

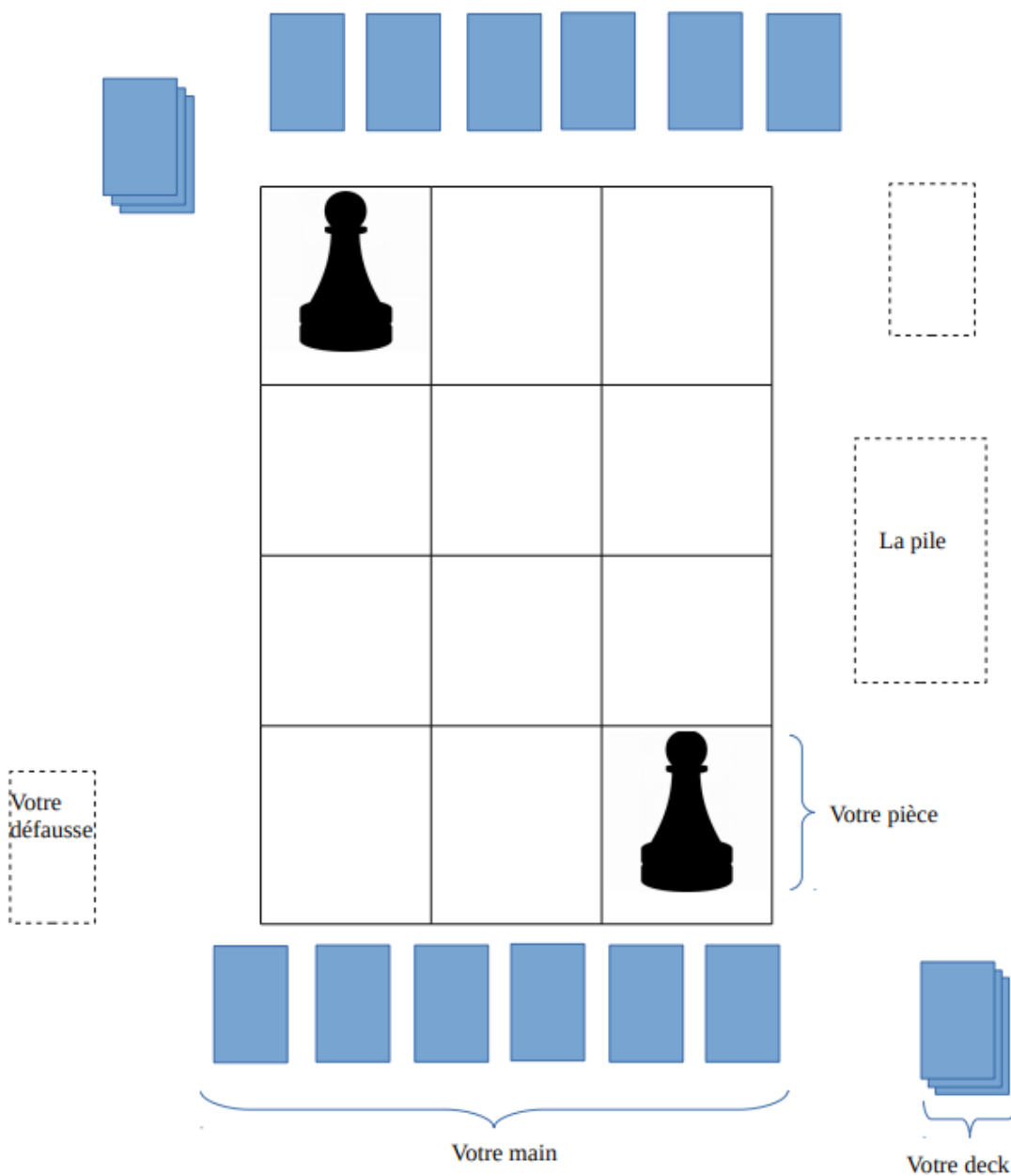
Projet informatique UE 2.4: Rapport de la première étape

Langlard Thibault
Vilasi Léa

I. Règles du jeu

Duel est un jeu de combat de un contre un. Le but du jeu est de réduire les points de vie adverses à 0 à l'aide de cartes d'attaque.

