DEVIDAL Antonin

Preuves projet 18 : Bataille navale

1) Je maitrise les règles de nommages en JAVA

Chaque nom de classe commence par une majuscule, le nom des méthodes est en minuscule tout comme les attributs d'une classe ou les packages.

2) Je sais binder bidirectionnellement deux propriétés JavaFX.

3) Je sais binder unidirectionnellement deux propriétés JavaFX.

J'ai utilisé plusieurs Property pour pouvoir binder des variables à des éléments JavaFX.

➔ GameViewComputer ligne 93

```
dialogu.textProperty().bind(game.texte1Property());
```

4) Je sais coder une classe Java en respectant des contraintes de qualité de lecture de code.

Chacune des classes à un seul et unique rôle, facilitant la compréhension et l'utilisation du code. Elles respectent au mieux les principes SOLID

5) Je sais contraindre les éléments de ma vue, avec du binding FXML.

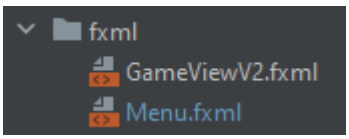
Ce texte est bindé sur le texte du bouton sélectionné dans la vue.

➔ Menu.fxml ligne 39

```
text="${difficulte.selectedToggle.text}"
```

6) Je sais définir une CellFactory fabriquant des cellules qui se mettent à jour au changement du modèle.

7) Je sais développer une application graphique en JavaFX en utilisant FXML.



Plusieurs vues ont été créées en FXML.

8) Je sais éviter la duplication de code.

J'ai créé des méthodes pour pouvoir éviter d'avoir de la redondance dans mon code

9) Je sais hiérarchiser mes classes pour spécialiser leur comportement

J'ai utilisé l'abstraction pour la classe Computer pour ensuite créer des classes filles ayant un comportement différent.

10) Je sais intercepter des évènements en provenance de la fenêtre JavaFX.

J'ai intercepté des évènements provenant de la fenêtre JavaFX comme l'appui d'un bouton.

➔ view.Menu.java ligne 136

```
@FXML
private void clickJoinButton(ActionEvent actionEvent) {
```

11) Je sais maintenir, dans un projet, une responsabilité unique pour chacune de mes classes.

Chaque classe à une responsabilité unique

12) Je sais gérer la persistance de mon modèle.

13) Je sais surveiller l'élément sélectionné dans un composant affichant un ensemble de données.

14) Je sais utiliser à mon avantage le polymorphisme.

Je downcast les classes ComputerEasy et ComputerNormal en Computer pour pouvoir les utiliser au même endroit sans avoir à réécrire du code.

➔ ComputerManager ligne 32

```
private Computer  computer;
computer = new ComputerEasy();
```

15) Je sais utiliser certains composants simples que me propose JavaFX.

J'ai utilisé des boutons dans ma page Menu.

➔ Menu.fxml ligne 59

```
<Label text="Join a local game: ">
```

16) Je sais utiliser certains layout que me propose JavaFX.

J'ai utilisé des verticalBox et des horizontalBox dans la fenêtre Menu

➔ Menu.fxml ligne 19

```
<VBox alignment="CENTER" prefHeight="200.0" prefWidth="100.0" BorderPane.alignment="CENTER">
   <children...>
</VBox>
```

17) Je sais utiliser GIT pour travailler avec mon binôme sur le projet.

J'ai utilisé GIT afin de pouvoir modifier mon travail depuis plusieurs poste (j'ai pas de binôme ☹). Et j'ai créé plusieurs branches correspondant à des modifications majeures de code.

18) Je sais utiliser le type statique adéquat pour mes attributs ou variables.

J'ai utilisé le type statique pour l'attribut IMAGESBATEAUX de ma classe Board pour ne pas avoir à définir le chemin des images à chaque création de cet objet.

➔ Board ligne 25

```
public transient static final Map<String,String> IMAGESBATEAUX= new HashMap<String, String>() {{
    put("front","/images/frontBoat.png");    // Coté de bateau
    put("body","/images/boatChest.png");     // Corp du bateau
    put("target","/images/target.png");      // Quand le bateau est touché
    put("redCross","/images/redCross.png"); // Quand on loupe un tire
}};
```

19) Je sais utiliser les collections.

J'ai utilisé une Map dans la classe Board pour pouvoir stocker les chemins des images utilisées.

➔ Board ligne 25

```
public transient static final Map<String,String> IMAGESBATEAUX= new HashMap<String, String>() {{
    put("front","/images/frontBoat.png");    // Coté de bateau
    put("body","/images/boatChest.png");     // Corp du bateau
    put("target","/images/target.png");      // Quand le bateau est touché
    put("redCross","/images/redCross.png"); // Quand on loupe un tire
}};
```

20) Je sais utiliser les différents composants complexes (listes, combo…) que me propose JavaFX.

Dans le menu, j'ai utilisé des radioButton pour pouvoir choisir la difficulté de l'ordinateur

➔ Menu.fxml ligne 27

```
<fx:define>
   <ToggleGroup fx:id="difficulte" />
</fx:define>
<RadioButton fx:id="RBfacile" mnemonicParsing="false" selected="true" text="Facile">
   <HBox.margin>
      <Insets left="25.0" right="25.0" />
   </HBox.margin>
</RadioButton>
<RadioButton fx:id="RBmoyen" mnemonicParsing="false" text="Moyen">
   <HBox.margin>
      <Insets left="25.0" right="25.0" />
   </HBox.margin>
</RadioButton>
```

21) Je sais utiliser les lambda-expression.

J'ai utilisé les lambda-expression pour pouvoir factoriser du code notamment lors de l'utilisation de la méthode runLater.

➔ Manager ligne 157

```
Platform.runLater(() ->setTexte1("Opponent won"));
```

22) Je sais utiliser les listes observables de JavaFX.

23) Je sais utiliser un convertisseur lors d'un bind entre deux propriétés JavaFX.

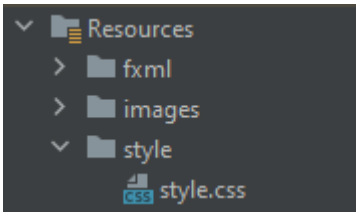
J'ai utilisé un convertisseur lors d'un bind pour pouvoir convertir un String en une image

➔ GameViewComputer ligne 245

```
imageView.imageProperty().bind(Bindings.createObjectBinding(() ->
    {
        if(c.getBoatImage() ==null)
            return null;
        return new Image(c.getBoatImage());
    } ,c.boatImageProperty()));
```

24) Je sais utiliser un fichier CSS pour styler mon application JavaFX.

J'ai utilisé un fichier style.css afin de styler mon application.



25) Je sais utiliser un formateur lors d'un bind entre deux propriétés JavaFX.

J’ai utilisé un formatteur lors d’un bind.

➔ GameViewComputer ligne 64