

## 7 Wonders version UTBM

Dans ce rapport, nous allons présenter la manière dont nous essaierons de réaliser ce projet dans le cadre de l'UV LO43. En nous inspirant du jeu de société « 7 Wonders », nous aimerions reprendre ses concepts et les transformer afin qu'ils s'appliquent à l'univers de notre école, c'est-à-dire l'UTBM. Afin de bien planifier ce que nous avons à faire, nous essayons d'utiliser la méthode AGILE :

- Tout d'abord, nous avons commencé par définir ce que nous devons rendre comme travail en Janvier. Nous avons donc réalisé le « Product Backlog » : liste des fonctions et éléments attendus à la fin du projet. Connaissant déjà notre date de fin et de « livraison » de projet, nous avons planifié des réunions.

- Lors des réunions (ou SCRUM), nous mettons au clair ce que nous avons fait, ce que nous devons faire et pour quand cela doit être fait. Ces réunions nous permettent voir ce qui est réalisable, ce qu'il faut revoir, ce que nous devons ajouter...

- Entre deux réunions, nous essayons chacun de faire ce que nous avons convenus lors de la réunion. Nous sommes toujours en contact si jamais une personne a besoin de précision sur un concept. Cette période est notre SPRINT.

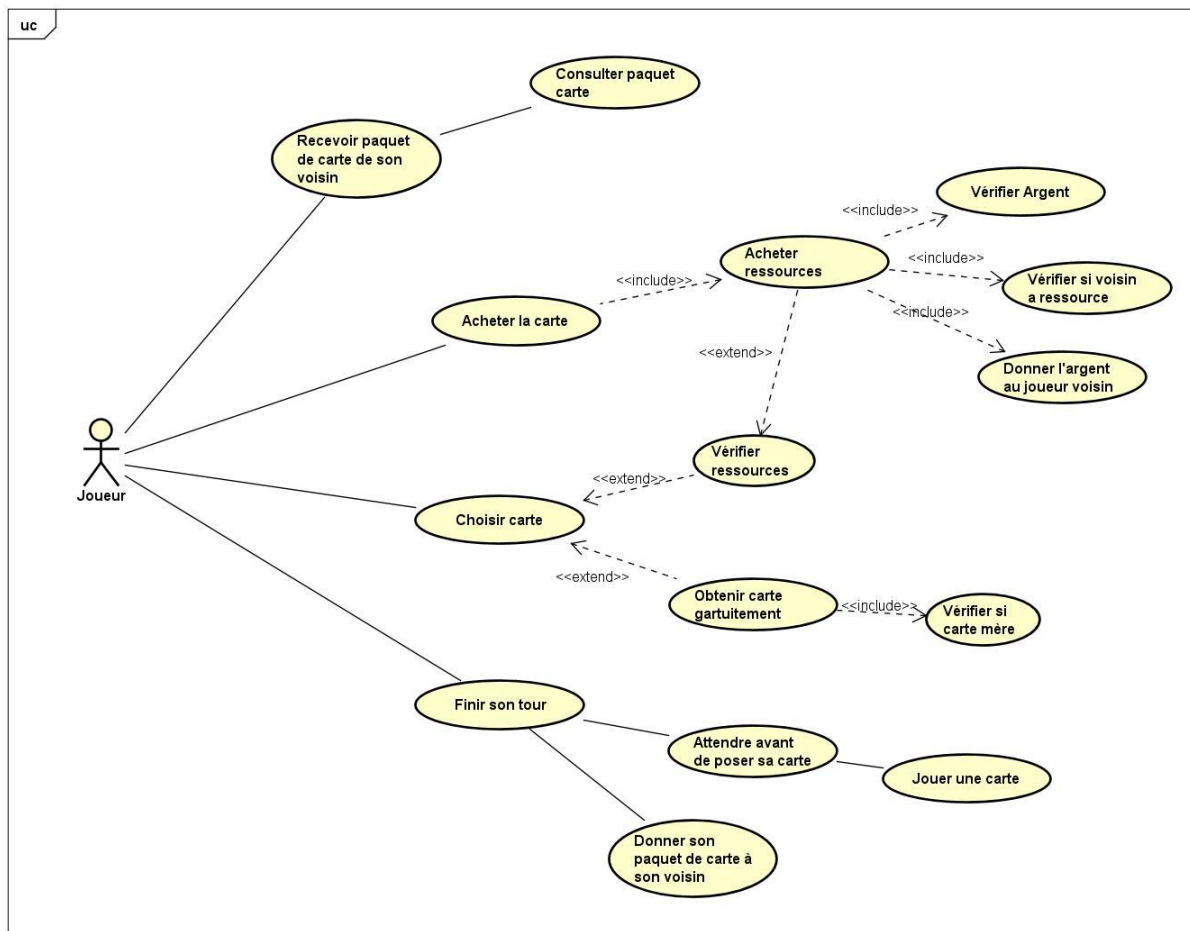
### Description de notre Product Backlog :

