

Projet de 3ème année : Jeu de cartes Yu-Gi-Oh

Réalisé par :
Romain CORNIC
Mateo CORTES DUARTE
Fahd RAFIK
Zeïck SOULE

Projet Logiciel Transversal
Option IS - ENSEA Promo 2024



14 octobre 2023

Table des matières

| | |
|---------------------------------------------------------|-----------|
| 1 Présentation Générale | 2 |
| 1.1 Archétype | 2 |
| 1.2 Règles du jeu | 2 |
| 1.3 Ressources | 4 |
| 2 Description et conception des états | 4 |
| 2.1 Description des états | 4 |
| 2.1.1 Etat de jeu selon les phases | 4 |
| 2.1.2 Etat général | 6 |
| 2.2 Conception logicielle | 7 |
| 3 Rendu : Stratégie et conception | 8 |
| 4 Règles de changement d'états et moteur de jeu | 8 |
| 4.1 Changements extérieurs | 8 |
| 4.2 Changement autonomes | 9 |
| 4.3 Conception logicielle | 9 |
| 5 IA | 10 |
| 5.1 stratégie | 10 |
| 5.1.1 Intelligence minimale | 10 |
| 5.1.2 Intelligence basée sur des heuristiques | 12 |
| 5.2 Conception logicielle : | 12 |
| 5.2.1 Intelligence minimale : | 12 |

1 Présentation Générale

1.1 Archétype

Ce projet a pour objectif de créer une version simplifiée du jeu de cartes Yu-Gi-Oh, un jeu de cartes de renommée mondiale qui a captivé des générations de joueurs par son mélange unique de stratégie, de magie, et de créatures mythiques. Dans Yu-Gi-Oh, deux joueurs s'affrontent sur un plateau virtuel, utilisant un jeu de cartes élaboré pour invoquer des monstres, lancer des sorts, et activer des pièges, tout cela dans le but de réduire les points de vie de leur adversaire à zéro.

La version simplifiée de Yu-Gi-Oh que nous développons conserve l'essence compétitive du jeu tout en éliminant certaines complexités. L'une des caractéristiques clés de cette adaptation est la suppression de certains types de monstres et de mouvements tactiques du jeu original. Cette simplification permet aux nouveaux joueurs de s'initier au jeu plus rapidement et sans être submergés par les règles et les mécaniques parfois complexes de la version complète.

Les joueurs pourront toujours profiter de l'excitation de la stratégie, de la construction de deck, et des duels épiques, mais de manière plus accessible. Cette version simplifiée est idéale pour ceux qui souhaitent découvrir l'univers riche de Yu-Gi-Oh sans nécessiter une connaissance préalable approfondie. Cependant, elle reste assez fidèle à l'expérience originale pour que les vétérans du jeu puissent y trouver du plaisir.

1.2 Règles du jeu

Le but du jeu est de réduire les points de vie de l'adversaire à 0 ou de faire en sorte qu'il ne puisse plus piocher de carte lorsqu'il doit le faire. Pour réduire les points de vie de l'adversaire, les joueurs disposent de cartes Monstre qui possèdent des points d'attaque ainsi que des points de défense. Ces cartes peuvent être utilisées en position d'attaque ainsi qu'en position de défense. Si un monstre en position d'attaque attaque un autre monstre en position d'attaque, celui qui possède le moins de point d'attaque est détruit, et la différence entre les points d'attaque des deux monstres est soustrait aux points de vie du joueur dont le monstre a perdu. Si un monstre en position d'attaque attaque un monstre en position de défense, on compare les points d'attaque de l'assaillant aux points de défense du défenseur. Dans le cas où l'assaillant remporte la confrontation, le monstre en position de défense est détruit mais le joueur ne subit aucun dégats. Dans le cas contraire, aucune carte n'est détruite, mais le joueur subit des dégâts égaux à la différence entre les caractéristiques d'attaque et de défense des monstres.

Les cartes détruites sont déposées dans une pile de cartes appelée le cimetière, ou elles ne pourront plus être utilisées (sauf dans certains cas précis).

En plus des cartes Monstre, nous avons également des cartes Magie et des cartes Piège. **Les cartes Magie** ne peuvent être activées que pendant la Phase Principale, et apportent différents effets tel que la destruction d'autres cartes ou le renforcement de monstres. Parmis les cartes Magie, il y a :

- **Cartes Magie Normale** : Elles sont activées en les plaçant face visible sur le terrain, puis vous résolvez l'effet écrit sur la carte. Une fois l'effet terminé, envoyez la carte au cimetière.
- **Cartes Magie Continue** : ces cartes restent sur le terrain une fois qu'elles sont activées. Il y a un risque que l'adversaire la retire du terrain avant que vous ne puissiez bénéficier de l'effet.
- **Cartes Magie d'Équipement** : donnent un effet supplémentaire à 1 monstre face visible de votre choix (le vôtre ou celui de votre adversaire, selon la carte). Ils restent sur le terrain après avoir été activés. Si le monstre équipé est détruit, retourné face cachée ou retiré du terrain, ses cartes d'équipement sont détruites.

