Questions sur la séance 3 de PCII

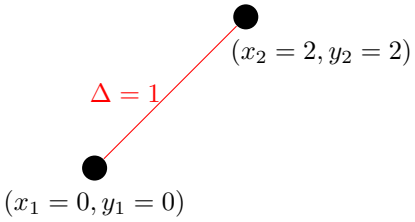
January 26, 2023

1 Génération des points

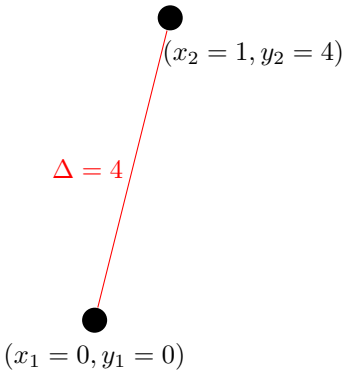
Tes points sont générés en effet de manière aléatoire. Tu dois cependant vérifier que la pente Δ entre chaque point n'est pas plus élevé que la vitesse de descente de ton ovale afin que ce soit réalisable par le joueur. Sachant que la pente Δ équivaut à :

$$\Delta = \frac{y_2 - y_1}{x_2 - x_1}$$

Par exemple :



Ici tes deux point générés aléatoirement dans le parcours ont une pente satisfaisante si la vitesse de descente est de 2 pixel. Par contre avec :



Ici la pente entre les deux point généré est de 4 alors que la vitesse de descente est de 2, c'est beaucoup trop grand il faut donc générer d'autres points.

Il te faut donc une fonction qui calcule ce Δ de type :

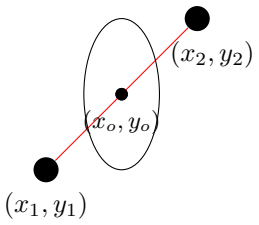
```
/**
 * Calcule la pente entre deux points
 * @param a le premier point
 * @param b le deuxieme point
 * @return la pente entre a et b
 */

 public float pente(Point a,Point b){
    return (float)(b.y-a.y)/(b.x-a.y);

 }
```

Ensuite tu utilise la fonction *pente* lorsque tu génères tes points pour vérifier que la pente est correcte. Si ce n'est pas le cas tu génères de nouveaux points.

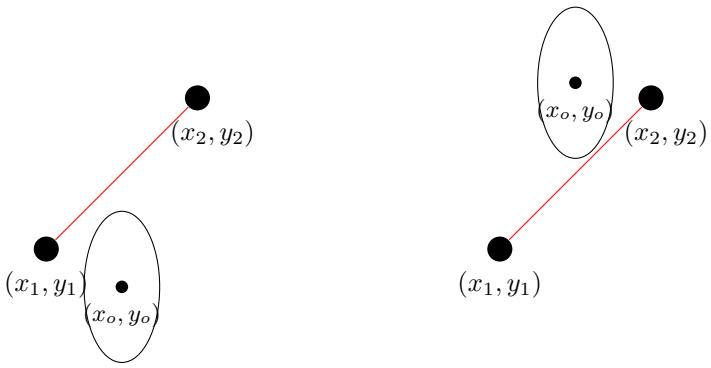
2 Vérifier si on a perdu

Ta deuxième question revient à savoir comment on détermine si on a perdu. Il y a finalement que deux cas :

- L'ovale est dans la ligne , le joueur ne perd pas :



- L'ovale est en dessous ou au dessus de la ligne, le joueur perd :



Tu connais :

- les coordonnées (x_o, y_o) de ton ovale.
- les deux points $(x_1, y_1), (x_2, y_2)$ entre lequel se trouve l'ovale
- la pente Δ entre (x_1, y_1) et (x_2, y_2)

Tu veux finalement déterminer si (x_o, y_o) est sur la droite entre (x_1, y_1) et (x_2, y_2) pour savoir si tu as perdu ou non. Il te suffit de calculer alors :

$$y = \Delta \times x_o + (y_1 - (\Delta \times x_1))$$

(regarde comment calculer les coordonnées d'un point connaissant la pente d'une droite si tu veux les détails de l'équation). Si ce y calculé est plus petit ou plus grand que y_0 en prenant en compte le rayon de l'ovale cela veut dire que tu n'est pas sur la droite et tu as perdu.

3 La variable position

La variable position est en effet le score du joueur. C'est un entier qui reflète jusqu'où le jouer est allé avant de perdre. Donc tu l'augment à chaque fois que tu fais avancer le parcours

4 Pour ton code....

Je pense que vu tous ce qu'il y a au dessus tu as assez d'éléments pour ajouter les choses nécessaires dedans :

- Une fonction qui calcule la pente
- Une génération qui prend en compte cette pente
- Une fonction qui calcule si l'ovale est sur la droite entre deux points avec la fonction pente