Tache 4 : Rédaction du rapport

2 Réalisation du jeu

Partie graphique

Avant de démarrer la construction de la partie graphique, il fallait réfléchir sur comment elle allait être et surtout comme elle devait être construite.

Au final, nous avons opté pour que l'interface graphique soit composée de deux principales

interfaces : le menu, qui sert à configurer le jeu et à initialiser une nouvelle partie, et la plate-forme, qui sert à afficher la partie en cours.

Le menu doit permettre de lancer une nouvelle partie en passant par différents sous-menus afin de configurer cette partie, tels que : choix du bâtiment, saisie des pseudos, nombre de tours, etc. C'est aussi dans le menu que quelques options seront disponibles, telles que : activer ou désactiver l'antialiasing ou le mode plein-écran.

La plate-forme doit, quant à elle, afficher toutes les informations nécessaires à l'utilisateur pour que la partie se déroule correctement. Elle doit donc afficher les informations relatives aux joueurs, les outils permettant d'interagir dans la partie (boutons, cases à cocher, etc) et la carte du jeu, c'est-à-dire le bâtiment.

Nous voulions aussi que la fenêtre du jeu puisse être redimensionnée. Il fallait donc faire en sorte que les éléments évoluent de façon cohérente lorsque la fenêtre change de taille.

Dans le menu et quelques parties de la plate-forme, nous avons décidé de ne pas utiliser de conteneur Swing. En effet, nous voulions pouvoir placer nos composants au pixel sans être « bridé » par les contraintes d'affichages (des fois très utiles) des conteneurs (« layout »).

Répartition des tâches

Comme nous formons une équipe de trois personnes, nous avons dû nous répartir les tâches afin d'éviter de faire le travail des autres et du coup, de perdre du temps.

Sur un projet comme celui-ci, il est difficile de bien se répartir les tâches en trois. En effet, chaque partie du projet est en lien avec les autres.

Les tâches se sont réparties de telle sorte :

- Antoine :
 - Interface graphique : le menu et zone de la plate-forme qui gère l'affichage des informations relative au joueur),
 - Contrôleur du menu et de la plate-forme.
- Baptiste :
 - Lecture et mise en mémoire du fichier correspondant à un bâtiment du jeu,
 - Création d'un petit éditeur de bâtiment,
 - Interface graphique : zone de la plate-forme affichant le bâtiment.
- Damien :
 - Écriture du modèle sans gérer tout ce qui relatif au bâtiment,
 - Établissement des règles du jeu.

3 Technologies utilisées

Swing

Pour l'interface graphique, nous avons utilisé Swing. C'est une bibliothèque graphique pour le langage de programmation Java. Swing offre la possibilité de créer des interfaces graphiques identiques quel que soit le système d'exploitation sous-jacent.

Git

Au départ, nous utilisions Dropbox afin de nous partager les fichiers et de travailler en collaboration. Cependant, Dropbox s'est retrouvé rapidement peu fonctionnel. Nous avons donc utilisé Git par la suite.

Git est un logiciel de gestion de versions. Il « agit sur une arborescence de fichiers afin de conserver toutes les versions des fichiers, ainsi que les différences entre les fichiers. Ce système permet par exemple de mutualiser un développement. Un groupe de développeurs autour d'un même développement se servira de l'outil pour stocker toute évolution du code source. Le système gère les mises à jour des sources pour chaque développeur, conserve une trace de chaque changement. Ceux-ci sont, en bonne utilisation, chaque fois accompagnés d'un commentaire. Le système travaille par fusion de copies locale et distante, et non par écrasement de la version distante par la version locale. Ainsi, deux développeurs travaillant de concert sur une même source, les changements du premier à soumettre son travail ne seront pas perdus lorsque le second, qui a donc travaillé sur une version non encore modifiée par le premier, renvoie ses modifications. » (source : wikipédia)

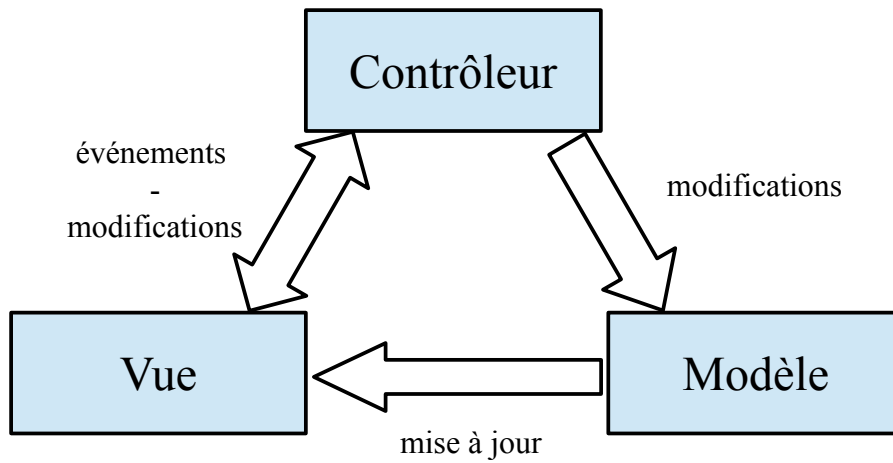
Architecture MVC

Cette architecture sépare en trois catégories les fonctions d'une application :

- le Modèle : ce qui contient les données, la mémoire de l'application ;
- la Vue : ce gère l'interface graphique pour l'utilisateur ;
- le Contrôleur : ce qui gère et contrôle les événements de l'interface (ex : clic sur un bouton)

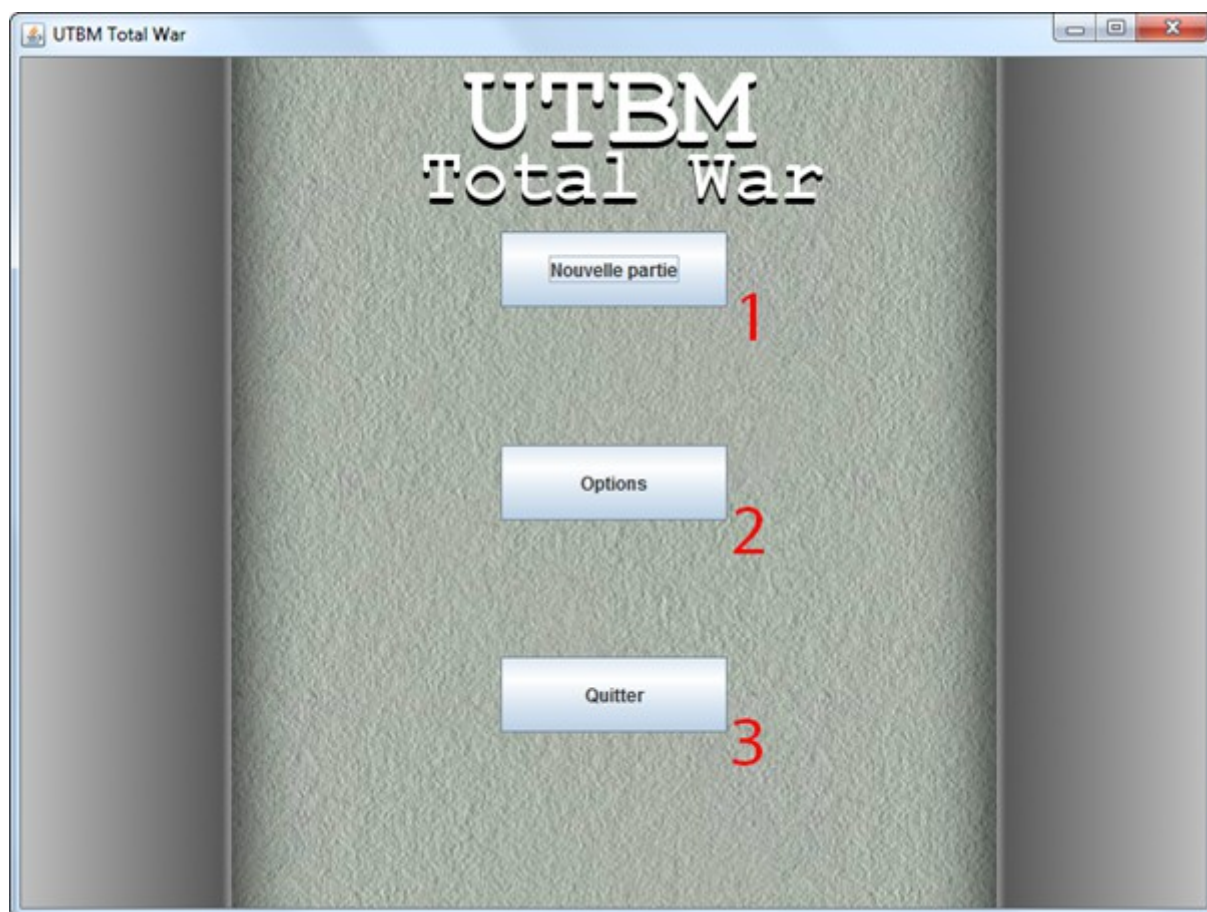
afin de synchroniser le modèle et la vue.

