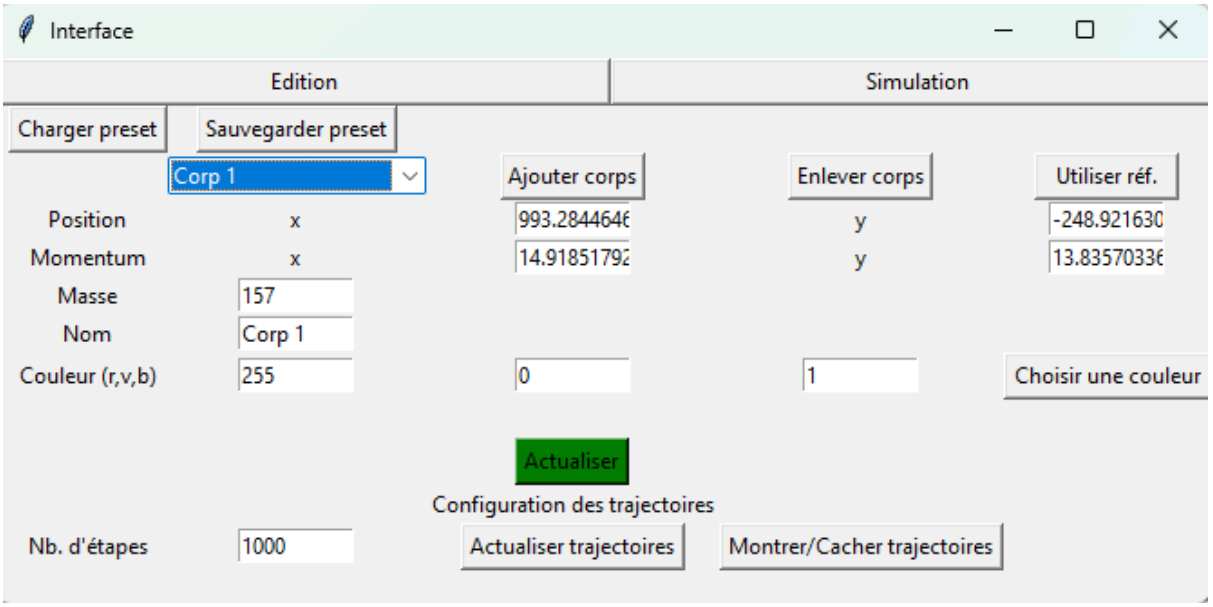
DOCUMENTATION

Ici on documente le mode d'utilisation et la structure générale du code.

Mode d'utilisation:

1) Démarrer le programme en lançant le fichier *main.py* avec python3. Deux fenêtres seront ouvertes, l'interface d'édition et l'interface graphique.

2) Utiliser l'interface d'édition dans l'onglet *Édition* pour construire l'environnement en plaçant des corps à la main ou en ouvrant un preset.



L'interface d'édition fourni les outils suivants:

- L'ajout ou l'enlèvement d'un corps avec les boutons *Ajouter corps* et *Enlever corps*
- La sélection d'un corps avec le combobox *Sélectionner corps*
- L'édition de la position, du momentum, de la masse, du nom et de la couleur du corps sélectionné avec les boîtes de texte.
  (L'édition de couleur ouvrira une nouvelle fenêtre permettant de séléctionner une couleur sans insérer les valeurs RVB à la main)
- L'édition du nombre d'étapes qui sera affiché dans l'interface graphique avec la boîte de texte *Nb. d'étapes*
- La visualisation des trajectoires avec le bouton *Montrer/Cacher trajectoires* (affiché dans l'interface graphique)
- L'actualisation des trajectoires avec le bouton *Actualiser trajectoires* (au cas oú quelques grandeurs ont été modifiés)
- Le chargement d'un preset avec *Charger un preset* (le gestionnaire de fichier sera ouvert dans un dossier *Presets*)

- Le sauvegardement d'un preset avec le bouton *Sauvegarder preset* (celui créé par l'utilisateur)
- La visualisation des trajectoires en relation au corps sélectionné avec le bouton *Utiliser réf.*
- et le bouton *Actualiser*

(Toute modification dans cet onglet devra être suivie du pressage du bouton *Actualiser*)

Il est aussi important de noter que quelques données comme la position et le momentum peuvent être éditées directement dans l'interface graphique avec la souris.

