

Synthèse IHM - BlackJack2022

Table des matières

1 Description de mon programme.....	2
Démarrage de l'application :.....	2
Démarrage de la Partie :.....	3
Fin de la partie :.....	3
1 Liste des options/particularités du jeu.....	4
2 Liste des options/particularités du développement.....	4

1 Description de mon programme

Démarrage de l'application :

Au lancement du programme, l'état du jeu est en MISE. Plusieurs actions sont réalisables par les joueurs, elles sont :

- Miser : Pour miser

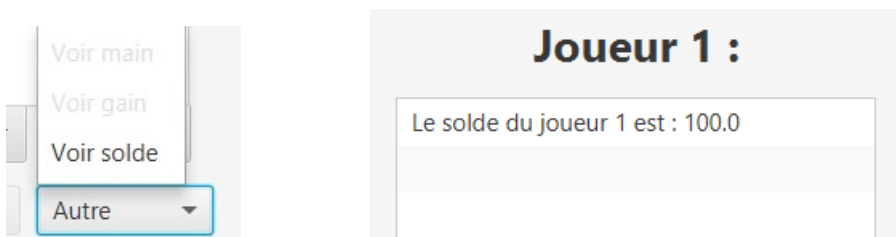


Le bouton « - » permet de baisser la mise

Le bouton « confirmer » permet de confirmer la mise

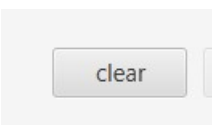
Le bouton « + » permet d'augmenter la mise

- Voir le solde :



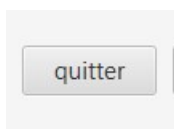
Permet de voir le solde du joueur

- Clear :



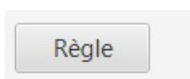
Permet de supprimer les messages des ListView

- Quitter :



Permet de quitter l'application

- Règles :

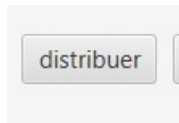


Permet de voir les règles du jeu

Démarrage de la Partie :

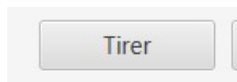
À cette étape du jeu, le l'état du jeu passe à JEU et différentes actions deviennent disponible pour les joueurs :

- Distribuer :



Permet de distribuer des cartes aux joueurs

- Tirer :



Permet de demander une cartes au croupier

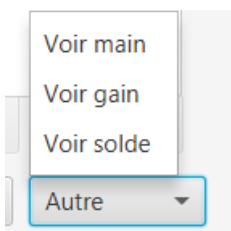
- Passer :



Permet de ne pas de demander de cartes au croupier

- Voir main :

- Voir gain :

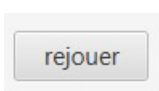


Permet de voir la main du joueur

Fin de la partie :

À cette étape, la partie vient d'être terminé car les joueur ont gagner ou perdu. L'état du jeu passe en GAIN, et certaines actions reste disponible :

- rejouer :



Permet de relancer une partie et d'obtenir ses gains ou perdre sa mise

2 Liste des options/particularités du jeu

```
@FXML
void augmenter1(ActionEvent event) {
```

Figure 1: fonction pour augmenter la mise du joueur

```
@FXML
void diminuerJ1(ActionEvent event) {
    int nb = Integer.parseInt( this.m
```

Figure 2: fonction pour diminuer la mise du joueur

```
@FXML
void confirmerJ1(ActionEvent event) {
    String info = ""
```

Figure 3: fonction pour confirmer la mise du joueur

```
@FXML
void distribuer(ActionEvent event) {
```

Figure 4: fonction pour distribuer des cartes

```
@FXML
void passerJ1(ActionEvent event) {
    String info = ""
```

Figure 5: fonction pour passer son tour

```
@FXML
void rejouer(ActionEvent event) {
    String info = "Nouvelle partie"
```

Figure 6: fonction pour rejouer une partie

```
@FXML
void tirerJ1(ActionEvent event) {
    String info = ""
```

3 Liste des options/particularités du développement

```
@FXML
void clear(ActionEvent event) {
```

Figure 7: fonction pour supprimer les messages

```
@FXML
void afficherRegle(ActionEvent event) {
```

Figure 8: fonction pour afficher les regles

```
void miseAJourEtat() {
```

Figure 9: fonction pour mettre a jour le textField de l'etat

```
@FXML
void quitter(ActionEvent event) {
    Alert alert = new Alert(AlertType
```

Figure 10: fonction pour quitter l'application

```
@FXML
void voirGainJ1(ActionEvent event) {
    String info = "Gain du joueur 1 :
```

Figure 11 fonction pour voir les gains du joueur

```
@FXML
void voirMainJ1(ActionEvent event) {
    String info = "Main du joueur 1 :
```

Figure 12: fonction pour voir la main des joueurs

```
@FXML
void voirSoldeJ1(ActionEvent event) {
    String info = "Solde du joueur 1 :
```

Figure 13: fonction pour voir le solde des joueurs