Projets Numériques master P2N
Cahier des charges

Bataille spatiale autour d'un trou noir

Description du projet :
Le but de projet est développer le moteur physique d'un mini-jeu de tir, où un vaisseau spatial tire sur une cible dans le voisinage d'un trou noir. Le moteur physique devra respecter autant que faire se peut les lois de la physique.

Le mini-jeu se présentera en 2D (plan équatorial du trou noir), par instructions rentrées au clavier et affichage de la dynamique des objets pris en compte.

Environnement de développement :
Le développement des programmes se fera exclusivement en language Python, avec si nécessaire l'utilisation des bibliothèques numpy, scipy, matplotlib (ou autre).
Des supports techniques peuvent être trouvés aux adresses suivantes :
`https://docs.python.org/`
`https://docs.scipy.org/doc/`
`https://matplotlib.org`

Description des livrables (description fonctionnelle du point du vue utilisateur) :

En version alpha, l'objectif est de présenter un mini-jeu dont le déroulement est le suivant :

- Le jeu se déroule sur un plan 2D correspondant au plan équatorial du trou noir.
- En début de partie, le vaisseau et la cible sont placés en opposition par rapport au trou noir (les centres du vaisseau, du trou noir et de la cible sont alignés, vaisseau et cible se trouvent de part et d'autre du trou noir). Le joueur choisit la distance à l'horizon des événements (HE) du trou noir (cf. notions physiques) à laquelle il souhaite commencer, et son vecteur vitesse initial. La distance à l'HE et la vitesse initiale de la cible sont choisies aléatoirement par le programme.
- À chaque tour de jeu, un intervalle de temps s'est écoulé. Le vaisseau et la cible sont déplacés sous l'effet de la gravité (la position courante ainsi que les trajectoires suivies jusque là sont affichées à l'écran).
- À chaque tour de jeu, le joueur peut choisir :
  - de ne rien faire pour attendre une position plus favorable ;
  - lancer un projectile lourd (le joueur pouvant choisir l'angle du projectile mais pas sa vitesse qui est prédéfinie) ; le projectile est alors géré comme le vaisseau et la cible par le jeu. Le nombre de projectiles lourds disponibles dans une partie étant limité ;
  - lancer un projectile léger explositif (en nombre limité également, avec une vitesse prédéfinie supérieure à celle du projectile lourd).
- Lors des calculs de trajectoires :
  - Si la cible rencontre un projectile lourd, si deux projectiles lourds se rencontrent, ou si le vaisseau rencontre l'un de ses projectiles lourds, la collision sera supposée élastique.

  - Si un projectile explosif rencontre la cible, celle-ci est supposée détruite est la partie se termine sur une victoire.
  - Si un projectile explosif revient sur le vaisseau, celui-ci est supposé détruit est la partie se termine sur un Game Over.
  - Si un projectile explosif rencontre un projectile lourd, ceux-ci sont supposés détruits, ils disparaissent du jeu et la partie continue.
  - Si un projectile arrive sur l'HE, il est disparaît dans le trou noir, et la partie continue.
  - Si la cible arrive sur l'HE, elle disparaît dans le trou noir, et la partie se termine sur une victoire.
  - Si le vaisseau arrive sur l'HE, il disparaît dans le tour noir, et la partie se termine par un Game Over.
  - Si un projectile dépasse la limite de la zone de jeu, il est supposé éjecté, il disparaît du jeu et la partie continue.
  - Si la cible dépasse la limite de la zone de jeu, elle est supposée éjectée, et la partie se termine sur une victoire.
  - Si le vaisseau dépasse la limite de la zone de jeu, il est supposé prendre la fuite, et la partie se termine sur un Game Over.

- Si le joueur épuise ses possibilités de tirs sans que la partie ne se soit arrêtée, le jeu calcule les trajectoires futures sur une longue période en appliquant les règles précédentes. Si aucun des cas d'arrêt du jeu ne se présente (vaisseau et cible sur des orbites stables sans collision possible), la partie est déclarée nulle.

Toutes les instructions sont rentrées pendant le jeu en mode console, et le graphe des trajectoires et positions est affiché à chaque tour, l'HE et la limite de zone de jeu sont également représentés. Une version pré-alpha sans les projectiles lourds et collisions élastiques associées pourra être d'abord développée.

La version beta pourra apporter les fonctionnalités supplémentaires suivantes :

- Meilleure gestion du lancer du projectile lourd : prise en compte du recule du vaisseau au lancer du projectile lourd (conservation de l'énergie et de l'impulsion).

- Meilleure gestion lors des sorties de la zone de jeu, suivant deux possibilités à choisir en début de partie :

  - Comparaison de la vitesse de l'objet avec la vitesse de libération au moment de la sortie de zone de jeu afin de déterminer s'il est effectivement libéré de l'attraction du trou noir. Si ce n'est pas le cas, un calcul long terme de la trajectoire est effectuée pour connaître l'instant du retour de l'objet en zone de jeu (l'objet devra refaire son apparition à ce moment de la partie ; s'il s'agit du vaisseau ou de la cible le temps de jeu est avancé à cet instant).
  - Retour immédiat de l'objet à l'opposé de sa sortie, sur le principe de conditions aux limites périodiques.

