# Projet Logiciel Transversal

**Maxime Marroufin, Quentin Chhean,**
**Abinaya Mathibala, Alban Benmouffek**

# Table des matières


**1 Présentation Générale** **1**
- 1.1 Archétype . . . 1
- 1.2 Règles du jeu . . . 1
  - 1.2.1 Zones de jeu . . . 1
  - 1.2.2 Conditions de défaite . . . 1
  - 1.2.3 Structure du tour . . . 1
  - 1.2.4 Déroulement de la partie . . . 2
- 1.3 Ressources . . . 2
  - 1.3.1 Textures . . . 2
  - 1.3.2 Cartes et graphiques propres à Magic . . . 2

**2 Description et conception des états** **4**
- 2.1 Description des états . . . 4
  - 2.1.1 Les zones . . . 4
  - 2.1.2 Tours de jeu . . . 5
- 2.2 Conception Logiciel . . . 5

**3 Rendu : Stratégie et Conception** **7**
- 3.1 Stratégie de rendu d'un état . . . 7
- 3.2 Conception logiciel . . . 7

**4 Règles de changement d'états et moteur de jeu** **9**
- 4.1 Conception logiciel . . . 9
- 4.2 Règles . . . 11

**5 Intelligence Artificielle** **13**
- 5.1 Stratégies . . . 13
- 5.2 Conception logiciel . . . 14

**6 Modularisation** **14**
- 6.1 Organisation des modules . . . 14
- 6.2 Conception logiciel . . . 15

**7 Glossaire** **15**

**8 Bibliographie** **15**

# 1 Présentation Générale

## 1.1 Archétype

Le jeu proposé que nous nommerons *MagiX*, se base sur Magic : The Gathering Online, la version numérique du jeu de carte à collectionner éponyme.

Dans *Magic : The Gathering*, les joueurs assemblent un deck contenant $60$ cartes et s'affrontent en utilisant celles-ci. *Magic : The Gathering* est séparé en deux étapes bien distinctes : la création de deck, et le jeu en lui-même. Nous nous focaliserons sur une seule de ces deux composantes : le jeu de carte.

## 1.2 Règles du jeu

### 1.2.1 Zones de jeu

Le Jeu est composé de sept zones :

**La Bibliothèque :** chaque joueur a la sienne, c'est la zone ou les joueurs placent leurs deck faces cachées. Lorsqu'un joueur pioche une carte, il pioche depuis cette zone.

**La Main :** propre à chaque joueur, c'est la zone depuis laquelle les joueurs peuvent jouer des cartes.

**Le Cimetiere :** propre à chaque joueur, c'est la zone ou les joueurs placent les cartes déffausées.

**L'exil :** propre à chaque joueur, c'est la zone où vont les cartes retirées du jeu par des effets de cartes.

**Le champ de bataille :** commun a tous les joueurs, c'est la zone ou l'on trouve des permanents qui sont soit des cartes soit des jetons crées par des effets de cartes.

**La Pile :** commune a tous les joueurs c'est la zone où vont les cartes jouées, effets de cartes et effets de permanents avant d'etre résolus.

**Zone de commandement :** commune a tous les joueurs, c'est une zone où vont des objets qui ne pourront pas être altérés par des effets de cartes.

Voir exemple de plateau : Figure 1.

### 1.2.2 Conditions de défaite

Un joueur perd la partie si au moins une des conditions suivantes est réalisée :
- Ses points de vie sont réduits a zéro
- Il pioche alors que sa bibliothèque est vide
- Un effet de carte lui fait perdre la partie
- Un effet de carte fait gagner un de ses adversaire

### 1.2.3 Structure du tour

Le tour est composé de cinq phases. Pendant les tours, les permanents peuvent être engagé ou dégagés.

**Phase de début de tour :** phase pendant laquelle le joueur actif pioche une carte et dégage les permanents qu'il contrôle.

**Phase principale :** c'est la phase durant laquelle le joueur actif peut jouer des sorts lents.

**Phase de combat :** c'est la phase durant laquelle le joueur actif va déclarer une ou plusieurs créatures comme attaquant un autre joueur. Le joueur défenseur pourra alors bloquer avec ses propres créatures.

**Phase principale post-combat :** le joueur actif peut de nouveau jouer des sorts lents.

**Phase de fin de tour :** c'est la phase durant laquelle le joueur actif va défausser des cartes s'il en a trop en main et où toutes les bléssures de combats sur les créatures sont retirées.

