Rapport Projet Rigel - Bonus

Bouilloud Mathias (309979) Mettler Julien (309999)

Bonus 1 : Dessin de la Lune en tenant compte de sa phase

Nous avons ajouté dans la classe SkyCanvasPainter une méthode privée appelée “drawMoonPhase” qui affiche la Lune différemment en fonction du pourcentage du disque illuminé. Ainsi la méthode dessine un cercle blanc plein si la phase est 1.

Pour tester la phase, nous avons ajouté un getter phase() dans la classe Moon.

Pour déterminer l’aspect de la Lune, nous avons repris les valeurs de phase lunaire données à la partie 1.2 “Cycle” de l’article “Phase de la Lune” de Wikipédia (https://fr.wikipedia.org/wiki/Phase_de_la_Lune#Cycle).

De plus, comme dans cet article, nous avons fait le choix de faire dépendre l’orientation du croissant de Lune selon que l’on se trouve dans l’hémisphère Nord ou dans l’hémisphère Sud, ce qui est déterminé par l’altitude de l’observateur sur Terre..

Pour y avoir accès, nous avons ajouté en paramètre de la méthode drawMoon de SkyCanvasPainter (qui elle-même appelle drawMoonPhase) le contenu de la propriété coordinates de la classe ObserverLocationBean.

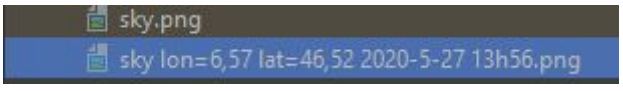
Cette distinction des cas permet de régler le “startAngle” lorsque l’on essaie de dessiner l’arc de cercle correspondant au croissant.

Bonus 2 : Bouton photo

Nous avons également ajouté dans la barre de contrôle un bouton permettant de prendre une photo du ciel observé, en créant un fichier png. Pour cela, nous avons récupéré le code de la classe “DrawSky” (utilisée comme test pour l’étape 8), en modifiant le nom du fichier créé.

Nous nommons notre fichier de la forme : “sky observed at position longitude d'observation lon latitude d’observation lat and date date d’observation”

Bouton photo

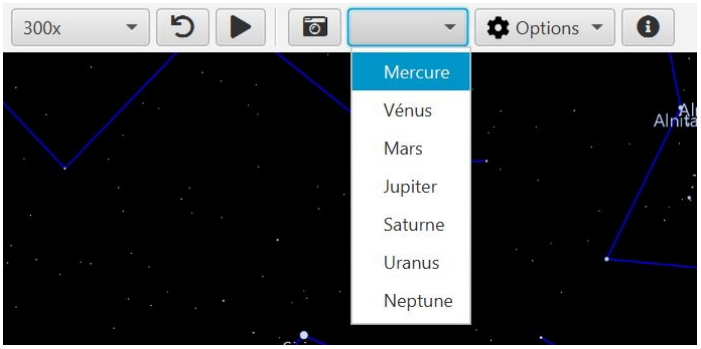


Exemple de fichier png généré

Bonus 3 : Sélection d’un objet céleste comme centre d’observation

Nous avons aussi ajouté à la barre de contrôle un menu déroulant contenant les planètes. En les sélectionnant, le centre d’observation se met aux coordonnées horizontales de ces objets célestes.Il permet de sélectionner, parmis les objets cité ci-dessus, un nouveau centre d’observation du ciel. Si l’objet en question n’est pas observable (c-à-d si leur latitude est supérieure à 90° ou inférieure à 0°), une fenêtre d’alerte apparaît, nous informant de l’échec.

Pour mettre en place ce menu, nous avons décidé d’utiliser une ChoiceBox. Pour la fenêtre d’alerte, nous avons utilisé une Alert.



Bonus 4 : Menu d'options

Bonus 4.1 Contrôle de l'affichage des astérismes

Nous avons ajouté un bouton dans la barre de contrôle, permettant d'activer ou de désactiver l'affichage des astérismes sur l'écran.

Pour cela, nous avons ajouté une SimpleBooleanProperty appelée "asterismEnable" dans SkyCanvasManager, et nous avons modifié la méthode "drawStars"de SkyCanvasPainter, de sorte à ce qu'elle accepte un paramètre en plus de type boolean qui décide si les astérismes sont oui ou non affichés.

Finalement, notre méthode de dessin des objets célestes dans SkyCanvasManager, appelle drawStars en rentrant "asterismEnable.get()" en dernier paramètre.

Le fait d'appuyer sur le bouton ne fait que changer la valeur contenue dans la propriété "asterismEnable".

