

Meunier Killian
Carpentier Tom

Oiram

Bug:

Lors de l'exécution du .jar, lorsque nous finissons un niveau une exception est levée "location is not set". Or l'exécution sur IntelliJ marche parfaitement.

Contexte:

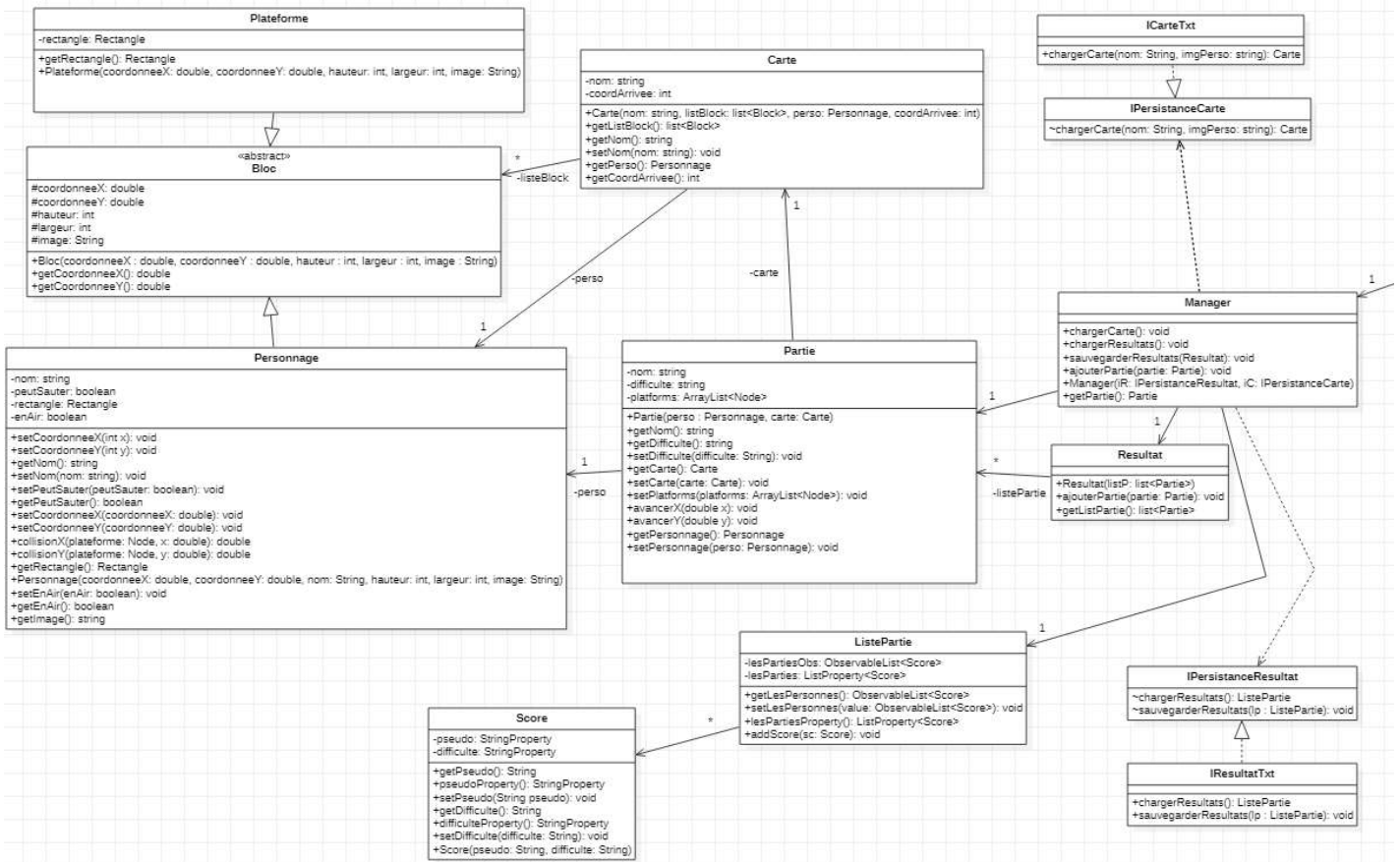
Oiram est un jeu de plateforme 2D. On peut y jouer:

-En solo: Le but du jeu est d'atteindre la fin du niveau sans mourir, pour cela on doit éviter des pièges et des trous.