Dès le départ, nous avons voulu utiliser cette architecture car elle permet de construire une application propre avec des parties bien distinctes. De plus, la répartition des tâches s'en est retrouvée plus facile.

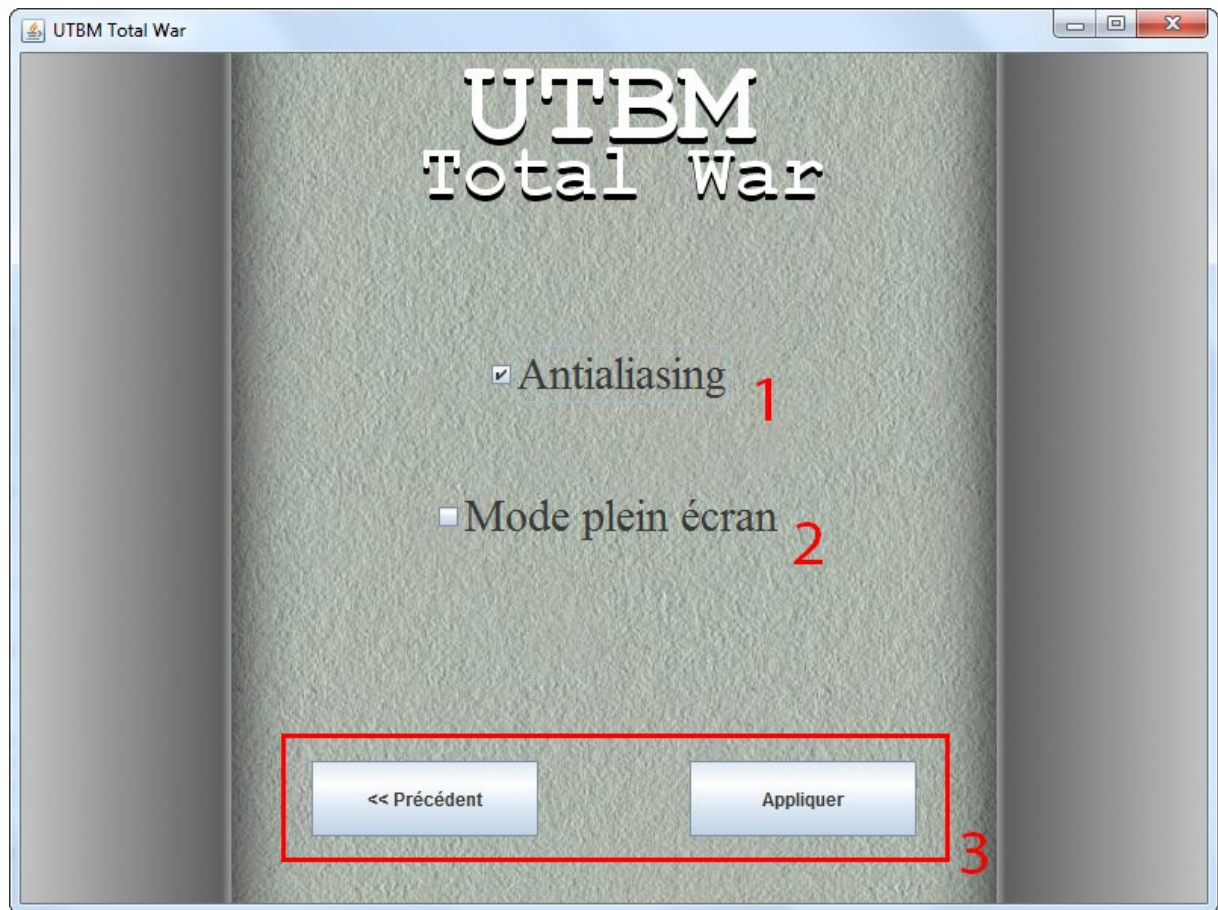


V. Résultats

1 Explications

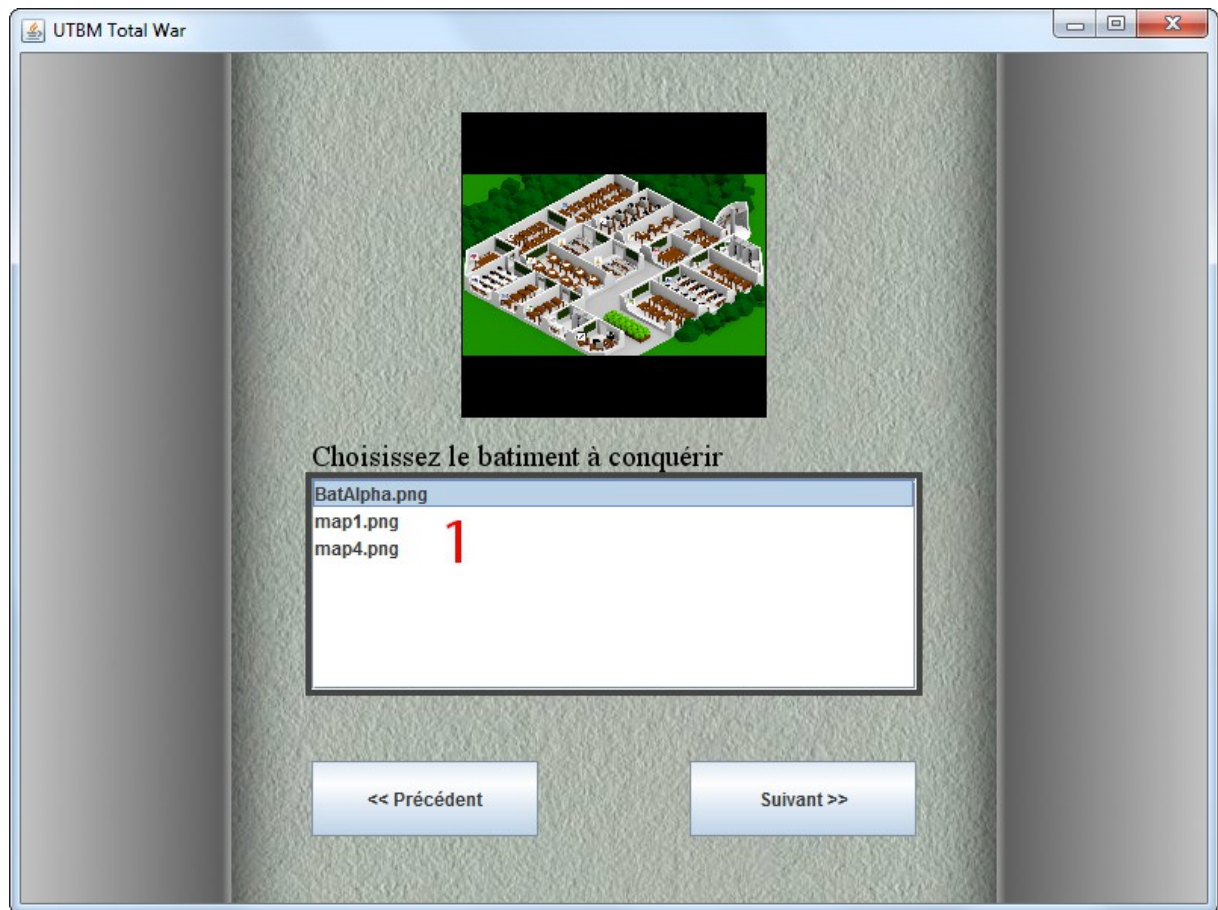


Au lancement du jeu, l'utilisateur arrive directement sur une fenêtre proposant un menu où l'utilisateur peut choisir de lancer une Nouvelle partie (1), d'accéder au menu Options (2), ou de Quitter (3) le jeu.