- Gestion des dégâts lors des collisions qui deviennent inélastiques.

- Mise en place d'un mode deux joueurs où la cible est remplacée par un second joueur aux mêmes possibilité que le premier à chaque tour.

La version gold pourra présenter les caractéristiques suivantes :

- Interface graphique pour gérer les instructions des joueurs.
- Remplacement des graphes statiques des positions et des trajectoires par des animations.

Description des attendus et des contraintes (description fonctionnelle et structurelle du point de vue développeur) :

- Un certain nombre de paramètres fixes doivent être ajustés : masses des objets, vitesse des projectiles, taille de la zone de jeu par rapport à l'HE, durée écoulée entre chaque tour. Des tests devront être effectués pour s'assurer de choix optimaux du point du vue de l'interêt du jeu.
- Il faudra veiller à trouver une façon adaptée pour l'utilisateur de rentrer la distance initiale au trou noir du vaisseau, le vecteur vitesse initial, et les angles de tir. Des valeurs typiques pourront être proposées.
- Il faudra définir les "hitboxes" (masques de collision) des différents éléments, c'est à dire la zone autour d'un élément où la collision se produit avec les autres éléments (si l'on réduit les éléments à des points, aucune collision ne se produira).

Concernant les calculs de trajectoires :

- En version alpha, on utilisera les expressions analytiques approchées des trajectoires (cf. notions physiques).
- En version beta, on intégrera les équations du mouvement post-Newtoniennes (cf. notions physiques).

Structurellement, le code doit se présenter sous forme d'un programme principal ("main") gérant la partie, et de fonctions python réalisant les différents événements (calcul de trajectoire, test de collision, changement des viteses aux collisions, sortie de zone de jeu, affichage graphique,...).

Notions physiques nécessaires au projet :
On note G la constante de gravitation universelle, c la vitesse de la lumière, et M la masse du trou noir.

Vitesse de libération

La vitesse de libération est la vitesse minimale que doit avoir un objet pour s'échapper de l'attraction gravitationnelle du trou noir. L'application du principe de conservation de l'énergie, nous apprend que celle-ci pour un objet se trouvant à la distance r du trou noir est

$$v_{lib}(r) = \sqrt{\frac{2GM}{r}} \tag{1}$$

Les trous noirs

Les trous noirs sont des corps tellement denses que la vitesse de libération à proximité dépasse la vitesse de la lumière. Les lois de la relativité interdisant que l'on puisse dépasser cette vitesse, les objets qui se trouvent sous une limite appelée Horizon des Événements (HE) ne peuvent se libérer de l'attraction du trou noir et sont donc définitivement piégés sour l'HE. Cela concerne les objets matériels, comme la lumière et même l'information. La zone sous l'HE est "noire" au sens où rien ne peut en ressortir.

L'HE est un sphère de rayon $R_S = \frac{2GM}{c^2}$ (on pourra vérifier qu'un corps qui se trouve à une distance $r < R_S$ d'un trou noir a bien une vitesse de libération supérieure à c). Par définition, pour être un trou noir, un corps doit avoir une densité suffisante pour que sa surface se trouve elle-même en dessous de R_S (dit rayon de Schwarzschild).

Les trous noirs présentent également une seconde frontière, une sphère de rayon $1.5R_S$, appelée sphère de photons (SP). Sous la SP, la force d'inertie d'entraînement relativiste due à la rotation autour du trou noir devient centripète au lieu d'être centrifuge. De plus, les photons peuvent présenter des trajectoires compactes. Sur la SP elle-même, les photons sont en orbite circulaire.

Les unités naturelles pour le problème sont donc les suivantes :

- les distances s'expriment en rapport au rayon de Schwarzschild (par exemple on parlera d'une longueur de $2.3R_S$) ;
- les vitesses s'expriment en fraction de la vitesse de la lumière (par exemple on parlera d'une vitesse de $0.3c$) ;

Donc dans le programme, par définition on aura $R_S = 1$ et $c = 1$ qui constituent la base des unités réduites pour les calculs.