### 1.2.4 Déroulement de la partie

Chaque joueur commence la partie avec un total de vingt points de vie et une main de départ de sept cartes. La partie commence dans la phase principale du premier joueur qui joue. À la fin de son tour, le tour d'un de ses adversaires commence. Ce cycle continue jusqu'à ce qu'un joueur gagne la partie.

## 1.3 Ressources

### 1.3.1 Textures

Nous avons mis beaucoup de textures dans le dossier `/res/textures`. Ces images sont libres de droit, à condition de citer la source : Toptal Subtle Patterns.

Nous utiliserons ces textures par exemple pour les couleurs de fond, les contours, les zones...

### 1.3.2 Cartes et graphiques propres à Magic

Les graphiques de *Magic : The Gathering* ne sont pas libres de droit, on ne peut donc pas les mettre dans le dépôt Git. À la place, nous utiliserons l'API de Scryfall pour télécharger les images et les placer dans la mémoire disque des utilisaturs.


FIGURE 1 – Plateau du jeu

# 2 Description et conception des états

## 2.1 Description des états

Le jeu dépend de plusieurs éléments : les joueurs, les zones, les objets contenus dans les zones et les tours de jeu.

### 2.1.1 Les zones

Le jeu Magic The Gathering est composé d'un ensemble de surfaces sur le plateau où sont les différentes cartes. On peut les qualifier ici d'éléments fixes. Ces « zones » sont réparties de la façon suivante :

**« Library » (FR : « Bibliothèque ») :** Chaque joueur en possede une, elle contient plusieurs cartes, cette zone ne peut être consultée par les joueurs et son contenu est ordonné. Lorsque qu'un joueur pioche, l'élement du haut de cette zone est déplacé vers la main. Au debut de la partie celle-ci est mélangé. Lorsqu'un effet mélange cette zone, l'ordre des cartes qu'elle contient est randomisé.

**« Hand » (FR : « Main ») :** Chaque joueur en possede une. Contient plusieurs cartes, les cartes piochées sont déplacées dans cette zone, de celle-ci un joueur peut jouer des cartes "terrain" et lancer des cartes "sort", le contenu de cette zone est visible par son propriétaire uniquement. Au debut de la partie 7 cartes sont piochés. Lorsqu'un joueur se défausse d'une carte, il choisi une carte et la déplace vers le cimetière. A la fin du tour d'un joueur si son nombre de cartes en main est supérieur à sa limite (par défaut 7) il doit se défausser de la difference.

**« Graveyard » (FR : « Cimetière ») :** Chaque joueur en possede un. Cette liste conserve les cartes défaussées du joueur. Elle peut être consultée par tous les joueurs. Un objet peut être envoyé au cimetière pour plusieurs raisons : si une carte "sort" se résout sans engendrer de permanent, si un permanent est détruit ou si des cartes sont directement déplacés depuis une zone vers le cimetière.

**« Exile » (FR : « Exil ») :** Un sort ou une capacité peut « exiler » une carte. Cette dernière est alors ajoutée dans cette liste. Cette carte y reste pour toujours à moins que l'effet qui l'y a mise soit capable de la récupérer. Cette zone est visible par tous les joueurs.

**« Stack » (FR : « Pile »)** La zone « Stack » est partagée par les joueurs. C'est ici qu'attendent les sorts et les capacités avant de se résoudre. Les joueurs peuvent alors décider de ne pas lancer de nouveaux sorts ou de ne pas activer des capacités. Quand c'est le cas, c'est le dernier sort ou la dernière capacité à être arrivé sur la pile qui se résout en premier, et les joueurs ont une nouvelle occasion de lancer des sorts et d'activer des capacités et ce jusqu'à ce que la pile soit vide.

**« Battlefield » (FR : « Champ de Bataille »)** C'est la zone qui contient tous les permanents.

**« Command » (FR : « Zone de commandement »)** Correspond a la zone de commandement, une zone où vont des objets qui agissent sur le reste du jeu mais pas l'inverse.

### 2.1.2 Tours de jeu

Le jeu peut se jouer en plusieurs tours. Chaque tour est composé d'une séquence de plusieurs phases. Ces dernières sont eux même composées d'étapes.

