GUERCI Richard *Groupe 2*
JEMMALI Fadoua *Groupe 4*

MINI PROJET TOP

I/ Introduction

Le sujet étant la réalisation dôun jeu de casino, cela nous a tout particulièrement motivé à développer une application fonctionnelle et amusante.

Lôapplication permet de jouer en mode console et en mode graphique. Lôinterface graphique a été réalisé avec les bibliothèques graphique *swing* et *awt*.
Nous avons également implémenté une extension permettant dôutiliser le capteur Kinect de Microsoft.

II/ Mode dôemploi

Comment jouer ?

Double-cliquez sur le fichier *top_guerci_jemmali.jar* et le programme se lancera automatiquement (Remarque: kinect ne fonctionne pas à partir du *.jar*).

Pour utiliser Kinect, il faut déplacer le ficher *libKinect.jnilib* (qui se trouve à la racine du projet) vers */Library/Java/Extensions/*.
Puis exécuter *swing.Main.java*.

A quoi sert une classe?

Nous avons édité une JavaDoc pour les principales classes de lôapplication afin de répondre à toutes les interrogations (elle se trouve dans le dossier *doc* à la racine du projet).

III/ Descriptif

Le programme se divise en 7 paquetages afin de structurer les classes et de rendre le programme plus compréhensible :

combinaison :

Le paquetage *combinaison* contient les énumérations de toutes les combinaisons de la roulette française (*ChanceSimple.java*, *ChanceCheval.java*, *ChanceCarre.java*, ...)

A chacune de ces énumérations est associée une classe héritant de *Combinaison.java* permettant de déclarer la liste des numéros qui compose la combinaison.
Pour chaque type de combinaison (Simple, Carré, Cheval, ...) est associée une mise minimum, une mise maximum, la mise du joueur et un multiplicateur (permettant de multiplier la mise en cas de gain).

Exemple:

GUERCI Richard *Groupe 2*
JEMMALI Fadoua *Groupe 4*

Dans la classe *CombinaisonCarre*, la combinaison *ChanceCarre._1_2_4_5* est une liste de 4 objets *Numero* ayant pour valeurs 1, 2, 4 et 5.
La mise minimum d'un Carré est 1.
La mise maximum d'un Carré est 120.
Le multiplicateur d'un Carré est 9.

La fonction *gain* de la classe *Combinaison.java* prend en paramètre un objet *Numero* qui correspond à un numéro de tirage. Ainsi, si ce dernier est présent dans la liste que contient la combinaison alors la combinaison est gagnante et la fonction retourne la mise multipliée par le multiplicateur.

exception :

Le paquetage comporte deux classes d'exceptions (*BesaceInsuffisanteException* et *MiseMaxException*) qui peuvent être déclenchées lors d'une partie à la roulette française.

jeu :

Le paquetage *jeu* rassemble les classes maîtresses de l'application:

La classe *Numero* qui représente un numéro de la roulette ou de la table (elle possède une valeur et une couleur)

La classe *Roulette* est une *ArrayList* des 37 numéros (de 0 à 36) qui constitue la roulette.
Elle contient la fonction tirage qui permet de retourner un objet *Numero* aléatoire.

La classe *Table* contient une instance de la classe *Roulette* et une *ArrayList* de *Joueur*.
Elle permet d'ajouter ou de retirer des joueurs, de faire miser les joueurs et de répartir les gains en fonction d'un tirage.

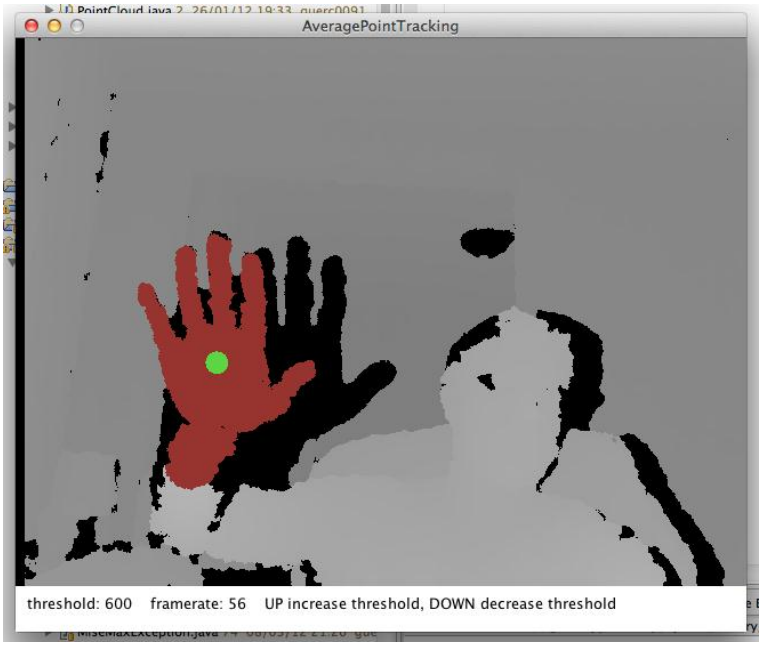
joueur :

