

Outils Info. - TD : Projet S2

Exercice 1 – DISQUE

Q1) Simuler un disque dont le rayon initial est 1 et qui augmente de 0.03 à chaque itération de la simulation (sans aucune attente). La simulation doit uniquement afficher, *dans le terminal*, l'aire du disque.

Exercice 2 – DIAG

Q2) Simuler un rectangle qui est, au début, un carré de côté 1 et dont un des côtés augmente de 0.03 et l'autre de 0.09 à chaque itération de la simulation (sans aucune attente). La simulation doit uniquement afficher, *dans le terminal*, la longueur de la diagonale du rectangle.

Exercice 3 – DISQUE (graphique)

Q3) Simuler un disque dont le rayon initial est 1 et qui augmente de 0.03 toutes les 10 millisecondes (utiliser `attendre_pendant`). La simulation doit afficher, au centre d'une fenêtre, le disque en vert.

Q4) Simuler maintenant un autre disque, mais en utilisant comme unité le centimètre. Le rayon initial est de 1cm et le disque augmente de 0.03cm toutes les 10 millisecondes. La simulation doit afficher le disque en bleu et en taille réelle, sachant que sur la plupart des écrans, il y a environ 38 pixels par centimètres.

Exercice 4 – MRU

Le mouvement rectiligne uniforme est défini par les lois suivantes (redondantes) :

- $\dot{x} = cst$
- $\dot{x} = v$ et $v = cst$

Le but est de simuler, à l'aide de la méthode d'Euler, avec un pas de simulation de 10ms, une balle dont la vitesse horizontale est constante à 5 m.s^{-1} . La simulation doit afficher la balle sous forme d'un disque rouge (qui démarre par exemple au centre de la fenêtre).

Q5) Quelle(s) variable(s) faut-il utiliser pour stocker l'état de ce système ?

Q6) Écrire la simulation en Python.

Exercice 5 – Cinétique chimique

On considère deux équations $A + B \mapsto C$ (vitesse k_1) et son inverse $C \mapsto A + B$ (vitesse k_2). Ceci définit les lois suivantes (en terme de concentrations, A est la concentration de l'espèce A etc) :

- $\dot{A} = -k_1 AB + k_2 C$
- $\dot{B} = -k_1 AB + k_2 C$
- $\dot{C} = +k_1 AB - k_2 C$

Le but est de simuler, à l'aide de la méthode d'Euler, avec un pas de simulation de 10ms, La simulation doit afficher les concentrations avec un print sous forme de rectangles dont la hauteur dépend de la concentration.

Q7) Quelle(s) variable(s) faut-il utiliser pour stocker l'état de ce système ?

Q8) Écrire la simulation en Python.

Exercice 7 – MRUA

Le mouvement rectiligne uniformément accéléré est défini par les lois suivantes (redondantes) :

- $\ddot{x} = cst$
- $\dot{x} = v$ et $\dot{v} = cst$
- $\dot{x} = v$ et $\dot{v} = a$ et $a = cst$

Le but est de simuler, à l'aide de la méthode d'Euler, avec un pas de simulation de 10ms, une balle soumise à la gravité (dont la vitesse verticale est uniformément accélérée à 9.80665 m.s^{-2}). La simulation doit afficher la balle sous forme d'un disque rouge (qui démarre par exemple au centre de la fenêtre).

Q11) Quelle(s) variable(s) faut-il utiliser pour stocker l'état de ce système ?

Q12) Écrire la simulation en Python.

Exercice 8 – MCU

Le mouvement circulaire uniformément accéléré est défini par les lois suivantes (où θ est l'angle et ω la vitesse angulaire) :

- $\dot{\theta} = cst$
- $\dot{\theta} = \omega$ et $\omega = cst$

Le but est de simuler, à l'aide de la méthode d'Euler, avec un pas de simulation de 10ms, l'aiguille d'une horloge dont la vitesse angulaire est constante à $\frac{2\pi}{60} rad.s^{-1}$ (radians par seconde). La simulation doit afficher l'aiguille sous forme d'une ligne allant du centre de la fenêtre au bout de l'aiguille.

Q13) Quelle(s) variable(s) faut-il utiliser pour stocker l'état de ce système ?

Q14) Quelles sont les coordonnées x, y du bout de l'aiguille sachant une valeur de θ ?

Q15) Écrire la simulation en Python.

Exercice 9 – Affichage à l'échelle

Q16) Reprendre vos simulations (MRU, MRUA, MCU) précédentes. Si ce n'est pas le cas, faire que vos variables d'état aient des unités physiques claires (mètres, secondes, etc) en utilisant des commentaires Python. Faire que l'affichage soit contrôlé par des variables qui définissent comment passer des unités physiques aux unités d'affichage (en $\frac{pixels}{metres}$).

Exercice 10 – Affichage d'une courbe

Le but est de tracer l'altitude de la balle qui tombe, en fonction du temps.

Q17) Reprendre la simulation MRUA. Faire qu'elle affiche dans une autre fenêtre : créer une deuxième fenêtre et la stocker dans une autre variable, appelée par exemple `f2`. Tracer à chaque itération (sans effacer `f2`) un point (petit disque) dont la coordonnée horizontale correspond à l'accumulation des temps écoulés (des dt) et la position verticale correspond à la hauteur de la balle.

Q18) Tracer aussi la vitesse.

Codes de Référence

Exemple d'attente simple :

```
1 # Attendre 42 millisecondes
2 attendre_evenement(f, 42)
3
4 print("42 millisecondes écoulées ?")
```

Exemple de gestion d'événements clavier :

```
1 # Attendre 42 millisecondes ou un événement (exemple : touche 'a')
2 attendre_evenement(f, 42)
3 ev = dernier_evenement(f)
4
5 if ev is None:
6     print("42 millisecondes écoulées")
7     # ...
8 else:
9     if ev == 65:
10        print("Touche 'a' appuyée")
```