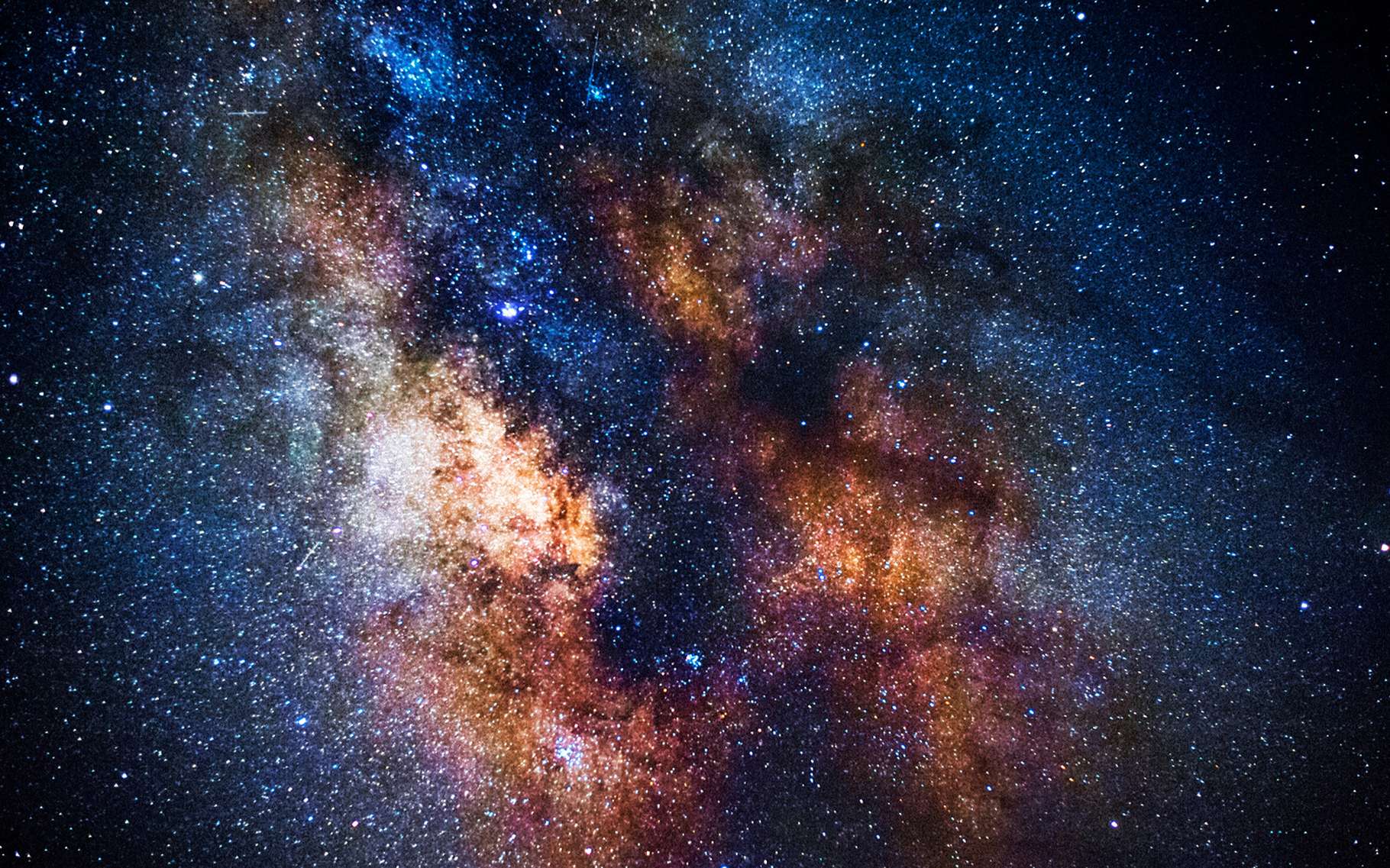