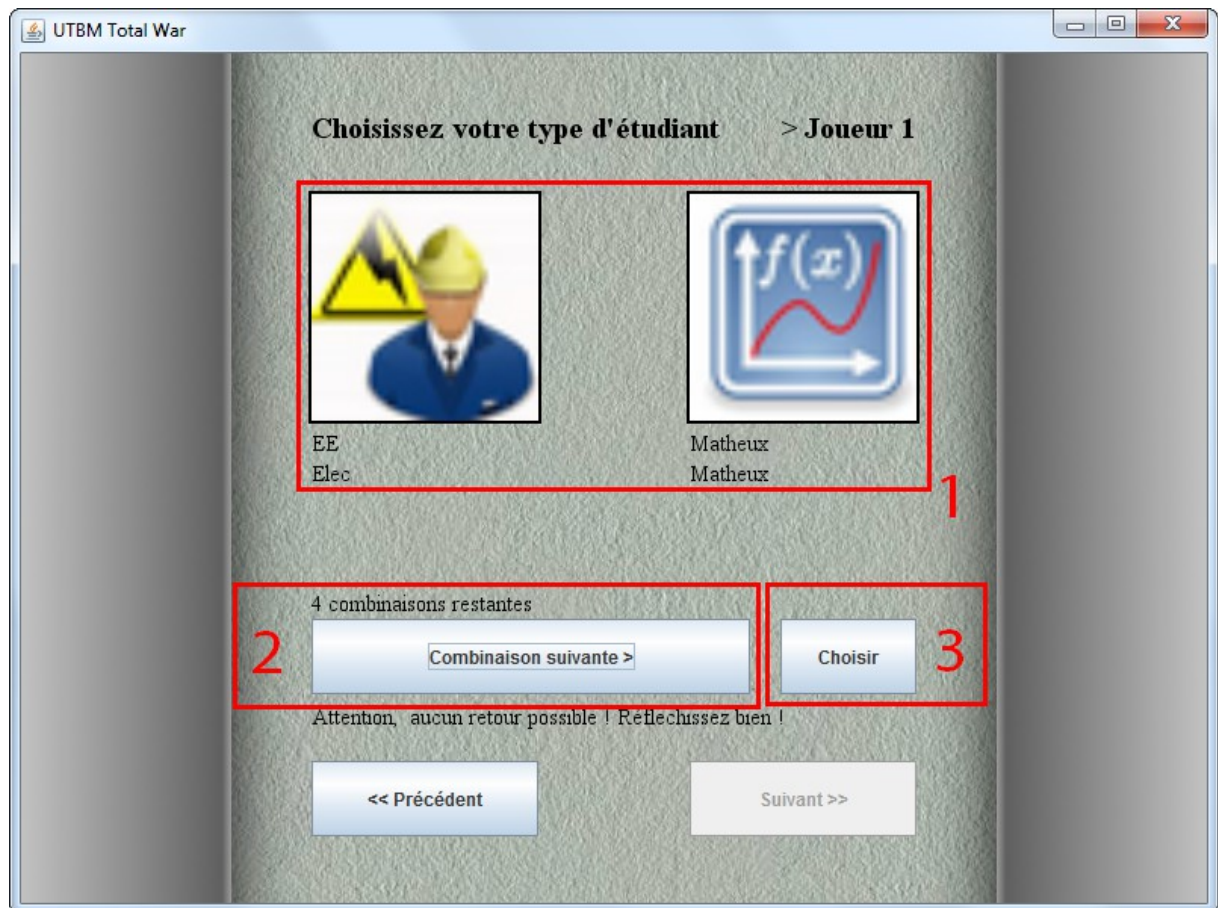


Quand l'utilisateur décide d'entrer dans le menu Options, un choix d'options s'offre à lui : il peut notamment décider d'activer ou non l'antialiasing(1) (ou antirénelage), ce qui permettra de lisser les polices d'écritures et d'éviter le crénelage, un phénomène qui survient lorsqu'on visualise certaines images numériques dans certaines résolutions. Le menu options ajoute également la possibilité à l'utilisateur de profiter pleinement de ce jeu en plein écran (2).

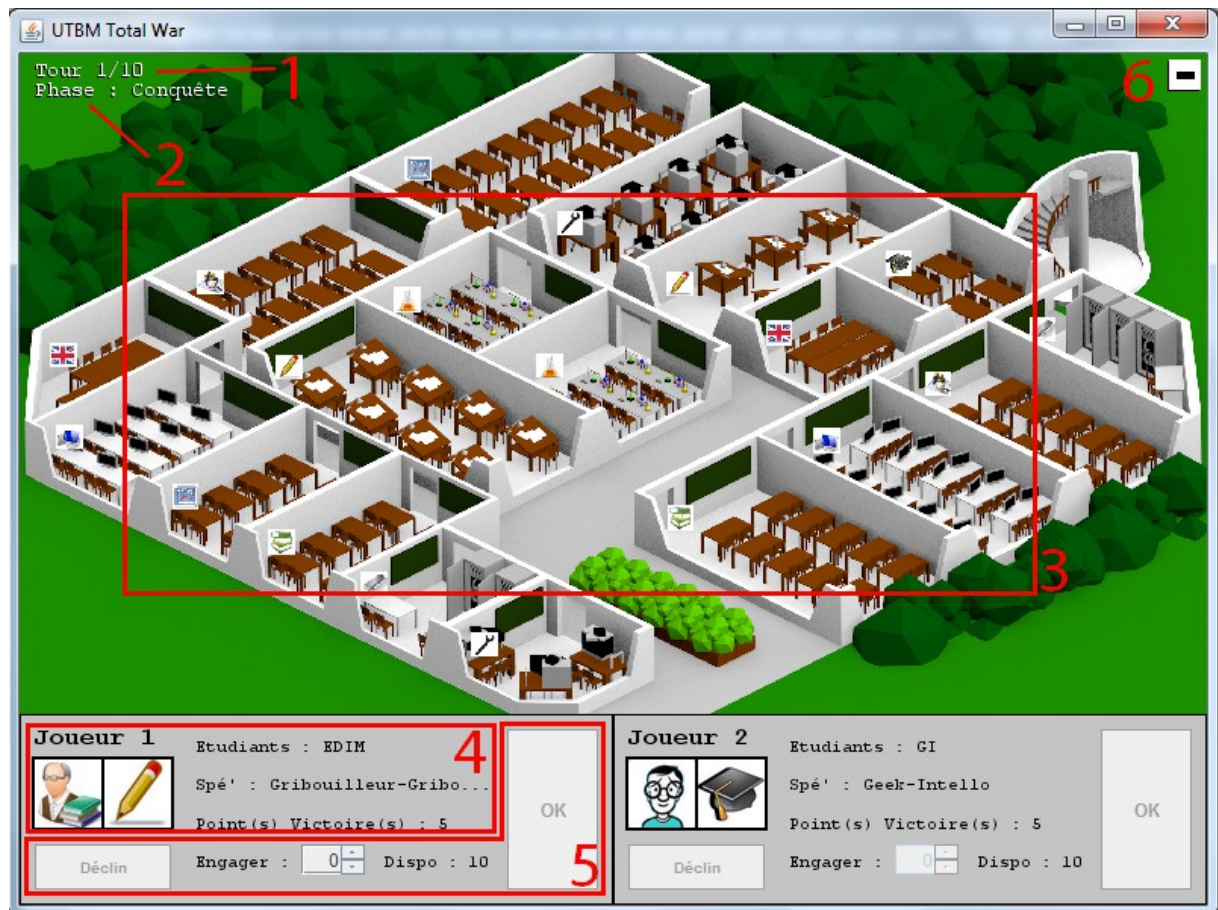
Deux boutons situés en bas de cette fenêtre permettent d'appliquer les éventuels changements et le retour sur la fenêtre de menu.



Lorsque l'utilisateur clique sur « Nouvelle partie », il passe directement sur une fenêtre où un panel des différentes cartes correspondant aux bâtiments lui est proposé.



L'étape suivante est le choix du couple étudiant/spécialité. On propose à l'utilisateur une combinaison d'une spécialité et d'un type d'étudiant (1). Là, deux solutions s'offre à l'utilisateur, il peut soit décider de choisir le couple qui lui est proposé en cliquant sur « Choisir » (3), ou alors il peut demander à voir la « Combinaison suivante » (2). Ainsi, cinq combinaisons maximum lui seront proposées.

