Licence Sciences et Technologies – 1ère année

Module Info2B - Interfaces Visuelles

RAPPORT FINAL

1. Les actions de transfert de cartes et de bataille

2. Application principale

Choukri Imane

MI2-I2

1. La hiérarchie de classes, Pourquoi cette hiérarchie est-elle utile ? Qu'apporte-t-elle ?:

La classe Action concerne un joueur, celui qui sera en train de jouer pendant le jeu. Elle comporte un descriptif pour des actions comme « Transfert de cartes » ou « Bataille » et un déroulement qui jouera le rôle d'un historique de ce qui s'est passé durant l'action et qui sera complété selon l'action réalisée.

Cette classe est la classe mère des deux classes filles Bataille et Transfert , la modélisation de ces actions est une hiérarchie qui sert à distinguer entre les deux types d'actions qui existent , car Bataille comme Transfert sont des sous-classes qui héritent de la super classe (ici Action).

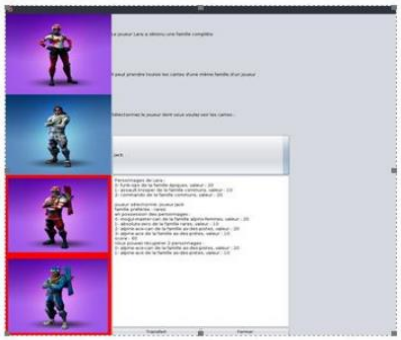
Cette classe est abstraite elle contient donc une méthode abstraite (ici execute()) qui est redéfinie deux fois et différemment dans les deux sous classes . Ainsi, on a l'assurance que les classes filles respecteront le contrat défini par la classe mère abstraite. Ce contrat est une interface de programmation.

La classe fille ou la sous-classe Transfert a trois attributs : Cible de type Joueur est qui est celui à qui les cartes vont être prises , fp de type String qui désigne la famille de la carte sélectionnée et carteTransferees de type LesPersonnages et qui désigne toutes les cartes de la famille choisies, prises au joueur cible .De plus, cette classe comporte des accesseurs en lecture de ses attributs et un constructeur avec paramètres qui , a son tour appelle le constructeur de la super classe « Action »,puis initialise les attributs à des valeurs par défaut .Dans cette classe ,l'implémentation de la méthode abstraite est essentielle , elle assure le transfert de cartes entre le joueur courant et la cible . Cette méthode fait appel a des méthodes de différentes classes comme LesPersonnages et Joueur afin de récupérer des informations dont on aura besoin au sein de cette méthode abstraite.

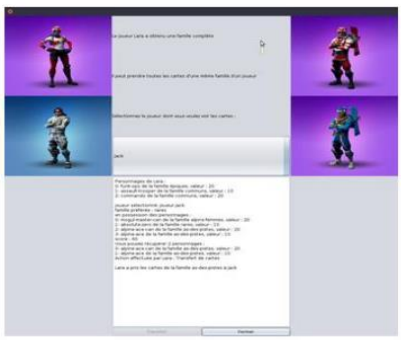
Elle vérifie d'abord si la famille « fp » existe et n'est pas null , puis dans ce cas,

- on retire les cartes de la famille « fp » du paquet du joueur cible en les stockant dans l'instance cartestransferees en utilisant des méthode externes (getPaquet()et getPersosFamille()).
- On ajoute dans le paquet du joueur courant les cartes qui ont été retirées du paquet du joueur cible.
- On affecte à l'attribut déroulement un message indiquant quel joueur a pris quelle famille de cartes à quel autre joueur.

Cela méthode renvoie une valeur res qui vaut soit 0, si aucune carte n'a été transférée, soit le nombre de cartes transférées.



