Plateforme Java

Nicolas DEVENET & Valériane JEAN

nicolas@devenet.info
jean.valeriane@gmail.com

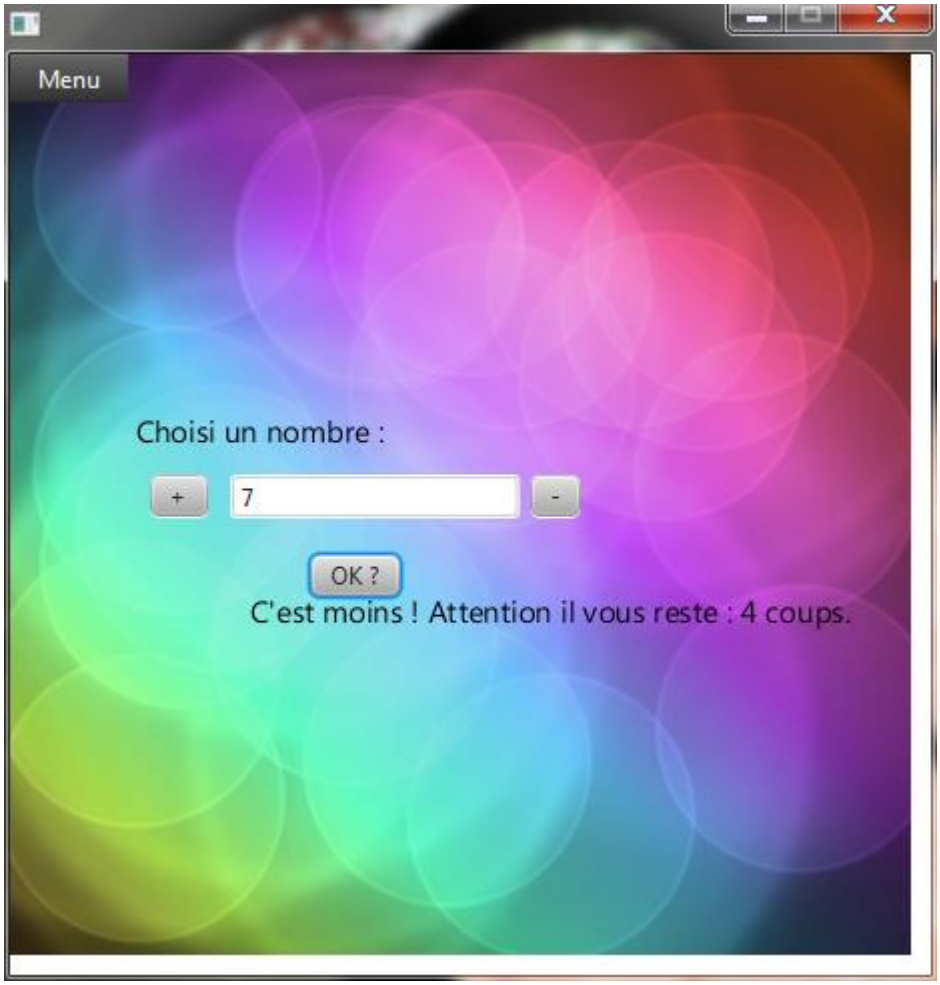
Introduction

Le but du projet est de créer quelques petits jeux pour enfants en utilisant, entre autre, les librairies et nouveautés vues en cours, et de coder une interface permettant de lancer ces jeux développés de manière indépendante.