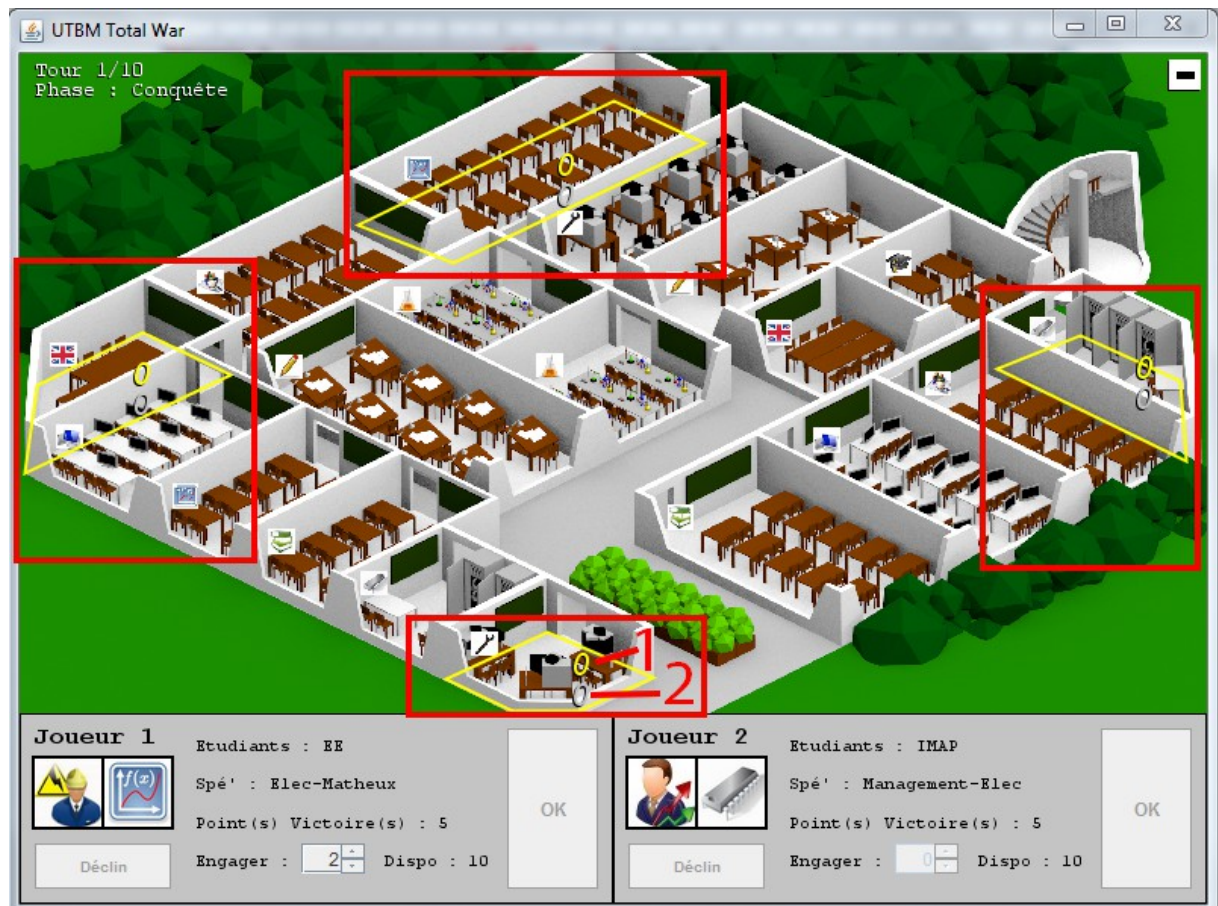


Une fois que les deux joueurs ont choisis leurs couples étudiant/spécialité, nous arrivons enfin sur la carte du jeu, ou nous pouvons apercevoir le contenu d'un bâtiment de l'UTBM (3).

Tout en haut à gauche sont indiqués le numéro du tour en cours (1) ainsi que le type de phase (2) correspondant au jeu, dans notre cas, comme nous venons de commencer cette partie, nous sommes dans une phase de conquête.

Plus bas nous pouvons retrouver deux cadres, qui sont les « tableaux de bord » respectifs aux deux joueurs, dans lesquels se trouvent les informations du joueur (4), dont le type d'étudiant, la spécialité et le nombre de points victoires. On trouve également dans ce tableau de bord les boutons (5) qui vont permettre aux joueurs d'attaquer, d'engager et de redéployer leurs étudiants. Chaque phase de terminant par l'appuie sur le bouton « OK ».

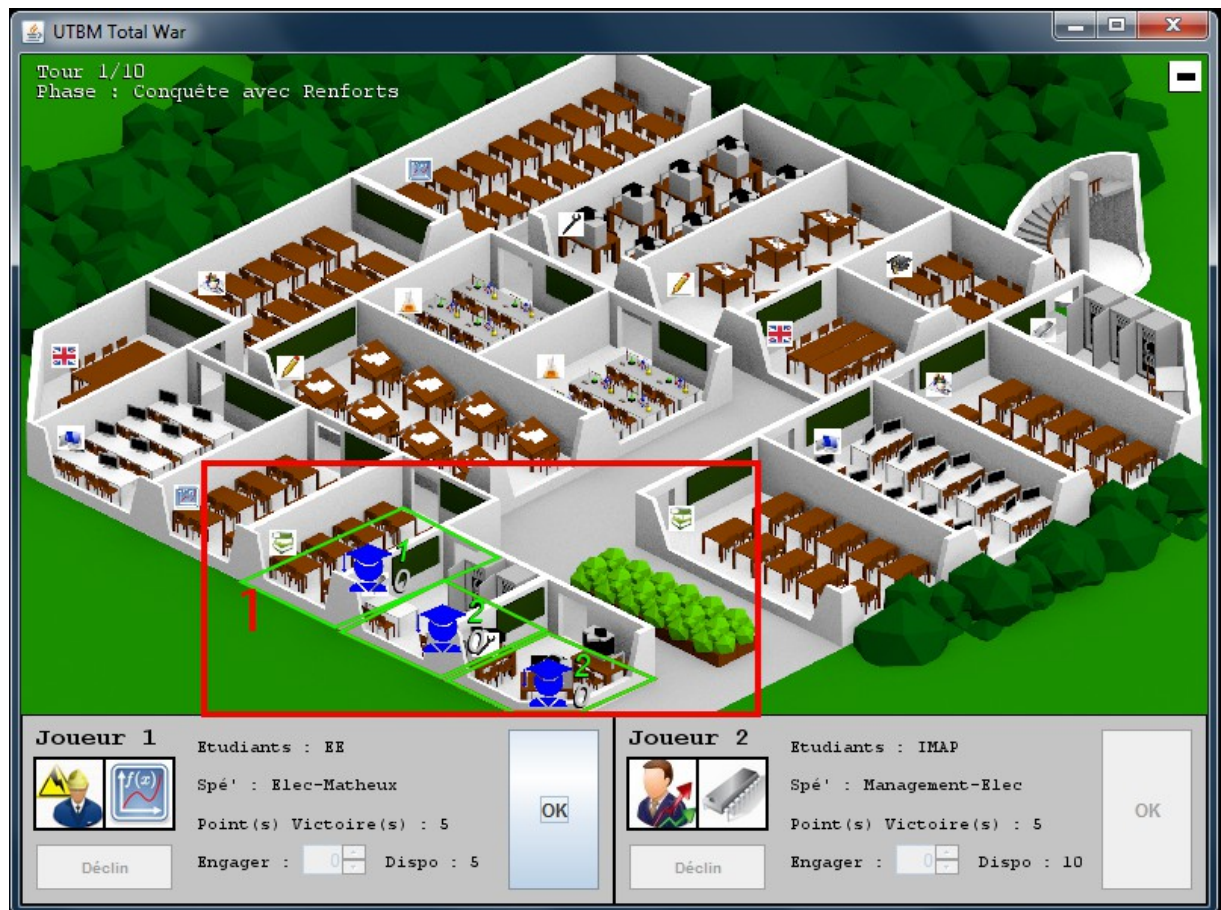
Notons tout en haut à droit le petit bouton (6) qui permet à l'utilisateur de mettre fin à la partie et ce peu importe la phase ou le tour en cours.



Dans la suite des explications, un petit code couleur s'impose, les salles délimitées à l'aide d'un polygone de couleur vert sont les salles possédées, celle délimitée avec un polygone de couleur jaune sont les salles attaquables, et celles délimitées avec un polygone de couleur rouge sont les salles appartenant aux joueurs adverses.

La partie commence, le premier joueur engage des étudiants sur les premières salles attaquables, ces dernières sont représentées par des polygones jaunes sur la carte. Le joueur choisit le nombre de joueur qu'il veut engager sur une salle, puis clique sur celle-ci qui se remplit automatiquement d'étudiants. L'utilisateur peut répéter l'opération autant de fois qu'il le souhaite, dans la limite de ses étudiants disponibles bien évidemment.

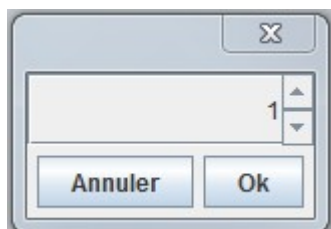
Sur chaque salle est écrit deux chiffres, le chiffre du haut en jaune (1) représente l'effectif actif de la salle, c'est-à-dire, le nombre d'étudiants actifs dans cette salle. Le chiffre du bas en gris (2) représente le nombre d'étudiants inactifs, donc les étudiants qui étaient dans cette salle avant qu'un joueur ne les mette en déclin.



