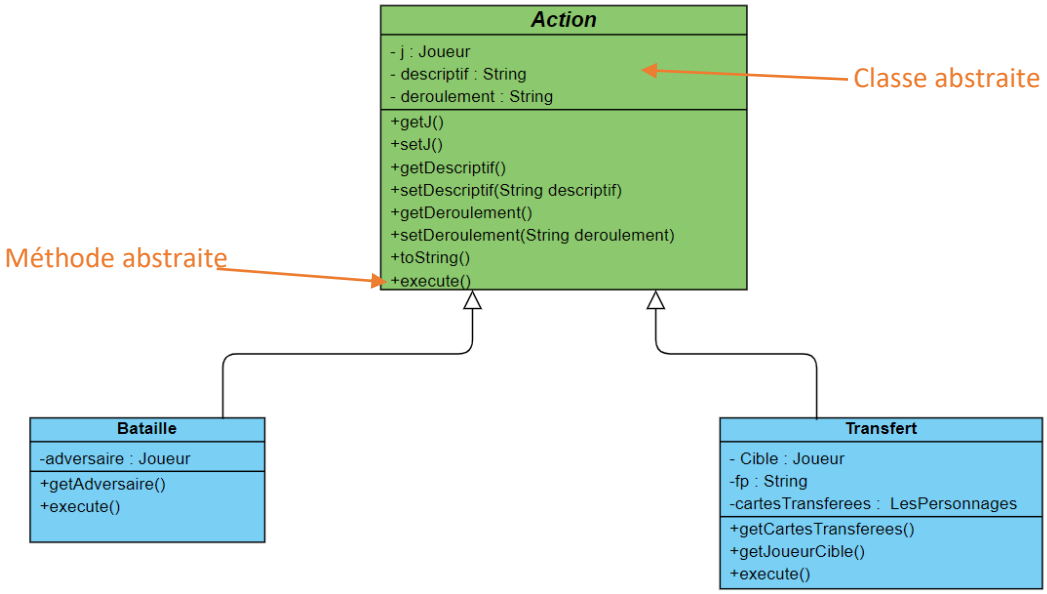
ISMAIL Saman **Groupe : IE2-021**

Rapport final projet Info2B

Le programme a été réalisé en binôme avec Hoarau Tristan

1) Hiérarchie des classes Action, Bataille et Transfert : Diagramme de classe



La classe mère Action est une classe abstraite qui contient une méthode abstraite execute()

execute() est ensuite définie dans les classes filles Bataille et Transfert.

Cela veut dire que les classes Bataille et Transfert héritent de la classe Action.

Cette structure hiérarchique est très utile car elle permet de ne pas répéter les attributs j, descriptif et deroulement dans les deux classes Bataille et Transfert.

De plus elle apporte une simplification pour que si le développeur de cette application veut ajouter d'autres actions par la suite il peut tout simplement ajouter une classe qui hérite de la classe Action et changer la méthode abstraite execute.

2) Gestion d'une action Transfert :
Une action transfert doit être réalisée dès lors un joueur gagne tous les personnages d'une des familles « rares » ou « communs ».

Quand un joueur appuie sur le deuxième personnage à révéler dans JeuMemory , la méthode verifPersos() est appelée. Si les deux boutons choisis par le joueur correspondent au même personnage, alors la méthode traiterTour() de la classe Jeu est appelée et son résultat est stocké dans une variable nommée bonus de type int.

La méthode traiterTour() permet de savoir si le joueur courant a gagné tous les personnages d'une même famille et selon la famille gagnée, elle renvoie un entier.

Si le joueur a gagné tous les personnages de sa famille préférée il gagne le jeu, sinon s'il gagné tous les personnages de la famille « rares » ou « communs » le bonus vaut 1.

Si le bonus vaut un 1 alors un transfert doit être réalisé.

Nous avons une boîte de dialogue TransfertDlg qui permet à l'utilisateur de faire des choix.

Lorsqu'un joueur gagne tous les personnages d'une des familles « rares » ou « communs » dans le jeu on informe les joueurs de cela en ajoutant du texte dans la zone d'édition nommée Edition :

```
Edition.append("Un transfert doit être réalisé");
```

On Déclare une instance de la boîte de dialogue TransfertDlg et on l'initialise avec son constructeur à 4 paramètres qui sont :

```
TransfertDlg transfert = new TransfertDlg(this, true, this.joueurs, monJeu.getIndC());
```

this : : le parent qui est JeuMemory (la fenêtre principale)
true : (Boolean modal) capture le focus de la fenêtre jusqu'à ce qu'elle soit fermée ou non
this.joueurs : Les joueurs présents dans le jeu
monJeu.getIndC() : l'indice du joueur courant

On rend la boîte de dialogue visible :
transfert.setVisible(true);

La boîte de dialogue TransfertDlg a un attribut de la classe Transfert nommé tc qui va nous permettre de gérer le transfert en interne.

Au chargement de la boîte de dialogue TransfertDlg, la JComboBox nommée ComboJoueurs est initialisée avec la méthode initCombo() avec le pseudo des joueurs y compris celui du joueur courant.

