

# Rapport projet Zuul

HOUEE Adrien - E3T

## Scénario

Le squelette d'un skateur légendaire répondant au nom de **Dem BONES** reprend vie et souhaite de nouveau "vivre" sa passion, **le skateboard**.

Pour cela, il devra dans un premier temps réunir toutes les pièces nécessaires pour remonter une planche de skateboard complète, c'est à dire :

- 1 plateau
- 1 grip
- 2 trucks
- 8 roulements à billes
- 4 roues
- la visserie

Pour chacune de ces pièces, des défis devront être réalisés.

Une fois la nouvelle planche de skateboard montée, Dem BONES devra **se rendre sur des spots** de skateboard connus pour **réaliser des tricks** (figures) et revivre sa passion.

## Réunir toutes les pièces du skateboard

### Le plateau et le grip

Le vendeur **prête un plateau** de skateboard à Dem BONES et lui demande de **réaliser quelques tricks**. Si tous les tricks sont réalisés, alors le vendeur **offre le plateau et le grip** à Dem BONES

**OU**

Dem BONES **achète** la plateau avec de **l'argent** et le vendeur lui **offre le grip**.

***Dépend de la conversation entre les 2 personnages***

### Les 2 trucks

Les roulements à billes

Les roues

La visserie

## Liste des spots et des figures à réaliser

### Figures

Nom de la figure	Commande
Ollie	sz
Nollie	zs
Pop Shuvit	sd
FS Pop Shuvit	sq
Nollie Pop Shuvit	zq
Nollie FS Pop Shuvit	zd
Kickflip	se
Heelflip	sa

### Lieux

- Palais de Tokyo, Paris
- Place de la République, Paris
- Place de la Bastille, Paris
- Cosanostra Skatepark, Chelles
- Skatepark intérieur boutique

# Rapport

## Auteur

HOUÉE Adrien - E3T

## Thème

Dans une boutique de skateboard, Dem Bones doit monter sa nouvelle planche de skateboard.

## Réponses aux exercices

### Exercice 7.5

1. L'exercice 7.4 est bien terminé.
4. Nous avons apporté cette modification afin d'optimiser le code et de ne pas le répéter. De plus, si nous devons effectuer un changement dans l'affichage des informations de la pièce (Room) courante, en changeant cette fonction seulement, la modification sera répercutée à tous les endroits nécessaires.

### Exercice 7.6

1. Quoi qu'il arrive une valeur sera affectée à chaque attribut, donc que la valeur soit null en paramètre ou null par défaut, cela ne change pas grand chose.

### Exercice 7.7

1. Les sorties d'une pièce (Room), sont relative à la pièce, elle n'ont donc aucun intérêt à ne pas être dans la classe Room.

### Exercice 7.8

4. La procédure setExits est devenue inutile car elle n'est pas assez généraliste et limite les évolutions que l'on peut apporter au projet. Elle a été remplacée par setExit.

### Exercice 7.9

3. la fonction `keySet` nous retourne l'ensemble des clés d'une `HashMap`

### Exercice 7.18.6

2) c. Nous n'avons plus besoin du scanner car nous allons passer par une interface graphique et donc récupérer la valeur des inputs.

### Exercice 7.20

En effet, les getters sont utiles pour récupérer les informations et les manipuler à l'extérieur de la classe `Item`.

### Exercice 7.21

Les informations relatives à un `Item` dans la classe `Room` seront produites grâce à la classe `Item`.

Les différentes `String` permettant les descriptions d'un `Item` seront donc produites grâce à la classe `Item`.

La classe qui affichera ces informations sera la classe `GameEngine` car c'est dans cette classe qu'on met à jour l'affichage de l'interface graphique.

### Exercice 7.26

La commande `back` fonctionne correctement car lorsque l'on exécute la commande, on procède à la vérification si la pile est vide ou non, donc si on peut retourner en arrière ou non.

### Exercice 7.29

Un `Player` devra avoir pour information unique sa pile de mouvements, son poids maximal, son inventaire et sa pièce courante.