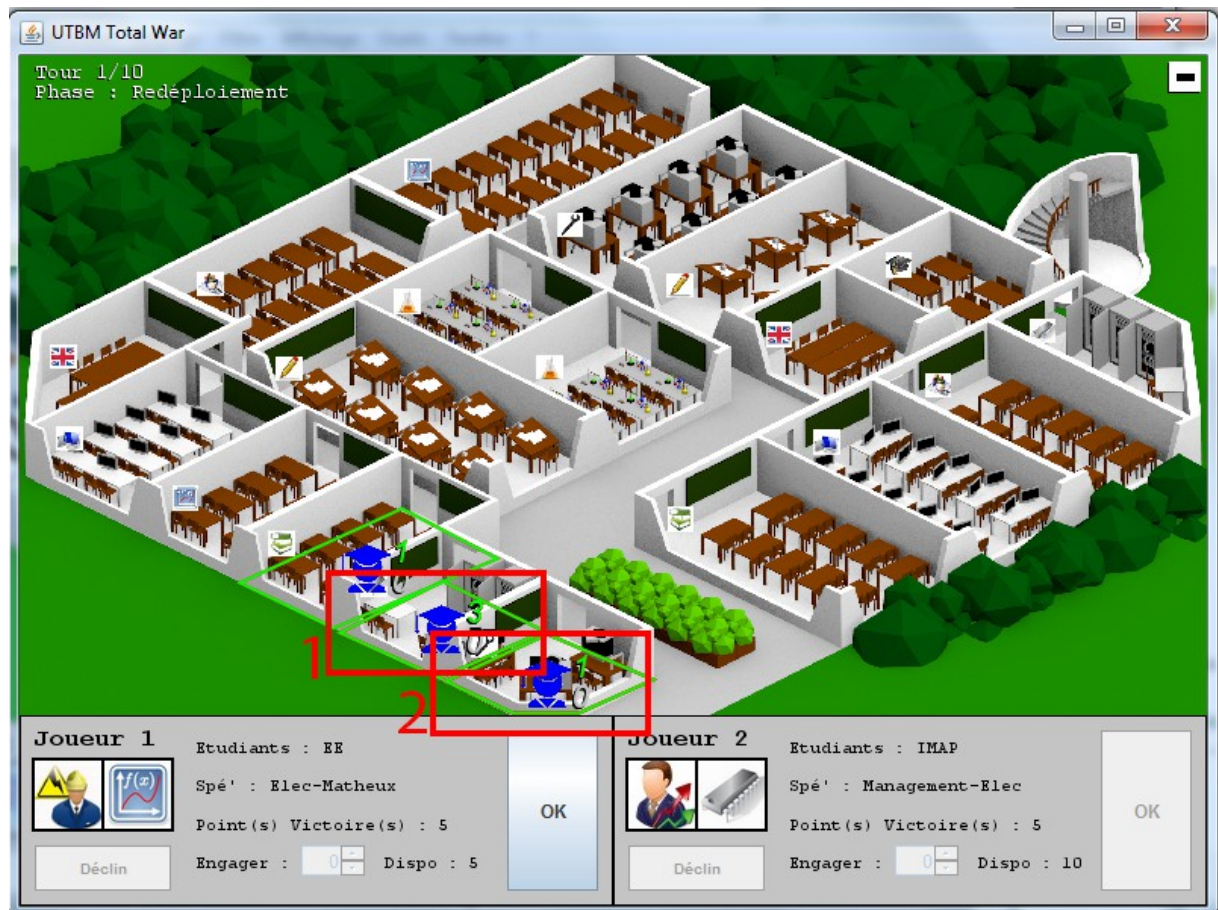
Une fois la « phase conquête » achevée, nous arrivons dans une nouvelle phase, la « phase conquête avec renforts » dans laquelle il est possible à l'utilisateur d'attaquer une salle non-attaquable en temps normal à l'aide d'un lancer de dé. Si le dé est supérieur à 0, la salle est prise sinon, rien ne se passe.

Dans notre cas, nous avons déjà deux salles chacune occupées par 2 étudiants, ensuite, nous avons engagé une personne sur une nouvelle salle (1).

La phase qui suit est une phase de redéploiement, si l'utilisateur souhaite modifier les effectifs de ses salles ou les redistribuer autrement, à ce moment-là il lui suffit de cliquer sur une salle, une fenêtre semblable à celle-ci va s'ouvrir :



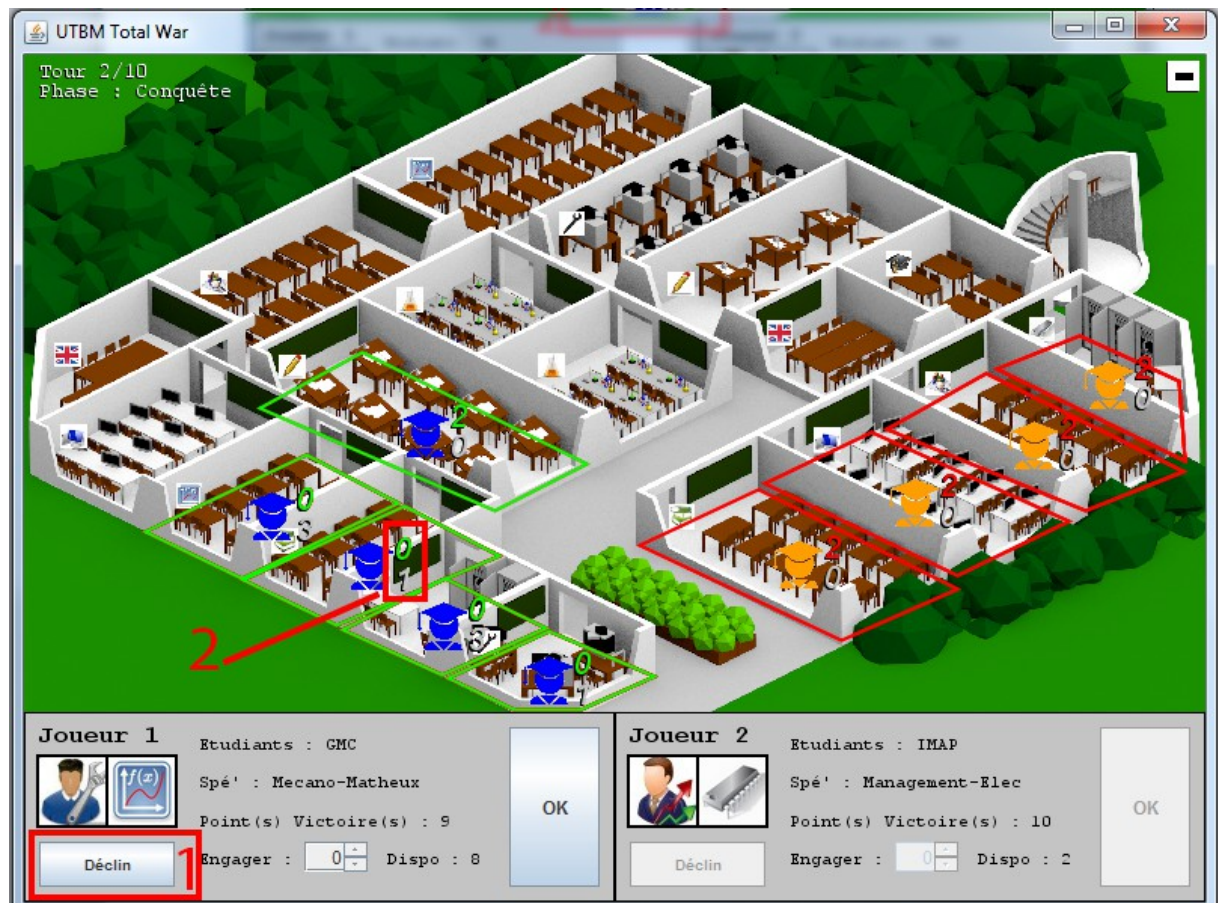
L'utilisateur n'aura alors qu'à indiquer le nombre d'étudiants qu'il veut retirer de cette salle et devra cliquer sur une autre afin d'engager ces derniers dans une autre salle.



Par exemple dans notre cas, nous avons pris un étudiant se trouvant sur la salle (2) que nous avons redéployé sur la salle (1).

Un tour vient de s'écouler, ces phases se répètent à chaque tour. C'est à présent au joueur 2 de jouer.

Une autre « phase spéciale » du jeu consiste à mettre ses étudiants en déclin. En effet, dans le cas où plus aucun d'étudiants n'est disponible, il est possible pour le joueur de mettre les étudiants de chaque salle qu'il possède en déclin, ainsi, ses étudiants continueront à lui rapporter des points tant que la salle n'a pas été conquise par l'adversaire.