Le joueur doit choisir un joueur dans ComboJoueurs , si ce joueur est le joueur courant on affiche dans la JTextArea au centre nommée Infos qu'il faut choisir un autre joueur sinon, le panneau à gauche s'initialise avec des boutons grâce à la méthode initPanneau() et on affiche tous les personnages du joueur cible avec la méthode affichePanneau().

Maintenant le joueur courant clique sur un des personnages du joueur cible dans le panneau à gauche, ainsi tous les personnages du joueur cible qui appartiennent à la même famille ont une bordure rouge (la famille de la carte sélectionnée est récupérée par l'intermédiaire du nom du bouton, qui correspond à la famille du personnage affiché, et stocké dans l'attribut « fs ») et on affiche dans « Infos » combien de personnages il peut gagner en faisant un transfert et tous cela se fait avec la méthode boutonActionPerformed().

Au click sur le bouton Transfert, on vérifie si le joueur courant a bien sélectionné un ou des personnages.
Si c'est le cas, une nouvelle instance de la classe Transfert est crée avec en paramètres le joueur courant j, l'adversaire et la famille des cartes sélectionnés.
La méthode execute() de la classe Transfert est appelée et son résultat est stocké dans une variable res de type int.
La méthode execute() retire tous les personnages sélectionnés du paquet de l'adversaire, les ajoute au paquet du joueur courant et met « ok »à true pour dire qu'un transfert a bien eu lieu.
la méthode renvoi la taille du paquet.

Coté TransfertDlg, grâce aux méthodes creePanneau() et dessinePanneau(), on initialise le panneau à droite avec le ou les personnages transférés (avec la méthode getcartesTransferees()) et on met à jour le panneau à gauche.

Le joueur courant peut faire un seul transfert donc le bouton Transfert se désactive ainsi le transfert se termine et l'utilisateur doit juste cliquer sur le bouton Fermer pour fermer la fenêtre et grâce à la méthode getDeroulement(), on affiche dans la zone d'Edition le déroulement du transfert s'il est effectué.

```
if(transfert.getOk())
      {
         Edition.append("\n"+transfert.getTc().getDeroulement());
      }
```

3) Gestion d'une action Bataille:

Une action Bataille doit être réalisée dès lors un joueur gagne tous les personnages d'une des familles « legendaires » ou « epiques ».
Son déclanchement est similaire à celui d'une action Transfert, juste la valeur du bonus change.
Nous avons une boîte de dialogue BatailleDlg qui permet à l'utilisateur de faire des choix.

Lorsqu'un joueur gagne tous les personnages d'une des familles « legendaires » ou « epiques » dans le jeu on informe les joueurs de cela en ajoutant du texte dans la zone d'édition nommée Edition :

```
Edition.append("Une bataille va commencer");
```

On Déclare une instance de la boîte de dialogue BatailleDlg et on l'initialise avec son constructeur à 4 paramètres qui sont :

```
BatailleDlg batailleDlg = new BatailleDlg(this, true, joueurs, monJeu.getIndC());
```

this : : le parent qui est JeuMemory (la fenêtre principale)
true : (Boolean modal) capture le focus de la fenêtre jusqu'à ce qu'elle soit fermée ou non
this.joueurs : Les joueurs présents dans le jeu
monJeu.getIndC() : l'indice du joueur courant

On rend la boîte de dialogue visible :
batailleDlg.setVisible(true);

La boîte de dialogue BatailleDlg a un attribut de la classe Bataille nommé b qui va nous permettre de gérer la bataille en interne.

Au chargement de la boîte de dialogue BatailleDlg, la JList nommée ListeJ est initialisée avec la méthode initListe() avec le pseudo des joueurs y compris celui du joueur courant, une seule sélection est possible.

Le joueur doit choisir un adversaire dans JList, si ce joueur est le joueur courant on affiche à gauche dans la JTextArea nommée infosCarte1 qu'il faut choisir un autre joueur sinon, afficher les personnages du joueur courant dans infosCarte1 à gauche, afficher les personnages de l'adversaire dans infosCarte2 à droite et une nouvelle instance Bataille est créée avec en paramètres le joueur courant (jc) et l'adversaire, tous cela grâce à la méthode ListeJMouseClicked().

Maintenant le joueur courant peut cliquer sur le bouton Démarrer pour commencer une bataille.
Cela déclenchera l'appel de la méthode DemarrerActionPerformed(), elle vérifie si la sélection de l'adversaire est valide et que les deux joueurs ont des cartes.
Le premier personnage de chaque joueur s'affiche dans les JButtons Carte1 et Carte2 et la méthode execute() de la classe Bataille est appelée et son résultat est stocké dans une variable res de type int.

```
int res = this.b.execute();
```

selon la valeur de la variable res :
0 : il y a égalité donc personne ne perd ou gagne de personnage.
1 : Le joueur courant gagne le premier personnage de l'adversaire
2 : L'adversaire gagné le premier personnage du joueur courant

On affiche dans chaque zone d'édition InfosCarte1 et InfosCarte2 le résultat de la bataille
On affiche les personnages de chaque joueur et on change le texte du label Vainqueur.