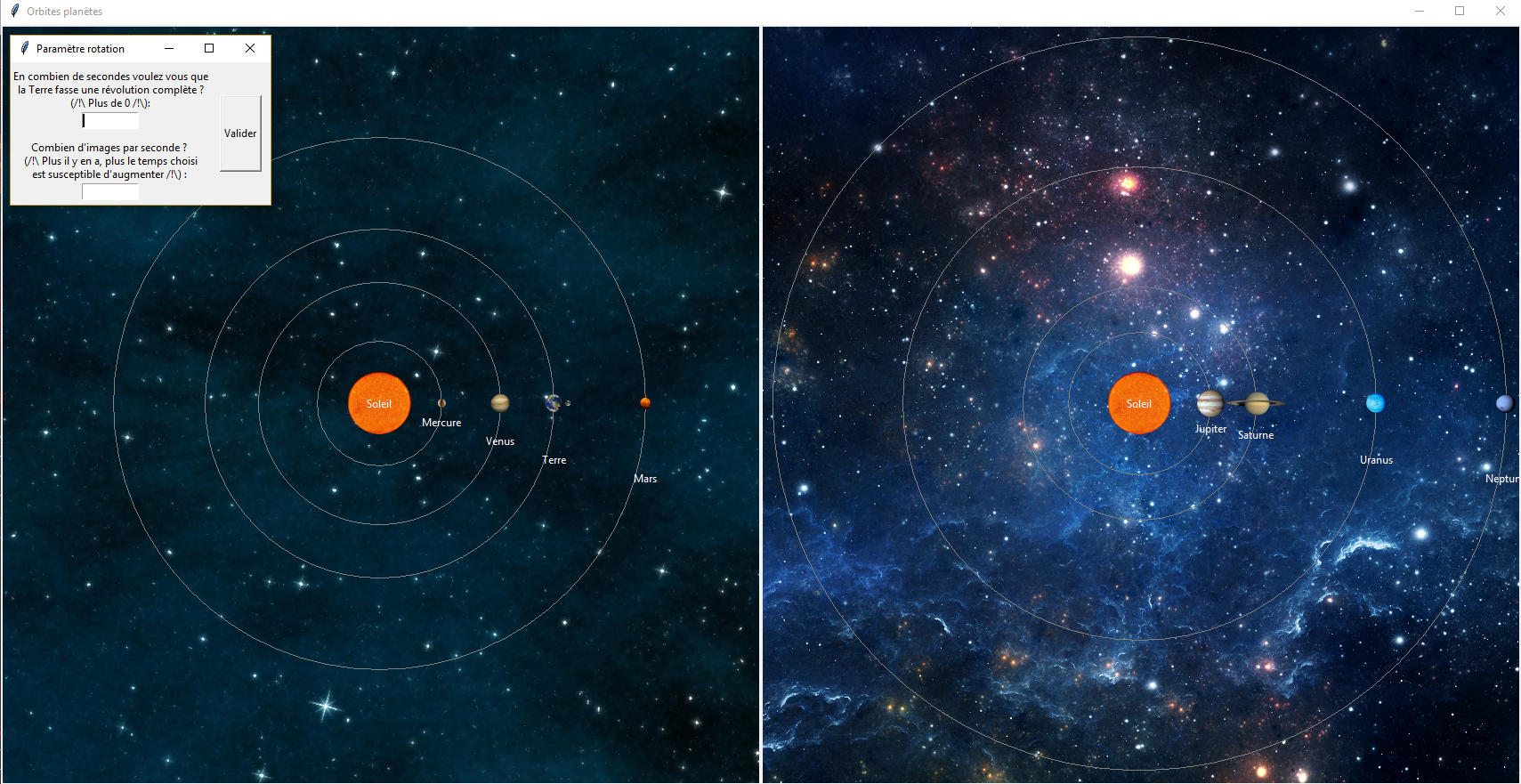
VINCENT FAIVRE, Terminale S