Après un clic sur l'un des personnages



Après un clic sur le bouton "Transfert"

Il est nécessaire d'assurer une réactivité entre l'application principale et la boîte dialogue nommée « TransfertDlg » . Lorsque le joueur courant gagne tous les personnages de la famille « rares » ou « communs » (et qu'il ne s'agit pas de sa famille préférée), la boite de dialogue s'ouvre pour permettre la réalisation d'un transfert de cartes ,lorsque la boîte de dialogue s'ouvre, elle permet au joueur courant de sélectionner un joueur dans une liste déroulante pour visualiser ses cartes dans le panneau gauche . Il peut ensuite cliquer sur un des personnages du joueur et toutes les cartes de la même famille possédées par ce joueur sont « encadrées » en rouge (regardez l'image ci-dessus) .

Si le joueur se choisit lui-même un message d'erreur lui est indiqué dans la zone d'édition, au milieu en bas de la boite, nommée « Infos ». Lorsque le joueur est satisfait de son choix, il clique sur le bouton « Transfert ». . Les cartes de la famille sélectionnée du joueur sont alors affichées dans le panneau droit et supprimées du panneau gauche.

Enfin le joueur courant et l'indice sont envoyés à la boite de dialogue au moment de la création de la boite de dialogue et la boite de dialogue renvoie l'attribut OK par la méthode isOK() pour savoir est ce que la boite de dialogue a été fermer par le bouton fermer ou non.

3- La classe « Bataille » est une classe fille de la clase mère Action .Elle contient un attribut de type Joueur nommé « adversaire »,et un constructeur à deux paramètres qui fait appel au constructeur de la super classe puis initialise l'attribut à une valeur par défaut . Cette classe implémente aussi la méthode public int execute() qui fait la comparaison entre les deux premières cartes respectives du paquet des joueurs.

Lors d'une bataille, le joueur courant est opposé à un adversaire que le joueur courant a préalablement choisi. Elle va consister à prendre la première carte du paquet de chacun des joueurs et à comparer la valeur des deux personnages. Le joueur possédant le personnage ayant la plus forte valeur remporte les deux cartes qui sont ajoutées à la fin de son paquet. Dans le cas où les deux cartes ont la même valeur, chacun remet sa carte à la fin de son paquet. La bataille s'arrête dès qu'un des joueurs n'a plus de cartes dans son paquet. Si l'un des joueurs n'a pas de cartes l'action ne peut pas se dérouler.

Elle renvoie la valeur 1 si le joueur courant gagne, la valeur 2 si l'adversaire gagne, la valeur 0 en cas d'égalité et -1 si la bataille ne peut pas être réalisée faute de carte d'un des joueurs

Si le joueur courant et son adversaire possèdent encore des cartes (taille de leur paquet >0) on procède à :

- L'initialisation de la valeur du résultat à -1, dans un entier nommé res .
- La Récupération de la 1ère carte (d'indice 0) du paquet du joueur courant dans une instance nommée c1, de type Personnage .
- La Récupération de la 1ère carte (d'indice 0) du paquet de l'adversaire dans une instance nommée c2, de type Personnage .
- La suppression de la carte c1, du paquet du joueur courant .
- La Suppression de la carte c2, du paquet de l'adversaire .

