

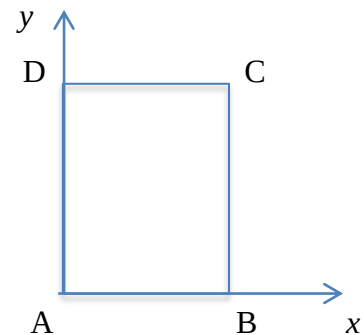
Billard

Objectif

L'objectif de ce projet est de simuler les déplacements d'une boule sur un billard.

Un billard est représenté par un rectangle ABCD.

A chaque fois que la boule touche un des bords verticaux, elle marque des points.



Données du problème

Les données du problème sont les suivantes :

- la largeur et la hauteur du billard
- le nombre de points marqués quand la boule rencontre un bord vertical
- la position de départ de la boule qui part forcément du bord horizontal bas
- l'angle avec lequel la boule est lancée

Trajectoire de la boule

- la boule est considérée comme un point
- elle est lancée à partir du point P avec un angle α avec Ax compris entre 0 et $\pi/2$ exclus
- elle chemine en ligne droite jusqu'à la rencontre d'un bord du billard
- sur chaque obstacle rencontré, elle rebondit symétriquement à la normale de l'obstacle
- quand elle arrive dans un angle du billard, elle repart en sens inverse dans la même direction

Pour ne pas gérer tous les phénomènes physiques, nous admettrons qu'il n'y a pas de frottements et que les impacts ne modifient pas la vitesse de la boule.

Résultats attendus

On demande :

- de **calculer** et de **tracer** la trajectoire de la boule. Pour l'affichage graphique, on pourra utiliser le module *turtle* de python.
- de **calculer** le nombre de points marqués par celle-ci
- sachant que :
 1. dans une première version, la boule s'arrête après avoir parcouru n **vecteurs** de droite avec n donné
 2. dans une seconde version, la boule s'arrête une fois qu'elle a touché chacun des bords du billard

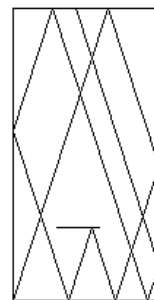


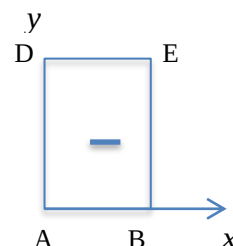
Figure 1 : Exemple de parcours de la boule

Quelques indications permettant d'arriver plus facilement à ce résultat sont fournies ci-après.

Le programme contenant les deux versions ne devrait pas dépasser la centaine de lignes de code (sans les docstrings et commentaires).

Question subsidiaire

On peut ajouter un ou plusieurs obstacles au sein du flipper



Aide

Déplacement

1) On peut remarquer que chaque vecteur :

- est porté alternativement par une droite d'équation $y = tx + r$ ou $y = -tx + r$ avec $t = \tan(\alpha)$ r déterminé par le point de départ du vecteur
- a de plus un sens montant ou descendant

Représentation

On propose de représenter :

- chaque point par le couple de ses coordonnées
- chaque bord du billard par ses deux points d'extrémité

Travail préliminaire

On propose de définir les deux fonctions suivantes facilitant ensuite l'écriture du programme général.

1) Intersection droite et segment horizontal

Ecrire une fonction qui, connaissant :

- l'équation d'une droite
- les points extrêmes d'un segment horizontal

retourne *None* si la droite ne coupe pas le segment et les coordonnées du point d'intersection où elle le coupe sinon.

2) Intersection droite et segment vertical

Ecrire une fonction qui connaissant :

- l'équation d'une droite,
- les points extrêmes d'un segment vertical

retourne *None* si la droite ne coupe pas le segment et les coordonnées du point d'intersection où elle le coupe sinon.