***Présentation du projet :***



***Simulation du mouvement des planètes du système solaire sur leurs orbites***

***FAIVRE Vincent, Terminale S ; CORBARD Nicolas, Terminale S ; BERSIA Julien, Terminale S***



Notre projet consiste à représenter graphiquement en deux dimensions les différentes orbites planétaires de notre système solaire, ainsi que le mouvement des planètes sur ces mêmes orbites.

Ce projet nous a permis d’appliquer des concepts de programmation Python à des connaissances scientifiques que l’on étudie en Mathématiques mais aussi en Physique. Ainsi, ce projet nous permet de revoir nos connaissances tout en en explorant de nouvelles dans un autre domaine.

Personnellement, ce projet m’a vraiment tenu à cœur et a été très important pour moi. Ayant toujours été intéressé par l’informatique, ce premier projet m’a permis de voir si je pouvais faire de la programmation mon futur objet d’étude dans le supérieur. Autrement, Nicolas, quant à lui voulait absolument que le projet tourne autour de l’astronomie. C’est donc tout naturellement que nous avons collectivement décidé que notre programme simulera notre système solaire.

Au niveau technique, ce projet ne présente pas d’amélioration par rapport aux programmes déjà existants, car ce domaine (la simulation de mouvements spatiaux) de la programmation est très développé.

En revanche, il nous a permis de faire face à de nouveaux problèmes que nous avons pu résoudre en appliquant des méthodes propre à la programmation.

***Analyse du besoin et recherche d’idées :***

Au début de notre réflexion sur notre projet, nous avions pour objectif de créer un programme capable de présenter une vulgarisation scientifique des mouvements qui animent notre système solaire. Nous voulions également réaliser un programme permettant à l’utilisateur de créer une planète, mais par manque de temps, nous avons dû abandonner cette idée. Nous nous sommes fixés différents objectifs à respecter :

* Fidélité raisonnable à la réalité (échelles, dimensions, temps…)
* Coder en Python
* Vue en deux dimensions
* Code compréhensible pour quelqu’un ne connaissant pas la programmation

Pour réaliser ce projet, nous avons donc utilisé la programmation Python, avec notamment les librairies graphique et mathématique. En effet, notre programme fait appel aux librairies « Tkinter », permettant la création de fenêtre d’affichage, ainsi que la mise en mouvement d’objets associés à des planètes, mais aussi à la librairie  « Math », permettant d’importer les fonctions mathématiques telles que les puissances. Enfin, nous avons mobilisé la librairie « Time » qui permet, comme son nom l’indique, d’apporter au programme la notion d’écoulement du temps. De plus, pour trouver les informations nécessaires à la mise en œuvre de ce projet demandant des données précises par rapport à la notion d’échelle, nous nous sommes référés à diverses sources d’internet (sites spécialisés ou non).

***Structure globale du projet :***

L’ensemble du projet est constitué d’un seul fichier python et de plusieurs fichiers images (Background, astres avec les planètes et le Soleil).

CREATION DES FENETRES D’AFFICHAGE ET DES REPRESENTATION DES ASTRES

(Import des images)

IMPORT DES LIBRAIRIES

(Tkinter, Math, Time)

***Répartition des tâches et démarche collaborative :***

LANCEMENT DE LA BOUCLE D’ANIMATION DES PLANETES

(Mobilisation des fonctions de mouvement/ de calcul)

INTERACTIONS AVEC L’UTILISATEUR

(Entrées du nombre d’images par seconde et de la vitesse de déplacement des planètes)

INITIALISATION DE LA POSITION DES ASTRES ET DE LEURS CARACTERISTIQUES

(Angles de départ pour chaque planète, propriétés de chaque astre, coordonnées de chaque astre, tracés des trajectoires de chaque planète)

