Rapport: Rigel étape 12

Graphes et ensembles

Le paquetage `math` est enrichi de `sets`, comptant nombre de définitions d'ensembles avec des opérations telles que la partition pour `PartitionSet` et l'équation dans `Equation`. Ceci facilite l'écriture de `graphs`, qui comprend, entre autres, la définition d'un arbre générique `Tree<T>`. Ces deux nouveaux sous-paquetages sont utiles dans l'implémentation de nombre des ajouts décrits ci-dessous.

Ajouts principaux

Barre de recherche

La barre de recherche permet à l'utilisateur de chercher un objet céleste via son nom. A chaque lettre tapée, l'utilisateur voit une liste des objets célestes commençant par la même séquence de lettres que celles entrées. Cliquer sur un nom centre la projection sur l'objet choisi.

L'implémentation de cette barre de recherche passe par `...gui.searchtool.SearchTextField`, classe abstraite qui représente une barre de recherche générale capable de fournir un objet (celui séléctionné au terme de la recherche). Le `Searcher` constitue une implémentation de cette classe qui se spécialise dans les objets célestes. Pour effectuer une recherche rapide, les noms des objets célestes sont entrés dans un `Tree<Character>` où chaque mot est décomposé par lettres, et appartiennent à une même branche les mots qui ont une séquence de lettres identique jusqu'à un certain indice.

Parallélisme

Afin d'améliorer la performance du programme et diminuer les saccades, `...parallelism.ThreadManager` génère un arbre de tâches à effectuer et assigne les tâches à une réserve de threads.

L'annotation `Requires` indique quels *stages* devront être complétés avant que ne soit exécuté celui au-dessus duquel elle se situe. Les tâches sont alors organisées dans un graphe, plus précisément une forêt. Dans chaque arbre, les feuilles constituent les premières tâches à effectuer. Les tâches plus proches de la racine nécessitent la complétion des précédentes avant d'être exécutées.

Chaque PN_i représente un *processus* qui sera exécuté par le gestionnaire. En rouge : une branche qui sera exécuté par les mêmes threads en parallèle de ses branches adjacentes mais qui attendra la complétion de la branche des $P1_i$ pour continuer en $P1_3$.

Ajouts secondaires

Prédiction d'orbite

La prédiction d'orbite permet de connaître, avec clic droit sur des planètes, la Lune ou le Soleil, leur trajectoire à venir. Ces futures positions sont modélisées par des points dont l'espacement est modifiable.

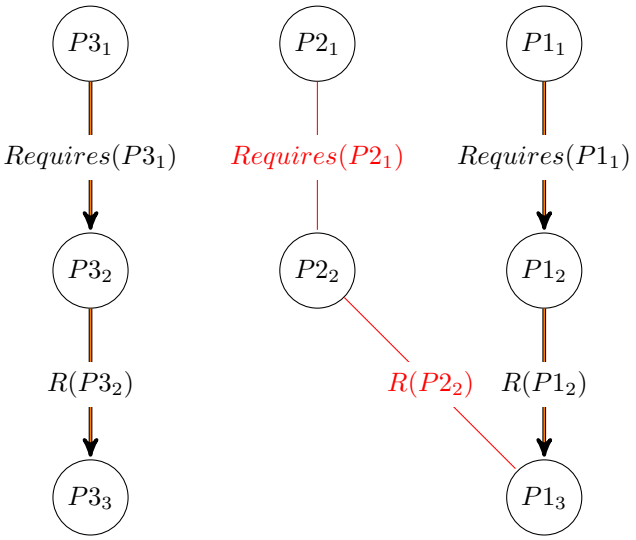

Figure 1: Un graphe possible des taches

Ces orbites font usage de `graphs` en héritant des `Cycles`, des graphes cycliques. Ainsi, l'orbite n'est qu'un `Cycle` de `Supplier<CelestialObjectModel>` pour limiter le calcul, et ce sont les paramètres d'espacement qui permettent de récupérer des positions avec une certaine périodicité. L'espacement entre les représentants de l'orbite est une visualisation de la vitesse de l'objet.

Gestion des paramètres d'interface graphique

Le bouton labellisé d'un rouage à coté de la barre de recherche ouvre un panneau de réglages graphiques : un mode *cinématique* cachant l'interface utilisateur pour ne laisser que le ciel, une touche permettant d'activer ou de désactiver le paneau d'information, un mode plein écran, et un bouton restreignant la vue au ciel réellement visible (interdiction d'observer sous la barre des 5° d'hauteur). Ensuite, deux paramètres de sensibilités de souris sont proposés. Pour finir, de nombreux paramètres graphiques gérant la couleur, les éléments à dessiner et l'espacement de la grille horizontale sont présentés.

Le plein écran joue sur un simple paramètre du BorderPane (`setFullScreen`); le masquage du panneau de droite, la restriction d'altitude et le mode cinématique sont gérés par des propriétés booléennes misent en place dans le `Main` et `SkyCanvasManager`. Les sliders agissent quant à eux sur des valeurs définies dans `SkyCanvasManager`. Enfin les paramètres de couleurs et d'affichages sont modélisés de manière similaire, le `SkyCanvasPainter` ayant la capacité de changer la couleur de certains de ces éléments et les `Checkboxes` activant ou désactivant des parties du ciel, s'aidant de l'enum `...gui.DrawableObjects`.

Informations sur les corps célestes

La barre de droite du BorderPane affiche, lorsqu'un corps céleste est séléctionné via clic droit, des propriétés générales telles que ses coordonnées équatoriales,

ainsi que des infos spécifiques à son type, par exemple : température et hipparcos pour les étoiles, ainsi que les étoiles dans la même constellation. Des sliders s'afficheront également pour modifier l'affichage de prédiction d'orbite des constituants du système solaire.

Pour trouver les étoiles dans une même constellation, la méthode `StarCatalogue.constellationOfStar(Star)` fournit un getter pour la map `Map<Star, AbstractMathSet<Star>> constellationsMap` qui relie, à chaque étoile, l'ensemble des étoiles dans la même constellation qu'elle. La `Map` permet un accès en $\mathcal{O}(1)$.

Mouvements de souris et rotation du ciel

Enfin, glisser la souris avec clic gauche enfoncé modifie le centre de projection, et la glisser avec roulette enfoncée modifie la rotation du ciel.

Plus d'informations sur les contrôles et fonctionnalités sont disponibles en cliquant sur le bouton d'aide, à droite de celui ouvrant les paramètres.

Vous remerciant pour un projet -et un cours!- passionnants malgré les contraintes exceptionnelles,

Alexandre Sallinen et Salim Najib.