Notre jeu se joue à 4 joueurs. Ceux-ci prennent le contrôle d'un endroit en rapport avec l'UTBM et à l'aide des ressources disponibles, ils développeront leurs relations commerciales et entrepreneuriales, affirmeront leurs suprématie au sein de l'école grâce à des points de courage et participeront à la marche en avant du progrès. Tout ceci dans le but de valider leurs 3 années de branche au sein de l'école.

Afin d'expliquer de manière plus détaillée comment sera fait notre jeu, nous avons utilisé différents diagrammes du langage UML. Le premier est un diagramme de cas d'utilisation. Il nous est important pour comprendre les besoins que le logiciel devra fournir. Nous nous plaçons du point de vue de l'utilisateur.

### Diagramme de Cas d'Utilisation :

Dans ce diagramme, nous n'avons défini qu'un seul acteur : le joueur. Puis nous avons essayé de lister toutes les actions que l'utilisateur fera au cours d'une partie. Comme nous avons prévu de faire le jeu sur un seul écran, beaucoup d'actions sont en rapport avec ce que le joueur pourra faire lors de son tour.



### Diagramme de Classe :

#### Cadre MVC : Regroupe les composants maîtres du système MVC

- La View permettra l'affichage des plateaux et cartes, ainsi que les informations qui seront superposées à ces images.
- La Controller permettra à l'utilisateur de sélectionner des cartes et de passer son tour.
- La Model déclenchera des appels de fonctions générales en fonction des retours de Controller, afin de coordonner toutes les fonctions présentes dans les autres classes.

#### Cadre composition de la classe Joueur : Permet d'obtenir une classe Joueur complète

- La Joueur contient tous les attributs nécessaires au stockage des données d'un joueur, ainsi que les pseudo-getter permettant de les modifier.
- La Plateau permet de regrouper les informations propres au plateau (ex : ressources du joueur) et ainsi de simplifier l'affichage fait par View.
- La SousListe correspond à la main du Joueur. On rappelle que celle-ci change entre chaque tour.