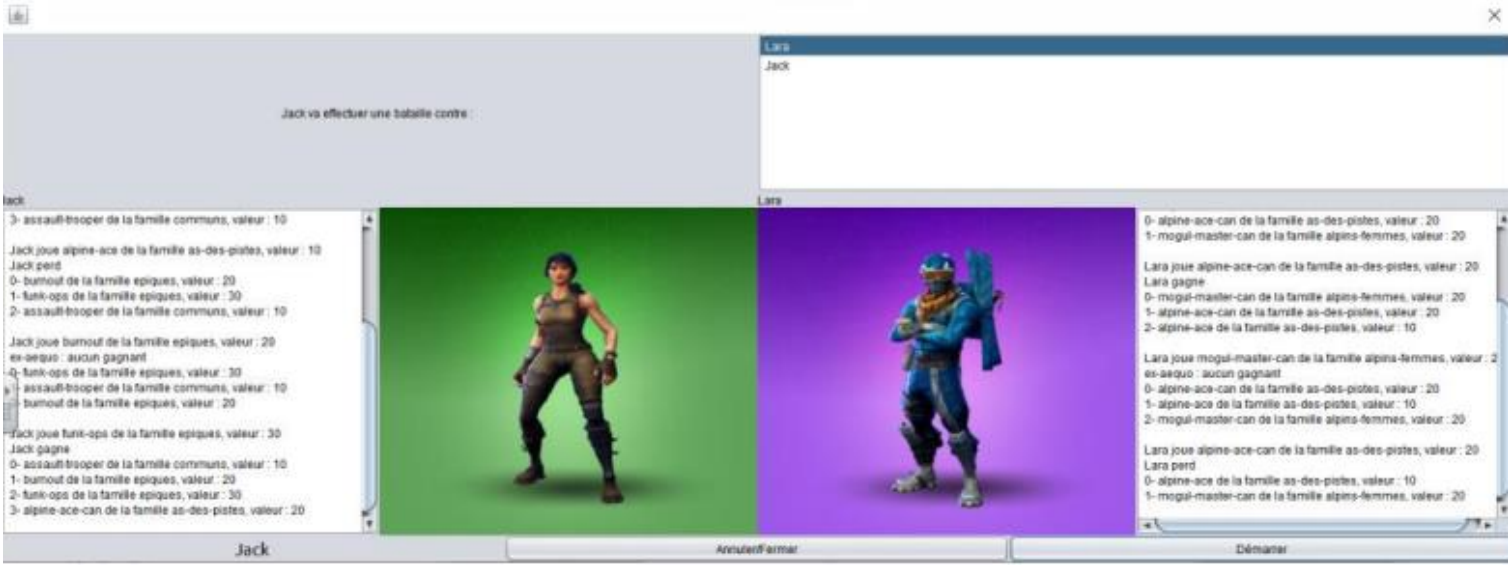
Si les valeurs des deux cartes sont identiques, le résultat est affecté à 0 donc une égalité et chacune des cartes sont remises au paquet du joueur auquel elle appartenait .

Cette classe utilise plusieurs méthodes externes d'autres classes afin de récupérer des informations utiles à son fonctionnement . Parmi ces méthodes on a : getPaquet() , getTaille() ,getPerso() ,getJ() ,retirePerso() , getValeur(),ajoutePerso() , getPseudo() et setDeroulement().

Il est nécessaire d'assurer une réactivité entre l'application principale et la boîte dialogue nommée «BataileDlg ».Cette boite s'ouvre lorsque le joueur courant gagne tous les personnages de la famille « légendaires » ou « épiques » :

- Elle propose au joueur courant de sélectionner un joueur dans une liste nommée « ListeJ » de type JList initialisée à l'ouverture par la liste des joueurs gérée dans le jeu.
- Le contenu du paquet de personnages du joueur courant est alors affiché dans une zone de texte nommée « InfosCarteDroite » (respectivement « InfosCarteGauche ») .
- Un jLabel nommé « Messsage » en haut à gauche s'affiche , pour indiquer le pseudo du joueur courant (qui est en train de joueur).
- Le clic sur le bouton BDemarrer fait appel à la méthode execute() défini dans la classe Bataille afin d'effectuer cette dernière .

- Les zones de texte donnent les détails des personnages joués.
- Le premier personnage du paquet de chaque joueur est alors affiché sur les deux boutons « Carte1 » et « Carte2 ».
- Le pseudo du vainqueur s'affiche sur le jLabel nommé « Vainqueur ».
- Le joueur peut mettre fin à la bataille en cliquant sur le bouton « Annuler » afin de fermer la boite de dialogue sinon le jeu se poursuit jusqu'à ce qu'un vainqueur et élu ou qu'un des deux joueurs n'aie plus de cartes , le bouton « Démarrer » est donc désactivé.
- Finalement ,le joueur courant et l'indice sont envoyés à la boite de dialogue dans le constructeur au moment de la création de la boite de dialogue et et la boite de dialogue renvoie l'attribut OK par la méthode isOK() pour savoir est ce que la boite de dialogue a été fermer par le bouton fermer ou non.