Bonus 4.2 Contrôle de l'affichage des satellites actifs en orbite autour de la Terre (voir Bonus 5)

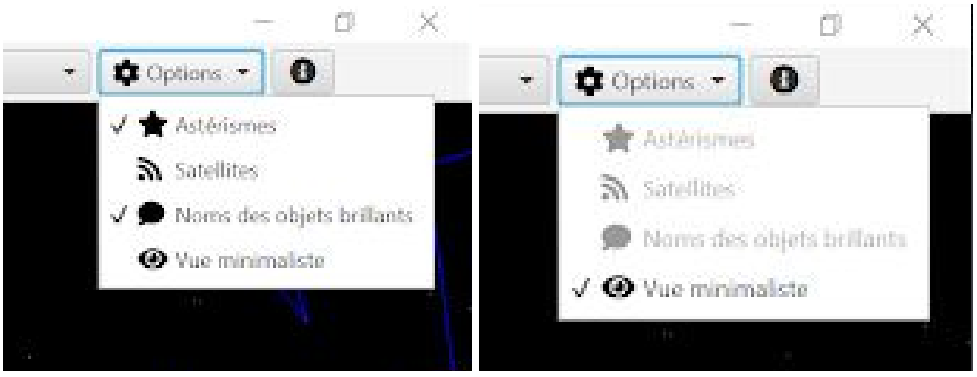
Fonctionnement similaire au bouton précédent.

Bonus 4.3 Contrôle de l'annotation des objets brillants (voir Bonus 6)

Fonctionnement similaire au bouton précédent.

Bonus 4.4 Vue minimaliste

Le choix "Vue minimaliste" permet de désactiver toutes les précédentes options. L'utilisateur ne peut alors activer ces dernières et doit d'abord désactiver la vue minimaliste pour y parvenir.



Bonus 5 : Affichage des satellites actifs en orbite autour de la Terre

Cette option permet d'afficher les 375 satellites actifs en orbite géostationnaire autour de la Terre. Pour cela, nous avons ajouté dans le dossier resources un fichier de type "comma-separated values" provenant du site Active Satellites in Orbit Around Earth et répertoriant ces satellites et leurs spécifications, à l'image du catalogue HYG utilisé à l'étape 6. Nous nous sommes inspirés de cette dernière et avons ajouté trois classes :

- La classe immuable Satellite représentant un satellite géostationnaire, héritant de CelestialObject
- La classe SatelliteCatalogue et son Builder, inspirées de StarCatalogue
- La classe SatelliteDatabaseLoader, inspirée de HygDatabaseLoader

De plus, nous avons modifié ObservedSky, SkyCanvasPainter, SkyCanvasManager…..
…….

Plusieurs explications sont nécessaires :

- Nous ne représentons que les objets en orbite géostationnaire, c'est-à-dire qui ont une période orbitale égale à un jour sidéral, et qui restent fixés au-dessus du même point au-dessus de la Terre en permanence. Nous avons fait ce choix car ils ont par définition une inclinaison nulle à l'équateur, ce qui permet de facilement représenter leur orbite dans le plan équatorial, d'où notre choix d'utiliser ce système de coordonnées pour leur position.
- De plus, cette orbite se situe à environ 36 000 km au-dessus de la Terre. Nous avons donc décrit les satellites par une même latitude (le fichier CSV ne fournit d'ailleurs que la longitude)
- Après plusieurs recherches, le seul catalogue que nous avons trouvé sous forme de fichier CSV date de Juillet 2016 et n'est donc pas exhaustif (Les satellites StarLink n'y figurent pas, par exemple)

<u>Bonus 6</u> : Annotations des objets brillants, des planètes, de la Lune, du Soleil

Le 6e bonus ajoute des annotations à côté des objets célestes les plus brillants. Ces derniers sont déterminés à leur construction, dans CelestialObject, à l'aide d'un attribut booléen isBright qui est **true** si la **magnitude est strictement inférieure à 1.**