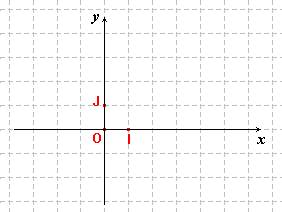
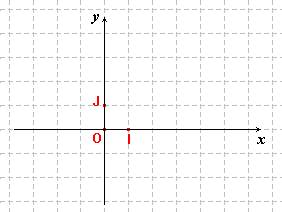
|  |  |  |
| --- | --- | --- |
| Tâches | Planning | Noms |
| Rédaction d'un cahier des charges simplifié | 13 janvier | Nicolas |
| - Recherche de l'équation de la trajectoire d'un mouvement elliptique  - Affichage d'un cercle à l'aide de la librairie Tkinter  - Questionnement sur la représentation à l'échelle ou non des orbites  - Utilisation de l'équation d'un cercle | 20 janvier | * Nicolas * Vincent * Julien * Nicolas/ Vincent |
| - Avancée dans l'interface graphique de notre projet de système solaire, avec une vue de haut et en 2D. On continue les recherches ! | 27 janvier | Vincent/ Julien/ Nicolas |
| - Affichage d'orbites de différentes planètes, mais pas du déplacement de la planète car nous n'avons pas encore trouvé la formule permettant de donner l'équation horaire des planètes. | 02 février | Nicolas |
| - Implémentation graphique du soleil, calcul des dimensions de son image, calcul des orbites des planètes affichables sur un écran. différents tests de mise à l'échelle.  - Explication claire dans le code afin de montrer l'importance de chacune des lignes de code. Différents tests pour l'affichage du nom des astres et provoquer un changement de couleur lors du passage du curseur sur le nom de l'astre | 03 février | * Nicolas * Julien |
| -mise en mouvement circulaire uniforme d’un cercle codé dans tkinter | 03 Mars | Vincent |
| - Application de la méthode trouvée le 3 Mars sur le code déjà réalisé où l'on voyait toutes les planètes immobiles. Désormais, les planètes sont en mouvement mais il reste encore à adapter les vitesses de déplacements. | 23 Mars | Vincent/ Nicolas/ Julien |
| - Calcul des nouvelles coordonnées d'une planète après une seconde, en utilisant les lois trigonométriques (cos x; sin x)  - Recherche et tests des principes d’animation et de déplacements des planètes dans le temps (nombre d’images par seconde, temps de pause entre deux déplacements) | 14 Avril | Nicolas |
| - Application des principes mathématiques trouvés à la séance précédente, tout en ajoutant la possibilité de faire varier le temps de pause entre chaque animation, ce qui fait varier la vitesse des planètes. | 20 Avril | Julien/ Vincent |
| - Ajout de la possibilité pour l'utilisateur de choisir la vitesse des planètes à travers une saisie. | 25 Avril | Vincent |
| - Ajout des images des planètes à la place des simples cercles qui représentaient jusqu'alors les planètes (suppression des diamètres des cercles présents dans le code)  - Ajout des textes sur la fenêtre des planètes gazeuses | 28 Avril | * Vincent * Julien |
| - Rédaction de la partie commune du dossier  - Rajout de la Lune | 5 Mai | Nicolas  Vincent |

Lors de la réalisation de ce projet, notre principal moyen de transmission des informations a été le système de messagerie Messenger, permettant des échanges rapides et simples d’informations à travers des pièces jointes, et la création d’un groupe commun. Cependant, nous avons également dû nous voir régulièrement hors des cours pour « brainstormer » et échanger nos idées. Enfin, on peut également citer la plateforme Github sur laquelle nous notions nos avancées personnelles et collectives.