#### Cadre déclaration des différentes cartes : Permet de définir les cartes de jeu

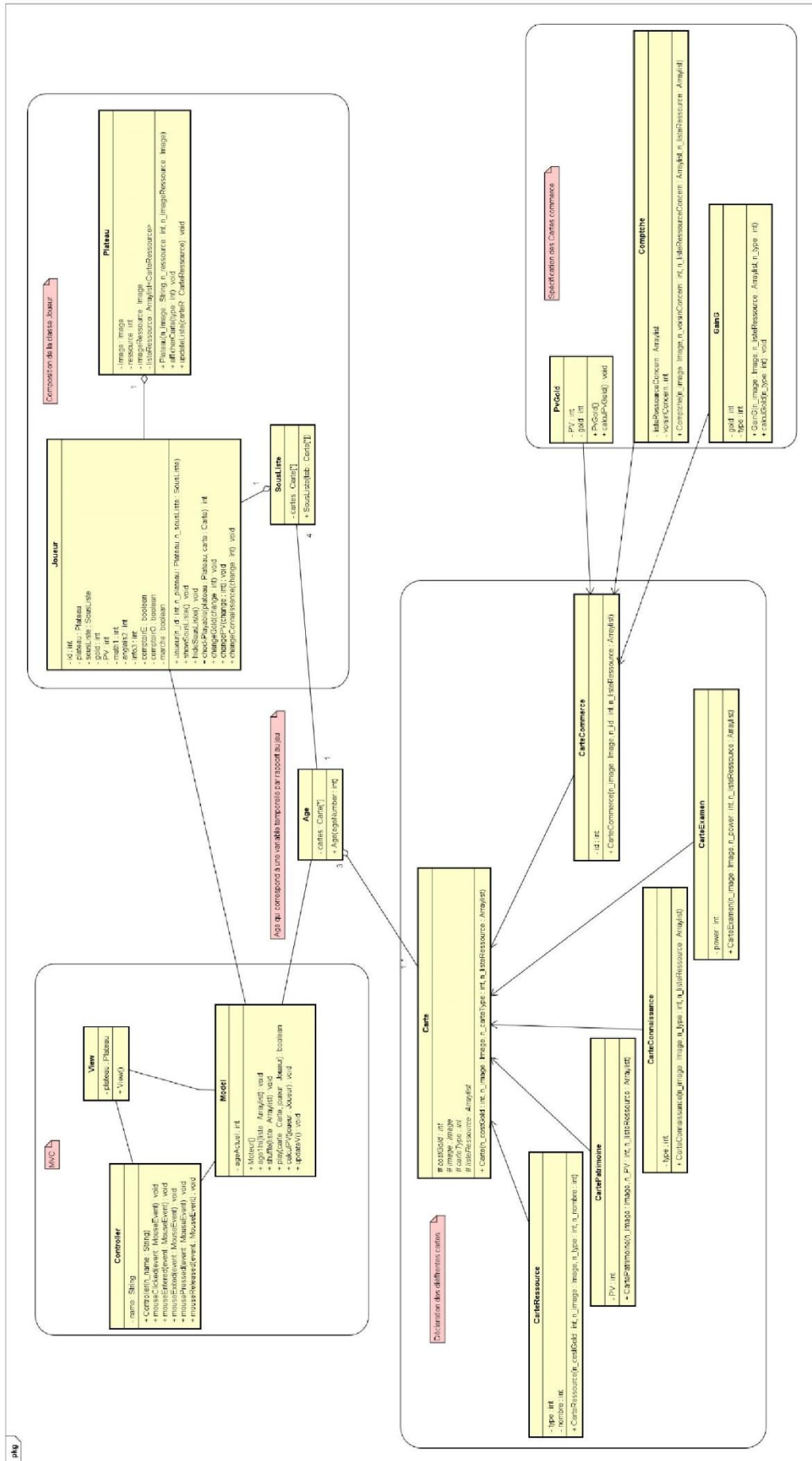
- La Carte est la classe mère de toutes les cartes. Elle permet simplement de faire un héritage pour alléger le code des attributs communs entre tous les types de carte.
- La CarteRessource contient toutes les cartes donnant des ressources du jeu 7 wonders (les ressources étant différentes du jeu de base pour s'adapter au sujet du projet).
- La CartePatrimoine correspond aux cartes bâtiment civil du 7 wonders.

- La CarteConnaissance correspond aux bâtiments scientifiques du 7 wonders.
- La CarteExamen correspond aux bâtiments militaires du 7 wonders.
- La CarteCommerce correspond aux bâtiments commerciaux du 7 wonders.

Cadre Spécifications des cartes commerce : Les cartes commerces ayant toutes une manière relativement unique de fonctionner, elles sont traitées en petits groupes.

- La PvGold correspond aux cartes donnant des points de victoires et de l'or en fonction des ressources d'un joueur
- La Comptche (comptoir + marché) correspond aux cartes permettant de réduire le coût d'achat d'une ressource à une cité voisine.
- La GainG (gain de gold) correspond aux cartes permettant de gagner de l'or (quantité fixe / dépendante de différentes variables)

La Age quant à elle, est une classe utilisée par plusieurs cadres. Son utilité est majoritairement de permettre aux différentes fonctions de s'adapter en fonction de l'avancement dans la partie (adapter les cartes disponibles, compter les victoires militaires etc.).



