Rapport de fin de Projet

Pour la modélisation, nous avons réalisés diagramme de classe, diagrammes de cas d'utilisation et un diagramme de communication.

Pour la partie code toutes les classes sont détaillées ainsi que nos Facilités et Difficultés / Problèmes rencontrés

I - Classe Personnage

En ce qui concerne les classes nous avons décidé de faire une classe Personnage qui gère tout les attributs de Personnage. Nous avons fait plusieurs méthodes comme les méthodes d'actions qui influe sur les attribut du personnage à chaque action (lorsqu'il se déplace ou non).
La fonction de gameOver() est aussi dans la classe Personnage ce qui paraissait le plus logique car la simulation s'arrête quand le personnage à un de ses attributs à 0.

II - Classe Batiment :

Pour la classe batiment, nous avons décider de faire une superClasse Batiment abstraite. Et de faire hérité Bar, Université, Maison, Fast-Food, Bibliothèque de celle-ci. Cela nous permet donc de redéfinir la fonction ressourcer() de Bâtiment pour chaque bâtiment en particulier.

Pour les obtentions de bonus de chance d'avoir son examen, nous avons fait un Random de 0 à 100 ce qui nous permet de détecter un pourcentage de chance d'avoir son examen.

III - Classe MoyenTransport :

La classe Moyen de Transport possède les différents moyen de déplacement ainsi qu'une méthode seDeplacer() qui gère la modification des attributs de Personnage à chaque déplacement.

IV - Classe Piege :

La classe Piège fonctionne de la même manière que Bâtiment, une classe abstraite Piège et deux classes PiegeRoute et PiegeTrottoir qui permette de redéfinir la méthode modifPiege() qui gèrent la modification des attributs du Personnage.

V - Classe Plateau / Case

Pour le plateau de jeux, nous avons fait un double tableau [] [] de Case.
Dans la classe case on créer une instance de chaque Bâtiment ainsi que de Route et de Trottoir.
Dans le plateau, on génère aléatoirement un plateau d'une taille prédéfinie de Case de bâtiments.

Sur le plateau il y a 1 bâtiment de chaque type entouré de Trottoir ou de Route

La classe Plateau gèrent les déplacement du personnage en récupérant sa position X et Y, à chaque fois que l'utilisateur saisie une touche de déplacement (Z Q S D) la position X/Y du Personnage change.
La classe Plateau détecte quand un Personnage se situe sur un Case de type bâtiment et applique aussi la fonction ressourcer() du bâtiment spécifique sur lequel il se trouve

Les facilités :

Ce qui était le plus « simple » à été la gestion des attributs du Personnage, et les modifications de ses attributs en fonction de ses déplacements, le bâtiment dans lequel il se trouve, etc …
De plus, la gestion des bâtiments n'était pas forcement un problème, le fait de les faire hérité d'une classe abstraite nous à permis d'avoir qu'une seule méthode que l'on redéfinissait à chaque fois , ce qui empêche des IF / ELSE disproportionné.

Les difficultés / Problèmes rencontrés :

La gestion du plateau était le plus complexe, en effet la génération aléatoire des batiments sur notre Map était difficile à mettre en place.
De plus la gestion des pièges et l'attribution aléatoire des pièges sur les Case Route ou Trottoir a posé un problème.
Faire en sorte que quand le personnage se déplace sur la map la case change pour faire en sorte que l'utilisateur voit ou se situe le personnage .
Interface graphique compliqué à faire, donc la map s'affiche dans le terminal,je pense qu'on s'y ai pris trop tard pour faire une interface graphique « propre » et que ça demander trop de modification dans le code déjà écrit.

Diagrammes assez compliqués à faire, assez complexe, je pense qu'il y a pas mal d'erreur sur les diagrammes surtout le use Case et le diagramme de Communication