- **Cartes Magie de Terrain :** Ces cartes sont placées dans votre zone de champ et y restent. Chaque joueur peut avoir 1 carte Magie de champ sur son propre côté du terrain. Pour utiliser un autre sort de terrain, envoyez le précédent au cimetière. De nombreux effets de sorts de terrain s'appliquent aux DEUX joueurs.

Les cartes Piège permettent également d'activer différents effets. La grande différence est qu'elles peuvent être activées pendant le tour de l'adversaire.

- **Cartes Piège Normal :** Pour l'activer, vous devez d'abord la placer sur le terrain. Vous ne pouvez pas activer un Piège pendant le tour où vous le placez, mais vous pouvez l'activer à n'importe quel moment par la suite - à partir du début du tour suivant. Une fois l'effet résolu, il sera envoyé au cimetière.
- **Cartes Piège Continu :** Les Cartes Piège Continues restent sur le terrain une fois activées et leurs effets continuent tant qu'elles sont face visible sur le terrain. Les Cartes Piège Continues peuvent avoir des effets qui limitent les options de votre adversaire, ou qui endommagent lentement les Points de Vie de votre adversaire.
- **Cartes Contre-Piège :** Ces cartes sont activées en réponse à l'activation d'autres cartes, et peuvent avoir des capacités telles que l'annulation des effets de ces cartes, ce qui est efficace contre les cartes de sorts normaux, qui sont autrement difficiles à arrêter. Cependant, de nombreuses cartes de contre-pièges nécessitent un coût pour être activées.

Afin d'utiliser correctement les cartes mentionnées ci-dessus, elles doivent être placées sur le plateau de jeu, chaque joueur ayant son propre espace de jeu pour placer ses cartes. Chaque type de carte dispose d'un espace spécifique sur le plateau, dont la répartition est illustrée dans l'image suivante.

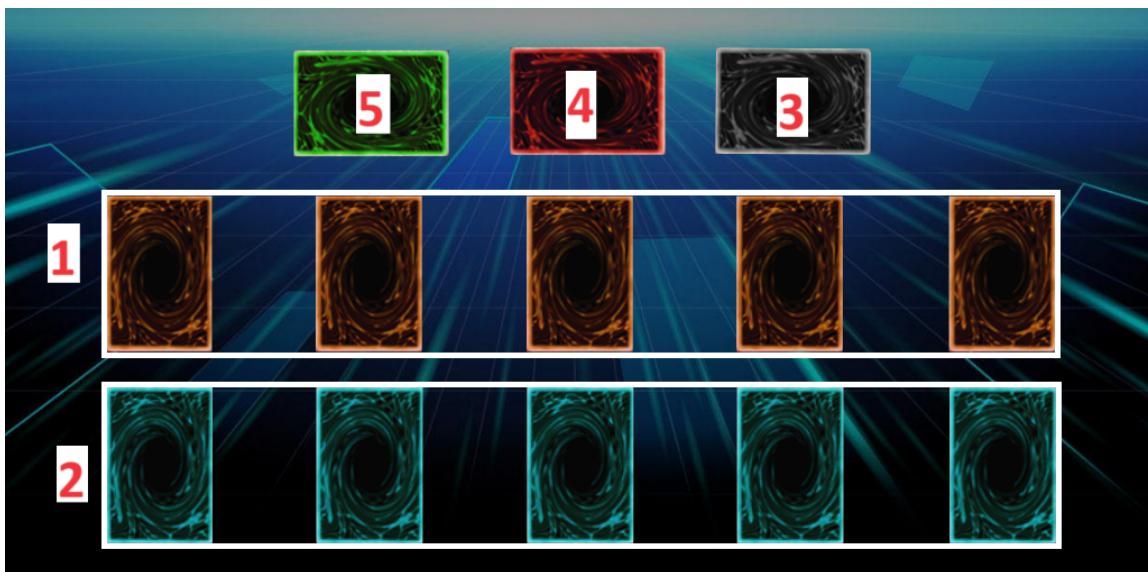


FIGURE 1 – Espace de jeu par joueur

1. **Zone Monstre :** Il y a trois façons de les placer : Position d'Attaque face recto, Position de Défense face recto et Position de Défense face verso. Placez la carte verticalement pour indiquer une Position d'Attaque, et horizontalement pour une Position de Défense