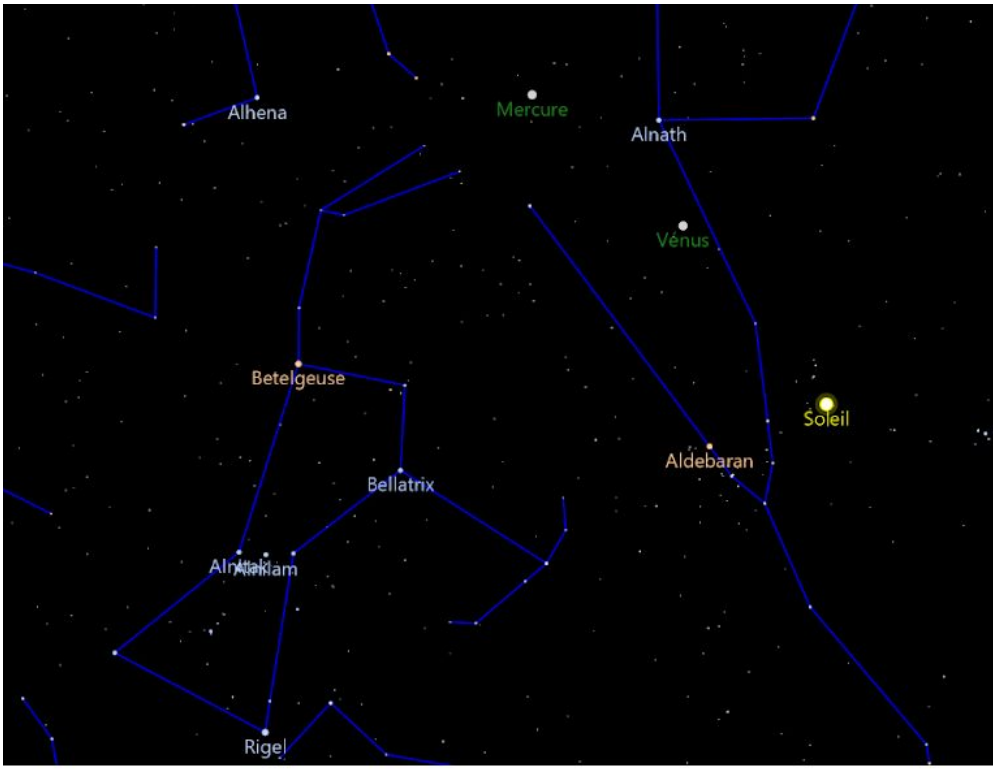
Nous avons choisi également d'annoter le Soleil, la Lune, et les planètes, mêmes celles qui ne sont pas visibles à l'oeil nu (comme Uranus et Neptune) pour voir leur position lorsque nous nous centrons sur elles, dans le bonus 4.

Des textes sont annotés à la position de tous ces objets sur le canevas dans la méthode drawAnnotation de SkyCanvasPainter, et leur contenu correspond à la méthode info() de Celestial Object.

La méthode drawAnnotation est appelée dans la méthode drawStars si le getter isBright() retourne true.

Un code couleur a été choisi pour chaque type d'objet :

- Pour les étoiles, la couleur correspondant à leur température
- Pour les planètes, le vert
- Pour le Soleil, le jaune
- Pour la Lune, le blanc



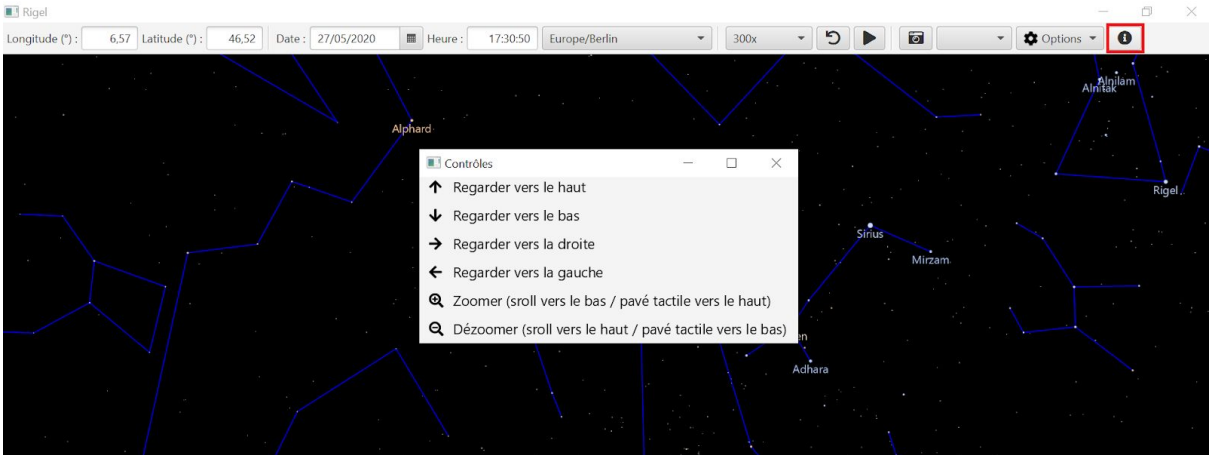
Un exemple d'annotations d'étoiles, de quelques planètes et du Soleil

Bonus 7 : **Bouton d'informations**

Dans l'optique de faciliter l'expérience de l'utilisateur lors de sa première utilisation du programme, nous avons ajouté comme dernier bonus un bouton qui une fois appuyé affiche une deuxième fenêtre (une nouvelle Scene) montrant les contrôles et leur utilité, c'est-à-dire les touches directionnelles et les scrolls de la souris, avec leurs icônes correspondantes.

Ces dernières sont des caractères Unicode provenant de la police FontAwesome
Pour bien indiquer l'association entre chaque caractère et sa description, nous avons eu recours à un GridPane, dont la deuxième colonne correspond à l'icône et la troisième colonne à la description.
Chaque ligne correspond à une touche du clavier ou de la souris

Afin d'avoir une marge tout à gauche du panneau, nous avons simplement laissé la 1ère colonne vide.
Les marges horizontales entre les deux colonnes et les marges verticales entre chaque ligne sont établies à l'aide des méthodes setHgap et setVgap de GridePane.



Le menu affichant les informations sur les contrôles utilisés dans le programme