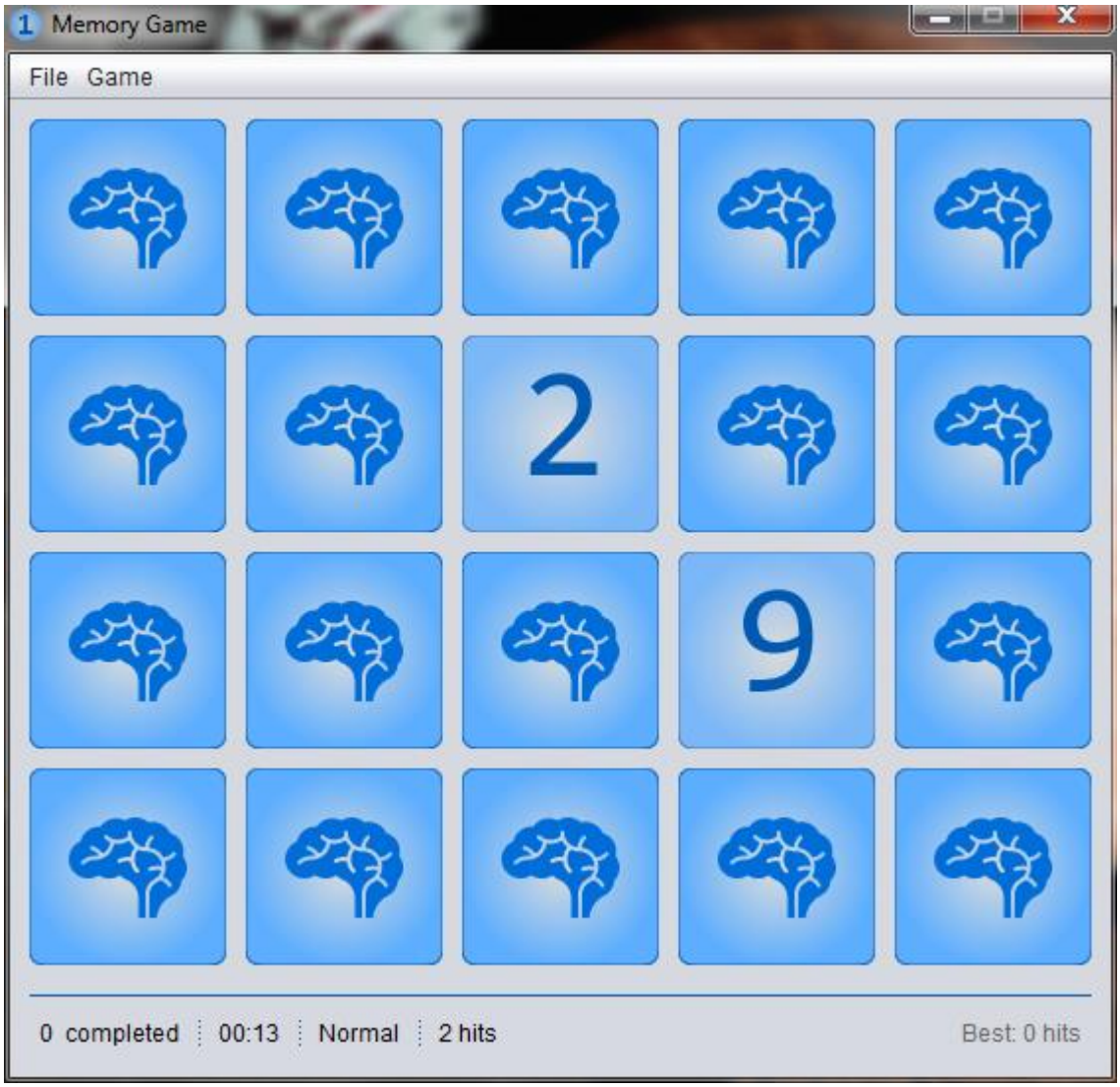
Présentation des jeux

Nous avons décidés de coder les jeux suivants :

- POM : dont le but consiste à deviner par un nombre d'étapes limité un nombre choisi aléatoirement par le jeu, ce dernier donnant pour seul indice si c'est plus ou moins que notre estimation ;