Les tours s'enchaînent jusqu'à ce qu'un qu'un joueur ne gagne la partie ou que tous les joueurs la perdent.

## 2.2 Conception Logiciel

Le diagramme des classes pour les états est présenté en Figure

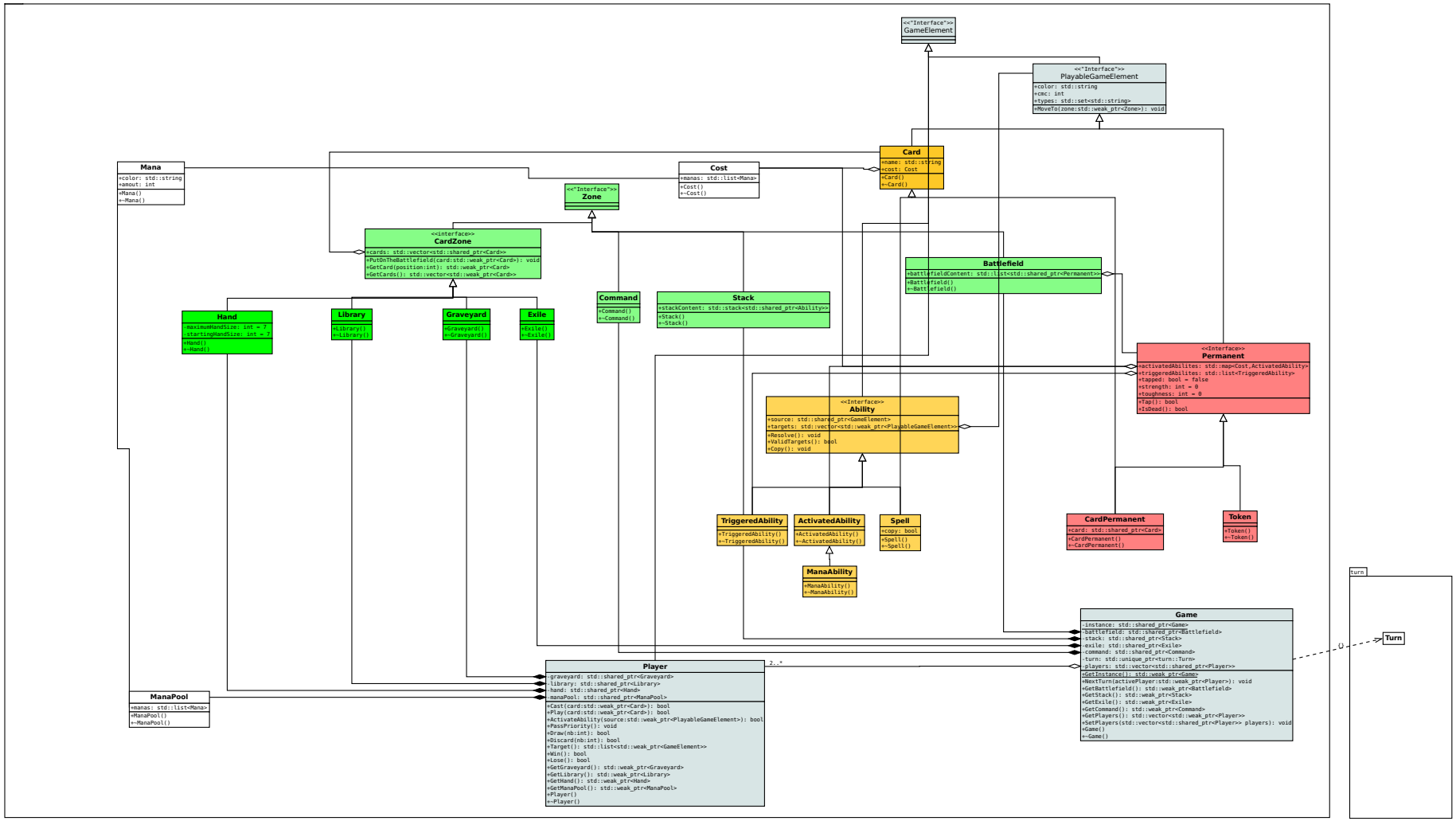


FIGURE 2 – Diagramme des classes d'état.

# 3 Rendu : Stratégie et Conception

## 3.1 Stratégie de rendu d'un état

Notre stratégie de rendu d'un état se base sur l'utilisation de l'interface SFML qui s'appuie sur OpenGL afin de générer un rendu en 2D..