2. **Zone Magie & Piège :** Vous pouvez les placer en mode recto ou verso pour les activer. Si une Carte Magie est déjà dans cette zone lors de son activation, vous ne pouvez pas jouer de Carte Magie supplémentaire tant que les 5 emplacements sont occupés.
3. **Cimetière :** Les Cartes Monstre détruites, ainsi que les Cartes Magie et Piège utilisées, vont ici face recto. Le contenu du Cimetière est public, accessible à votre adversaire à tout moment. L'ordre des cartes dans le Cimetière reste inchangé.
4. **Deck :** Votre Deck est placé face verso dans cet emplacement. Les joueurs y piochent les cartes et les ajoutent à leur main. Si une carte demande de révéler des cartes de votre Deck ou de regarder son contenu, mélangez le Deck et replacez-le à cet endroit après avoir résolu l'effet.
5. **Zone Terrain :** Des Cartes Magie de Terrain sont jouées dans cet emplacement. Chaque joueur peut en avoir 1 à la fois. Pour utiliser une autre, envoyez la précédente au Cimetière. Elles ne comptent pas dans la limite de 5 cartes de votre Zone Magie & Piège.

Une fois le jeu ouvert, les éléments qui seront affichés pendant le jeu sont présentés dans la figure ci-dessous [2](#).

1. **Boîte d'informations sur le joueur :** Cette case indique les points de vie restants du joueur.
2. **Nombre de cartes restantes pour le deck :** Le joueur aura des cartes en main et d'autres cartes dans lesquelles il pourra piocher. Cette case indique combien de cartes, il reste en réserve pour piocher.
3. **Cartes en main**
4. **Nombre de points d'attaque/défense :** Ce carré est rouge lorsque des points sont attaqués et vert lorsque des points sont défendus.
5. **Carte en attaque ou défense :** La carte est verticale lorsqu'elle est en attaque et horizontale lorsqu'elle est en défense.

1.3 Ressources

Les ressources pour ce projet constituent l'ensemble des cartes Yu-Gi-Oh de type Monstre, Piège et Magie, dont des exemples se trouvent en figure 2, 3 et 4. Il y a également un plateau sur lequel le joueur pourra déposer ses cartes.

2 Description et conception des états

2.1 Description des états

2.1.1 Etat de jeu selon les phases

L'état de jeu dépend de la phase dans laquelle les joueurs se trouvent. Une partie se divise en plusieurs phases :

1. **Phase de pioche :** La phase de pioche correspond à la phase durant laquelle le joueur (dont c'est le tour actuellement) extrait des cartes de son deck. Cette phase est automatique, afin de faire en sorte que le joueur ait 6 cartes au début de son tour. Le joueur voit les cartes s'ajouter aux cartes déjà présentes dans sa main. Deux boutons lui permettent de passer à la phase suivante, ou d'activer une carte Magie ou Piège.



FIGURE 2 – Espace de jeu

2. **Phase d'attente** : La phase d'attente est une phase intermédiaire qui permet de donner des informations au joueur. Cela se fera à travers une fenêtre pop-up qui informera le joueur de la résolution ou de l'action des effets en cours. Par exemple, une carte Magie qui empoisonne un joueur en lui retirant des points de vie s'activera dans cette phase. Trois boutons permettent au joueur de consulter les effets en cours, de passer à la phase suivante ou d'activer une carte Magie ou Piège.
3. **Première Phase principale** : La première phase principale permet au joueur de déposer des cartes et/ou de modifier la position des cartes sur son plateau. Il peut retourner des cartes se trouvant en phase cachée, changer la position d'une carte monstre (position d'attaque ou de défense) et activer une carte Magie. Le joueur aperçoit son plateau ainsi que les cartes dans sa main. En cliquant sur



(a) Carte Monstre : "Red Eyes Black Dragon"



(b) Carte Magie : "Monster Reborn"



(c) Carte Piège : "Mirror Force"



(d) Plateau de jeu Yu-Gi-Oh

FIGURE 3 – Éléments de jeu

une carte qu'il possède, il peut consulter les caractéristiques de la carte (nom, description, effet, points d'attaque, points de défense...), et déposer la carte en choisissant dans quelle position (si elle se trouve dans sa main).

4. Phase de combat : La phase de combat correspond à la phase durant laquelle les joueurs peuvent infliger des dégâts à l'adversaire. Cette phase n'est pas disponible pour le premier tour du premier jour. Elle se décline en 4 parties :

- A *Etape de début* : La première étape de la phase de combat se déclenche quand le joueur clique sur le bouton pour débuter la phase de combat. Elle affiche seulement un message pour confirmer le début de la bataille.
- B *Etape de bataille* : Lors de la deuxième étape, le joueur peut choisir un monstre avec lequel il souhaite attaquer. Il peut ensuite choisir le monstre qu'il souhaite attaquer. Dans cette phase, le joueur peut attaquer une fois avec tous les monstres qu'il possède en position d'attaque. Il n'est néanmoins pas obligé d'attaquer avec tous les monstres qui sont sur son plateau.
- C *Etape de dommage* : La troisième étape correspond au calcul des dommages. Elle permet de comparer le monstre assaillant et le monstre défenseur, et permet d'afficher les dommages reçus. Après cette étape, le joueur revient à l'étape précédente pour attaquer à nouveau. Ce schéma peut se répéter autant de fois qu'il y a de monstres en position d'attaque sur le terrain.
- D *Etape de fin* : Cette étape se lance lorsque le joueur a attaqué avec tous ses monstres, ou s'il souhaite s'arrêter et passer à la prochaine phase. Elle décrète la fin de la phase de combat.
- E *En fin de partie* : le joueur doit choisir le moment où il décide de terminer son tour, car il n'y a pas d'heure fixe pour terminer le tour.