***Réalisation personnelle :***

Lorsque nous avons commencé le projet, la première tâche dont je me suis chargé a été de comprendre comment utiliser une interface graphique en Python. J’ai donc trouvé que Tkinter ferait parfaitement l’affaire par rapport aux contraintes de notre projet n’étant pas trop gourmand niveau ressource

J’ai effectué quelques essais avec Tkinter pour essayer de cerner les différentes fonctions que proposait ce module et lesquelles pourrait potentiellement nous être utiles. Le site « apprendre-python.com » m’a été d’une grande aide pour la compréhension première de Tkinter. J’ai donc commencé à créer un canevas dans lequel je pourrais placer plein de widget différents (exemple : texte, image, formes géométriques, etc…). C’est en créant des cercles (avec la méthode create-oval()) que je me suis rendu compte que le repère orthonormé (en pixels) utilisé par Tkinter était inversé par rapport à celui conventionnel :





Repère conventionnel Repère Tkinter

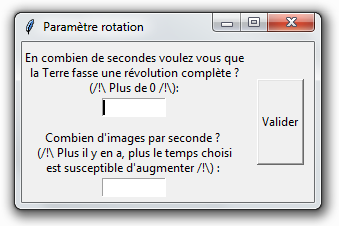
Cette découverte bien qu’étrange ne s’est pas révélé déterminante dans la réalisation de notre projet.

Très rapidement, je me suis occupé de représenter le mouvement des planètes dans le système solaire. Cela a été source d’énormément d’essai peu fructueux de ma part mais ça nous a permis d‘avancer.   
Donc, après en avoir parlé avec Nicolas et Julien et après nous être entretenu avec Emmanuel Erlich (notre prof de Physique-Chimie)

J’ai donc commencé à utiliser le cosinus et le sinus. En testant par tâtonnement, j’ai réussi à simuler un mouvement circulaire uniforme. Cependant, je n’arrivai pas à comprendre et à expliquer comment la fonction du mouvement fonctionnait. Mais, comme le mouvement était fonctionnel et réaliste, nous avons conservé mes formules pendant longtemps.   
Cependant, lorsque Nicolas s’est intéressé à la simulation du temps, nous avons  
changé mes formules pour une simples fonctions trigonométriques du type x = xSoleil + distanceAstre\_Soleil\*cos(angle) et pareil pour le y mais avec un sinus.  
Ce faisant, notre code est devenu plus claire et plus simple à changer par la suite.

Il a donc fallu trouver le moyen de représenter la dynamique de déplacement des planètes en maîtrisant l’échelle du temps d’affichage par rapport au temps de révolution des planètes. Pour cela, nous avons décidé tout d’abord de prendre comme référence le temps souhaité par l’utilisateur (en secondes) pour une révolution de la Terre autour du Soleil.

Encore une fois, j’ai commencé à tester de faire une simulation du temps

« En m’inspirant des principes d’animation des jeux vidéo, j’ai compris qu’en définissant la fréquence d’affichage en images par seconde, nous pouvions calculer les vitesses de déplacement de chacune des planètes en angles ainsi que le temps de pause nécessaire entre deux déplacements. Ainsi, j’ai réalisé des tests en utilisant la librairie python Time et ses fonctions « sleep() » et « time() ». »

Une fois que notre programme était presque fini, je me suis occupé de créer une petite fenêtre utilisateur dans laquelle l’utilisateur pourrait rentrer les paramètres du programme.  
Pour cela, j’ai fait appel aux méthodes Frame(), Label(), Entry() et Button() de Tkinter