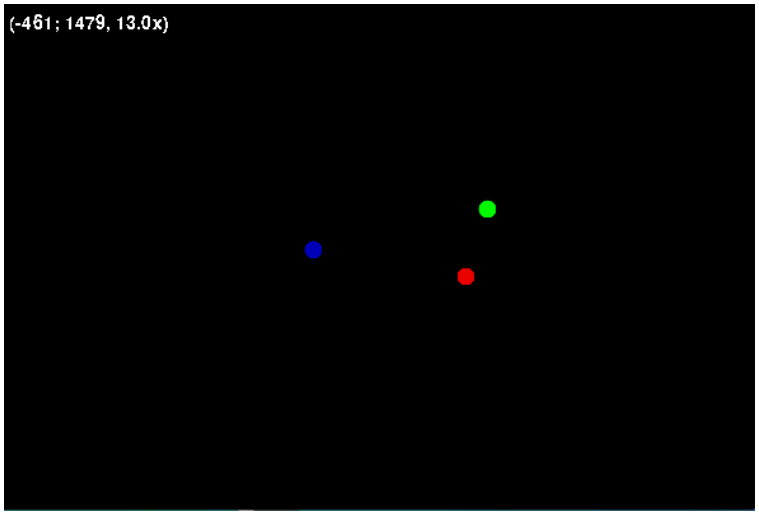
3) Ouvrir l'onglet *Simulation* dans l'interface d'édition et démarrer ou pauser la simulation avec le bouton *Pause/Play*.

Quelques autres outils sont présents dans cet onglet, tels comme:

- L'édition de la vitesse des pas, c'est à dire la vitesse de changement d'étapes (refreshrate)
- L'addition d'une étape en plus
- et le bouton *Actualiser simulation*

(Toute modification dans cet onglet devra être suivie du pressage du bouton *Actualiser simulation*)

4) Vous pourrez visualiser la simulation dans l'interface graphique ainsi que vous déplacer dans l'espace 2D avec les touches WASD et zoomer avec la roue de la souris.



5) Pour fermer la simulation, il suffit d'appuyer la touche Q ou ESC dans la fenêtre graphique ou fermer l'une des fenêtres par le bouton *Exit*.

Structure du code:

Le code est divisé en plusieurs scripts, voici un aperçu des fonctions primaires de chacun.

- **main**
  C'est *main.py* qui exécute tout le projet. D'ici sont créées deux instances, une pour l'Interface et l'autre pour le Jeu. Pour la communication on utilise deux objets *multiprocessing.Queue()* qui permet d'envoyer des valeurs entre l'interface et le jeu dans les fonctions *envoyerValeurMultiprocessing* et *recevoirMultiprocessing*.
- **Interface**
  C'est la partie qui contrôle la simulation: l'utilisateur paramètre la simulation dans l'interface, qui envoye ces valeur à Jeu.py.
- **Jeu**
  C'est le coeur du projet, c'est où sont faites la simulation gravitationnelle et tout la projection graphique, grâce à des classes et fonctions dans autres fichiers.
- **Regles/LoiGravitation**
  C'est l'application de la formule de gravitation à tous les corps envoyés par le Jeu. Il y a un fichier *LoiGravitationLeapfrog*, une optimisation non implémentée de la même loi.
- **Classes**

Comprend toutes les fonctions des calculs des dessins des sprites, ainsi que les classes définissant les corps et les caméras.

- **Trajectoires**
  Comprend les calculs de trajectoires grâce à la LoiGravitation et le dessin de celles-ci.
- **Collisions**
  Ce sont où les collisions et projections de collisions sont calculées. C'est aussi là que les corps en collision sont fusionnés.
- **FonctionsPreset**
  Toute la partie qui traite de sauvegarder et de charger les preset grâce au module *pickle* propre à python.