Notre affichage sera distribué entre différents objets, objets qui ont pour tache d'afficher une partie du state , ces objets seront appelés Renderers et seront gérés par le RenderingManager qui comme le Game ( dans state) est un singleton.

L'idée est que les Renderers vont être mis à jour lorsque les éléments de state changent ( cf : https ://en.wikipedia.org/wiki/Observer_pattern ) et une fois mis à jour vont appeler le RenderingManager pour que celui ci redessine le contenu de la fenêtre.

## 3.2 Conception logiciel

Notre RenderingManager sera le propriétaire de notre sf : :RenderWindow sur la laquelle les Renderer seront dessinés.

Notre RenderingManager va créer dans son constructeur les Renderers pour afficher un state et vas ensuite stocker des références vers ceux ci pour pouvoir les dessiner au moment venu.

Nos Renderers héritent de sf : :Drawable et redéfinissent la méthode draw pour pouvoir être dessiné sur une sf : :RenderWindow, certains Renderers n'ont pas besoin de redéfinir cette méthode,dans ce cas ils héritent directement de sf : :Sprite.

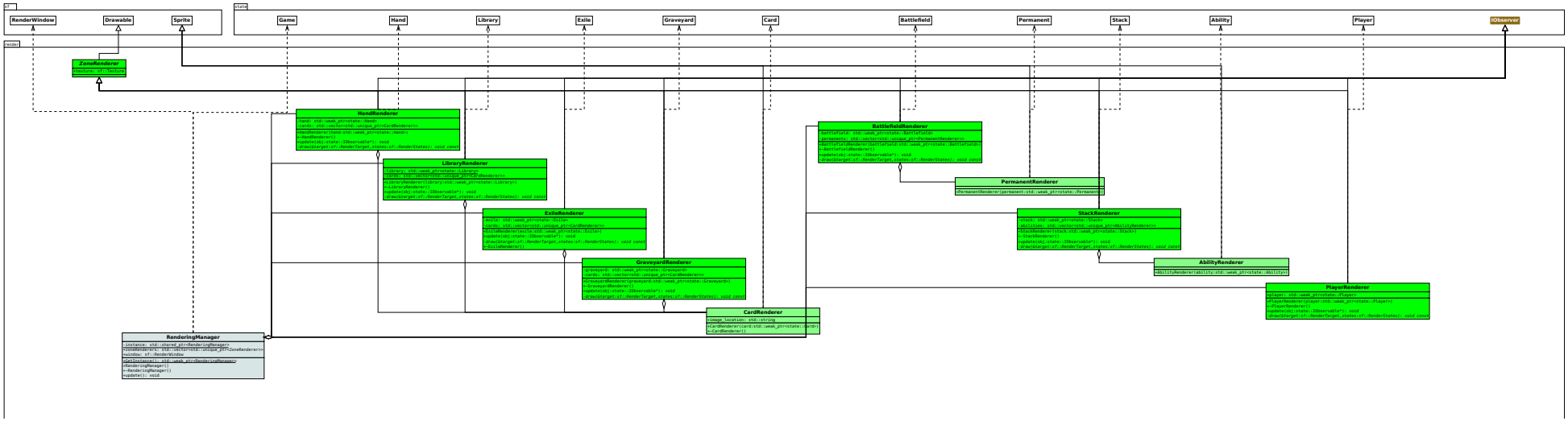


FIGURE 3 – Diagramme des classes de rendu.

# 4 Règles de changement d'états et moteur de jeu

## 4.1 Conception logiciel

Notre conception se découpe en 3 classes principales : (voir figure 4 pour les détails)

**Engine :** gère l'état du jeu (changements d'état, interrogations des instances de Actor) : vérifications et exécution des commandes

**Actor :** génère des commandes à la demande de l'Engine. Cette classe représente donc une IA ou un joueur.

**Command :** décrit une commande émise par un Actor, peut être exécuté par l'Engine pour changer l'état du jeu. Cette classe contient les données qui transiteront sur le réseau.

