

# RAPPORT

Surfman (<https://git.esiee.fr/surf-game/surf-man.git>)

(A3P 2015/2016 J4)

# auteur

- Lehmoudi Kamel

# Thème

- Un surfeur dans une plage doit récupérer tout les planches de surfeur perdues pour gagner .

# Résumé du scenario

# Plan

# Scénario détaillé



Lieux, items, personnages



Situation gagnante et perdantes

commentaire

# Réponses aux exercices

- 7.5 : printinfo location : on règle le problème de cohésion en ajoutant une methode qui retourne l'information sur la salle courante.
- 7.6: getexit: on modifie les attribues de la classe room en private et on les encapsules ensuite on fait un accesseur pour decoupler la classe Room et Game
- 7.7: getexitstring: on ameliore la cohésion en separant l'info produite et l'affichage , l'info est produite dans Room et l'affichage se fait dans game

# Réponses aux exercices

- 7.8: on remplace les 4 parametre de sortie par HashMap pour par la suite pouvoir rajouter d'autre direction plus facilement
- La méthode devenue maintenant inutile est getExit() car on peut accéder a la description d'une room dans le hashmap .
- 7.8.1:deux nouvelles directions ont était ajouté “bas” “haut” et une pièce utilisant ses deux directions vHole
- 7.9: la méthode keyset renvoie l'ensemble des clés que l'on a défini pour le hashmap donc ici les directions .

# Réponses aux exercices

- 7.10: la méthode `GetExitString()` du code 7.7 doit retourner une string,
- Ensuite la première ligne initialise une string `returnString`, après on définit le type des clés du hashmap `exits` qui sont des string, on fait par la suite une boucle `for` qui parcourt tout le hashmap et qui va ajouter au string `returnString` tout les directions en forme de string de la Room sur laquelle on a appelé la méthode , enfin on retourne `returnString`.
- 7.10.2: la java doc est ajouté , la classe `game` ne sert qu'à afficher de l'information alors que la classe `room` doit manipuler des données d'où la différence de nombre de méthodes.

# Réponses aux exercices

- 7.14: on ajoute un tableau avec les commandes valides qu'on pourra modifier plus tard pour ajouter d'autres commandes.
- On ajoute des instructions à la méthode `processCommand()` pour effectuer un affichage lorsque une commande valide est reconnue.
- (je me suis arrêté à cet exercice, j'ai un problème avec la méthode `processCommande`.)

Mode d'emploi

# Déclaration anti-plagiat