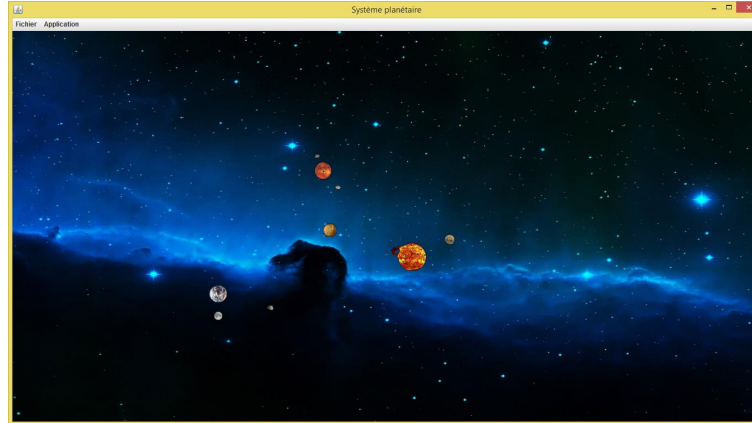


## PROJET MOP 2015

Le but de ce projet est de concevoir en Java une application qui réalise une **animation graphique** en 2D représentant un ensemble de planètes et leurs satellites en orbite autour d'une ou plusieurs étoiles. Pour des raisons pratiques (notamment sur la taille des astres et leurs distances respectives), votre affichage ne respectera pas la réalité du cosmos : vous êtes libre d'imaginer des systèmes quelconques.



Vous devrez créer des classes permettant de modéliser des **astres**. Certains astres, qu'on appellera **étoiles** seront fixes, déterminés par leur position ( $x, y$ ). D'autres qu'on appellera **satellites** seront mobiles, en orbite autour d'un autre astre.

Nous simplifierons les lois de Kepler pour considérer que la trajectoire orbitale d'un satellite est une ellipse qui donnera à un instant  $t$  la position ( $x, y$ ) d'un satellite selon les équations

$$x = a \cdot \cos(t/p) + x_{\text{AstreRéfèrent}}$$

$$y = b \cdot \sin(t/p) + y_{\text{AstreRéfèrent}}$$

où  $a$  est le demi-grand axe de l'ellipse,  $b$  est le demi-petit axe de l'ellipse et  $p$  est la période de rotation.

Un satellite sera donc caractérisé par l'astre autour duquel il tourne et ses 3 propriétés  $a, b$  et  $p$ .

L'application comportera un menu permettant

- de créer un astre (étoile ou satellite), qui sera ajouté à l'ensemble des astres de l'application
- de supprimer un astre de l'application
- d'enregistrer sur disque une étoile et ses satellites
- de recharger un fichier d'une étoile (et ses satellites) préalablement enregistré.

Chaque astre sera représenté graphiquement par une image différente qui sera chargée, à la création de l'astre, à partir d'un fichier .png

Pour positionner des composants à une position déterminée, vous devrez utiliser un composant JPanel. Ensuite,

- ou bien :

Vous utilisez la propriété Icon du composant JLabel pour placer l'image sur un objet JLabel. Pour positionner ce composant à une position ( $x, y$ ), vous pourrez ensuite utiliser la méthode `setBounds(x, y, width, height)`.

Fixer dans ce cas le gestionnaire de disposition comme suit : `setLayout(null)`

- ou bien :

Dessinez les images (de type `ImageIcon`, à l'aide de la méthode `paintIcon`), en redéfinissant la méthode `paintComponent(Graphics g)` du JPanel.

Pour que votre animation puisse se dessiner correctement, nous vous suggérons d'utiliser la méthode [Thread.sleep\(t\)](#) pour laisser au système le temps de rafraîchir l'affichage, avec du code ayant la forme suivante :

```
while (true)
{dessineEtoiles();
try {Thread.sleep(20);} catch (Exception e) {}
}
```

Pensez également à utiliser de manière judicieuse la méthode [repaint](#)

Critères de notation du projet :

- Modélisation objet et structure des données
- Algorithmique
- Ergonomie et interface utilisateur
- Commentaires et javadoc