

L'objectif de ce travail qui va s'étaler sur plusieurs semaines est double :

- _ Découvrir JavaFX
- _ Réaliser un projet avec JavaFX

C'est une situation que vous rencontrerez certainement dans votre vie professionnelle : découvrir un nouvel outil qui suit des principes similaires à un que vous connaissez déjà et apprendre à l'utiliser pour réaliser un nouveau projet.

Pour atteindre ces objectifs vous allez procéder en deux temps. Dans un premier, vous allez vous documenter sur JavaFX et réaliser quelques exemples simples et ensuite vous vous attaquerez au projet lui-même.

Vous travaillerez au choix seul ou par deux et remettrez votre travail (exercices et projet) après la dernière séance consacrée au projet (le vendredi 5 mars 2021). Cela constituera la dernière évaluation.

Vous ne devez pas ici utiliser de générateur d'interfaces mais bien écrire le code vous-même.

Découverte JavaFX

Il existe plusieurs manières de construire des interfaces via JavaFX, soit en programmant l'ensemble de votre application en Java (c'est ce que vous allez faire pour ce travail), soit en utilisant aussi des fichiers FXML (un dérivé de XML) pour décrire votre interface.

Voici quelques éléments à garder en tête tout au long de ce travail :

- Un des objectifs est de voir comment transposer ce que vous avez appris en Swing vers JavaFX et donc de répondre aux questions suivantes :
 - _ Dans une application JavaFX, quelle est ou quelles sont les fenêtre top-niveau (équivalent notamment de JFrame en Swing)
 - _ Quelle est la structure d'une telle fenêtre, que peut-on mettre dedans ?
 - _ De quelles composants graphiques dispose-t-on ? (en Swing, des conteneurs comme des JPanel et des composants graphiques simples comme des JLabel ...)
 - _ De quels algorithmes de placement dispose-t-on et comment les utilise-t-on ? (en Swing, BorderLayout, GridLayout etc.)
 - _ Comment gère-t-on les événements utilisateur (en Swing, ajout d'écouteurs sur des composants)
 - _ Quels sont les événements proposés par JavaFX ?
 - _ Comment peut-on dessiner sur un composant ?
 - _ Comment est calculée la taille des composants graphiques, que se passe-t-il lorsqu'on redimensionne une fenêtre ?
- Voici quelques liens où vous pouvez trouver des informations (la plupart de ces liens est en relation avec la version 8 de javafx mais les informations sont encore valides dans les version ultérieures) :
 - _ Documentation disponible sur le site d'Oracle : https://docs.oracle.com/javase/8/javafx/get-started-tutorial/javafx_get_started.htm et <https://docs.oracle.com/javase/8/javase-clienttechnologies.htm> avec aussi des éléments sur :

- les gestionnaires de placement : <https://docs.oracle.com/javase/8/javafx/layout-tutorial/>
 - la gestion des événements : <https://docs.oracle.com/javase/8/javafx/events-tutorial/events.htm>
 - le dessin de composants : <https://docs.oracle.com/javase/8/javafx/graphics-tutorial/canvas.htm#JFXGR214>
 - des démonstrations : https://docs.oracle.com/javase/8/javafx/get-started-tutorial/get_start_apps.htm
- Les plus récentes versions de JavaFX sont accessibles sur <https://openjfx.io>

Il existe bien d'autres sources sur JavaFX que vous êtes bien entendu libres de consulter.

À noter que JavaFX ne s'utilise pas de la même manière selon la version de Java que vous utilisez. Avant la version 11, il faisait partie du JDK, depuis il faut l'installer en plus. Vous pouvez trouver des instructions pour IntelliJ (que vous pouvez adapter pour tout autre environnement de développement) sur <https://www.jetbrains.com/help/idea/javafx.html>.

Une fois les choses bien installées, vous pouvez créer vos projets JavaFX.

Pour commencer, voici quelques exercices, que vous devriez pouvoir réaliser en vous aidant de la documentation, des tutoriels et du code exemple fourni sur UPdago.

Exercice 1 : Hello World

En suivant le tutorial https://docs.oracle.com/javafx/2/get_started/hello_world.htm, découvrez les bases et écrivez votre première application JavaFX

Exercice 2 : Placement

Reprenez le code Swing, fourni sur UPdago pour la feuille 9 (Section Retour des événements), `SemanticEvents.java` et créez un composant JavaFX dérivant d'un **BorderPane** qui soit composé d'éléments similaires agencés de la même manière.

Changez les textes des 3 boutons de la zone centrale (par exemple, indiquez "bouton 1", "bouton 2" et "bouton 3") et faites en sorte que lorsqu'on clique sur n'importe lequel d'entre eux son texte soit affiché dans le label de la zone nord.

Pour tester votre programme, écrivez, en vous inspirant du premier exercice, une classe qui dérive de la classe **Application** qui permet d'afficher le composant que vous venez de créer.

Exercice 3 : Dessin

Écrivez un composant qui dérive de la classe **Region**, avec pour taille préférée 100 par 100 et sur lequel est dessiné soit un carré vert, soit un disque rouge, selon l'argument qui a été passé à son constructeur.

Pour tester votre programme, écrivez une classe qui dérive de la classe **Application** qui affiche deux instances du composant que vous venez de créer, l'une avec le dessin du carré vert, l'autre avec le dessin du disque rouge.

Projet

Le projet consiste à créer un jeu de Memory : on dispose d'un ensemble d'images allant par paires. Tous les images sont mélangées et étalées face cachée, chaque joueur, à son tour, retourne 2 cartes. Si elles sont identiques, elles sont retirées du jeu et il marque un point. Si elles sont différentes, elles sont à nouveau retournées faces cachées. Lorsque toutes les paires ont été retrouvées, le gagnant est le joueur comptabilisant le plus de points.