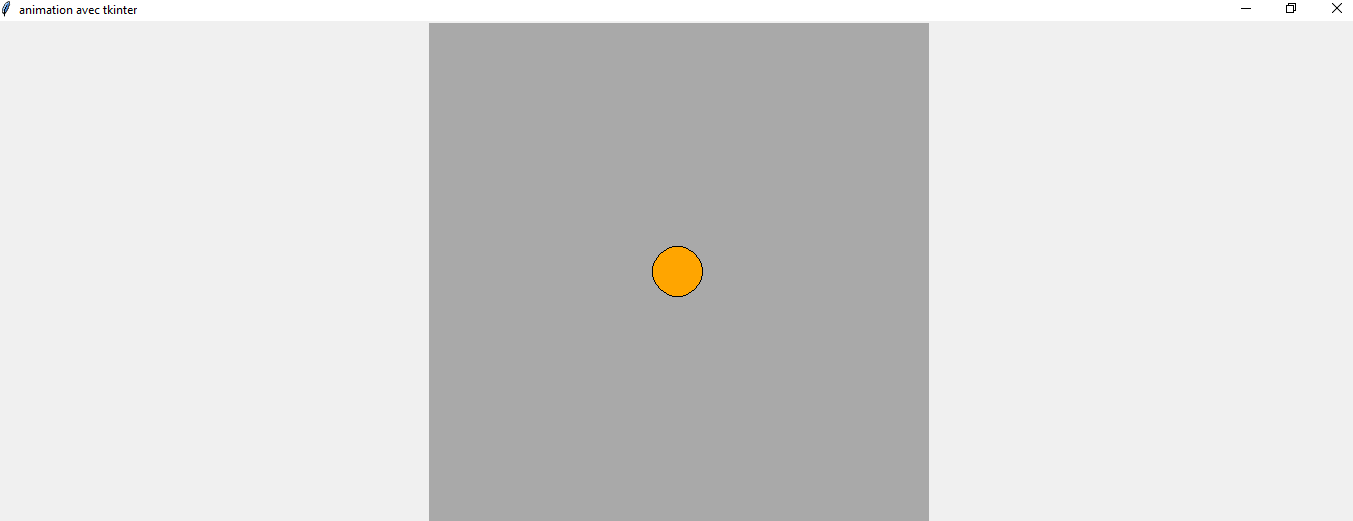
Enfin, Je me suis chargé d’implanter la Lune dans le code parce que cela me tenait à cœur. J’ai donc créé un astre comme toutes les autres planètes sauf que j’ai remplacé toutes les variables au Soleil par des variables correspondant à la Terre.



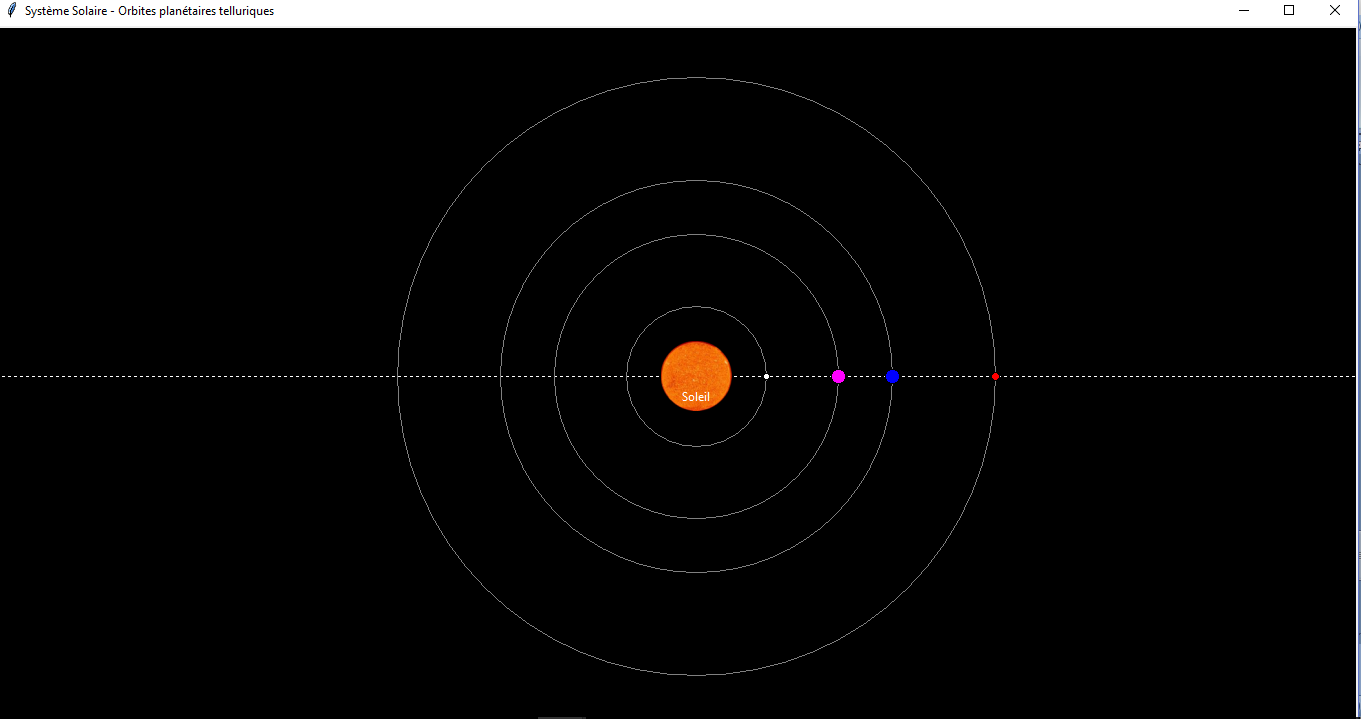
Sur cette image, on peut voir l’une des deux frames (frame des planètes telluriques)