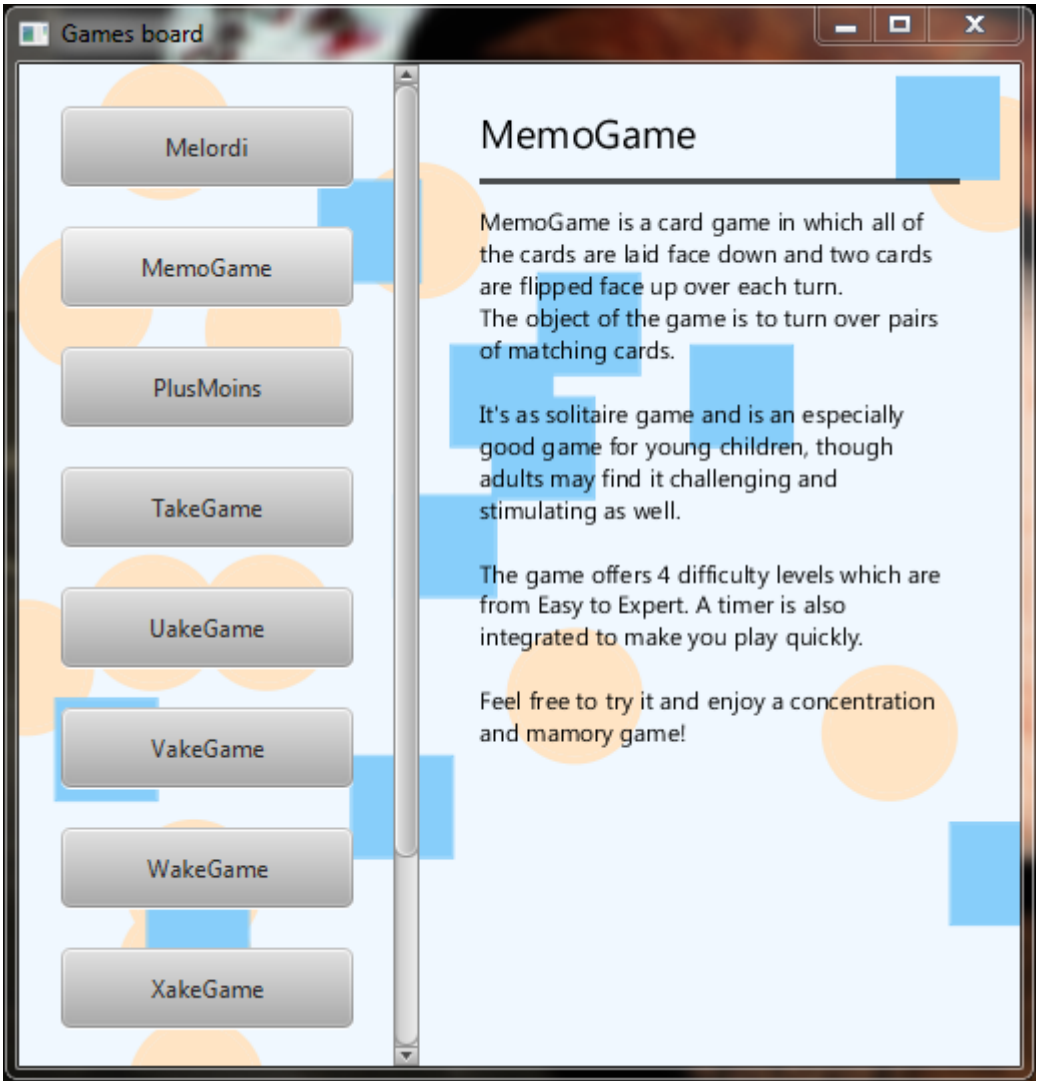


- MemoGame : dont le but est de trouver en moins de coups possibles les paires d'une même carte, sachant que les cartes sont disposées aléatoirement face cachées et qu'une fois deux cartes différentes retournées face visibles, elles sont remises face cachée pour continuer ;



- Melordi : qui consiste en un petit piano à 8 touches générant des sons lors de l'appui sur les touches.

Présentation du luncher



Sur la gauche, on trouve une liste déroulante des jeux répertoriés dans un dossier. Lors du survol sur le bouton d'un jeu, la description de celui-ci est affichée. Un clic sur le bouton du jeu choisi permet de le lancer.

Mise en œuvre

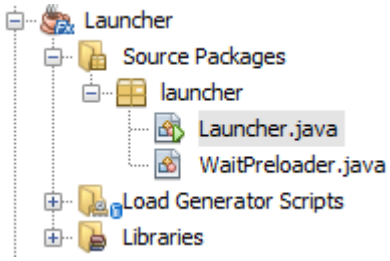
Tous les projets ont été réalisés avec NetBeans.

