

Itération 1 groupe 1

Exercice 7.1 :

Que fait cette application?

L'application est un jeu d'aventure basé sur du texte écrit en lignes de commandes.

Quelles commandes le jeu accepte-t-il?

Le jeu accepte les commandes : help, go, quitte.

Que fait chaque commande?

Help :

Imprime des informations d'aide.

Nous imprimons ici un message stupide, cryptique et une liste des mots de commande

go :

Essayez d'aller dans une direction. S'il y a une sortie, entrez la nouvelle salle, sinon affiche un message d'erreur.

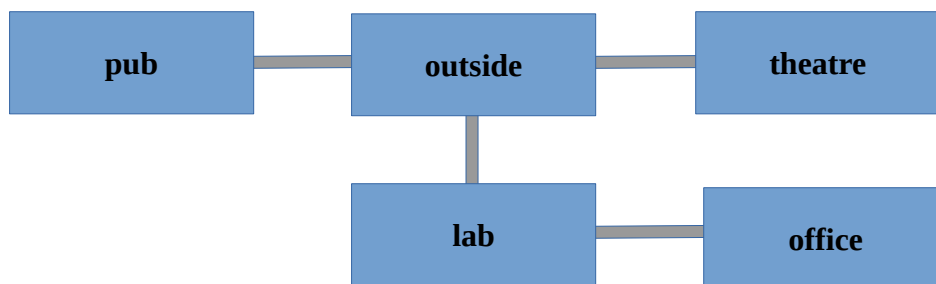
quitte :

Vérifie le reste de la commande pour voir si nous avons vraiment quitté la partie.
retour true, si cette commande quitte le jeu, false sinon.

Combien de pièces y a-t-il dans le scénario?

Il y a directions dans le jeu : north, south, east, west.

la carte des salles existantes :



Exercice 7.2 :

Game.java :

Cette classe est la classe principale de l'application "World of Zuul".

Pour jouer à ce jeu, créez une instance de cette classe et appelez la méthode "play" .

Cette classe principale crée et initialise toutes les autres, elle crée le parseur et toutes les chambres, et démarre le jeu.

Room.java :

Une "salle" représente un endroit dans le décor du jeu, elle est connectée à d'autres salles via des sorties. Les sorties sont étiquetées nord, est, sud, ouest.

Pour chaque direction, la pièce stocke une référence de la pièce voisine, ou null s'il n'y a pas de sortie dans cette direction.

Command.java :

Cette classe contient des informations sur une commande émise par l'utilisateur.

Une commande est actuellement composée de deux chaînes: un mot de commande et un autre.

La façon dont ceci est utilisé est:

- Les commandes sont déjà vérifiées pour être valide.
- le mot de commande : Si l'utilisateur écrit une commande non valide (un mot qui n'est pas connu) alors le mot de commande est null.
- Si la commande n'a qu'un mot, le deuxième mot est null.

Parser.java :

Chaque fois qu'il est appelé, il lit une ligne du terminal et tente d'interpréter la ligne comme une commande de deux mots. Il retourne la commande en tant qu'objet de la classe Command. Le parseur possède un ensemble de mots de commande connus. Il vérifie les entrées de l'utilisateur avec les commandes connues, et si l'entrée n'est pas une des commandes connues, elle retourne un objet de commande qui est marqué comme une commande inconnue.

CommandWords.java :

Cette classe contient une énumération de tous les mots de commande connus du jeu.

Il est utilisé pour reconnaître les commandes au fur et à mesure de leur saisie.

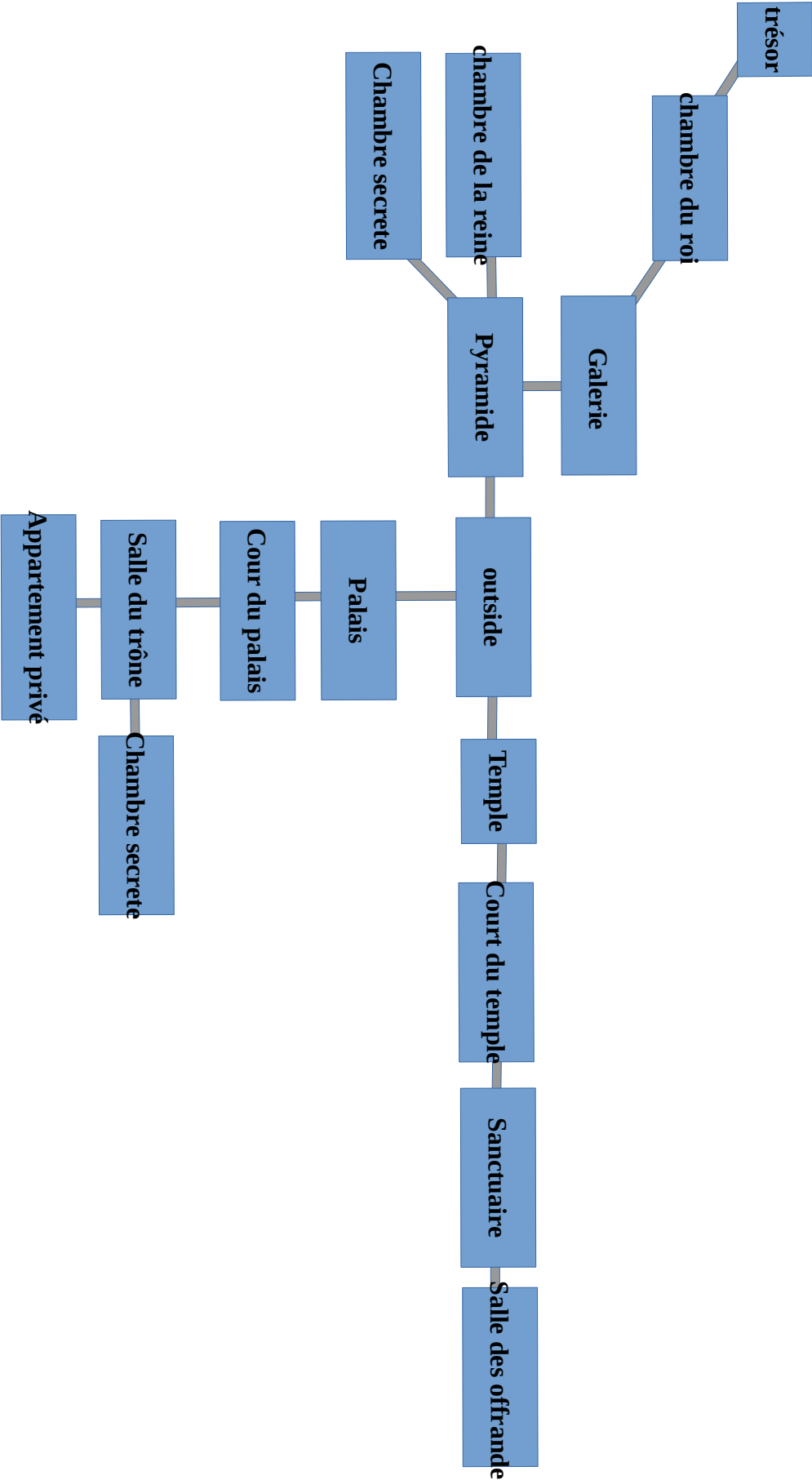
Exercice 7.2.1 :

Fait.

Exercice 7.3 :

Pour l'instant on a imaginé un personnage qui se balade dans une cité égyptienne ancienne, scénario à développer prochainement...

Exercice 7.3.2 :
le plan du scénario :



Exercice 7.4 .. Exercice 7.17.2 :

les modifications ont été apporté directement au code.