Mini-projet: Rush Hour. 1ère partie

Rush Hour est un casse-tête qui simule un parking à l'heure de pointe. Le but du jeu est d'extraire le véhicule rouge du parking dans lequel plusieurs autres véhicules bloquent la sortie. Le plateau de jeu est un carré de six cases de côté, comportant une sortie et des rainures imposant des directions aux véhicules: ceux-ci ne peuvent qu'avancer et reculer, donc vers le haut et le bas, ou vers la droite et la gauche, selon leur direction initiale. Il est interdit de soulever les véhicules. Les véhicules sont des voitures (deux cases de long) ou des camions (trois cases de long). Une des voitures est rouge ; c'est celle qu'il faut faire sortir du parking. La sortie est à droite sur la 3ème ligne.

Écrire une classe `Vehicule`, qui représente un véhicule du jeu. Celui-ci est caractérisé, à sa création, par son nom, sa longueur, sa direction, sa position dans la grille, sa couleur. Le nom sert à désigner les véhicules. Un nom est composé d'un seul caractère majuscule. La voiture rouge à sortir s'appelle toujours "X". Pour les autres attributs, on a par exemple, sur l'image ci-dessus :

- pour la voiture rose :
  - longueur : 2
  - direction : verticale
  - position : x=2, y=0
  - couleur : rose
- pour le camion jaune :
  - longueur : 3
  - direction : horizontale
  - position : x=3, y=3
  - couleur : jaune

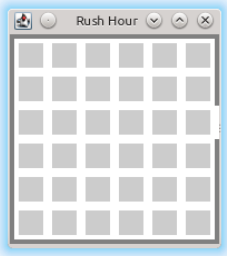
Cette classe doit fournir des méthodes permettant de visualiser le véhicule dans une fenêtre et de le déplacer. Pour le déplacement, on fournit les paramètres suivants: direction (haut, bas, droite ou gauche) et distance (en nombre de cases).

Vous utiliserez les classes données, développées à partir des classes du projet `shapes` vu en cours :

- `Canvas` : permet d'afficher des formes sur une fenêtre
- `Rectangle` : un rectangle coloré avec un texte, qu'on peut afficher. Servira à représenter un véhicule. Pour le déplacement du véhicule, utilisez les méthodes `slowMoveXXX` qui permettront de mieux suivre le déplacement.

On donne également une interface `Constants`, qui définit un certain nombre de constantes utilisées dans les différentes classes de l'application.

Écrivez une classe `Grid` qui permet d'afficher la grille du jeu : les bords tout autour, sauf à l'endroit de la sortie, et les cases de la grille, avec les rails le long desquels les véhicules se déplacent. Ci-dessous un exemple d'affichage du parking :



Écrivez une classe `Parking` qui permet d'afficher la grille du jeu et les voitures qui occupent le parking. Cette classe doit fournir les méthodes suivantes :

- une méthode permettant d'ajouter une voiture sur le parking
- une méthode qui retourne en résultat une voiture du parking étant donné son nom

Écrivez une classe `RushHour1` qui permettra de tester la première partie de votre application. Cette classe contiend une méthode `main` qui crée un parking, y ajoute des voitures, puis réalise une série de déplacements.

Les voitures à placer :

- "X" : voiture rouge, horizontale, à la position x=1, y=2
- "A" : voiture verte, horizontale, à la position x=1, y=3
- "B" : voiture orange, verticale, à la position x=1, y=4
- "C" : voiture bleue, horizontale, à la position x=2, y=5
- "O" : camion jaune, vertical, à la position x=3, y=2
- "P" : camion violert, vertical, à la position x=5, y=3

Les déplacements : PU3, OU2, CR2, AR3, OD3, XR1, BU4, XL1, OU3, CL3, AL3, PD3, OD3, XR5. On donne le nom du véhicule, le sens de déplacement (Up, Right, Left, Down), et la distance en nombre de cases).