Au début de la partie, chaque joueur choisit un personnage, pioche 6 cartes dans le deck qui lui est attribué et place sa figurine dans le coin inférieur droit du plateau par rapport à lui.



Chaque joueur débute avec 40 points de vie. Le premier tour commence ensuite. Le plus jeune joueur a l'initiative, c'est-à-dire qu'il décide s'il souhaite agir ou non. S'il ne souhaite pas agir, le deuxième joueur doit décider s'il souhaite agir ou non. S'il ne le souhaite pas non plus, le joueur qui a le plus de cartes en main doit agir. Si les deux joueurs ont le même nombre de cartes en main, le joueur qui a l'initiative est obligé d'agir. Pour agir, un joueur a plusieurs options :

- Il peut déplacer ou orienter sa figurine
- Il peut jouer une carte d'attaque, reconnaissable à son fond bleu
- Il peut jouer une carte d'action, reconnaissable à son fond rouge

Il est à noter qu'on ne peut se déplacer que d'une case à la fois, y compris en diagonale. Lorsqu'un joueur agit, on dit qu'un échange commence. Son action ou sa carte est alors placée sur la pile. Un joueur peut alors choisir d'agir si son action a une vitesse supérieure à la dernière action placée sur la pile. Un déplacement a une vitesse de 1, une orientation a une vitesse de 3 et la vitesse d'une carte correspond au nombre marqué sur la flèche en bas à droite de celle-ci. Toutefois, une carte d'attaque peut être placée sur la pile après un déplacement ou une orientation même si sa vitesse est égale à ces derniers, et une carte d'action peut être placée sur la pile après un déplacement, une orientation ou une carte d'attaque même si sa vitesse est égale à celle de ces derniers. Par exemple, après une orientation peuvent être joués un *Dans ta face* ou une *Soif de sang*, mais pas une *Frappe verticale* ou un *Charger l'arbalète*.

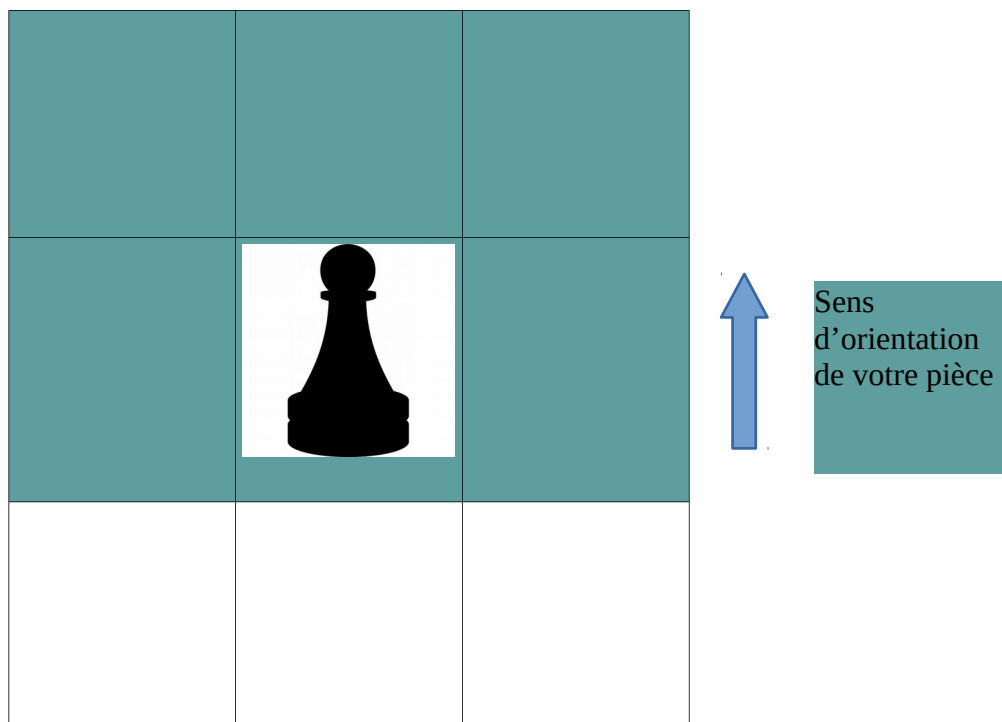
Lorsque plus aucun joueur ne désire agir durant l'échange, les actions se résolvent en commençant par celle qui a été placée en dernière sur la pile. Cependant, si un joueur subit des dégâts par une carte d'attaque, ou si une carte dit de mettre fin à l'échange, les actions sur la pile ne sont pas résolues. Les cartes qui étaient sur la pile sont ensuite placées dans la défausse de leur propriétaire. L'échange prend ensuite fin, les joueurs ont le droit de s'orienter sans que leur adversaire puisse répondre à cette action. Puis, si un joueur s'est déplacé durant l'échange, il doit laisser à son adversaire la possibilité d'agir. A tout moment, vous pouvez choisir de vous défausser de votre main. Cette action a une vitesse de 1 et votre adversaire subit 1 dégât par carte défaussée.