Application principale :

La classe JeuMemory est la classe principale du projet elle contient, des attributs :

- persos de type LesPersonnages qui sert stocker des personnages selon la taille du jeu.
- joueurs de type LesJoueurs qui sert stocker les joueurs qui vont jouer.
- monjeu de type Jeu, cet attribut est le médiateur entre l'application principale et les autres classes,il veille sur la gestion d'un tour de jeu.
- l1,c1,l2,c2 des entiers pour les cartes sélectionnées à chaque tour de jeu, ils sont les positions (ligne et colonne) des cartes cliquées par le joueur courant.

  Un constructeur :
- public JeuMemory() un constructeur par défaut qui fait appel à la méthode initComponent() , initialise l'attribut persos avec le constructeur par défaut de la classe LesPersonnages et l'attribut joueurs avec celui de la classe LesJoueurs .

  Des méthodes :
- La méthode « public void startTimer() » permet d'avoir un temps d'attente, après le clic sur la 2ème carte, pour que le joueur voit les 2 cartes avant leur traitement. Après ce délai, le gestionnaire appelle la méthode « verifPersos() ».
- La méthode « public void verifPersos() » est la méthode qui vérifie les personnages des deux cartes retournées, traite le tour de jeu, met à jour l'interface et passe au joueur suivant.
- BRecommencer.setEnabled(false) sert à désactiver le bouton Recommencer avant de joueur.
- Edition.setEditable(false);interdir la modification sur la zone de texte Edition(JTextArea) .
- Le clic sur une carte du jeu, qui correspond à l'exécution du gestionnaire « private void boutonActionPerformed(ActionEvent evt) » .
- Le clic sur le bouton « Démarrer » débute la partie de jeu.

les différentes étapes du déroulement du jeu :

- ***Avant le démarrage :***

Premièrement et avant de commencer le jeu il faut choisir les joueurs ainsi que le niveau du jeu via la sous option « option » ou grace à la sous option « AjoutJoueur » (afin de créer votre propre joueur avec un pseudo image personnalisé , l'utilisateur choisi aussi sa famille préférée) , pour l'instant le bouton recommencer est désactivé et les deux sous options cartes et joueurs n'affichent rien.

- Pendant le jeu :

Au cours du jeu , le joueur peut visualiser les cartes et aussi les joueurs qui jouent , il peut choisir deux cartes parmi celle présentes(une zone de texte affiche les informations sur le joueur courant) .Le bouton démarrer est désactivé ainsi que les sous option ajoutJoueur et Option cependant , le bouton recommencer est activé .

Si un joueur réussit à obtenir tous les personnages d'une famille qui n'est pas sa famille préférée, il peut alors, selon la famille complète en sa possession, effectuer une action supplémentaire :

- récupérer les cartes d'un autre joueur
- faire une bataille ou un combat

Si le joueur a obtenu tous les personnages de la famille :

- « rares » ou « communs », il peut prendre à un autre joueur, toutes ses cartes d'une famille au choix.
- « légendaires » ou «épiques », un jeu de bataille de cartes se fait entre le joueur courant contre un autre joueur. Cet autre joueur est choisi par le joueur courant.
- « alpins-femmes » ou « as-des-pistes », un combat est engagé entre le joueur courant et un autre joueur de son choix.

- l'arrêt du jeu :

  Le joueur peut mettre fin à la bataille en cliquant sur le bouton « Annuler » afin de fermer la boite de dialogue sinon le jeu se poursuit jusqu'à ce qu'un vainqueur et élu ou qu'un des deux joueurs n'aie plus de cartes , le bouton « Démarrer » est donc désactivé.

Merci