## Diagramme de Séquence :

Ordre de Lecture :

1. Séquence Diagramme : Game Standard
2. Différents Cas pour les types de Cartes
  - a. Case Ressource ou Commerce
  - b. Case Bâtiment

Le Joueur va commencer par « Start the game », il y aura ainsi la génération des Plateaux Individuels avec leur ressource attitrée et un espace prévu pour les pièces. On va ainsi afficher les 4 Plateaux Individuels sur le Plateau Principal (appelé par la suite Plateau). Les joueurs peuvent voir tous les plateaux individuels.

*Note : Les flèches colorés sont pour l’affichage :*

*- en Rose sont ce qui est vue par le joueur.*

*- en Or sont ce qui a été effacé de sa vue.*

Par rapport aux cartes : On configure les Cartes par Age. (3 « paquets » par Age). Après avoir répartis, l’Age n’est plus utilisé.

Du côté Joueur, chaque Joueur reçoit 3 pièces en début de partie. Elles sont affichées sur le Plateau dans la zone spécifiée.

On rentre ainsi dans la 1<sup>ère</sup> boucle qui dépend de l’Age (1<sup>er</sup>, 2<sup>ème</sup> ou 3<sup>ème</sup> Age). On mélange donc les cartes. On crée 4 Sous Listes qui seront les Cartes pour chaque Joueur. A la fin, on supprimera les 4 Sous Listes créées pour cet Age. Et on « bloque » dans les Cartes l’Age correspondant.

On rentre dans la seconde boucle qui est un Age (on va jouer toutes les cartes des Sous Listes créées jusqu’à ce qu’il ne reste qu’une carte dans la Sous Liste de chaque Joueur). Cette boucle est créée pour tourner les Sous Listes des Joueurs, ainsi pendant un Age, les Joueurs ne voient pas toujours les mêmes cartes.

3<sup>ème</sup> boucle qui est le choix du Joueur, elle boucle jusqu’à ce que tous les Joueurs aient pris une Carte de leur Sous Liste. C’est un tour de partie.

On affiche la Sous Liste créée au Joueur. On attend le choix du Joueur par un clique souris sur la Carte voulue. A partir de ce moment-là, il y a plusieurs options possibles en fonction du type de Carte. (*Cf Ordre de Lecture*).

A) Si le Joueur choisit Ressource ou Commerce :

On va vérifier si le Joueur peut prendre la Carte voulue. On va vérifier en premier les ressources de son Plateau et ceux de ces voisins directs. Puis son argent si suffisant. Si tout va bien, on affiche les icônes de ces ressources gagnées sur le plateau (petite image rajoutée sur le plateau avec l’icône de cette ressource). On fait ainsi la mise à jour de son argent sur le Plateau et sur son jeu.

B) Si le Joueur choisit Patrimoine

Même cas que Ressource et Commerce mais il y a, en addition, la mise à jour des points de victoires associés au patrimoine sur le Plateau et sur le Joueur.

Ensuite, il y a l'affichage de cette action A ou B.

On efface après avoir cliqué sur la carte, la Sous Liste du Joueur et on affiche le bouton « next player ». On le force à appuyer sur le bouton « next player », il ne se passe rien s'il ne clique pas. L'affichage permettant ainsi que le nouveau joueur ne voit pas les Cartes du joueur précédent. Quand le bouton est cliqué, on « efface » de la vue le bouton.

Les boucles sont toutes finies, on est donc à la fin des 3 Age, les Cartes sont « bloquées » et les Sous Listes sont supprimées, on supprime à la suite les Cartes.

On affiche sur le Plateau « End » et c'est la fin du jeu.