Pour la réalisation de l'interface utilisateur, tous les jeux ainsi que le luncher ont été codés en JavaFX, sauf le jeu MemoGame qui a été codé en Swing.

Le modèle métier est séparé de l'implémentation graphique.

Luncher

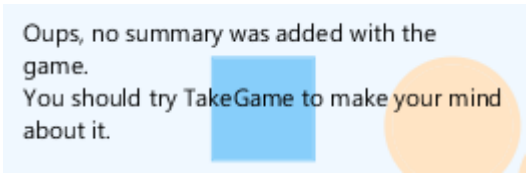
Le luncher, lors de son lancement, va automatiquement et dynamiquement répertoriés tous les fichiers « .jar » présents dans le dossier « games ».
Comme la liste des jeux disponibles peut-être assez longue, nous avons optés pour une liste déroulante permettant ainsi une navigation et un aperçu faciles.



S'en suit ensuite la création des boutons, et de leur gestion d'évènements. Le panneau de droite est initialement sans texte.

Au survol d'un bouton, ce dernier insère dans le panneau de droite le nom de son jeu ainsi que la description qu'il a pu trouver le concernant.

La description du jeu est stockée dans un fichier texte nommé « summary.txt » et qui est stocké directement dans le JAR, dans le package racine « res ».
Le fichier est ainsi directement généré lors de la compilation du JAR du jeu, il n'est pas nécessaire d'ajouter un fichier supplémentaire dans le dossier « games » du luncher pour en bénéficier, et ce le luncher qui va chercher le fichier dans le JAR.



Notons que si le fichier n'a pas été trouvé, un texte générique lui est substitué.

Lors d'un clic sur un bouton, le luncher exécute la commande pour lancer le fichier du jeu correspondant. Comme il peut y avoir un délai entre le moment du clic et le moment où l'application est effectivement lancée et affichée, nous avons ajouté un texte d'information en bas du panneau de droite pour donner un retour à l'utilisateur et l'informer que son action a bien été prise en compte.

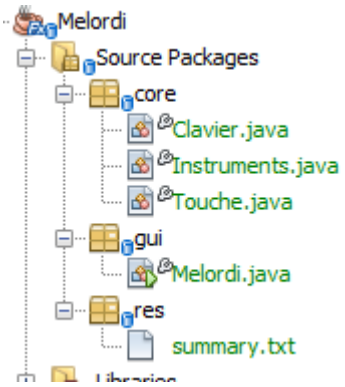
Le luncher est donc générique dans le sens où il suffit que des jeux au format « .jar » soient présents dans le dossier pour que le luncher les reconnaisse.
Notons que la liste des jeux n'est plus dynamiquement mise à jour une fois l'application chargée, mais il suffit de la relancer pour que les modifiions au dossier « games » soient prises en compte.

Il a aussi été ajouté un « Preloader » qui se charge (rapidement) avant l'application et permet de voir la progression de son chargement. Cependant, avec une machine récente, il est parfois difficile de note sa présence.

Mélordi

Il s'agit d'une petite application destinée aux jeunes enfants afin qu'il ait une première notion de musique.

Dans le package « core », on retrouve tous les éléments qui compose un piano



La classe Touche s'occupe de la gestion des touches, c'est elle qui implémente le son lors de l'appui sur la touche et des effets graphiques liés aussi à l'appui et au relâchement de la touche.

La classe Clavier correspond à l'ensemble de toutes les touches à la manière d'un clavier de piano. Cette classe permet la construction du tableau de touches.

La classe Instruments permet de définir plusieurs instruments de musique pour l'instant seul le piano est implémenté mais un piste d'amélioration serait d'y ajouter d'autre instruments comme le violon, la guitare, …

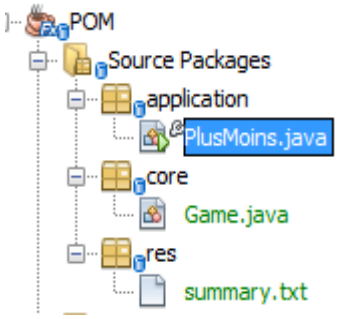
POM

Le jeu POM est un jeu de réflexion dans lequel il faut trouver le nombre mystère choisi aléatoirement par l'ordinateur.