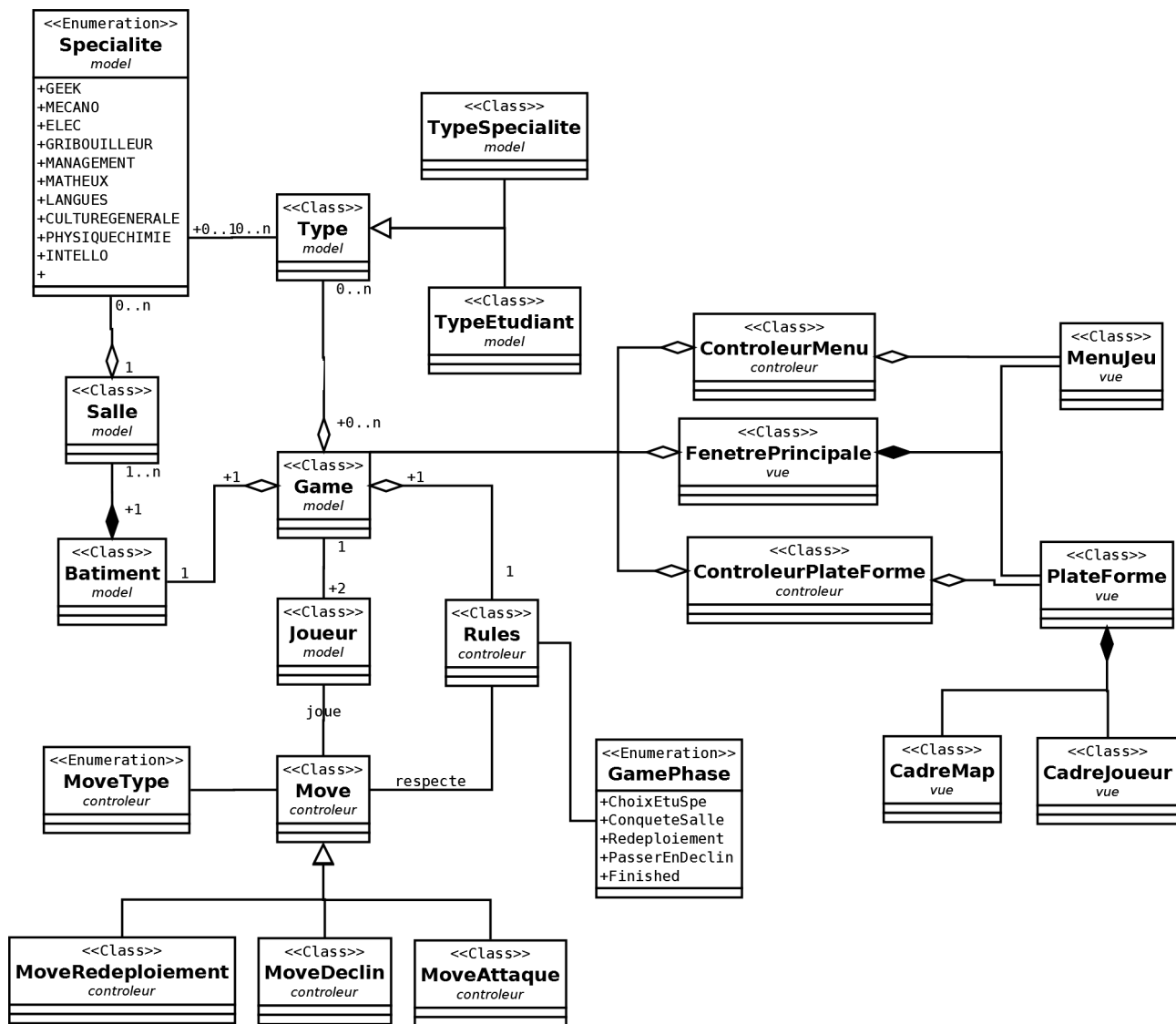
Pour cela, il suffit d'appuyer sur le bouton (1) « Déclin ». On remarque également que les effectifs (2) de chaque salle occupée ont changé et que les étudiants actifs (vert) sont passés en déclin (gris).

Il est possible de passer en déclin deux fois maximum par tour et par joueur.

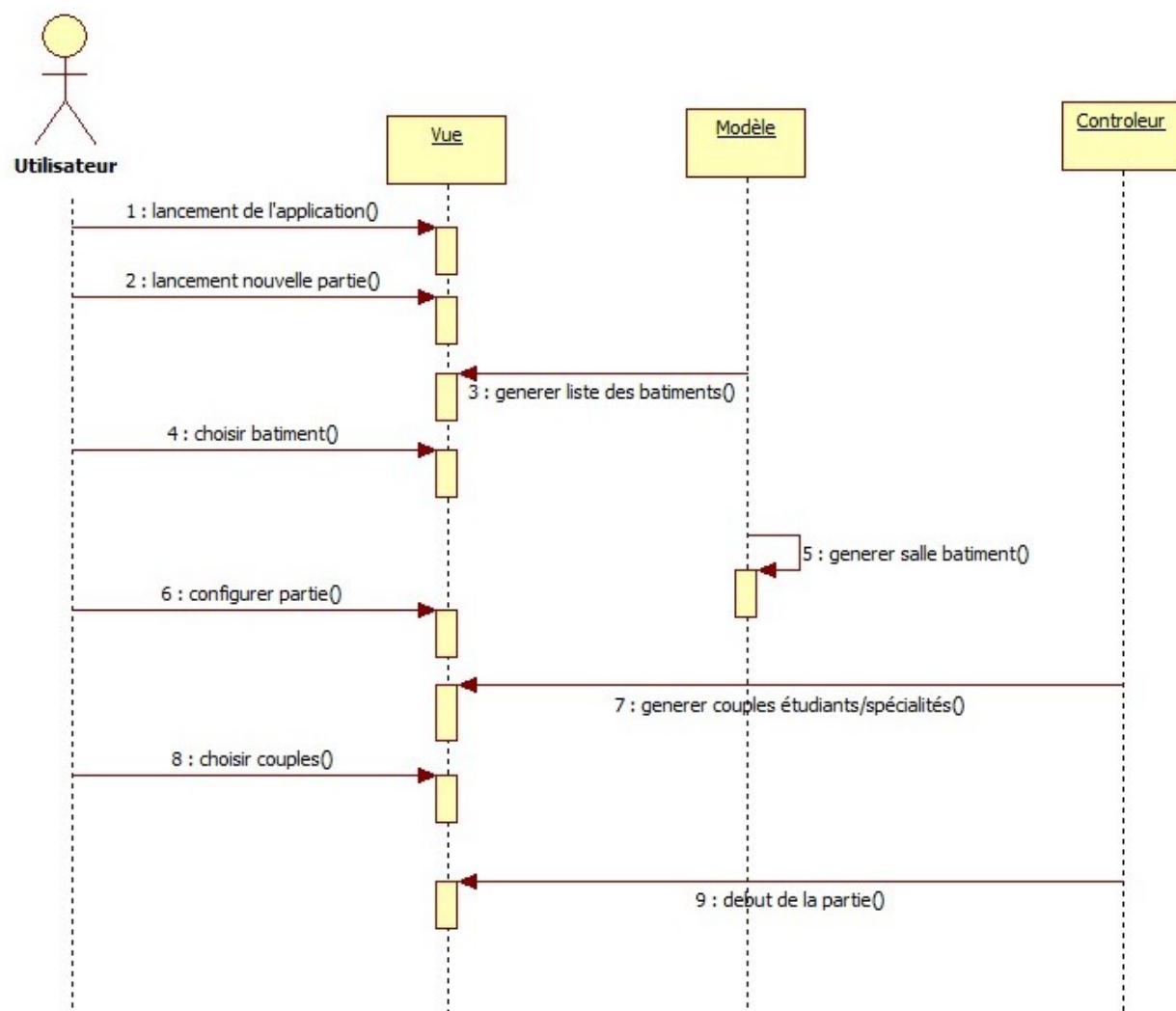
Une fois la partie finie, un message s'affiche à l'écran avec le nom du vainqueur ainsi que le nombre de points de vie de chacun :