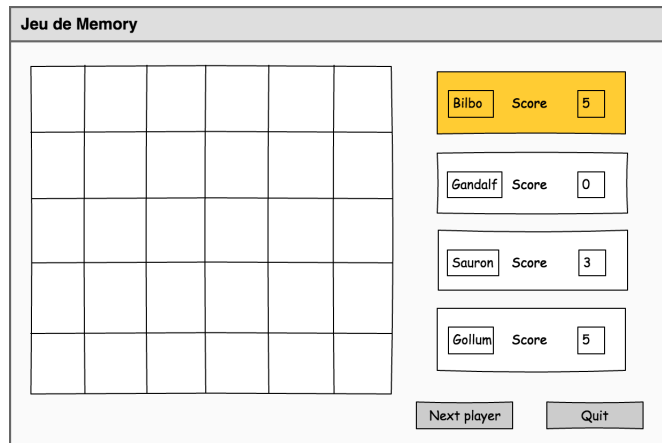


FIGURE 1 – Exemple de ce à quoi pourrait ressembler la fenêtre principale du jeu, ici le joueur dont c’est le tour est signalé par un fond jaune.

Le projet comporte deux niveaux : le premier que vous devez tous avoir fait dans son intégralité. Le second propose des extensions. Vous devez en choisir au moins une et la réaliser.

Premier niveau

Lors du lancement du jeu, une fenêtre doit permettre de choisir le nombre de joueurs et le nombre de paires de cartes pour la partie (vous pouvez laisser le choix entre des valeurs que vous aurez prédéfinies).

Pendant le jeu, la fenêtre principale du jeu permet de visualiser le plateau (souvent les cartes sont rangées comme sur une grille, mais vous pouvez les disposer différemment si vous le souhaitez), le score actuel des différents joueurs, ainsi que le joueur dont c’est le tour. Un bouton permet de quitter l’application, un autre de donner la main au joueur suivant.

L’interface pourrait ressembler à la figure 1.

Les noms des joueurs doivent pouvoir être renseignés, soit avant le début du jeu dans une fenêtre ad hoc, soit dans la fenêtre principale.

Lorsqu’un des joueurs a gagné, la possibilité est offerte de rejouer.

Sur les cartes, vous pouvez mettre des images, des dessins ou un mélange des deux. Les cartes n’ont pas besoin d’être rectangulaires. Si vous utilisez des images, vous pourrez tirer profit du tutoriel Oracle sur la manière de créer des icônes (<https://docs.oracle.com/javase/tutorial/uiswing/components/icon.html>) et d’inclure des images dans un projet (<https://docs.oracle.com/javase/tutorial/uiswing/components/icon.html#getresource>)¹.

Pour corser un peu la partie, le joueur qui a le plus petit score peut, lors de son tour échanger deux cartes de son choix sur le plateau (sans se cacher des autres joueurs). Cela peut se faire, au choix à l’aide de la souris, ou bien avec des raccourcis clavier (pour utiliser des raccourcis clavier, il peut être utile de désigner les cartes par leurs coordonnées sur le plateau comme dans un jeu de bataille navale par exemple et de faire apparaître cette information sur la fenêtre).

1. Si vos images ne sont pas incluses de la bonne façon, je ne les verrai pas quand je testerai votre application.

Second niveau

Choisir parmi :

- ajout d'un menu avec différentes actions :
 - abandonner la partie
 - sauvegarder la partie
 - mélanger les cartes et recommencer
 - changer l'apparence du jeu (couleur, police de caractère, etc.)
- gestion d'une bibliothèque de cartes :
 - pouvoir choisir les cartes parmi une bibliothèque d'images ou de dessins existant
 - pouvoir ajouter des éléments dans cette bibliothèque
- ajout d'une autre version du jeu. Cette fois, outre les paires de cartes, peuvent être présentes des cartes spéciales dont le nombre est fixé au début de la partie. Chaque fois qu'un joueur retourne une carte spéciale, une marque est faite en face de son nom, lorsqu'il en a retourné 3 il perd un point. Ajoutez ce mode de jeu et modifiez l'application pour que l'on puisse choisir de jouer à l'une ou l'autre version.

Consignes générales

Vous rendrez, dans une archive (tar.gz ou zip), un répertoire **NOM1_NOM2** contenant votre code (dans un sous-répertoire **src/**) ainsi qu'un (petit) rapport en pdf et un manuel utilisateur pour le projet (dans un sous-répertoire **documentation/**). Le code sera développé dans les packages **training** pour les exercices et **memory** pour le projet.

On attend un code modulaire et commenté. Pour le projet, il peut être judicieux de définir plusieurs composants graphiques *ad hoc* héritant de composants standards de JavaFX (par exemple, un composant pour un panneau de score, un composant pour le plateau de jeu, ...).

Pour le projet enfin sont attendus :

- un manuel utilisateur qui indique comment lancer l'application (où est la classe principale, quels sont les éventuels arguments nécessaires, etc.), quelles sont les interactions possibles avec l'application etc.,
- un petit rapport de quelques pages présentant l'architecture du programme et toutes les informations utiles pour un programmeur souhaitant reprendre votre projet et l'enrichir.

Dans le répertoire Documentation, vous fournirez également une bibliographie (plus probablement une « webographie ») contenant l'ensemble des sources sur lesquelles, vous vous êtes appuyés pour réaliser ce travail.