Lorsque plus aucun joueur n'a de carte en main et qu'aucun échange n'est en cours, le tour se termine. Chaque joueur pioche alors 6 cartes de sa pioche, le joueur qui avait l'initiative se déplace de façon à être distant de deux cases en ligne droite de son adversaire, on oriente les pièces de façon à ce qu'elles se fassent face et un nouveau tour commence, identique au précédent à ceci près que le joueur qui possède l'initiative est celui qui ne l'avait pas au tour précédent.

Les cartes d'action et d'attaque se lisent comme ceci :



La portée de la carte d'attaque correspond à la distance et la direction pour lesquelles l'attaque est valide. Par exemple, un **Abandon du fair-play** infligera des dégâts si on peut atteindre l'adversaire depuis sa pièce en se déplaçant au plus le nombre de fois indiqué de case en case selon l'une des directions indiquées, ou par une diagonale qui serait le produit de deux de ces directions, soit si l'adversaire se situe dans l'une des cases suivantes :



Lorsqu'une attaque se résout, le joueur qui reçoit l'attaque subit les dégâts de l'attaque et un dégât supplémentaire pour chaque autre carte dans la pile. Ces dégâts supplémentaires sont appelés dégâts de fatigue. Si un joueur subit au moins deux dégâts de fatigue, il est mis à terre. Si le joueur est de dos au moment où il reçoit l'attaque, il subit deux dégâts supplémentaires (cinq si l'attaque vient d'un assassin).

Il existe plusieurs mots-clés écrits en gras sur les cartes :

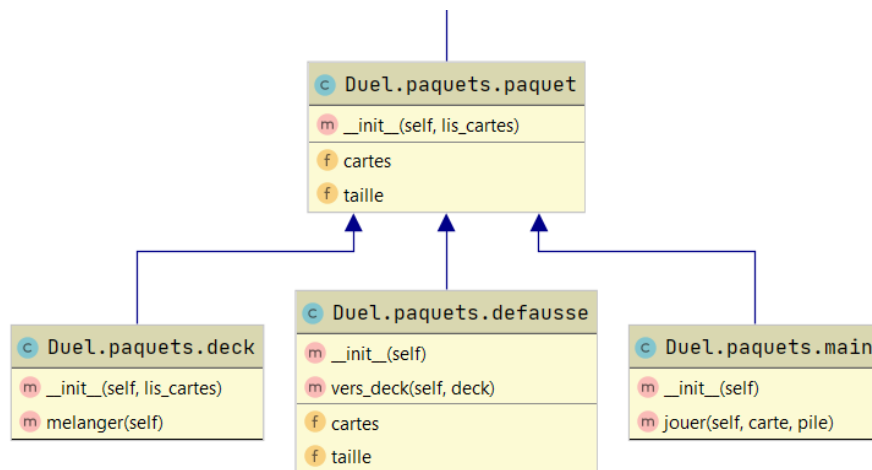
- si vous êtes **étourdi**, lors du prochain échange, si vous jouez une carte, lancez un dé. Si vous obtenez un nombre pair, annulez votre carte et défaussez-la.
- Si vous êtes **à terre**, lors du prochain échange, si vous jouez une carte, elle perd 2 de vitesse.
- Pour une **attaque à distance**, comptez la portée minimale que doit avoir votre attaque pour se résoudre. Divisez ce nombre par 2, arrondissez par excès et lancez un dé à 6 faces autant de fois que votre résultat. Si vous obtenez 1 au moins une fois, annulez l'attaque et défaussez-la.

