

Compétences devant être maîtrisées le 20 janvier 2020

Documentation

Demande	Preuve
Je sais concevoir un diagramme UML intégrant des notions de qualité et correspondant exactement à l'application que j'ai à développer.	Fichier StarUML dans le dossier documentation
Je sais décrire un diagramme UML en mettant en valeur et en justifier les éléments essentiels.	Voir documentation StarUML
Je sais documenter mon code et en générer la documentation.	Voir documentation JavaDoc
Je sais décrire le contexte de mon application, pour que n'importe qui soit capable de comprendre à quoi elle sert.	Voir documentation
Je sais faire un diagramme de cas d'utilisation pour mettre en avant les différentes fonctionnalités de mon application.	voir documentation/useCase.png

Code

Demande	Preuve
Je maîtrise les règles de nommage Java.	Utilisation du CamelCase (package=minuscule, classe=majuscule, méthode et attribut = minuscule, utilisation majuscule pour marquer changement de mot dans les noms: exemple: romainOlivier)
Je sais binder bidirectionnellement deux propriétés JavaFX.	Utilisation dans l'intro du jeu: main.java.view.controller.KnownStart ou StartPage
Je sais binder unidirectionnellement deux propriétés JavaFX.	Utilisation dans l'intro du jeu: main.java.view.controller.KnownStart ou StartPage
Je sais coder une classe Java en respectant des contraintes de qualité de lecture de code.	Respect des règles d'écriture (camelCase, ordre des définitions: attributs,constructeurs,méthodes), indentations, nommage clairs des variables
Je sais contraindre les éléments de ma vue, avec du binding FXML.	Validation du prénom dans la vue StartPage
Je sais définir une CellFactory fabriquant des cellules qui se mettent à jour au changement du modèle.	//TODO Add CellFactory (EndGame when adding sentence ?)
Je sais développer une application graphique en JavaFX en utilisant FXML.	Dossier main.java.view.controller & resources.views
Je sais éviter la duplication de code.	Multiple exemples mais le plus intéressant est la manière dont sont fait les déplacements (main.java.core.control)
Je sais hiérarchiser mes classes pour spécialiser leur comportement.	Multiple exemples avec les Actions en particulier
Je sais intercepter des événements en provenance de la fenêtre JavaFX.	main.java.core.control.KeyBinder
Je sais maintenir, dans un projet, une responsabilité unique pour chacune de mes classes.	Visible à travers l'ensemble du projet où chaque classe n'a qu'une responsabilité dédiée
Je sais utiliser à mon avantage le polymorphisme.	Chaque item, pnj, interaction peut être downcast vers son interface adapté (visuel, collision, interactif) nous permettant de tout traiter de la même manière
Je sais utiliser certains composants simples que me propose JavaFX.	Utilisation de Label, Button

Demande	Preuve
Je sais utiliser certains layout que me propose JavaFX.	utilisation de VBox, HBox, GridPane
Je sais utiliser GIT pour travailler avec mon binôme sur le projet.	utilisation de branches, Merge, Revert, etc ... (principe GIT de base)
Je sais utiliser le type statique adéquat pour mes attributs ou variables.	Utilisation pour des informations telle que le niveau de folier des PNJ, partager entre les PNj. !Attention! le static ne doit pas servir à simplement donner un accès global pour simplifier le code
Je sais utiliser les collections.	utilisation de List (Array/linked), et de Set (HashSet)
Je sais utiliser les différents composants complexes (listes, combo...) que me propose JavaFX.	utilisation de TimeLine
Je sais utiliser les lambda-expression.	utiliser naturellement à travers tout le code (exemple: eventhandler)
Je sais utiliser les listes observables de JavaFX.	//TODO Add usage of them
Je sais utiliser un convertisseur lors d'un bind entre deux propriétés JavaFX.	//TODO Add usage of one
Je sais utiliser un fichier CSS pour styler mon application JavaFX.	KnownStart et StartPage utilise tout deux des fichiers CSS
Je sais utiliser un formateur lors d'un bind entre deux propriétés JavaFX.	Utilistion dans controller.KnownStart pour afficher le prénom du joueur

Théorie

Vu durant l'examen écrit

Rédigé par et avec la vision de Romain Olivier