- 5. Deuxième Phase principale :** La deuxième phase principale permet de se préparer au tour adverse. Il est possible de faire la même chose qu'au premier tour, à condition que l'action n'a pas déjà été faite au à la première phase principale.
- 6. Phase de fin :** Cette phase marque la fin de la phase de fin. Elle permet de résoudre certains effets. Il est encore possible pendant cette phase d'activer des cartes Piège.

2.1.2 Etat général

Certains éléments de l'interface sont fixes et correspondent à l'environnement de jeu que peuvent voir les deux joueurs, peu importe la phase de jeu en cours.

Plateau de jeu : Au nombre de deux, les plateaux permettent aux joueurs de déposer leur cartes et d'organiser leur jeu. Ils sont composés de plusieurs emplacements fixes portant un symbolisme précis :

- Le cimetière (Une carte retournée et grise)
- La zone Monstre (5 cartes retournées et marron)
- La zone Magie et Piège (5 cartes retournées et bleues)
- La zone Deck (Une carte retournée et rouge)

Interface joueur : L'interface des joueurs permet d'afficher le nom et le nombre de points de vies restants pour chaque joueur. Il permet également de visualiser les cartes en main et d'accéder au menu du jeu.

2.2 Conception logicielle

Le diagramme de classe, se trouvant en figure 9, permet de mettre en évidence les classes et les relations entre elles dans ce projet. Nous les détaillons ci-dessous :

Les Classes Cartes : Les cartes constituent la classe la plus importante du jeu. La classe Cartes est une classe abstraite implementée par les classes Carte Piège, Carte Magie et Cartes Monstre. Cette classe possède les attributs communs à tous les types de cartes, tel que le nom, l'ID, la description, l'état et le type. Son ID permet d'identifier une carte. Sa description et son nom ont pour but de caractériser le monstre et ses effets de manière plus accessible et ludique. *L'attribut état permettra de savoir dans quel état se trouve la carte (est-elle détruite ? en main ? sur le plateau ?).*

Les cartes Monstre ont un niveau (qui permettra d'implémenter certaines améliorations futures), des points d'attaque et de défense qui permettent d'affliger des dégâts au joueur adversaire et de combattre contre d'autres monstres, un attribut effet, ainsi que deux méthodes qui leur permettront d'attaquer et d'être affectées par un sort.

Les cartes Magie possèdent un attribut effet de type spellEffect et une méthode isOn qui permet de savoir si l'effet de cette carte est activée ou pas. Le type énuméré spellEffect permet d'implémenter différentes actions dans la méthode activerEffet en fonction du type d'effet de la carte.

Les cartes Piège sont similaires aux cartes Magie, si ce n'est qu'elles peuvent être activées à tout moment durant le tour de l'adversaire. Elles possèdent un attribut effet de type trapEffect. Comme pour les cartes Magie, le type énuméré trapEffect permet d'implémenter différentes actions dans la méthode activerEffet en fonction du type d'effet de la carte.

La classe Boards est la classe grâce à laquelle se déroule les étapes de la partie. Suite au lancement d'une carte, celle-ci s'ajoutent aux cartes appartenant au plateau. Il est possible d'attaquer les

La classe States est la classe grâce à laquelle se déroulent les étapes de la partie. Elle permet d'effectuer la gestion du jeu. C'est dans celle-ci que s'effectuent les actions d'incrémantation des tours, d'ajout et de suppressions des joueurs.

La classe Players est une classe qui permet de gérer les méthodes et les attributs de chaque joueur. Chaque joueur possède un deck, un plateau, des points de vie, et un status. Status est de type GameStatus qui est un type énumération, qui peut prendre trois valeurs : En jeu, gagné et perdu. Cette variable permet donc de savoir qui est le vainqueur et qui est le perdant. Le joueur peut être soit un humain, soit une IA. On crée donc un type énumération TypePlayer.

La classe Calculation est une classe qui permet de calculer les dégâts réalisées par les joueurs durant la phase de combat.

La classe Deck permet de référencer toutes les cartes d'un joueur. Elle permet également d'accéder aux cartes dans les mains des joueurs.