Pour la carte duel de regards, lorsqu'elle se résout, piochez une carte, puis chacun des joueurs place sa main devant lui face cachée et fait un duel de regard avec son adversaire. Le joueur qui perd doit défausser une carte de sa main de son choix. Au bout de 15 secondes, si personne n'a perdu, chaque joueur a le droit d'agir, mais en participant toujours au duel.

II. Diagramme de classes

Notre programme est composé de 6 classes principales :

a) La classe paquet dont héritent les classes deck, defausse et main.



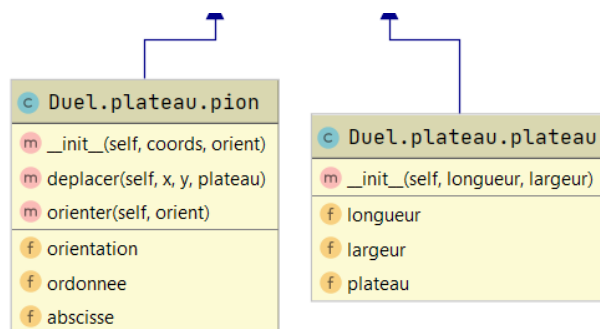
Paquet permet d'accéder à la liste d'un ensemble de cartes et à la taille de cette liste.

Deck correspond au paquet de cartes de chaque joueur, et contient une méthode pour mélanger les cartes.

Defausse contient la méthode vers_deck(), qui va servir à remettre les cartes de la défausse dans le deck du joueur lorsque le deck est vide.

Main correspond aux cartes que le joueur a en main. La méthode jouer() permet au joueur de poser une carte sur la pile.

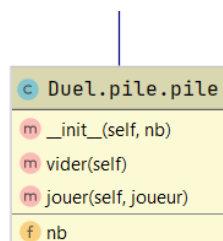
b) La classe plateau et la classe pion



Plateau permet de définir les dimensions du plateau pour une longueur et une largeur donnée.

Pion prend en paramètres des coordonnées et une orientation. Les méthodes déplacer() et orienter() permettent au joueur de le bouger.

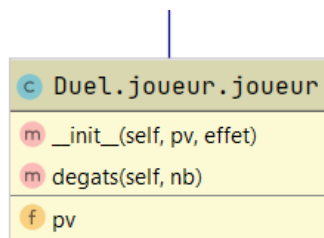
c) La classe pile



La méthode jouer() donne l'opportunité à chaque joueur d'interrompre l'action du joueur adverse en posant une carte sur la pile. Cette méthode est récursive. Chacun des joueurs sera questionné à nouveau jusqu'à ce qu'aucun d'entre eux ne souhaite plus interrompre. A ce moment, les cartes de la pile seront résolues une par une.

