

Yanis MERCIER-TALLET

S1B

Choix algorithmique

J'ai choisi de parler de la classe Obstacle, celle ci se trouve dans l'onglet 'modes_speciaux' de la ligne 14 à la ligne 23, et permet de créer un obstacle.

Un objet de cette classe a quatre attributs :

- obsX : l'abscisse du coin haut gauche de l'obstacle
- obsY : l'ordonnée du coin haut gauche de l'obstacle
- obsLargeur : largeur de l'obstacle
- obsHauteur : hauteur de l'obstacle

La valeur de chaque attribut est définie avec un entier aléatoire borné en minimum et en maximum, pour permettre une apparition des obstacles au niveau du centre du terrain avec une taille plus au moins grande.

Afin de générer plusieurs obstacles, j'utilise un tableau que je remplis d'objets de la classe Obstacle, une autre façon de faire possible est de se contenter d'un tableau rempli d'entiers.

En effet je pourrais faire un tableau où 4 éléments à la suite représenteraient abscisse, ordonnée, longueur et hauteur d'un obstacle. Pour cela :

Tout d'abord je crée un tableau obstacle[] de mon nombre d'obstacles souhaité fois 4

(exemple 12 pour 3 obstacles). Au moment d'initialiser mes obstacles je pourrais faire une boucle pour i de 0 à mon nombre d'obstacle fois 4 par pas de 4 en définissant à chaque tour de boucle :

- obstacle[i] comme mon abscisse
- obstacle[i+1] comme mon ordonnée
- obstacle[i+2] comme ma longueur
- obstacle[i+3] comme ma largeur

Avec des entiers aléatoires comme dans ma classe.

Le changement est ensuite similaire pour les autres fonctions (dessinerObstacle et collisionObstacleBall) en changeant les boucles c'est à dire boucle de 0 à nombre d'obstacle fois 4 par pas de 4 et en appelant obstacle[i] ([i+1]/[i+2]/[i+3],) au lieu de obstacle[i].obsX(.obsY/.ObsLargeur/.ObsHauteur).

L'avantage majeur d'avoir une classe ici est de «manier» des objets, qui représentent chacun un obstacle individuellement, le code est ainsi beaucoup plus compréhensible et lisible.

Si on utilise un tableau d'entier en revanche le code est légèrement plus compacte.

Selon moi, la différence se fait donc dans la compréhension, dans cette seconde version on ne comprend pas vraiment ce que l'on utilise, un obstacle[i].obsY nous indique bien l'ordonnée d'un obstacle tandis que obstacle[i+1] est beaucoup moins parlant.

Je vous invite à aller voir cette version modifiée en lançant : sketch_Pong_autre_obstacle → onglet 'modes_speciaux'

Portfolio

En partant d'un code calculant le rebond d'une balle dans une «fenêtre», nous avons pour objectif de re-cr  er le jeu Pong. Dans ce projet j'ai donc cr     le menu avec les diff  rents modes de jeux, les raquettes, ainsi qu'un syst  me de calcul du score. Ce projet m'a surtout permis de d  couvrir l'int  r  t des classes ainsi que de nombreuses fonctions sous Processing. Le choix du nom des variables et la documentation se sont r  v  l  s   tre des aspects cl  s de la programmation.