3 Rendu : Stratégie et conception

Il s'agit d'un jeu en 2D en un contre un. Par conséquent, l'image du jeu sera divisée en deux, une moitié pour chaque joueur. Chacun de ces éléments sera affiché par le biais d'un système de couches. Les premières couches contiennent les éléments statiques tels que l'image de fond, ainsi que les cartes face cachée qui marquent l'espace alloué à chaque type de carte.[??](#)

Ces couches seront suivies d'objets animés dont l'apparence et la position changeront au cours du jeu. Sur ces calques se trouvent les différents types de cartes utilisés dans le jeu.

Sur les dernières couches se trouveront les blocs d'information comme ceux montrés dans l'image [2](#). Ceux-ci permettent de connaître le nom des joueurs, le nombre de cartes restantes, les points d'attaque, etc.

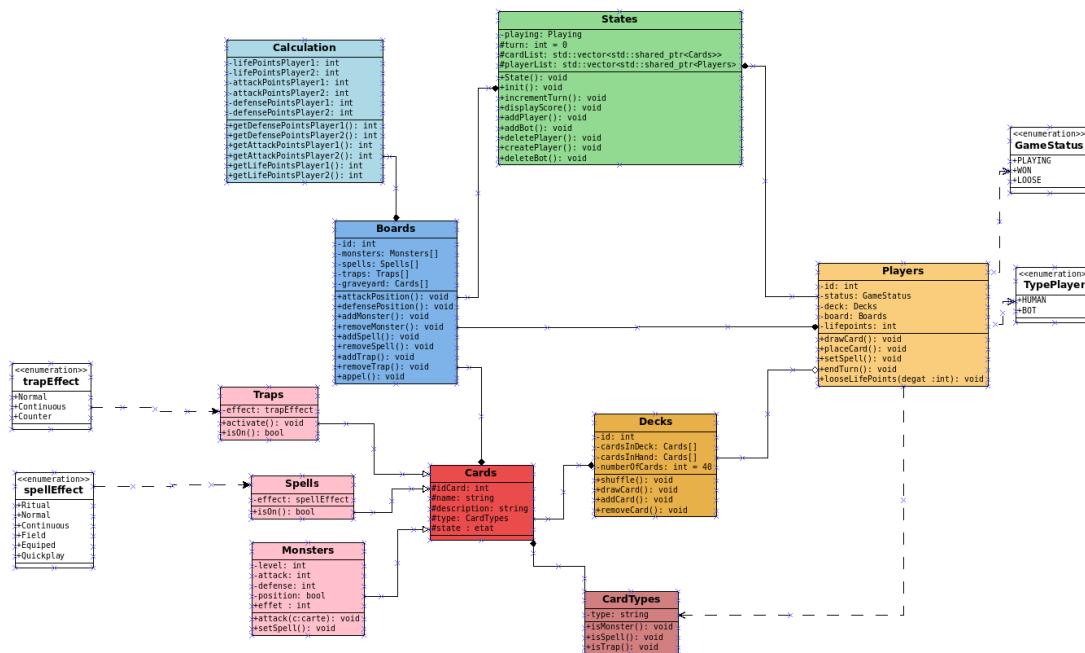


FIGURE 4 – Diagramme de classe du projet"

4 Règles de changement d'états et moteur de jeu

4.1 Changements extérieurs

À l'aide de la souris, le joueur choisit l'action qu'il souhaite effectuer pendant le match. Chaque tour de jeu se déroule en trois étapes majeures : la modification de la Main d'un joueur, la modification du Board, et une dernière étape qui nous permet d'attaquer un monstre ou un joueur. Les changements de ces états se font en réponse aux actions des joueurs à travers les commandes et le rendu. Par exemple, en fonction de la position du clic, le rendu enverra une instruction à l'engine qui nous permettra de choisir la carte à placer. Les actions possibles au cours d'un tour de jeu comprennent :

- Tirer une carte de la pioche : En cliquant sur la pioche pendant la phase principale, le joueur peut tirer une carte.
- Placer une carte dans le champ : En cliquant sur la carte souhaitée dans la main, le joueur peut décider de l'activer ou non. Si la carte n'est pas activée, elle apparaîtra dans le champ face cachée. Si la carte est un monstre, sa position sera horizontale au lieu de verticale. Lorsqu'elle est activée, elle restera face visible.
- Activer une carte : Lorsque la carte monstre est placée sur le terrain en mode défense, elle peut être activée en mode attaque simplement en cliquant sur la carte. Pour les cartes pièges, elles peuvent être activées en cliquant sur la carte, même si ce n'est pas le tour du joueur. Quand la carte est activée, elle se retourne pour révéler son action.
- Attaquer un autre monstre ou un joueur : Pendant la phase d'attaque, le joueur peut choisir un monstre pour attaquer un autre monstre. S'il n'y a pas de monstre sur le terrain du rival, le joueur peut attaquer directement les points de vie du rival.

