

# Rapport du projet final PRO 2

DA: 1333099:

Jérémi Pedneault

Travail présenté à :

Monsieur Louis Marchand

Dans le cadre du cours :

Programmation Orientée par objet II

420-PRB-DM

Cégep de Drummondville

# Guide d'installation, compilation, configuration et utilisation

## **Installation:**

Pour faire fonctionner le jeu, il faut:

que l'exécutable soit dans le même répertoire que les fichiers ressources

Compilation:

Pour compiler le jeu, il faut premièrement le code source, disponible sur ce lien:

[https://github.com/jerped/jeu\\_de\\_8.git](https://github.com/jerped/jeu_de_8.git)

Deuxièmement, il faut avoir installé le logiciel EiffelStudio, disponible sur ce lien:

<http://sourceforge.net/projects/eiffelstudio/>

Troisièmement, il faut la librairie de jeu de Louis Marchand pour pouvoir compiler

le projet. Suivre les instructions sur ce fichier .PDF pour savoir comment faire:

<http://www.mediafire.com/view/9zp5ct7kb2bl9nb>

## Configuration:

Il n'y a pas de configurations vraiment en tant que tel. Comme toutes les valeurs sont prédéfinies directement dans le code, il n'y a pas d'options disponibles. À noter que le port par défaut pour l'application est le 25565, ouvert en TCP. En changeant le code directement, il est possible de modifier cette valeur pour définir le port voulu. Il faut simplement, si le code source est à disposition, ouvrir le fichier `ENGIN_DE_RESEAU` et changer la valeur à la ligne:

```
port : INTEGER = 25565
```

pour la valeur voulue. De plus, il est possible de changer les images du jeu en remplaçant celles par défaut dans le fichier `[ressources/images]`. Pour les fichiers audio, ils sont accessibles dans le répertoire `[ressources/musique]`. Il faut respecter les noms et le nombre de fichiers. Si il manque une image ou un fichier audio, l'application ne démarrera pas. Même chose si un des fichiers n'a pas le bon nom. Pour une expérience optimale, il est aussi recommandé de garder les images à ces tailles:

850px X 850px pour l'image de fond

77px X 107px pour les cartes et l'endos des cartes

## **Utilisation:**

Afin de simplifier ce guide d'utilisation, le terme client désignera le joueur qui a rejoint la partie, et le terme serveur désignera le joueur qui héberge la partie.

Une fois l'application lancée, il est possible soit d'héberger la partie, soit d'en rejoindre une. Il faut se déplacer à l'aide des flèches et appuyer sur « Return » pour sélectionner une option.

Si l'option de rejoindre est sélectionnée, il va falloir entrer l'adresse IP de l'adversaire, puis appuyer sur entrer. Dans le cas où c'est l'option d'héberger qui est sélectionnée, le jeu va figer et se mettre en attente d'une connexion d'un opposant. Il n'est malheureusement pas possible d'annuler l'attente d'un adversaire en ce moment. Une fois que le client a rejoint la partie, le jeu se lance de lui-même.

Après le lancement du jeu, l'application serveur va se figer et se mettre en attente de son tour à jouer. Une fois son tour arrivé, c'est l'application client qui se fige pour attendre son tour. Il est à noter que les actions de l'autre joueur n'apparaissent qu'une fois que celui-ci a terminé son tour.

Le jeu fonctionne comme suit:

Le joueur a à sa disposition un paquet de cartes, appelé main. Pour se déplacer dans la main, il faut utiliser les flèches gauche et droite du clavier. Une fois la bonne carte sélectionnée, il faut appuyer sur la touche « Return » pour jouer la carte.

L'action de jouer une carte termine le tour du joueur actuel. Il n'est donc pas possible de changer de décision en cours de route. Une fois la carte jouée, il n'est pas possible de la modifier.

## **Règlements**

Le but du jeu est de ne plus avoir de cartes en main.

Pour jouer une carte, celle-ci doit respecter l'une des règles suivantes:

avoir la même valeur que celle au centre (ex: jouer un 6 de trèfle sur un 6 de carreaux)

avoir la même sorte que celle au centre (ex: jouer du trèfle sur du trèfle)

Les seules cartes où ces règles ne s'appliquent pas sont les 2 jokers et les cartes 8, qui peuvent être jouées à n'importe quel moment.