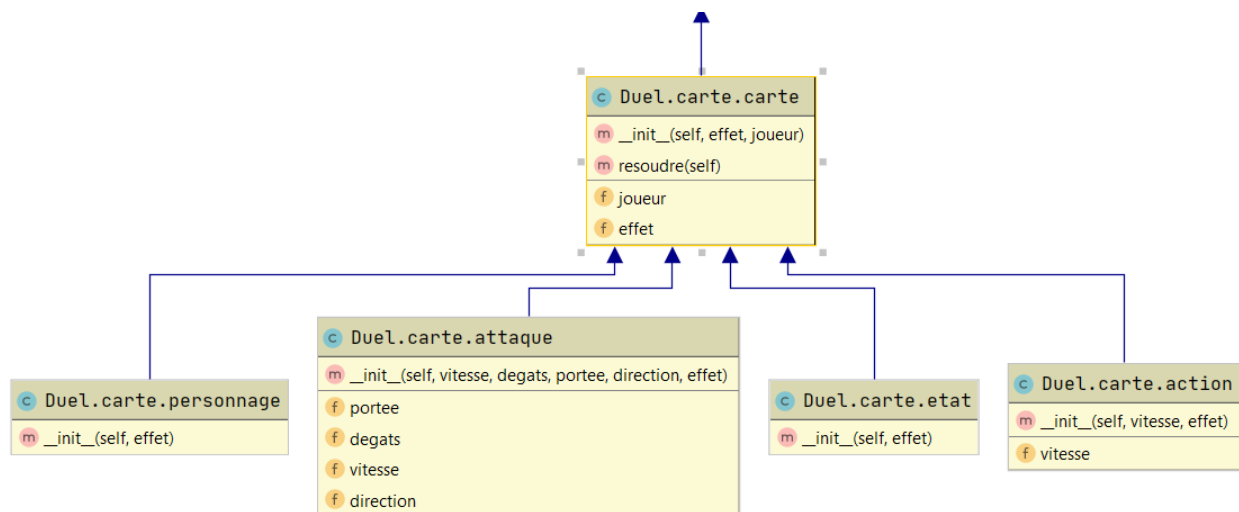
La méthode `vider()` va être utilisée pour vider la pile une fois que toutes les cartes ont été résolues.

d) La classe joueur



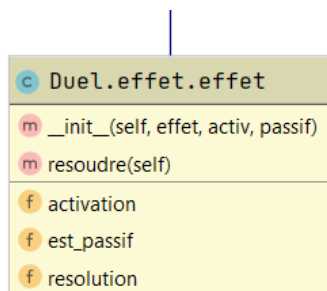
Elle donne accès au nombre de points de vie de chaque joueur, qui diminue en cas de blessure par l'intermédiaire de la méthode `degats()`.

e) La classe carte dont héritent les classes action, attaque, etat, personnage.



Chaque carte a pour attribut le joueur auquel elle appartient et une liste de ses effets, ainsi que ses différentes valeurs. Chaque sous-classe correspond à un type de carte différent. La méthode `resoudre()` donne accès à la méthode `résoudre()` de la classe `effet`. Il est à noter que les déplacements et orientations seront codés comme des cartes pour des raisons de praticité, mais ils ne sont pas considérés comme tels dans le jeu.

d) La classe effet



Nous avons envisagé plusieurs solutions pour résoudre les effets. Une distinction doit être faite entre les effets passifs, qui agissent à plusieurs reprises sur une durée donnée (parfois jusqu'à la fin

du tour, parfois jusqu'à la fin de l'échange) et les effets qui ne s'appliquent qu'une seule fois, cette distinction étant faite grâce à l'attribut `est_passif` qui prendra la forme d'un booléen.

La méthode que nous avons retenue pour l'instant est la suivante : les effets seront divisés en deux types : les effets qui appliquent des états aux joueurs et les effets qui modifient la valeur des cartes (ces deux sous-classes ne sont pas encore codés). Lors d'une partie, chaque joueur aura une liste qui comprend tous les effets actifs de ce joueur, et à chaque événement susceptible de voir un effet se terminer, on parcourt ces listes et on en retire tous les effets dont la condition d'arrêt (l'attribut `résolution`) correspond à l'événement.

Les effets qui appliquent des états auront seulement un attribut `activation` en plus, décrivant ce que fait concrètement l'effet, tandis que les effets modifiant les cartes auront un attribut `par` caractéristique de la carte qu'ils modifient (des sous-classes supplémentaires seront peut-être à prévoir en fonction du type de carte). Ces attributs seront des entiers, il n'y aura qu'à les additionner ou les soustraire aux valeurs des cartes.

III. Tests menés

On effectue plusieurs tests sur les classes pion, deck, defausse, action et attaque. On a vérifié que attaque et action étaient bien des instances de carte et pas l'inverse, et que les coordonnées du pion étaient bien enregistrées. De plus, on s'est assuré que deck et defausse étaient des instances de paquet, que les tailles de ces paquets étaient correctes et que même si les listes de cartes de ces deux paquets étaient égales, il ne s'agissait pas du même objet.