4.2 Changement autonomes

En plus des actions générées par le joueur, il y a aussi des réactions qui seront générées par le moteur du jeu. Ces réactions sont considérées comme des changements automatiques gérés par le render en fonction de l'état du jeu :

- Le changement de couleur et de valeur sous les cartes de monstre lors du choix entre le mode attaque et le mode défense.
- Le changement de la valeur des points de vie du joueur.
- L'envoi d'une carte au cimetière après une attaque, dans le cas où les points de vie de l'adversaire sont égaux ou supérieurs à ceux de la carte qui a reçu l'attaque.
- Changement du nombre indiqué sur la pioche.

4.3 Conception logicielle

Le diagramme de classe du moteur de jeu (Figure 5) symbolise la traduction des actions d'un joueur en commandes et méthodes à appeler.

La classe Command a pour rôle de représenter les commandes. Ses attributs, à savoir player, opponent, state et Board, facilitent la récupération des informations de la partie en cours de manière plus claire et efficace que si l'on se servait uniquement du state. Ces attributs sont utilisés comme arguments pour indiquer un changement de phase, un déplacement des cartes sur le board, une attaque, le choix d'un monstre, ou encore une perte de vie.

Les classes Attack, ModifyBoard et ModifyHand héritent de la classe Command et résument les actions d'un joueur énumérées précédemment. C'est au sein de chacune de ces classes que l'on appelle la fonction appropriée, déjà implémentée dans notre State, en fonction du cas de figure ; selon la phase et les points de vie disponibles, le status du Board : des monstres, sorts et pièges disponibles. Par ailleurs, les variables des ces différentes classes sont en protected et sont modifiées directement au sein de la classe engine.

La classe engine est la classe "centrale" puisqu'elle contient le vecteur des commandes, qui s'exécuteront dans l'ordre et en fonction des choix des joueurs. Lorsqu'un tour de jeu démarre, c'est-à-dire lorsqu'on a appelé la méthode updateState() après un laps de temps suffisant, le moteur appelle la méthode execute(). Il choisit quelle commande, décidée par le joueur, doit être exécutée. Ensuite, les commandes agissent

directement sur les paramètres du state; elles incrémentent le nombre de tours et mettent à jour les statistiques des personnages et du plateau. De plus, la plupart des variables sont en private et sont fournies directement aux commandes pour éviter tout problème de triche en forçant des entrées à partir du state ou du rendu.

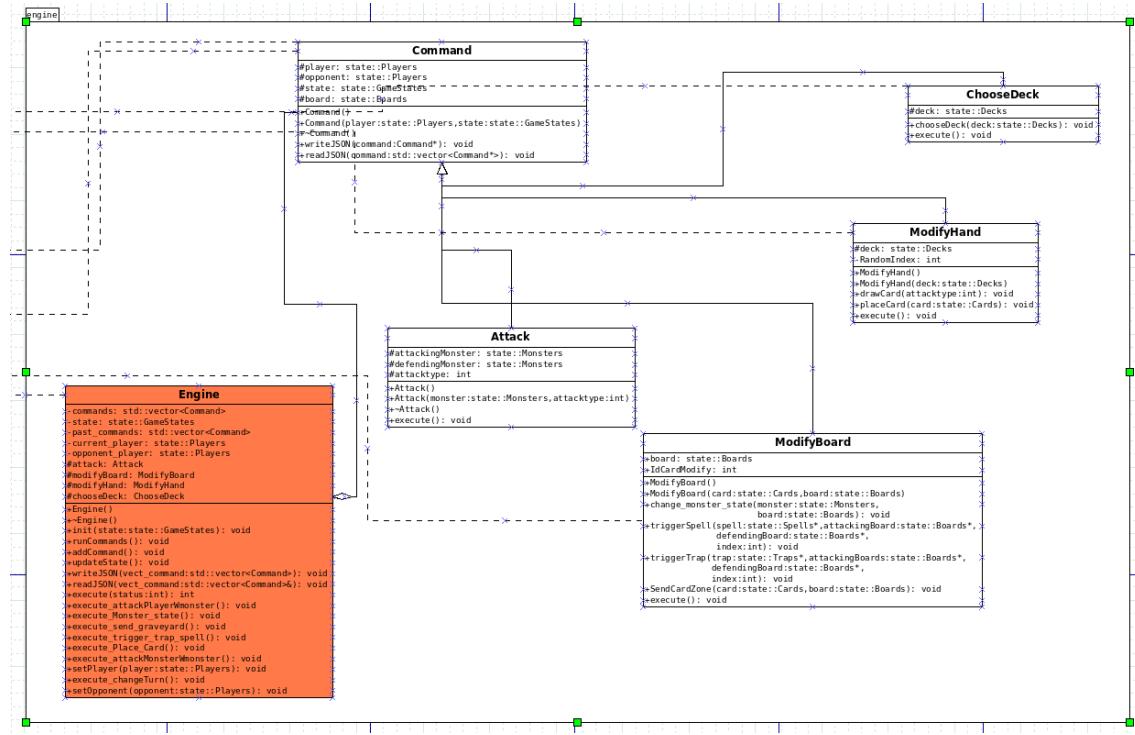


FIGURE 5 – Diagramme de classe du moteur de jeu

5 IA

5.1 stratégie

5.1.1 Intelligence minimale

La stratégie d'IA que nous avons développée jusqu'à son intégration dans le moteur de jeu consiste en une approche aléatoire. Cette intelligence artificielle crée un objet de type Joueur et effectue automatiquement des choix pour sa main et les monstres placés sur le tableau de jeu. Une fois positionnée sur le plateau au cours des tours suivants, l'IA suit les trois étapes fondamentales du jeu : piocher, placer des cartes et attaquer.