S'il n'est pas possible de jouer une carte, le joueur doit obligatoirement en prendre une dans le paquet de cartes au centre

De plus, certaines cartes ont un effet spécial:

les 2 font piger 2 cartes et font passer le tour du joueur suivant sauf:

le 2 de pique qui fait piger 4 cartes au joueur suivant

les 10 font changer le sens du jeu (horaire ou antihoraire)

les 11 font passer le tour du joueur suivant

les jokers font piger 5 cartes au joueur suivant et permettent de changer la sorte qu'il est possible de jouer

les 8 permettent de changer aussi la sorte qu'il est possible de jouer

les As permettent de jouer toutes les cartes de même sorte d'un seul coup

## Évolution du projet

J'ai eu l'idée de ce projet après une soirée se carte en famille. Au début, le projet était beaucoup plus gros. J'avais en tête d'essayer de permettre à jusqu'à 8 joueurs de jouer sur la même partie, de permettre d'ajouter des paquets de cartes et beaucoup d'autres options. Le problème, c'est qu'au début la motivation n'était pas vraiment là, donc ce n'est qu'après la mi-session que le projet a vraiment commencé à avancer. Aussi, avec tous les autres travaux dans les autres cours et ma mauvaise habitude à procrastiner et à attendre vers la fin m'ont obligé à rétrécir de beaucoup le projet. Maintenant, il n'est possible de jouer des parties

qu'en 1 contre 1, les règlements ne sont pas aussi flexibles que je l'aurais voulu et il n'est plus possible d'ajouter de la musique à volonté.

Au début, j'ai commencé par chercher les images des cartes et du fond du jeu. J'ai passé peut-être un bon 2 semaines avant d'avoir trouvé le thème que je cherchais.

Après ça, une fois le plateau de jeu affiché, je n'ai rien touché pendant un bon 2 ou 3 semaines. Avant la remise 2, je me suis efforcé d'avoir au moins le graphique et du son qui jouait, ce qui a été le cas. Ensuite, rien n'a vraiment changé avant les 3 dernières semaines de la session. Une fois l'écran titre terminé, j'ai pu

m'embarquer dans la fonctionnalité réseau. En travaillant dessus et après avoir interrogé mon enseignant, il est devenu évident qu'il allait me falloir du multithread.

Maintenant que le réseau fonctionne et que le multithread est complété, le reste du jeu a été beaucoup plus rapide à travailler. Par contre, il manque encore l'effet spécial des cartes qui n'est pas fait.

## Exemple des « designs » objets demandés

### **Constructeur:**

Je m'en sers pour presque toutes mes classes. Le meilleur exemple est celui de la classe ECRAN\_JEU, où je ne fais pas qu'instancier l'objet, je crée aussi les joueurs et je fais la génération des paquets de cartes.

### **Héritage Simple:**

L'exemple parfait est celui de mes cartes, où j'hérite de la classe CARTE, mais en changeant l'effet.

### **Héritage Multiple:**

Aucun, malheureusement :X

### **Polymorphisme:**

Mes cartes sont gérées de manière à ce que je puisse faire un paquet carte sans avoir à spécifier le type de chacune. Ainsi, mon paquet de cartes est une `ARRAYED_LIST[CARTE]`. Ainsi, je n'ai qu'à faire `mon_paquet.at(i).effet` pour exécuter l'effet de n'importe quelle carte peu importe le type de celle-ci.

### **Classe Abstraite:**

La seule classe abstraite est CARTE. Toutes mes différentes cartes en héritent et utilisent la fonction effet, qui est définie dans CARTE, mais en changeant le contenu en fonction de la carte jouée.



### **Méthode Redéfinie:**

Dans ma classe JOKER, je redéfinis le constructeur de la classe CARTE pour attribuer différentes variables.

### **Autocritique:**

Je pense que le jeu en tant que tel était une excellente idée, tout peut être facilement adapté pour respecter les demandes de Louis. J'aurai dû implanter l'héritage multiple au début du projet par contre, puisque maintenant ça reviendrait à refaire une grosse partie du code. De plus, m'être pris plus tôt, j'aurai eu le temps de programmer les effets des cartes. Présentement, on peut les jouer, mais elles ne font rien de spécial. Sinon, je trouve que mon jeu est vraiment affreux, l'affichage a l'air botché (surtout pour l'écran titre) et je dois vraiment améliorer le degré de détail dans mes interfaces. Parfois, la musique arrête de jouer ou le joueur peut faire plus qu'une action, ce qui est vraiment étrange et inexplicable. À part ça, je suis content du résultat, même si je dois m'appliquer plus sur mon code et apprendre à travailler tout le long de la session au lieu d'attendre à la fin.