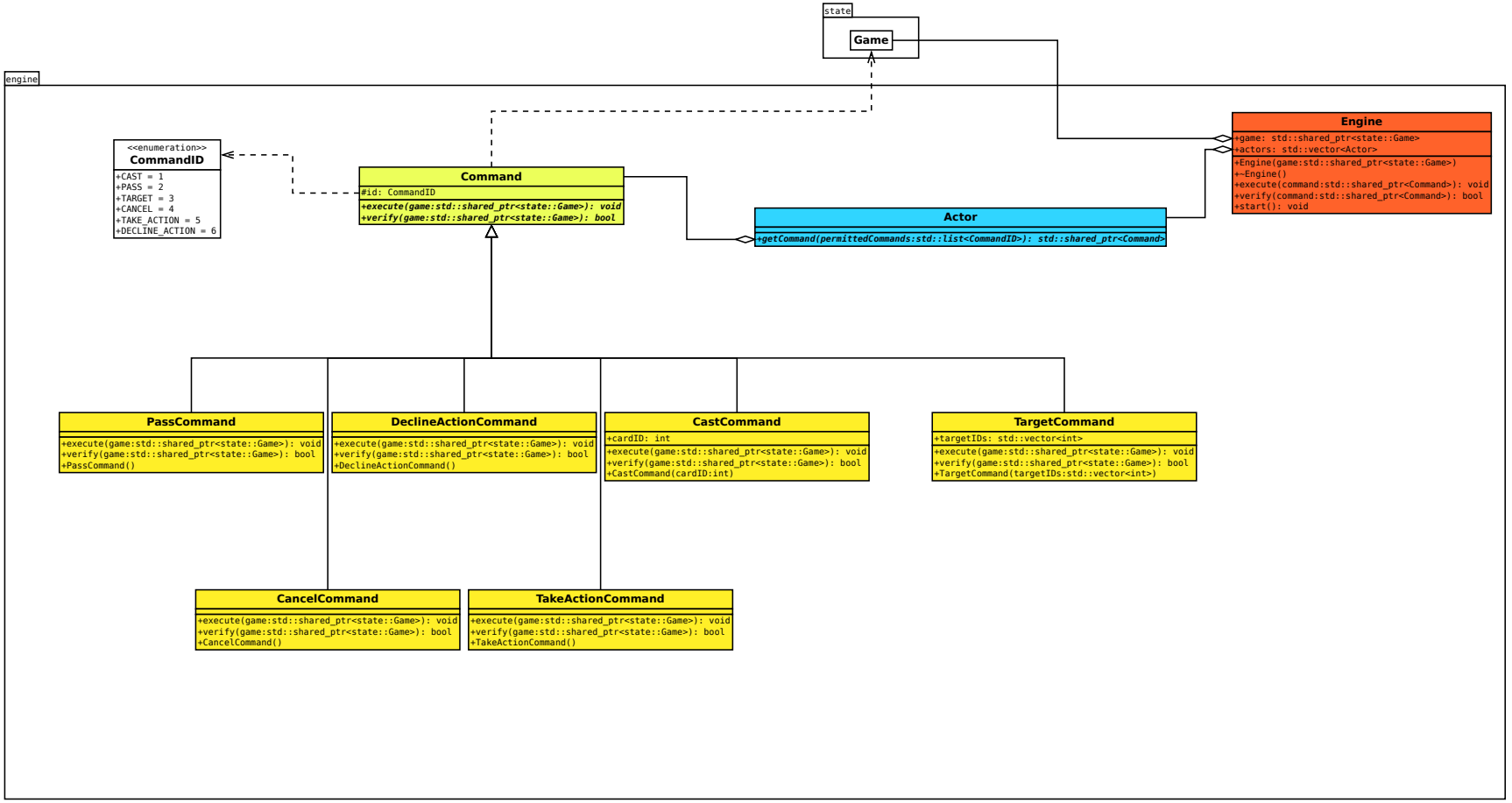


FIGURE 4 – Diagramme des classes du moteur du jeu.

## 4.2 Règles

Au début du jeu, la méthode `Engine::start()` est appelée pour démarrer l'Engine.

L'Engine interroge les Actor tour par tour (en respectant les priorités) en appelant `Actor::GetCommand()`. Les actors savent quels types de commande ils peuvent faire (grâce au CommandID). `Actor::GetCommand()` renvoie une instance de Command qui décrit une action, qui modifiera l'état du jeu.

Avant d'exécuter une commande, l'Engine la vérifie (avec `Engine::verify()`). Si la commande est valide, elle est exécutée grace à `Engine::execute()`.

