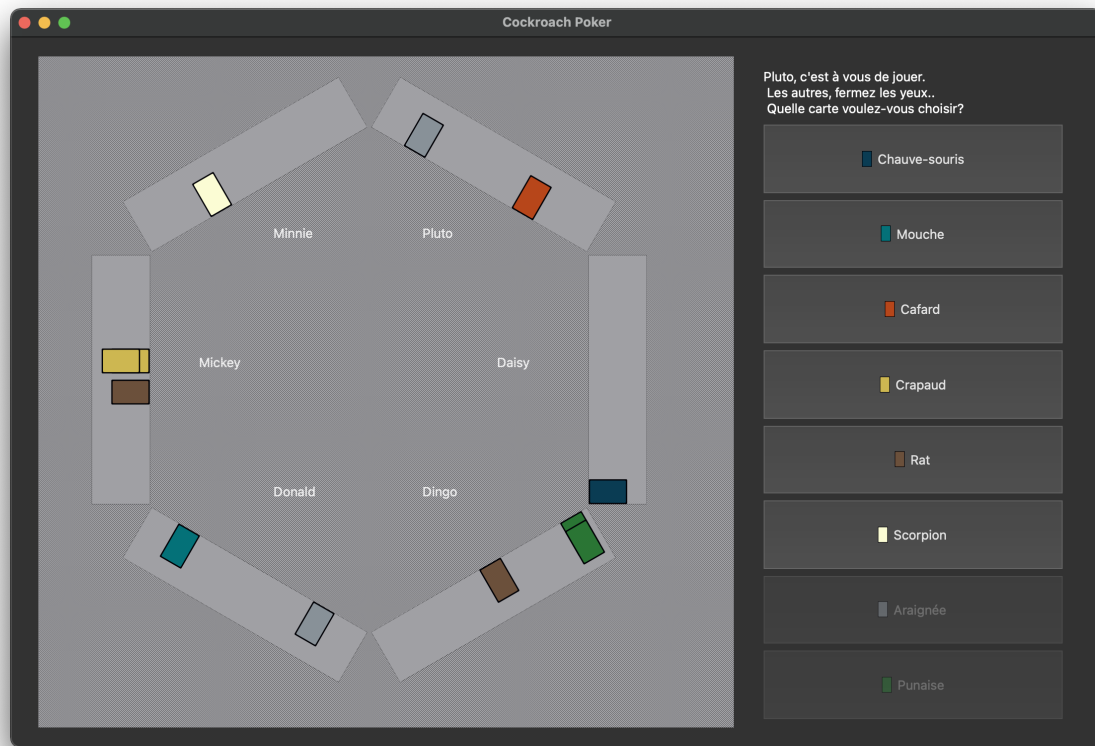


[DEV4] Le poker des cafards



Cockroach Poker is a reverse set collection game that has nothing to do with poker – except that the game is all about bluffing, with cards that show cockroaches, rats and stink bugs. The goal is to force another player to collect 4 of any one type of critter.

L'arborescence du projet

Le projet respecte la structure MVC ainsi que le pattern "Observateur/Observé"

```
.
├── src
│   ├── controller
│   │   ├── controller.cpp
│   │   └── controller.hpp
```

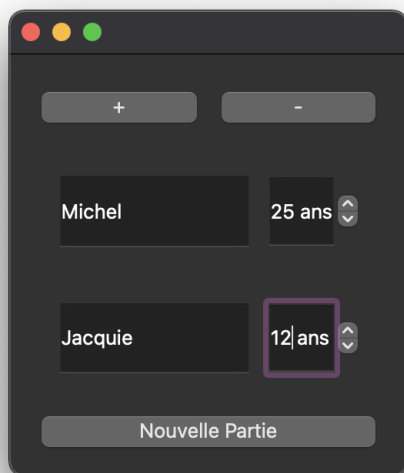


```
| | | cards
| | |   | cardcollectiontest.cpp
| | |   | cardtest.cpp
| | |   | game
| | |   |   | playertest.cpp
| | |   | maintest.cpp
| | CMakeLists.txt
| | LICENSE
| | Makefile
| | README.md
| | doxygen
```

Mode d'emploi

Pour la compilation, veuillez vous referer au README.md

Etape 1

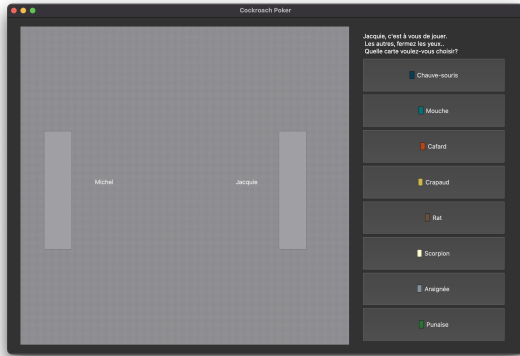


Entrez le noms des joueurs ainsi que l'age de chaque participant. Le plus jeune commencera la partie.

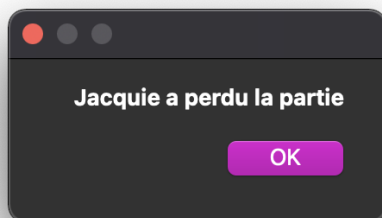
Cliquez sur "**Nouvelle Partie**"

Etape 2

Jouez en suivant les instructions sur la droite de l'ecran.



Etape 3



Celui qui aura 4 cartes identiques sera perdant. Son nom sera affiché sur l'écran.

(Egalement si il n'a plus de carte lorsque c'est son tour)

Etape 4

Recommencez autant de fois que vous voulez. Mais n'oubliez pas d'aller dormir de temps en temps.

Ecart par rapport à l'énoncé

- Une carte peut être transférée même si elle a fait le tour de tous les joueurs et également quand il n'y a que deux joueurs

Une justification des avertissements (warnings) qui subsistent

- Aucun warning a la compilation
- Warning de Qt sur MacOS (à l'exécution) lorsque la souris quitte la zone de la fenêtre.
- Fuites mémoires ⇒ CLEAN 🔥 , *aucun problème* détecté par Valgrind


Les problèmes et difficultés rencontrés lors de la réalisation du projet

Modele peu complexe, mais beaucoup d'événements à gérer dans la vue.. J'ai finalement décidé d'utiliser des lambdas expressions que je transmets de mon contrôleur à ma vue et je crée des signaux dans Qt que je relie à ces actions. Comme ça, la vue ne modifie jamais le modèle.

Pour ce qui est de l'observateur observé, j'ai décidé d'ajouter un champ pour spécifier la propriété qui a été update en m'inspirant de :

PropertyChangeSupport (Java SE 11 & JDK 11)

```
public class MyBean { private final PropertyChangeSupport pcs = new PropertyChangeSupport(this); public  
void addPropertyChangeListener(PropertyChangeListener listener) {  
this.pcs.addPropertyChangeListener(listener); } public void
```

 [https://docs.oracle.com/en/java/javase/11/docs/api/java.desktop/java/beans/PropertyChangeSupport.ht
ml](https://docs.oracle.com/en/java/javase/11/docs/api/java.desktop/java/beans/PropertyChangeSupport.html)

Une liste des bogues restant dans l'application

- Si on utilise deux fois un même nom pour deux joueurs différents, le jeu ne fonctionne pas..