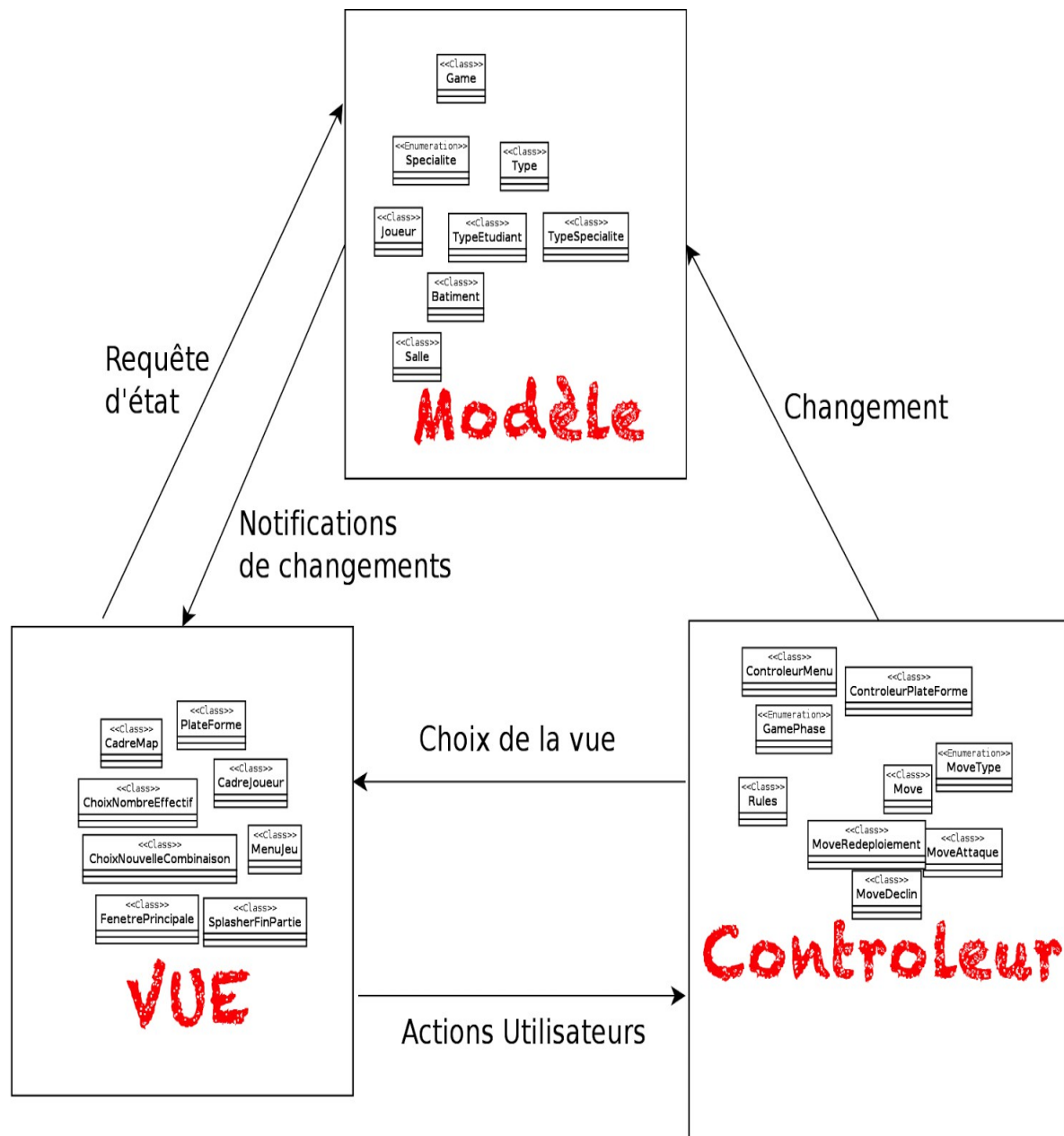
2 Structure



3 Fonctionnement



4 Diagramme MVC



5 Contrôleur

Dans ce package se situe toutes les classes qui vérifient les règles métiers qui définissent le jeu, c'est la structure du jeu.

ControleurMenu : Cette classe est le contrôleur de MenuJeu. Tous les événements sur MenuJeu sont retransmis et interprétés par cette classe.

ControleurPlateForme : Cette classe est le contrôleur de PlateForme donc de CadreMap et de CadreJoueur. Tous les événements sur PlateForme sont retransmis et interprétés par cette classe.

Ils sont ensuite envoyés à Rules.

GamePhase: Dans cette classe va être défini les différentes phases du jeu sous forme d'énumération. L'énumération permet de rajouter des phases de jeu si nécessaire dans une éventuelle évolution du jeu.

Move: c'est la classe mère qui regroupe MoveAttaque, MoveDeclin, MoveRedeploiement.

MoveAttaque: l'instance qui représente un coup joué par l'utilisateur lorsqu'il veut conquérir une salle. Dans cette classe, on aura aux accès aux informations telles le nombre de jetons qu'il a engagé pour la conquête de la salle.

MoveDeclin: cette classe représente le coup joué lorsque l'utilisateur veut passer son étudiant actif en déclin afin de choisir un nouveau couple étudiant/spécialité.

MoveRedéploiement: c'est l'instance qui représente le coup qui va être joué par l'utilisateur lorsqu'il considérera qu'il n'a plus assez de jetons pour attaquer ou conquérir une salle. Un renfort aléatoire de 1,2 ou 3 sera ajouté à son nombre de pions engagés.

MoveType: c'est une énumération qui présente tout les types de Move possibles. Elle permet de différencier par exemple lors d'un MoveAttaque, une conquête de salle ou une attaque de salle (des pions ennemis sont déjà présents dans la salle).

Rules: c'est la classe qui est le béton de la structure du jeu, elle représente les règles du jeu. Pour toutes actions d'un joueur sur la vue, elle contrôle que cette action est permise par les règles définies. Un autre rôle de cette classe, par exemple, on calcule le nombre de points victoire est attribué à chaque joueur.

6 Modèle

Dans ce package se situe toutes les classes qui vont rendre le jeu fonctionnel, c'est le cœur du jeu.

Batiment : c'est la « map » du jeu, un bâtiment est défini par les salles qui le composent : une liste des salles et une liste des premières salles accessibles seront stockées dans cette classe.

Joueur : c'est la partie qui représente un joueur, son nom, son effectif disponible, ses points victoires dont il n'aura connaissance qu'à la fin de la partie

Salle : la salle est l'objet de la lutte entre les joueurs de UTBM Total War. A partir d'une salle, on peut savoir son atout, si elle est occupée et ses salles adjacentes.

Specialite : c'est une énumération des spécialités. L'énumération a été mise en place pour l'évolutivité du jeu et permettre une meilleure visibilité du code., c'est à dire que si l'utilisateur désire ajouter de nouvelles spécialités, les spécialités en question seront à ajouter dans

l'énumération.

Type : Classe mère qui regroupe le TypeEtudiant et le TypeSpecialite.

TypeEtudiant : c'est la classe qui représente l'étudiant que choisira le joueur.

TypeSpecialite : c'est la spécialité qui sera associé à l'étudiant.

7 Vue

Dans ce package se situe toutes les classes qui vont rendre le jeu interactif, c'est l'interface du jeu.

FenetrePrincipale : Cette classe est la fenêtre principale de l'application. Tout ce qui sera affiché, sera dans cette fenêtre.

MenuJeu : Il est une des deux grandes parties de FenetrePrincipale. C'est lui qui gère l'affichage du menu principale du jeu ainsi que de ses sous-menus.

PlateForme : Elle est la seconde grandes parties de FenetrePrincipale. Elle gère l'affichage du plateau du jeu ainsi que l'affichage des informations des joueurs.

CadreMap : Cette classe fait partie de PlateForme. C'est un cadre qui gère l'affichage du plateau du jeu (le bâtiment).

CadreJoueur : Cette classe fait partie en double de PlateForme. C'est un cadre générique qui permet d'afficher les informations d'un joueur.