Plateau principal (canvas)

Cartes

Sous Liste Joueur

## Cartes déjà triées par Age

1: Achat Possible ?()

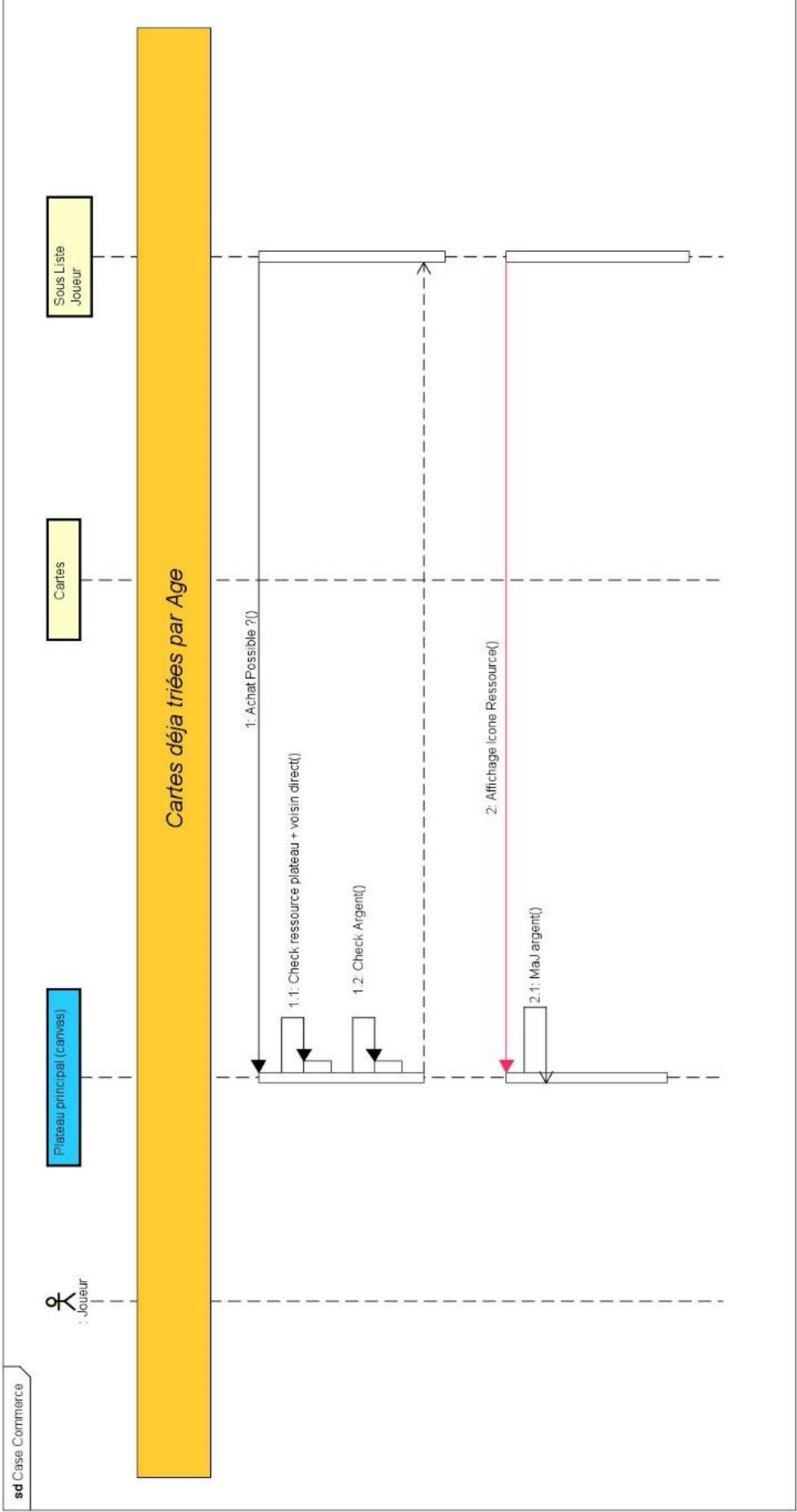
1.1: Check ressource plateau + voisin direct()

1.2: Check Argent()

2: Affichage Icône Ressource()

2.1: MaJ argent()

2.2: MaJ Point()







:Joueur

Plateau principal (canvas)

Cartes

Sous Liste  
Joueur*Cartes déjà triées par Age*

1: Achat Possible ?()

1.1: Check ressource plateau + voisin direct()

1.2: Check Argent()

2: Affichage Icone Ressource()

2.1: MaJ argent()

