

Jeu Civilisation

Rapport de conception

8 novembre 2012

AMANDINE LE CAHAIN
ADRIEN BRUNELAT

Introduction

Civilization est un jeu vidéo en tour par tour très connu des amateurs de jeux de stratégie. Pour ce projet d'Analyse, de Conception et de P.O.O, nous développerons une version minimale du jeu Civilization dans un univers un peu différent.

En effet, ici, les deux civilisations disponibles seront représentées par les deux départements EII et INFO. Les unités seront différenciées en 3 types à savoir « Directeur de Département », « Etudiant », « Enseignant » chacune ayant ses particularités.

Ce rapport présentera tous les documents réalisés lors de notre modélisation du jeu Civilization.

Contenu

Rapport de conception	1
Introduction.....	2
Modélisation du jeu.....	4
1. Fonctionnalités illustrées par des cas d'utilisation.....	4
❖ Use Case 1: Units	4
❖ Use Case 2: Player	4
2. Modélisation des données et patrons de conception.....	5
❖ Diagramme d'interfaces	5
❖ Diagramme des classes d'implémentation	5
3. Modélisation des comportements du jeu	5
❖ Diagrammes d'interactions	5
❖ Diagrammes d'états-transitions	5

Modélisation du jeu

1. Fonctionnalités illustrées par des cas d'utilisation

❖ Use Case 1: Units

Le cas d'utilisation représentant les différentes actions des unités est relativement basique puisqu'une unité ne peut que se déplacer, attaquer pour certaine et créer une ville pour d'autres.

Une unité peut être soit un Directeur de département, un étudiant ou un enseignant comme représenté par l'héritage.

Figure 1 : Unit Use Case

❖ Use Case 2: Player

Un joueur possède de nombreuses actions à réaliser autant au niveau des menus du jeu que de la partie en elle-même. Un joueur peut donc créer une map pour une partie et choisir sa taille. Il choisit sa civilisation puis il possède ensuite toutes les actions relatives à la partie. Les différents cas d'utilisation disponibles pour un joueur sont représentés sur la figure 2 de la page suivante.

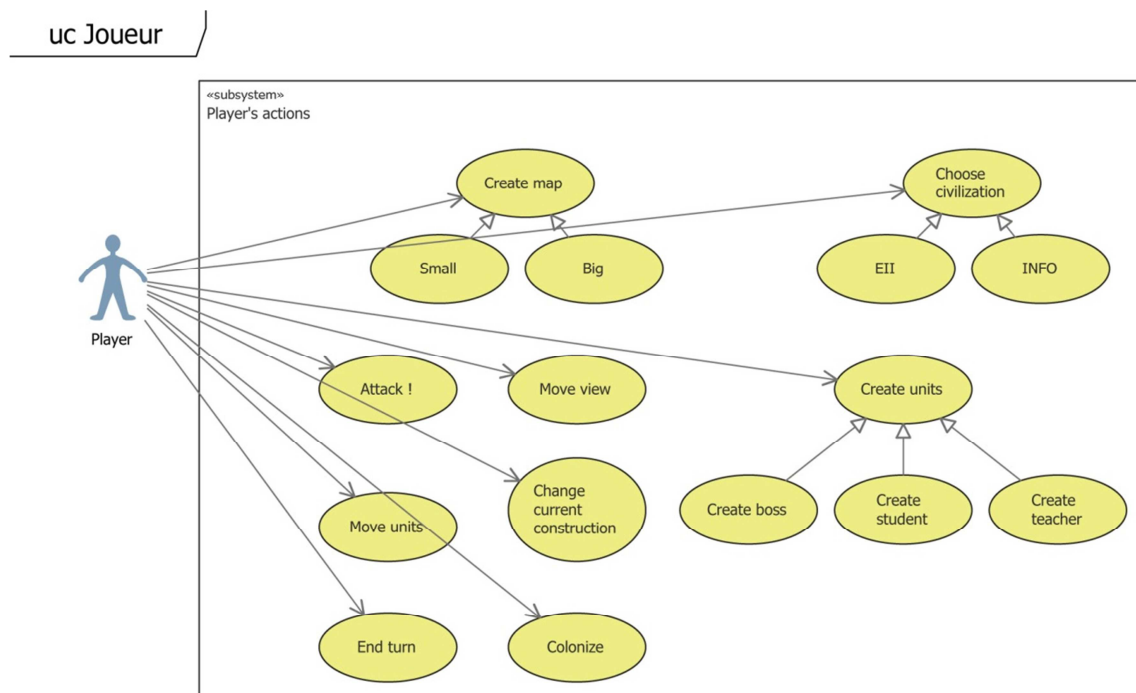


Figure 2 : Player Use Case

2. Modélisation des comportements du jeu

❖ Diagrammes d'interactions

❖ Diagrammes d'états-transitions

3. Modélisation des données et patrons de conception

❖ Diagramme regroupant les interfaces (API)

Fabrique abstraite : Création des unités

Nous avons utilisé le patron de conception Fabrique Abstraite permettant la fabrication d'objets concrets pour la gestion de la création des unités. Notre interface **CivilizationFactory** a donc pour rôle de créer les unités d'une civilisation. Elle donnera deux implémentations : INFOFactory et EIIFactory.

Monteur : Création de la partie

Le monteur est utilisé pour la création d'un objet complexe. GameBuilder va ici nous permettre de gérer la création d'une partie comme représenté sur la figure suivante:

Poids-mouche et décorateur : Création des cases et gestion de leurs ressources

La gestion des cases de la carte et notamment leur création sera attribuée à un poids-mouche pour faire apparaître une nouvelle classe de construction des cases. Chaque case possède un nombre de ressources. L'attribution des ressources additionnelles aux différentes cases sera gérée par un décorateur comme le représente la figure suivante :

Stratégie : Création de la carte

La carte peut être créée de deux tailles différentes ; une petite de 25 x 25 cases ou une grande de 100 x 100 cases. Pour gérer la création de ses cartes et pour que le joueur puisse choisir la taille de la carte sur laquelle il veut jouer, le patron Strategy sépare l'algorithme de création de carte de la carte elle-même comme représenté sur le diagramme suivant :

Finalement, le diagramme de classes en annexe donne l'ensemble des interfaces qui seront nécessaires à l'implémentation de notre jeu.

❖ Diagramme des classes d'implémentation

S'ajoutant à notre diagramme regroupant les interfaces, nous avons créé un diagramme des classes implémentant ces interfaces. Il est encore assez difficile de déterminer l'ensemble des

attributs de chaque classe ainsi que le réel besoin de celle-ci. Nous n'hésiterons pas pendant notre implémentation à retoucher certains des attributs et ajouter voir supprimer des classes qui nous semblerait à l'avenir inutiles.

Dans notre choix de conception, nous avons ajouté deux énumérations ; l'une pour le type de production des villes et l'autre pour le type de la map comme présenté sur les deux figures suivantes :