Déroulement d'une étape de jeu (Les tours de jeu sont composés de phases qui sont elles composées d'étapes) Les détails du déroulement d'un step sont dans l'algorigramme figure 5.

Détail de l'algorigramme :

Dans cet algorigramme, Actor1 est le joueur actif et Actor2 le joueur réactif.

Au début d'une étape, une logique est appliquée ,cette logique dépend de la phase en question, pour une phase de pioche , le joueur actif piocherait une carte par exemple.

Ensuite Le joueur actif est interrogé par l'engine, tant qu'il n'émmet pas de commande PASS , il garde la priorité, une fois qu'il passe cette priorité c'est le joueur réactif qui prend le relais, ensuite si le joueur réactif a émit plus d'une action valide, autrement dit s'il n'a pas fait que passer la priorité, le joueur actif reprend la priorité et peut à nouveau ajouter des effets sur la pile.

Une fois que les deux joueurs ont passé la priorité sans ajouter d'effets sur la pile, si la pile est vide alors l'étape est terminé , sinon on résout l'effet du haut de la pile et on redonne la priorité au joueur actif.

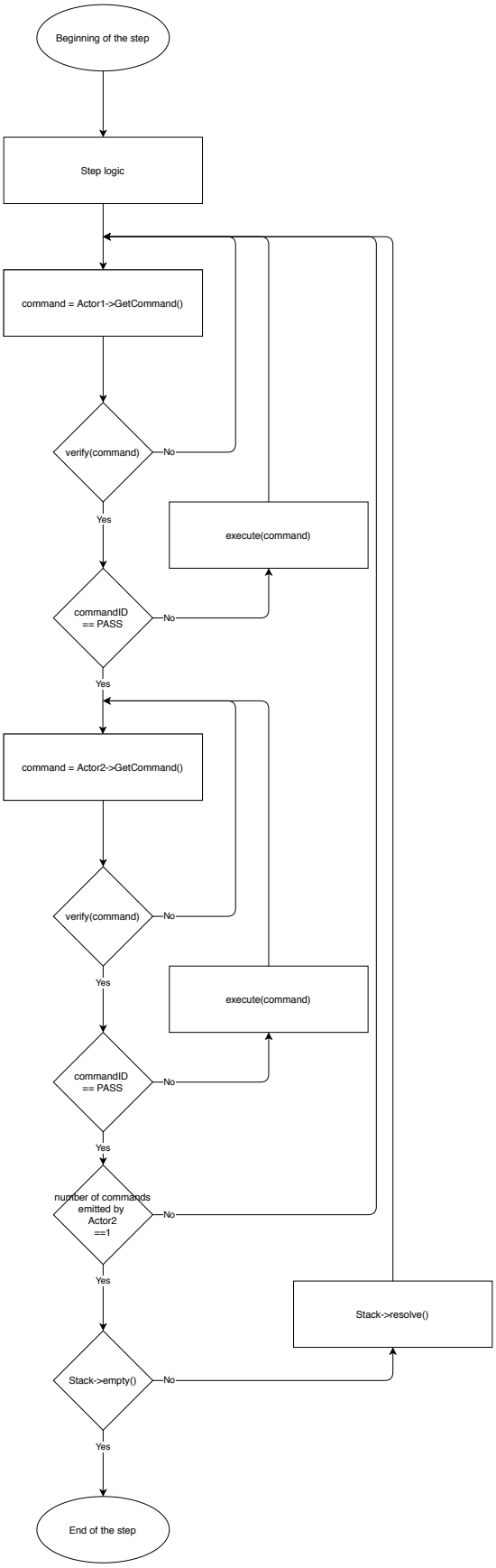


FIGURE 5 – Déroulement d'un Step

# 5 Intelligence Artificielle

## 5.1 Stratégies

### 5.2 Conception logiciel

## 6 Modularisation

### 6.1 Organisation des modules

### 6.2 Conception logiciel

# 7 Glossaire

**Joueur actif :** joueur dont c'est le tour

**Engage/dégagé :** un permenant est dit *engagé* lorsque la carte qui le représente est tournée de 90 degrés vers la droite, il est dit *dégagé* lorsqu'il ne l'est pas

**Deck :** ensemble de cartes qu'un joueur utilise pendant la partie

# 8 Bibliographie

**Règles du jeu officielles :** https ://media.wizards.com/2014/docs/FR_M15_QckStrtBklt_LR_Crop.pdf