Avec cette capture d’écran, on se rend compte que la taille des planètes est arbitraire, car impossible à reproduire avec une échelle, mais que les distances Soleil-Astres sont, quant à elles, parfaitement à l’échelle. Les planètes, qui n’étaient à l’origine que des cercles en mouvement, sont représentées par des images leur correspondant. On peut également voir les différentes trajectoires affichées en gris, ainsi que les planètes restant sur le tracé de leur trajectoires grâce à la méthode « cosinus/ sinus ».

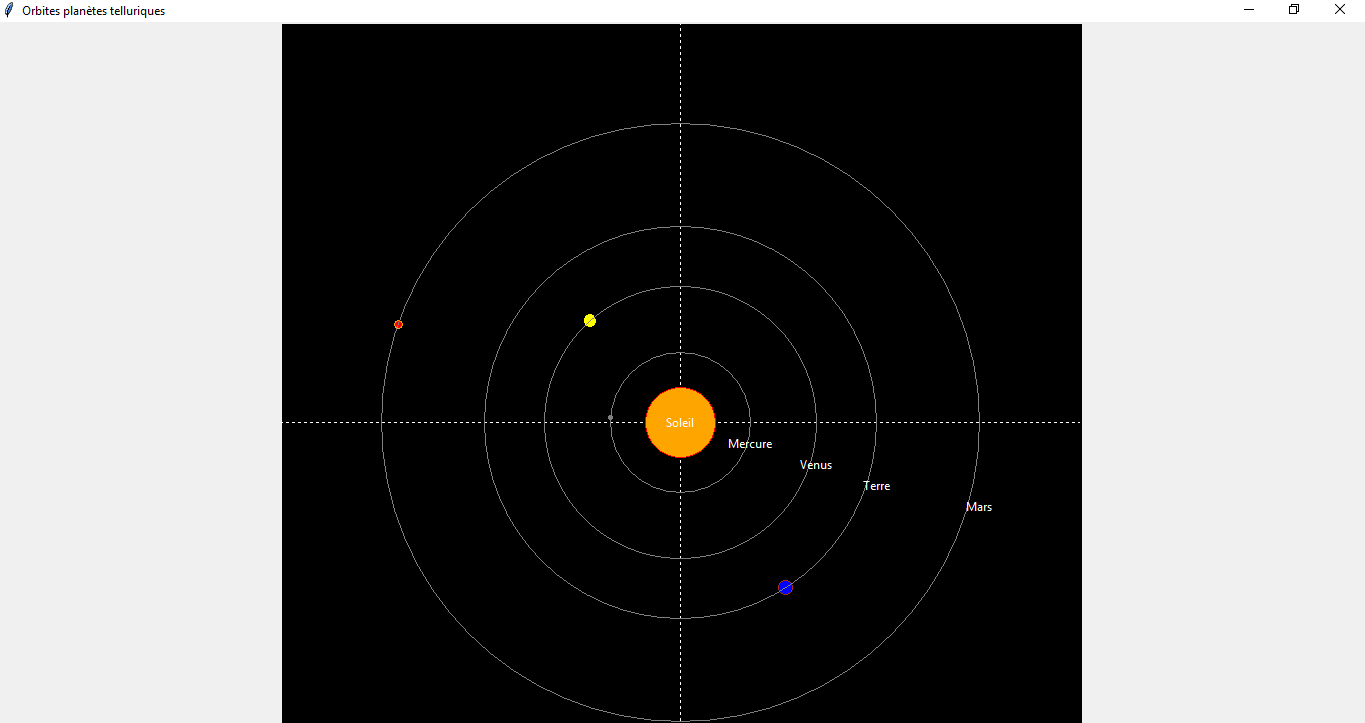
***Présentation des tests :***



Sur cette image, on peut voir notre premier test d’affichage d’un cercle grâce à Tkinter.



Ici, on peut voir la version immobile du système solaire (que Nicolas et Julien ont réalisé quand j’étais malade), où nous testions encore les tracés de trajectoire et les échelles de distances. De plus, les planètes ne sont pas encore associées à des images, mais seulement à des cercles de couleur.



Sur ce test, on peut voir que les planètes suivent bel et bien leurs orbites après avoir introduit les fonctions de mouvement mobilisant cosinus et sinus. Cependant, à l’époque de ce screenshots, les mouvements étaient encore codées avec mes fonctions inexplicables



Enfin, cette dernière capture d’écran montre l’aspect final du projet, avec l’interaction entre l’utilisateur et le programme, la totalité des images, ainsi que le mouvement des planètes autour du Soleil. On peut également noter l’ajout de dernière minute correspondant au mouvement de la Lune autour de la Terre.

***Améliorations possibles :***

Ce programme restant tout de même assez simpliste par rapport aux phénomènes se déroulant réellement, de nombreuses améliorations restent possibles.