Lors de la phase de pioche, elle tire automatiquement une carte de la pioche. Ensuite, pendant la phase de placement des cartes (**main phase**), elle vérifie si la carte est un monstre ou un sort/piège. Si c'est un monstre, elle positionne la carte dans son état le plus puissant ; si c'est un sort ou un piège, elle la place en position de désactivation.

Dans l'étape suivante (attaque), l'IA vérifie la présence de cartes monstre, en choisit une au hasard et l'utilise pour attaquer un monstre sur le tableau de l'adversaire. En cas d'absence de monstre chez l'adversaire, elle attaque directement les points de vie. Cette logique est illustrée dans le diagramme

ci-dessous 6.

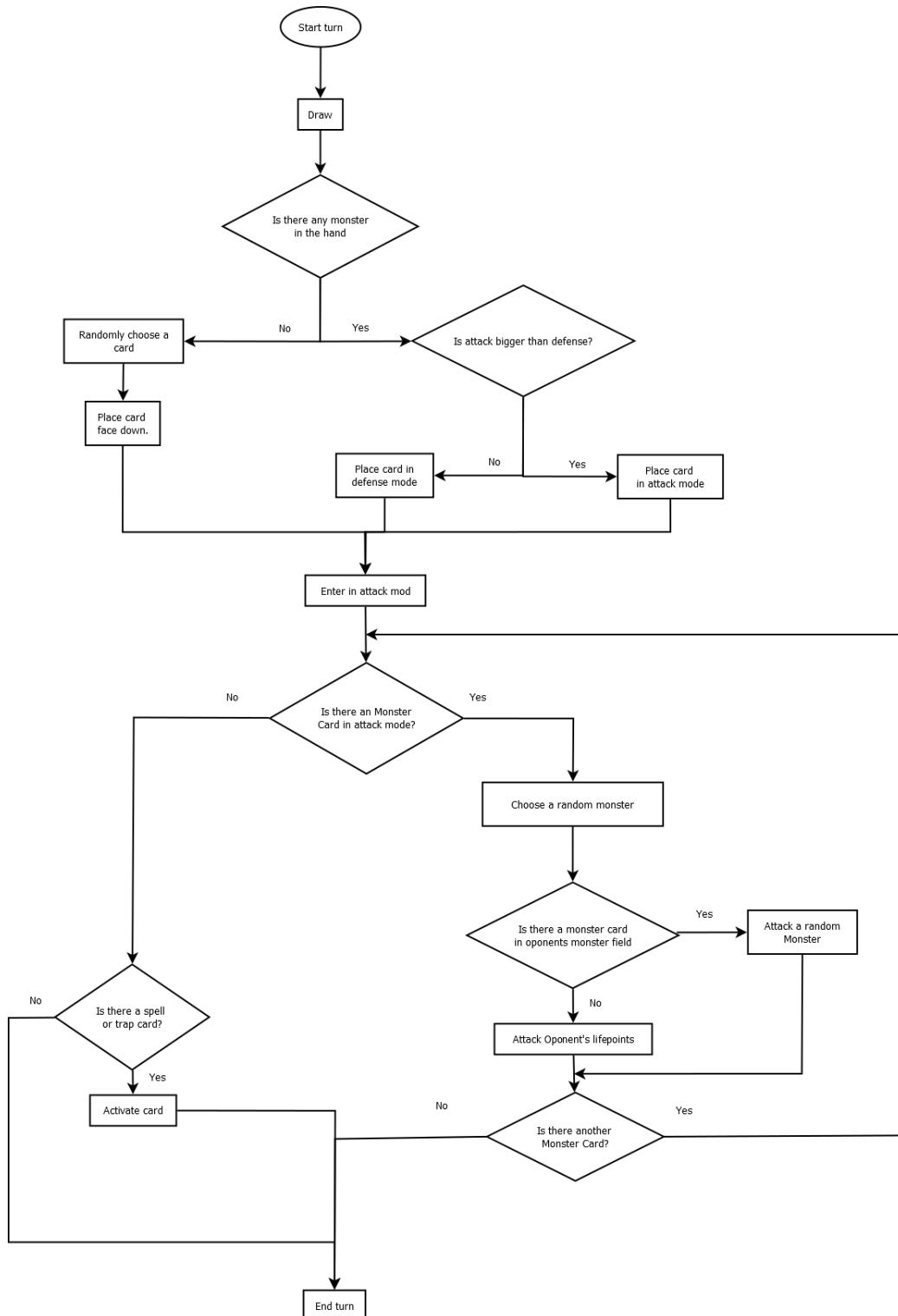


FIGURE 6 – Diagramme de flux basique

5.1.2 Intelligence basée sur des heuristiques

L'idée consistait à introduire une deuxième intelligence artificielle afin de rehausser la difficulté pour le joueur. En plus d'analyser les types de cartes du joueur et du bot, l'IA devait organiser les monstres dans sa propre main de manière que leurs points d'attaque augmentent dans l'ordre croissant. De la même manière qu'elle organise la main de l'adversaire, nous