

## **Rapport CS108**

Dans ce rapport, nous allons décrire les différentes améliorations que nous avons apportées au programme Rigel. Celles-ci sont majoritairement des nouvelles fonctionnalités qui améliorent l'expérience utilisateur, en facilitant la prise en main, et en ajoutant des options pour que le programme soit plus complet. Voici une liste exhaustive de celles-ci : Ajout d'une navigation par drag and drop, un menu déroulant afin de sélectionner quels éléments afficher, un bouton pour mettre la fenêtre en plein écran, l'ajout de différentes représentations graphiques pour la lune et l'amélioration des graphismes de l'application.

### **Le Drag and Drop**

Jusqu'à l'étape 11, la seule façon de changer l'orientation de l'observateur était d'utiliser les flèches du clavier. Dans la partie 12, nous avons implémenté un changement de vue avec la souris. De plus, il est maintenant possible de faire un tour de plus de 360 degrés sur l'azimut. Notre implémentation a été réalisée dans le *skyCanvasManager*. Comme le modèle du code pour les coordonnées azimutales a été conçu pour les angles de 0 à 360 degrés, nous appliquons une fonction *reduce* afin de pouvoir utiliser le modèle, même après un tour complet.

Concernant le mécanisme de drag and drop, nous récupérons les coordonnées fournies par la méthode *setOnMouseDragged*, afin de calculer la différence par rapport aux coordonnées actuelles et de les additionner pour modifier le canvas. Il est possible d'utiliser cette fonctionnalité avec tous les boutons de la souris.

### **Les phases de la Lune**

La lune était représentée jusqu'à maintenant à l'aide d'un disque blanc. Nous avons amélioré cette représentation en proposant cinq dessins différents pour illustrer les différentes phases de la lune : La nouvelle lune (0% à 20%), le croissant de lune (20% à 40%), la demi-lune (40% à 60%), la lune gibbeuse (60% à 80%) et la pleine lune (80% à 100%). Pour chaque phase, un disque noir est d'abord dessiné, il sert de base pour la lune. La partie visible est ensuite superposée à la base, créant ainsi l'effet désiré.

### **Le bouton de plein écran**

Nous avons implémenté un bouton qui permet à l'utilisateur de passer du mode fenêtré en mode plein écran et inversement. son fonctionnement est similaire à celui du bouton play/pause. Etant en mode fenêtré, un click sur celui-ci va faire permuter la fenêtre en mode plein écran et le texte, initialement "Fullscreen", va être

changé en “windowed”. Nous avons également utilisé les logos de la police fournie *font awesome* afin de rendre ce bouton plus élégant.

### **Le menu show/hide**

Le menu déroulant que nous avons ajouté, permet de choisir quels éléments graphiques l'utilisateur souhaite voir dans le canvas. Il a le choix d'afficher, ou non, les astérismes, les étoiles, le soleil, la lune, les planètes et l'horizon. Le menu s'ouvre en cliquant sur le bouton “Option”. Chaque paramètre du menu est relié à un bean qui va être lu par le *skyCanvasManager* afin de savoir quel élément dessiner. De plus, les éléments non dessinés ne sont plus affichés dans la barre d'information.

### **Le style des boutons**

Lors de la mise en place du menu déroulant, nous avons constaté que son style graphique était différent des autres boutons. Nous avons donc décidé de tous les changer afin qu'ils se ressemblent. Nous avons créé une feuille Cascade Style Sheet (CSS) dans le but de changer la mise en page. Nous avons décidé pour des raisons de clarté de changer la couleur en bleu à chaque fois que le curseur passe sur un élément cliquable du graphical user interface (GUI) à l'aide de l'attribut *hover*.

### **L'icône d'application**

Nous avons rajouté une icône d'application afin d'améliorer le côté esthétique. Il s'agit d'une capture d'écran de la constellation d'Orion. Le logo est au format png et s'intitule “icon.png”. La couleur de la constellation a été modifiée afin qu'elle soit plus visible pour l'utilisateur.