Il existe plusieurs niveaux :

- Facile : le nombre mystère se trouve entre 0 et 10 et le joueur a 5 coups pour trouver le nombre mystère.
- Moyen : le nombre mystère se trouve entre 0 et 100 et le joueur a 9 coups pour trouver le nombre mystère.
- Difficile : le nombre mystère se trouve entre 0 et 1000 et le joueur a 9 coups pour trouver le nombre mystère.

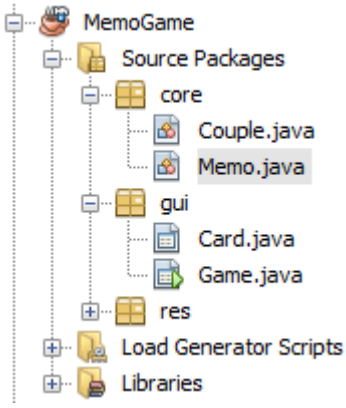
Le jeu contient deux packages :



- Application : contient la classe PlusMoins qui produit la fenêtre graphique et qui s'occupe de la communication de tous les composants graphiques. Lors du lancement d'un niveau cette classe crée tous les composants graphiques du jeu.
- Core : contient le modèle métier du jeu. La classe Game effectue tous les calculs liés à la gestion du jeu, notamment c'est elle qui s'occupe de dire quand le joueur gagne ou perd et quand le joueur valide une réponse. C'est aussi via cette classe que le jeu répond si c'est plus ou si c'est moins.

MemoGame

C'est un jeu de mémory qui se joue seul.



Un Memo est constitué d'un ensemble de couples, et ces derniers sont constitués de deux cartes identiques (dans le sens où elle représente la même valeur).
La classe Game permet ainsi de lancer un jeu Memo.

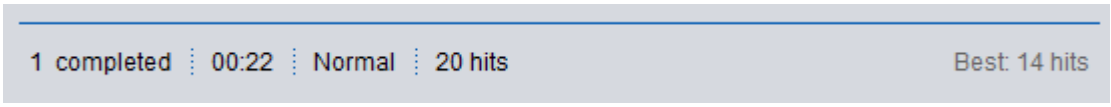
Le package « core » contient l'implémentation de la représentation du jeu, le package « gui » contient la fenêtre principale et la représentation d'une carte (dont l'implémentation de sa partie de jeu a été codée à l'intérieur), et le package « res » contient les ressources nécessaires au jeu (images, description, …).

Il y a possibilité de jouer selon 4 niveaux de « easy » à « expert », ce qui implique un nombre de cartes plus important à retrouver (respectivement de 3 à 24 paires).

Lors d'une nouvelle partie, toutes les cartes sont retournées face cachées. Lors d'un clic sur une carte, l'événement de retournement vers la face visible n'est faisable que si aucune carte n'a déjà été retournée ou si une seule carte l'est.
Dans ce dernier cas, une comparaison est alors effectuée pour s'avoir s'il s'agit de la même paire. S'il ne s'agit pas de la même paire, un clic sur le dos d'une autre carte provoquera leur retournement vers leur face cachée et permettra ainsi au joueur de continuer. S'il s'agit de la même paire, les cartes restent figées face ouverte, et le joueur continue.

Une partie se termine quand toutes les paires ont été retrouvées et le joueur est notifié qu'il a gagné avec le temps mis et le nombre de coups utilisés.

Une ToolBar située en bas permet au joueur de connaître le niveau actuel de la partie, de suivre le temps mis et le nombre de coups utilisés depuis le début de la partie.



Il connaît aussi le nombre de partie complète finie, et parmi toutes les parties terminées (tous niveaux confondus), le nombre minimum de coup qu'il a fait pour gagner sa meilleure partie.

Conclusion

L'objectif de réaliser un lanceur de jeu est atteint, nous avons voulu offrir plusieurs jeux de différents niveaux de difficulté (Melordi, POM, MemoGame). Les joueurs pourront facilement ajouter d'autres jeux en glissant des « .jar » dans le dossier « games » de l'application Launcher.