souhaitons que le bot cible les cartes ayant des points d'attaque plus faibles pour infliger des dégâts à l'adversaire. En l'absence de monstres, les pièges seront activés, suivant le modèle de la stratégie aléatoire.

5.2 Conception logicielle :

5.2.1 Intelligence minimale :

L'idée consistait à introduire une deuxième intelligence artificielle afin de rehausser la difficulté pour le joueur. En plus d'analyser les types de cartes du joueur et du bot, l'IA devait organiser les monstres dans sa propre main de manière que leurs points d'attaque augmentent dans l'ordre croissant. De la même manière qu'elle organise la main de l'adversaire, nous souhaitons que le bot cible les cartes ayant la plus faible puissance d'attaque pour infliger des dégâts à l'adversaire. En l'absence de monstres, les pièges seront activés, suivant le modèle de la stratégie aléatoire.

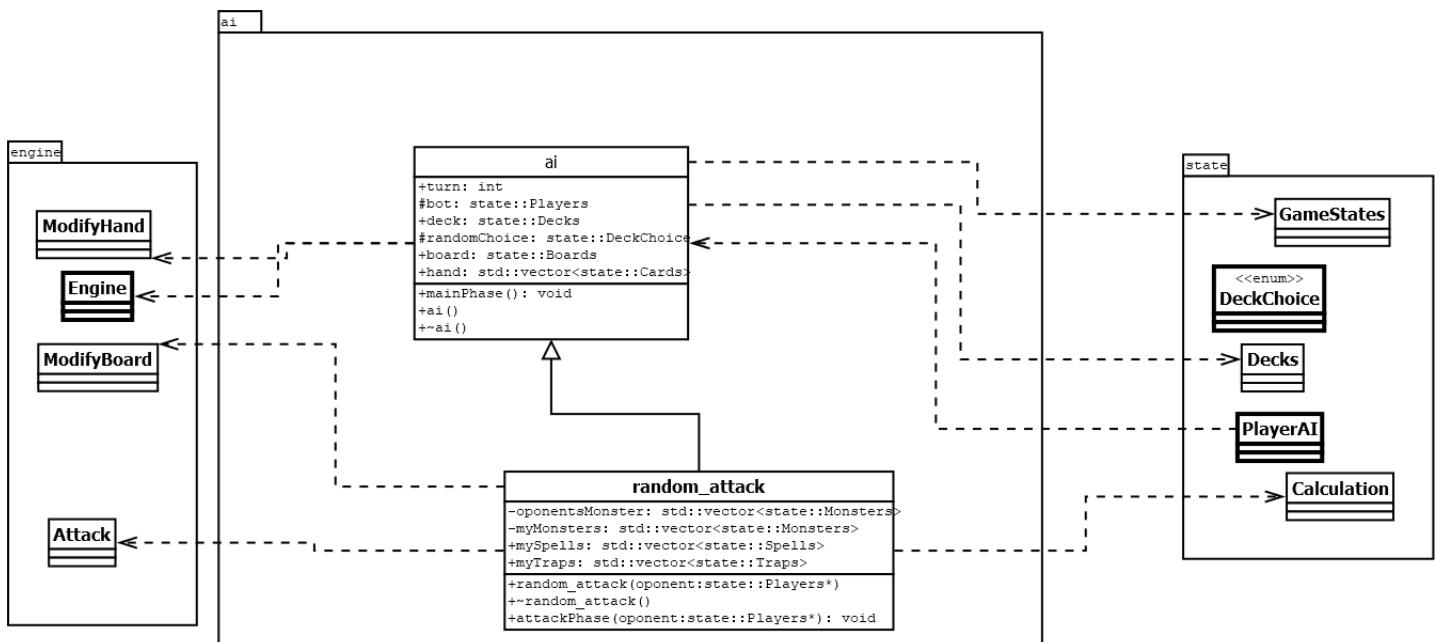


FIGURE 7 – Diagramme UML de l'ia

Bien que l'intelligence de base de l'heuristique ait été décrite dans la stratégie, nous n'avons pas été en mesure de la mettre en œuvre à temps avec les états du jeu pour qu'elle fonctionne en conjonction avec le reste du jeu.