Le package contient les classes qui définissent les différents types de joueurs se basant sur leurs stratégies du jeu, à savoir *JoueurAlembert*, *JoueurHawks*, *JoueurEsial* et *JoueurHumain*.
Ces classes héritent toutes de la même classe mère *Joueur*.
Un joueur *possède* un nom, une besace et une *ArrayList* de *Combinaison*.

kinectsensor :

Le paquetage rassemble l'ensemble des classes nécessaires pour exploiter le capteur Kinect. *AveragePointTracking* et *KinectTracker* écrit en Processing permettent de détecter les profondeurs de champs via Kinect en fonction de la sensibilité (*threshold*). Le centre de la surface pris en compte (affiché en rouge) définit un point.

GUERCI Richard *Groupe 2*
JEMMALI Fadoua *Groupe 4*



La classe *Mouse* permet de bouger le curseur de la souris et dœeffectuer un clic si la zone sélectionnée par la main de lôutilisateur a été sélectionnée depuis au moins 3 secondes (la classe *PositionHistory* permet de sauvegarder les points des 3 dernières secondes et de calculer sôils sont dans la même zone).

simulation :

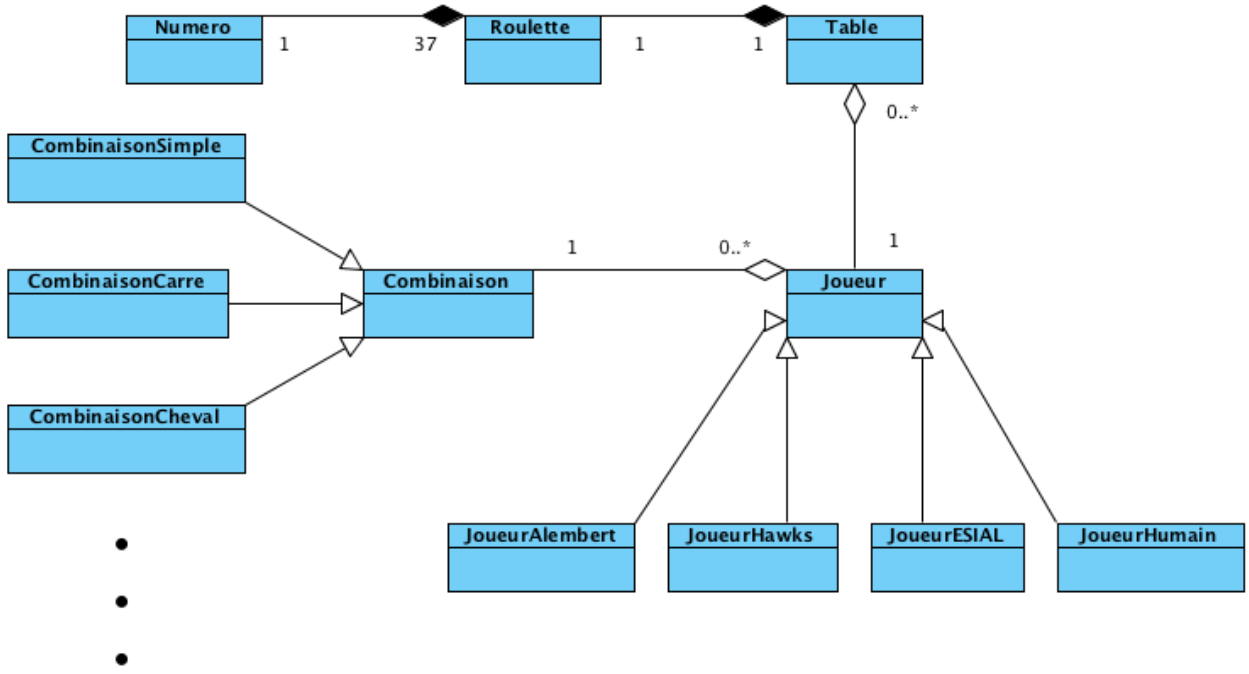
Ce paquetage simulation contient deux classes :
JeuEnConsole.java qui permet de jouer en mode console.
LaSoiree.java qui permet de simuler la soirée des joueurs de lœESIAL au casino dôAmnéville.