Intégration de la dynamique :
Pour calculer les trajectoires, on intégrera le principe fondamental de la dynamique de Newton en repère polaire centré sur le trou noir :

$$\vec{a} = \frac{\sum \vec{F}}{m} \tag{2}$$

On rappelle que la vitesse et l'accélération dans le repère polaire sont données par

$$\vec{v} = \dot{r}\vec{e}_r + r\dot{\theta}\vec{e}_\theta \tag{3}$$
$$\vec{a} = (\ddot{r} - r\dot{\theta}^2)\vec{e}_r + (2\dot{r}\dot{\theta} + r\ddot{\theta})\vec{e}_\theta \tag{4}$$

Dans l'approximation post-Newtonienne (application des lois de Newton auxquelles on ajoute des corrections relativistes), les forces extérieures sont les suivantes :

- la force gravitationnelle issue la loi d'attraction universelle de Newton :

$$\frac{\vec{F}_G}{m} = -\frac{GM}{r^2}\vec{e}_r \tag{5}$$

- la correction relativiste de premier ordre :

$$\frac{\vec{F}_R}{m} = -\frac{3}{2}R_S\dot{\theta}^2\vec{e}_r \tag{6}$$

(Pour simplier le problème, on ne considère pas les effets de décalage d'horloges induits par la vitesse et par la gravité, on néglige les interactions gravitationnelle entre le vaisseau, la cible et les projectiles).
On obtient ainsi l'équation radiale :

$$\ddot{r} = -\frac{GM}{r^2} + (r - \frac{3}{2}R_S)\dot{\theta}^2 \tag{7}$$

On peut montrer qu'il existe des solutions analytiques approchées des équations post-Newtoniennes. Ces solutions fournissent les équations polaires suivantes pour les orbites :

$$r_{\pm}(\theta) = \frac{\frac{q}{2} \pm \sqrt{\frac{q^2}{4} - \frac{3}{2}qR_S(1 + e\cos\theta)}}{1 + e\cos\theta} \tag{8}$$

avec q et e des paramètres définis par les conditions initiales (paramètre de longueur et excentricité). On remarquera que si les effets relativistes sont négligeables, i.e. $R_S \ll q$ (par exemple pour la Terre R_S est de l'ordre de quelques centimètres alors que q est supérieur au rayon terrestre), on retrouve l'équation polaire d'une conique (ellipse ($e < 1$), parabole ($e = 1$) ou hyperbole ($e > 1$)). La vitesse angulaire est définie par la conservation du momement cinétique :

$$r^2\dot{\theta} = \frac{L_0}{m} \tag{9}$$

La valeur de $\ell_0 = \frac{L_0}{m}$ étant elle aussi définie par les conditions initiales.
La solution $r_+(\theta)$ est à appliquer si $r > 3R_S$, par contre on applique la solution $r_-(\theta)$ si $r < 3R_S$. Ces solutions approchées négligent le phénomène de précession du péricentre qui peut dans la réalité être très important.

Collisions :
Les collisions élastiques obéissent aux lois de conservation de l'énergie cinétique et de la quantité de mouvement. Pour les collisions inélastiques, c'est l'énergie mécanique totale qui est conservée (incluant le défaut d'énergie utilisé pour créer des dégâts sur les corps en collision).

Algorithmes à utiliser :
On intégre les équations du mouvement sur une durée T (durée écoulée entre deux tours de jeu). L'intervalle de temps $[t_i, t_i + T]$ (où t_i est l'instant initial – date du tour précédent –) est divisé en N instants (avec N grand), $t_i < t_{i+1} < ... < t_{i+N} = t_i + T$, avec $t_{k+1} - t_k = \frac{T}{N} = \Delta t$ pas de discrétisation temporel.

Calcul de $\theta(t)$:
On calcule $\theta_k = \theta(t_k)$ en intégrant l'équation 9 en utilisant un schéma de premier ordre donné par le pseudo-code suivant :

```
Pour k de i à i + N − 1 faire
    θ_{k+1} ← θ_k + (ℓ_0 / r_k^2) Δt
Fin pour
```

La valeur de θ_i ayant été calculée au tour précédent ou étant la condition initiale $\theta_0 = 0$ au premier tour, ℓ_0 est la constante calculée à partir des conditions initiales. La valeur de r_k est soit obtenue à partir de la formule analytique approchée équation 8 comme $r_{\pm}(\theta_k)$ (version alpha), soit obtenue à partir de l'algorithme ci-après (version beta).

Calcul de $r(t)$:
On calcule $r_k = r(t_k)$ en intégrant l'équation 7 par l'algorithme dit du Leapfrog qui est donné par le pseudo-code suivant :

```
Pour k de i à i + N − 1 faire
    r_{k+1} ← r_k + v_{k+1/2} Δt
    v_{k+3/2} ← v_{k+1/2} + a(r_{k+1}) Δt
Fin pour
```

avec $v_{k+1/2} = \dot{r}(t_k + \Delta t/2)$ la vitesse dans le référentiel accompagnant la rotation de l'objet autour du trou noir, les valeurs de r_i et $v_{i+1/2}$ ayant été calculées au tour précédent, ou étant définies par la condition initiale r_0 et par $v_{1/2} = v_{r0} + a(r_0)\Delta t$ (v_{r0} étant le composante radiale de la vitesse initiale). L'accélération dans le référentiel en rotation est donnée par $a(r) = -\frac{GM}{r^2} + (r - \frac{3}{2}R_S)\frac{\ell_0^2}{r^4}$.

Référent : D. Viennot, `mailto:david.viennot@utinam.cnrs.fr`