On peut déplacer le personnage à droite et à gauche, et même sauter (le niveau se déroule de gauche à droite).

Le joueur pourra accéder à plusieurs options, comme choisir la difficulté du niveau, changer la langue, choisir le thème de l'application et du niveau.

Point de vue du modèle:



Point de vue de la vue:

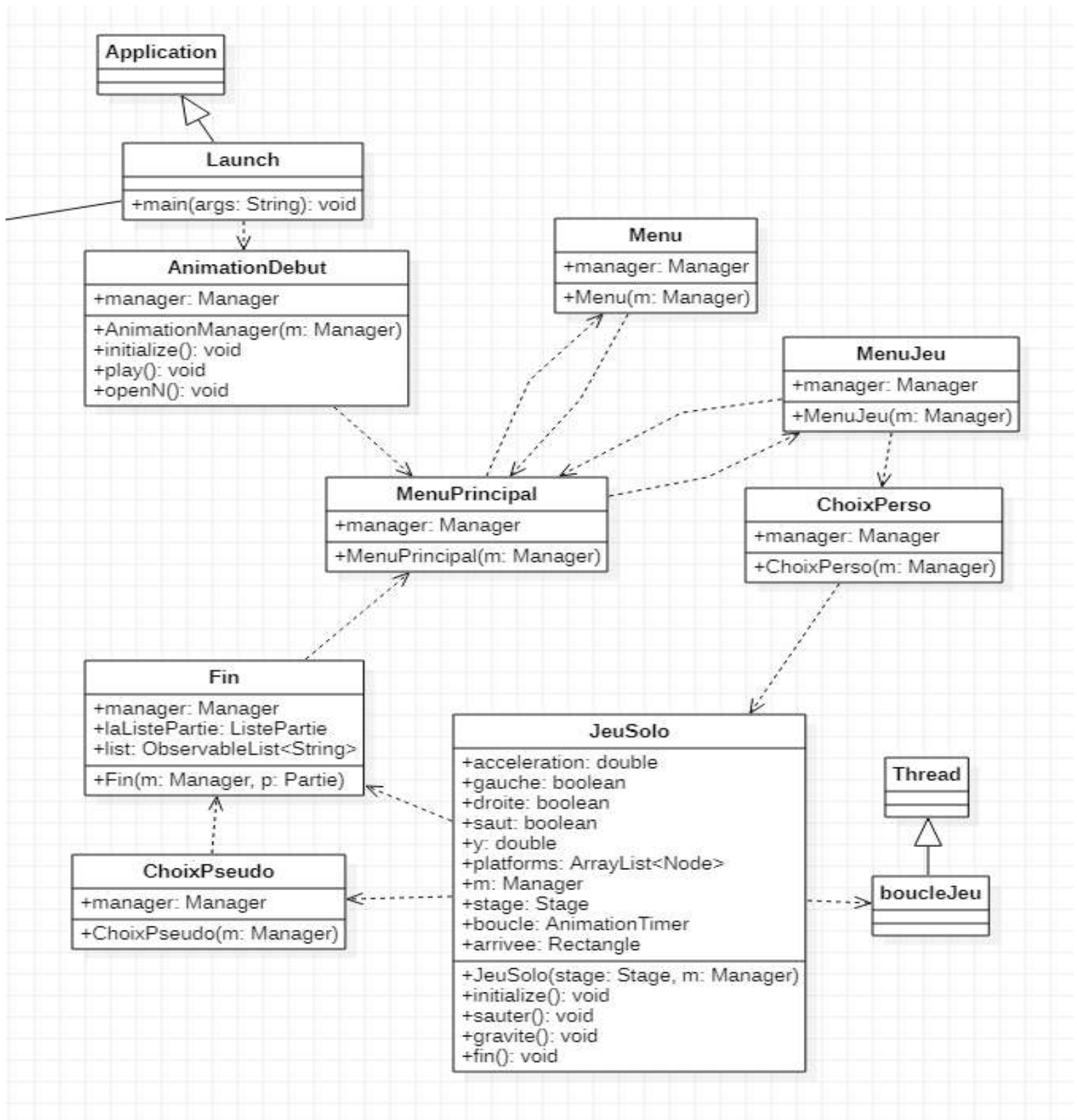
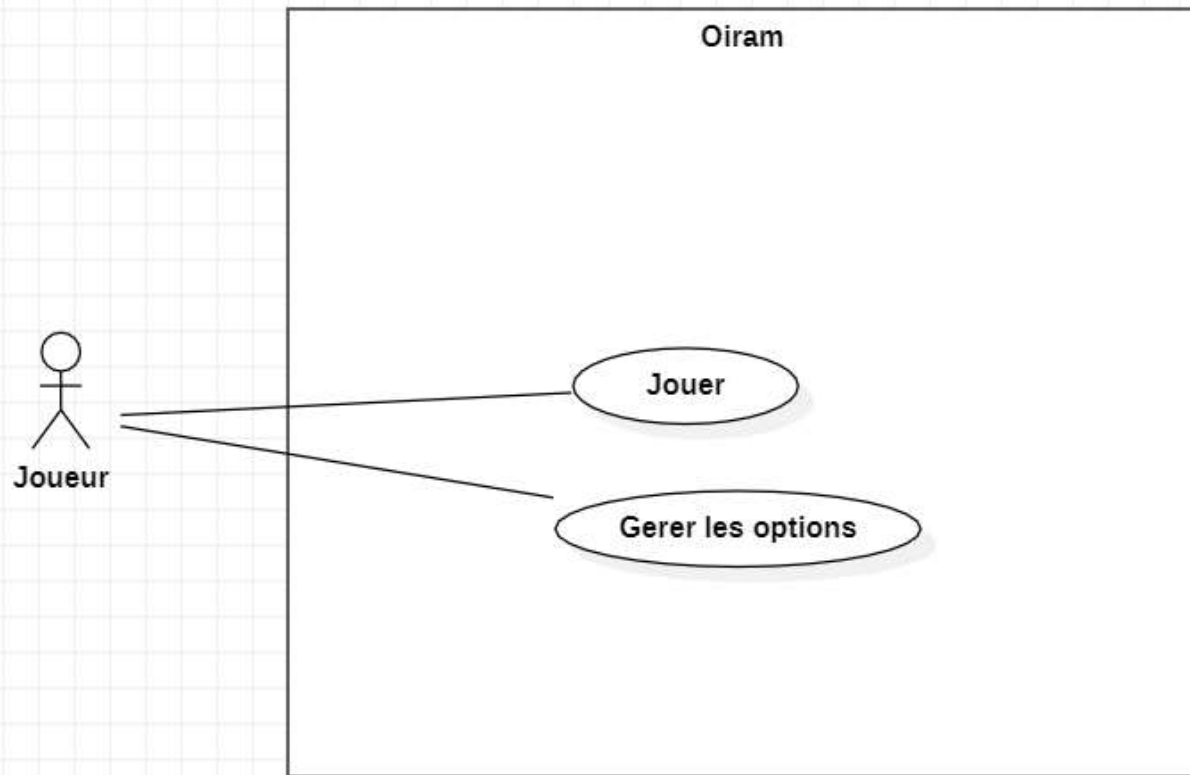