swing :

Ce paquetage rassemble toutes les classes utilisées par lôinterface graphique.

Pour résumer les relations entres les principales classes de lôapplication nous avons réalisé un diagramme de classes (pour simplifier la représentation nous nôavons pas représenté les classes des paquetages *exception*, *simulation*, *swing* et *kinectsensor*) :

GUERCI Richard *Groupe 2*
JEMMALI Fadoua *Groupe 4*



IV/ Kinect

Lœextension Kinect utilise la librairie de Daniel Shiffman qui elle-même utilise Processing et la librairie OpenKinect (Remarque: Elle ne fonctionne uniquement que sur Macintosh).

Processing est un langage de programmation basé sur la plateforme Java, il est donc possible de le programmer directement en langage Java. Processing est tout particulièrement adapté à la création graphique interactive. (http://fr.wikipedia.org/wiki/Processing)

OpenKinect est un ensemble de librairies dont le code source est libre et gratuit, et qui permettent à Kinect de fonctionner avec Windows, Linux et Mac (http://openkinect.org/wiki/Main_Page/fr).

Pour implémenter cette extension nous avons suivi le tutoriel ñKinect + Processing + Eclipseò (http://cs.smith.edu/dftwiki/index.php/Kinect_%2B_Processing_%2B_Eclipse)

Nous avons ensuite étendu les fonctionnalités de bases de la librairie de Daniel Shiffman afin de pouvoir bouger le curseur et déclencher un clic de la souris en fonction du mouvement détecté par Kinect.

V/ Réalisation et difficultés rencontrées

Le programme répond à toutes les attentes du sujet (jeu en mode console, interface graphique, jeu à plusieurs (en mode graphique), utilisation de Kinect, simulation de la soirée des

aventuriers de lôESIAL, joueurs utilisant différentes stratégies, hall of fame, calibrage de la roulette, ...)

On peut estimer notre temps de travail sur le projet à au moins 60h (conception, programmation et tests).

Nous avons rencontré de nombreux problèmes tout au long de l'élaboration du projet :
- Åintégration de subvertion à eclipse pour pouvoir lôutiliser en mode graphique.
- Åinsérer une image en image de fond.
- Åpositionnement des composants avec swing.
- Åtrouver une librairie permettant lôutilisation de kinect facilement.
- Åimplémenter du processing en java.
- Åredimensionner et coloriser les cellules dôun tableau (JTable).
- Åactualiser un tableau après une modification de son contenu (JTable).
- Åutilisation des images dans un executable .jar

Nous nôavons malheureusement pas eu le temps dôimplémenter une fenêtre permettant dôafficher une roulette en train de tourner.

Un seul bogue a été trouvé sur la version finale du projet (un jeton reste affiché lorsque lôon fait une mise interdite) cependant ce nôest quôun bogue mineur puisque quôil affecte uniquement lôaffichage, le déroulement du jeu nôest pas perturbé.

VI/ Conclusion

Le mini projet TOP nous a permis de mieux nous familiariser avec la Programmation Orientée Objet, et de mettre en pratique les différentes logiques de programmation enseignées jusquôà ce moment dans ce langage.

Il nous a permis également dôentreprendre les démarches dôun informaticien, de la conception à la programmation en passant par de nombreuses phases de tests.
La réalisation de lôinterface graphique nous a permis de nous familiariser avec la librairie *swing* et *awt*. Lôimplémentation de kinect nous a permis dôentreprendre des recherches et dôadapter des solutions dans un domaine très récent et peu documenté.

Bien que les difficultés rencontrées et les bogues étaient nombreux, le programme obtenu est bel et bien fonctionnel sans bogue majeur connu.

Pour conclure, le mini-projet TOP a été une bonne expérience pour nous, vues les nombreuses compétences acquises par chacun de nous.