Tant que les deux joueurs ont des personnages, la bataille peut continuer avec le bouton Démarrer, sinon le texte du bouton Annuler devient Fermer et le bouton Démarrer est désactivé

Coté JeuMemory, elle vérifie si une bataille a bien eu lieu avec l'accesseur getOk(), si c'est le cas on ajoute le déroulement de bataille à la zone d'édition du jeu principal.

```
if(batailleDlg.isOk())
        {
           Edition.append("\n"+batailleDlg.getBataille().getDeroulement());

        }
```

4)

LesJoueurs joueurs : ensemble des joueurs de la parite

LesPersonnages persos : ensemble des personnages de la partie

int l1,c1,l2,c2 : Les coordonnées des boutons cliqués

Jeu monJeu : instance qui gère la partie, grâce à elle on peut connaitre l'état du plateau, le joueur courant, les personnages restants etc…

Le constructeur JeuMemory ne prend pas de paramètres, il commence avec l'appel de la méthode initComponents(), qui permet l'affichage de la fenêtre; Initialise joueurs et persos avec leurs constructeurs standards respectifs. Les quatre coordonnés sont initialisés à -1, car c'est le démarrage et aucun bouton n'a été cliqué encore. L'attribut monJeu est initialisé à null, car l'instance de Jeu sera créée après que le joueur ait choisi les joueurs, les personnages et la niveau de difficulté dans OptionDlg. Le bouton Recommencer est invisible pour le démarrage.

5)

a) Avant de démarrer le jeu :

Le joueur doit aller dans le menu « Paramètres » puis « Options ».
Une nouvelle fenêtre OptionDlg s'affiche, dans cette boîte de dialogue, le joueur peut choisir le niveau de jeu (La difficulté) et les joueurs créés par défaut parmi (Lara, Jack, Jean-Sébastien et Mozart).
S'il a fait son choix ,il doit cliquer sur le bouton « Valider », « ok » de la JDialog est mis à true, Le joueur est renvoyé vers le jeuMemory (fenêtre principale).

JeuMomory vérifie que les personnages ont bien été créés avec l'accesseur getOk(), dans ce cas on ajoute à la liste des joueurs de JeuMemory ceux créés dans la JDialog, on récupère le niveau sélectionnée dans la JDialog, on construit un paquet de personnages correspondant au niveau sélectionné grâce au constructeur de la classe LesPersonnages prenant comme paramètre un entier et on initialise l'attribut monJeu avec en paramètres les personnages, les joueurs et le niveau.

Le joueur peut aussi ajouter un autre joueur, il doit aller dans le menu « Paramètres » puis « Ajout Joueur ». Cette JDialog prend en paramètre « joueurs ». Grâce à cette boîte de dialogue, l'utilisateur peut créer un nouveau joueur avec le pseudo de son choix, une famille préférée qu'il peut choisir dans une JList nommée « ListeFamilles » parmi la famille des joueurs déjà créés pour la partie et enfin une photo pour le joueur. Il doit ensuite cliquer sur « Valider », ce qui va créer nouveau joueur et l'initialiser dans l'attribut « j » de type Joueur et ferme la JDialog. Le JeuMemory récupère le booléen de la JDialog, s'il est à true, le joueur créé dans la JDialog est récupéré et ajouté à l'attribut « joueurs » de la JFrame JeuMemory. Le joueur peut maintenant appuyer sur le bouton « Démarrer ».

b) Durant la partie :

Le joueur clique sur deux boutons, c'est-à-dire 2 cartes différentes, si les personnages cliqués sont les mêmes alors le joueur gagne le personnage sinon c'est le tour du joueur suivant.

Il y a plusieurs actions qui peuvent être réalisés, le transfert si le personnage a gagné une famille complète parmi « rares » ou « communs », la bataille si le joueur gagne une famille complète parmi « legendaires » ou « epiques », les informations concernant ces actions sont ajoutés dans la zone d'édition à gauche de la fenêtre principale nommée Edition.

Durant le jeu, le joueur peut regarder quels personnages il a gagné et son score actuel, il doit aller dans le menu « Visualiser » puis « Cartes ».
Il peut aussi connaître le score des autres joueurs, leur nom, leur famille préférée et leur photo, il doit aller dans le menu « Visualiser » puis «Joueurs».

c) Fin du jeu :

Le jeu se termine dans l'un des cas suivants :

1) Le joueur courant a gagné tous les personnages de sa famille préférée plateau (Le ou les vainqueurs sont affichés dans la zone d'édition)

2) S'il n'y a plus de cartes à dévoiler sur le plateau (Le ou les vainqueurs sont affichés dans la zone d'édition)

3) Si le joueur clique sur le bouton recommencer en bas à gauche pour recommencer un nouveau jeu (le jeu actuel est abandonné) et donc il doit retourner dans le menu « Paramètres » puis « Options » pour créer un nouveau jeu.

Fin