En effet, on peut tout d’abord penser au fait que les orbites planétaires ne sont pas exactement circulaires, puisqu’elles sont en réalité elliptiques. Les ellipses étant beaucoup plus complexes à interpréter mathématiquement en raison de leur excentricité par rapport à un cercle, nous avions pris la décision de coder des mouvements circulaires. Cela entraine également le problème des vitesses, car dans un mouvement circulaire, la vitesse est constante, alors que dans un mouvement elliptique, elle ralentie ou accélère selon la position de l’astre sur l’ellipse.

De plus, bien que les distances entre les astres soit à l’échelle, la taille des astres a été, quant à elle, réglée de manière arbitraire, car cette échelle ne pouvait pas être transcrite sur une petite fenêtre (les astres seraient plus petit que des pixels par rapport au Soleil). Sans compter le fait qu’une fenêtre unique pour toutes les planètes serait préférable, mais là encore, les distances étaient trop grandes.

D’un point de vue plus esthétique, une amélioration pourrait consister en la vision 3D du mouvement en ajoutant un axe de profondeur (Nous devrions donc nous séparer de Tkinter qui ne gère que des affichages 2D). Nous avons également la possibilité d’ajouter un affichage des données de chaque planète qui apparaitrait après un simple « clic » sur la planète. Enfin, il aurait été possible d’ajouter une interaction supplémentaire avec l’utilisateur à travers la saisie de caractéristiques menant à la création et à la simulation du mouvement d’une planète imaginaire.

***Bilan :*** Ce projet a été pour moi extrêmement important. En effet, c’est grâce à la spécialité ISN que je me suis découvert une passion pour la programmation et c’est pourquoi mes études dans le supérieur porteront sur l’informatique. Je suis donc très heureux de ce premier programme que nous avons réalisé tous les 3, Julien, Nicolas et moi. Bien que l’on ait rencontrés pas mal de difficultés, nous avons toujours réussi à avancer en améliorant le programme et en trouvant de nouvelles idées (Lune, Fenêtre utilisateur).

Lien du projet : <https://github.com/VincentFaivre/SolarSystem/projects/1>