

---

## Projet : Rush Hour

---

### 1 Présentation du jeu

**Rush Hour**<sup>1</sup> 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.



**Règles** Le plateau de jeu est un carré de six cases de côté, comportant une sortie, à droite sur la 3<sup>ème</sup> ligne. Le plateau de jeu comporte des rainures entre les cases 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).

Les véhicules sont identifiés chacun par une lettre et une couleur. Les déplacements sont identifiés par une lettre (Up, Right, Left, Down) et un chiffre (nombre de cases). Ainsi, AU2 signifie qu’on déplace la voiture de couleur “light\_green” de deux cases vers le haut. Le tableau ci-dessous contient les descriptions de tous des véhicules existants :

voitures	X	red	A	light_green	B	orange	C	blue
	D	pink	E	dark_purple	F	green	G	dark_grey
	H	grey	I	light_yellow	J	dark_brown	K	dark_green
camions	O	yellow	P	purple	Q	dark_blue	R	green

On dispose d’un certain nombre de cartes représentant des configurations initiales du parking, de différents niveaux correspondant à leur difficulté. On donne en fichiers joints des descriptions de ces cartes, dans lesquelles chaque véhicule est représenté par sa lettre, sa direction (Horizontale ou Verticale) et sa position (x,y). Ainsi, AV03 représente la voiture de couleur “light\_green” placée verticalement à la position (0,3).

**Fin de partie** La partie se termine quand la voiture rouge sort du parking.

---

<sup>1</sup>cf [https://fr.wikipedia.org/wiki/Rush\\_hour\\_\(casse-t%C3%A0te\)](https://fr.wikipedia.org/wiki/Rush_hour_(casse-t%C3%A0te))

## 2 Développement du jeu

**Boucle du jeu** Au début du jeu, on choisit une des configurations et on l’affiche. Ensuite, la boucle principale du jeu consistera, à chaque itération, à :

- demander au joueur d’entrer (au clavier) son prochain déplacement,
- calculer le nouvel état du jeu : déplacer le véhicule si possible, ...
- afficher le nouvel état du jeu,

On sort de la boucle quand le joueur a gagné.

**Découpage du code** La mise en œuvre du jeu sera découpée en deux parties : les classes implémentant le modèle du jeu (i.e. la logique ou les règles du jeu) et les classes permettant d’afficher le jeu (i.e. la vue ou la représentation graphique du jeu). Les classes du *modèle* doivent offrir au minimum des méthodes publiques permettant aux classes de la *vue* de transmettre le déplacement entré par le joueur (pour lancer une nouvelle itération de jeu) et d’obtenir l’état du jeu (pour l’afficher).

*Remarque :* vous pourrez utiliser les classes fournies aux TP précédents pour la partie graphique.

**Tâches à réaliser** Vous développerez le jeu en équipes. Vous devrez donc :

- définir ensemble les principales interfaces du jeu en précisant leurs contrats. Ces interfaces sont celles représentant le modèle du jeu : entités composant le jeu, et les relations entre elles.
- vous répartir le développement : implémentation de ces interfaces mettant en œuvre la logique du jeu, des classes permettant d’afficher le jeu, et de la boucle principale du jeu.
- développer chacun votre partie en testant bien qu’elle respecte sa part du contrat.
- mettre vos classes en commun pour obtenir un jeu complet et fonctionnel.

## 3 À rendre

Vous devez rendre un fichier zip contenant :

- un diagramme de classes UML détaillé décrivant les classes que vous avez implémentées
- le code source de vos classes bien documenté
- les éventuels fichiers de données ou images utilisés
- un document pdf décrivant le fonctionnement du jeu et la façon d’y jouer (manuel utilisateur)