Diagramme de cas d'utilisation:



Preuves:

Je maîtrise la notion d'immuabilité de la classe String. [sur 0.5 point]

PREUVE :

/

Je maîtrise les règles de nommage Java. [sur 1 point]

PREUVE :

Partout

Je sais binder bidirectionnellement deux propriétés JavaFX. [sur 1 point]

PREUVE :

/

Je sais binder unidirectionnellement deux propriétés JavaFX. [sur 1 point]

PREUVE :

src/view/Fin→ligne50

Je sais coder une classe Java en respectant des contraintes de qualité de lecture de code. [sur 2 points]

PREUVE :

src/model->toutes les classes du model

Je sais contraindre les éléments de ma vue, avec du binding FXML. [sur 2 points]

PREUVE :

ressource/fxml/AnimationDebutfxml→ligne15

Je sais définir une CellFactory fabriquant des cellules qui se mettent à jour au changement du modèle. [sur 2 points]

PREUVE :

src/view/Fin→ligne44

Je sais développer une application graphique en JavaFX en utilisant FXML. [sur 2 points]

PREUVE :

ressource/fxml

Je sais éviter la duplication de code. [sur 1 point]

PREUVE :

src/model/Persistence

Je sais hiérarchiser mes classes pour spécialiser leur comportement. [sur 2 points]

PREUVE :

src/model

Je sais intercepter des événements en provenance de la fenêtre JavaFX. [sur 2 points]

PREUVE :

src/view/JeuSolo→ligne80

Je sais maintenir une encapsulation adéquate dans mes classes. [sur 2 points]

PREUVE :

src/model/Bloc-> attribut protected

Je sais maintenir, dans un projet, une responsabilité unique pour chacune de mes classes. [sur 2 points]

PREUVE :

src/model

Je sais gérer la persistance de mon modèle. [sur 2 points]

PREUVE :

src/model/Persistence

Je sais surveiller l'élément sélectionné dans un composant affichant un ensemble de données. [sur 2 points]

PREUVE :

src/view/Fin

Je sais utiliser à mon avantage le polymorphisme. [sur 2 points]

PREUVE :

src/model/persistence/ICarteTxt -> ligne 52

Je sais utiliser certains composants simples que me propose JavaFX. [sur 0.5 point]

PREUVE :

src/view/Menu→ligne22

Je sais utiliser certains layout que me propose JavaFX. [sur 1 point]

PREUVE :

src/view/JeuSolo→ligne43

Je sais utiliser GIT pour travailler avec mon binôme sur le projet. [sur 2 points]

PREUVE :

Voir forge.

Création de plusieurs branches:

Master

Test

Tom

Killian

Création d'un .gitignore et utilisation de merge.

Je sais utiliser le type statique adéquat pour mes attributs ou variables. [sur 1 point]

PREUVE :

src/model/Stub→ligne4

Je sais utiliser les collections. [sur 1 point]

PREUVE :

src/view/JeuSolo→ligne46

Je sais utiliser les différents composants complexes (listes, combo...) que me propose JavaFX. [sur 1 point]

PREUVE :

src/view/Fin→ligne27

Je sais utiliser les lambda-expression. [sur 1 point]

PREUVE :

src/view/JeuSolo→ligne57

Je sais utiliser les listes observables de JavaFX. [sur 1 point]

PREUVE :

src/model/ListePartie→ligne10

Je sais utiliser les packages à bon escient dans un projet. [sur 1 point]

PREUVE :

launch, model, view

Je sais utiliser un convertisseur lors d'un bind entre deux propriétés JavaFX. [sur 1 point]

PREUVE :

/

Je sais utiliser un fichier CSS pour styler mon application JavaFX. [sur 1 point]

PREUVE :

src/CSS

Je sais utiliser un formateur lors d'un bind entre deux propriétés JavaFX. [sur 1 point]

PREUVE :

/