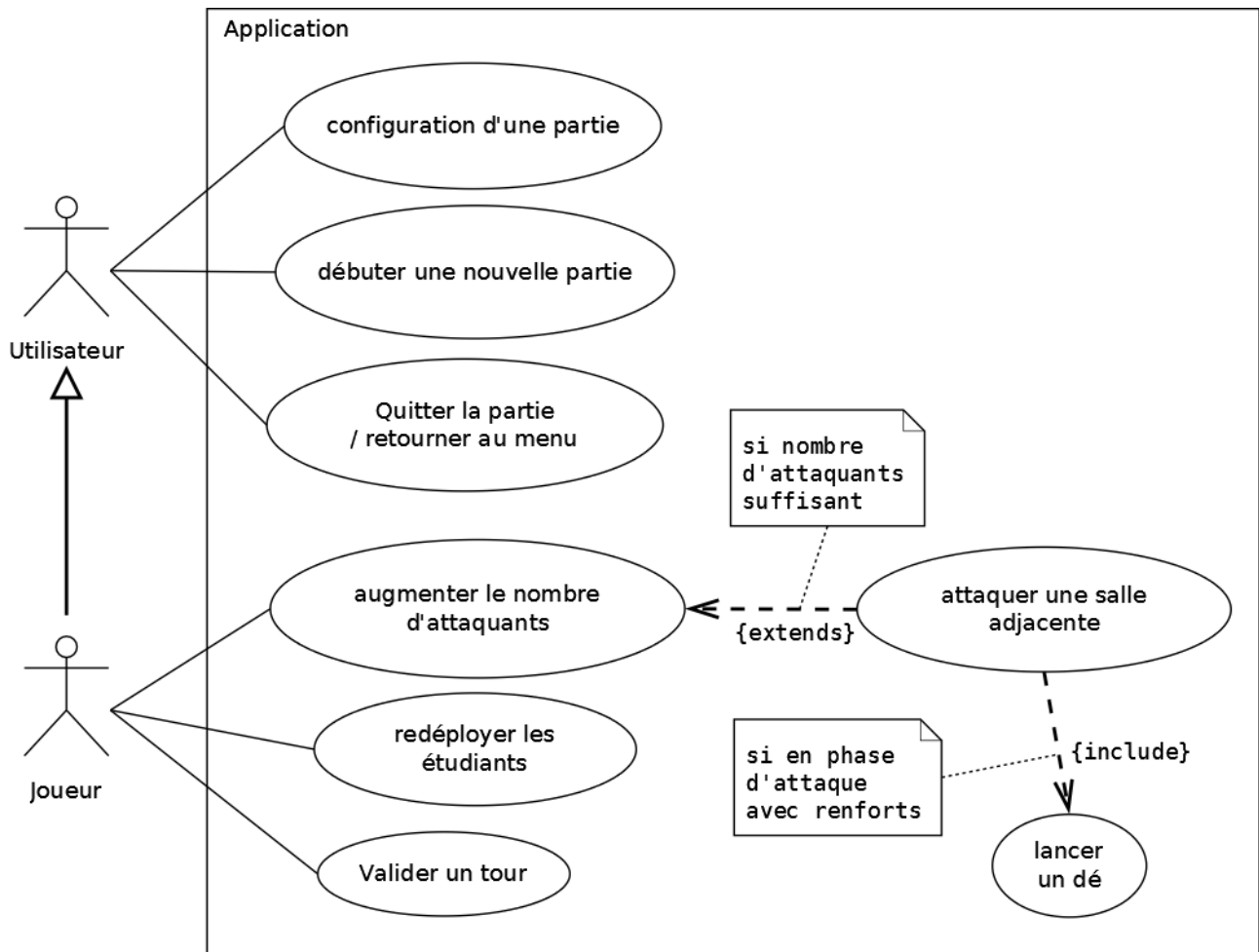
SplasherFinPartie : C'est un splasher qui s'affiche à la fin d'une partie en cours. Elle permet d'afficher le score des joueurs et le vainqueur de la partie.

ChoixNombreCombinaison : Cette classe gère une fenetre pop-up demandant à l'utilisateur si il veut changer de combinaison Etudiant-Spécialité. Elle lui en propose 5 comme dans un des sous-menus de MenuJeu.

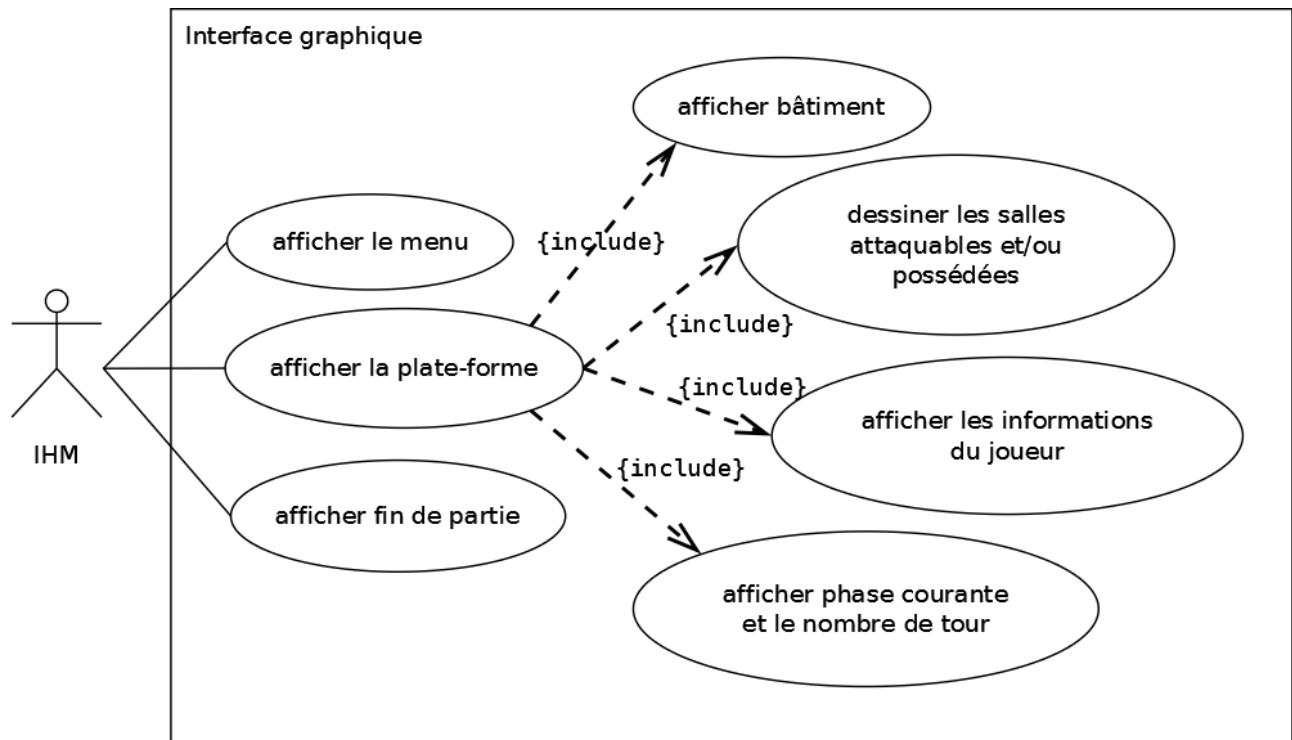
ChoixNombreEffectif : Cette classe gère une fenêtre popup qui s'affichera en dessous du curseur pour demander à l'utilisateur, combien d'étudiants il veut déplacer. Cette fenêtre n'est disponible que pendant la phase de redéploiement.

8 Diagramme de cas d'utilisation

Afin de bien montrer comment fonctionne l'application, nous avons construit un diagramme de cas d'utilisation.



Ci-dessus, un premier diagramme expliquant les actions que peut effectuer l'utilisateur. Ci-dessous, un second diagramme expliquant quelles sont les fonctions de l'interface graphique.



VI. Améliorations possibles

Tout au long de la réalisation de ce projet, nous avons pensé à plusieurs améliorations possibles, notamment :

- Le choix de plusieurs bâtiments : il est normal pour un tel jeu de proposer proposer plusieurs cartes différentes à l'utilisateur.
- Augmenter le nombre de joueurs : plus on est de fous, plus on rit.
- Un panel d'étudiants et de spécialités plus important.
- L'ajout de musiques d'ambiance et de sons dans le jeu afin de rendre la partie encore plus prenante.
- Quelques animations, pour animer des personnages dans le bâtiment, rendant ainsi notre carte plus vivante.
- L'ajout d'une intelligence artificielle pour permettre une partie à un joueur.
- Une sauvegarde des préférences à la fermeture du jeu pour éviter de changer les options à chaque nouvelle partie.
- La mise en place d'un réseau + chat, permettant ainsi aux joueurs de pouvoir dialoguer entre eux.
- La récupération des scores personnels, afin de créer des profils, établis ainsi un "guinness book" des records propre à Small UTBM War pour permettre une compétition entre les joueurs du monde entier !
- L'évolution vers un mode 3D
- L'implémentation vers une version mobile pour pouvoir jouer partout.

VII. CONCLUSION

Cette expérience nous a offert une bonne préparation à notre insertion professionnelle car elle fut pour nous une expérience enrichissante et complète qui conforte notre désir d'exercer le métier d'ingénieur dans le domaine de l'informatique.

Ce projet nous a permis d'appliquer des connaissances personnelles, mais nous avons également pu découvrir de nouvelles choses telles que certaines subtilités du langage Java. Il nous a également permis de travailler au sein d'un groupe où chacun a su s'organiser et prendre ses responsabilités. Ainsi, nous garderons un très bon souvenir de cette expérience.