**Rapport de Projet JAVA**

Hermel Jean-Alexis et GHYS Robin  
  
Sujet : Western

Introduction :  
  
 Nous avons choisi de prendre le sujet sur le Western car nous avons trouvé que c’était un sujet très ouvert et possiblement plus créatif que les autres, mais aussi parce que nous avions un petite base dans le cours pour nous lancer et voir ce qu’il était possible de faire grâce au langage JAVA.

Pour débuter nous avons d’abord, avant de coder, mit en place ce que nous voulions qu’il y ait dans notre projet sur papier. Le but étant de savoir où l’on va lorsque qu’on commence à écrire du code et de ne pas avancer à l’aveugle. Ainsi nous avons réalisé un premier diagramme de classes (disponible en PDF), peaufiné au fil du temps, qui nous a été très utile pour nous repérer à travers nos différentes classes. Le Western que nous avons créé reflète l’histoire classique qui se déroule dans les films de western, avec des personnages de type cowboy, brigand, banquier, barman, indien… Ainsi il est possible de générer quelques histoires en fonction des différents choix qui s’offrent à l’utilisateur comme nous allons l’expliquer.

Sujet et Description :

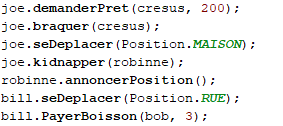
Le but du projet est de créer un environnement de classes dans lequel il va être possible de générer des histoires selon les décisions prises par l’appel de méthodes de ces différentes classes.  
Notre projet s’articule autour d’un genre de village Western, où des lieux vont être créés (Classe Enumération), sur cette base de lieux, nous allons construire des personnages de différents types. Certains auront un lieu attitré, d’autre non. Un personnage verra sa classe héritée d’une ou plusieurs autre classe, en fonction de s’il possède des propriétés communes avec d’autres personnages. Exemple avec le personnage indien qui hérite de la classe Homme (arme, tirer, force…) qui elle-même hérite de la classe Humain (nom, prénom, Position, âge...).

Nous avons fait le choix de nous restreindre dans le nombre et les différences entre les personnages, ce malgré nos nombreuses idées, car sinon le temps nécessaire au projet n’aurait pas suffi pour avoir un programme avec un fonctionnement correct. Finalement nous avons 7 types de personnages pouvant interagir dans 7 lieux différents.

Les personnages pourront être : Banquier, Cowboy, Barman, Indien, Dame, Brigand, Sherif.

Les lieux sont : Le campement, la rue, la prison, la maison, le désert, le bar, la banque.

Un personnage va être décomposé en plusieurs classes (héritées), dont certaines abstraites. Comme dit précédemment les caractéristiques partagées par différents personnages sont réunies dans des classes mère (cf. exemple humain>homme>civil…) afin de factoriser le code. Le barman et le banquier partagent une interface commune, la classe du barman est composée avec une classe Boisson pour pouvoir gérer et entretenir un stock. Enfin les lieux sont regroupés dans une classe Enumération pour y accéder facilement dans le code.

Le principe de notre projet va être d’instancier des personnages avec leurs caractéristiques propres (cf. diagramme) dans le *main*, pour ensuite pouvoir leur faire effectuer des actions et interagir entre eux. Ceci va avoir la forme ci-contre, pour connaître les options possible pour chaque personnage créé, une méthode *help(personnage)* renvoie chaque action disponible.

Notre programme est fait de sorte que l’utilisateur ne puisse pas faire ce qu’il veut avec toutes les méthodes, afin de garder un aspect cohérent et logique à la génération de l’histoire. Par exemple si l’utilisateur souhaite commander une boisson, il devra être vivant (donc avoir au minima gagné un précédent combat), et se trouver au bar. Pour cela il devra s’y déplacer et discuter avec le barman en particulier. Nous avons voulu impliquer l’utilisateur dans les choix s’offrant à lui plutôt que simplement appeler une méthode « d’affichage d’information » lorsqu’il effectue une action avec un personnage.

Ainsi, nous avons dans quelques classes et méthodes introduit le hasard, qui fait varier et augmente considérablement les issues des histoires possibles. Il est envisageable que si, par exemple, le brigand décide de combattre le cowboy, celui-ci réplique et tue le brigand. Et ce à la place d’un méthode qui tuerait atomiquement le personnage qui se fait tirer dessus. Notre programme est donc plus vivant et permet une plus grand durée d’utilisation.

Parmi les différentes méthodes et classes disponibles, certaines sont au cœur du projet et ce sont celles-ci qui vont augmenter les interactions et scénarios possibles :

- La fonction *tirer (Homme)*, agit avec un système de force (0-10) des personnages de type Homme, l’issue du *combat (Homme1, Homme2)* va être décidée par le hasard, pondérée par la force de chaque personnage. L’utilisateur va pouvoir découvrir qu’il ne peut pas faire ce qu’il veut et doit respecter des conditions pour s’il veut faire combattre des personnages (ex. le cowboy est gentil et ne tire donc que sur les méchants).

- La fonction *négocier (Homme)*, est en lien avec la fonction *tirer* car elle va avoir des effets principalement sur le Cowboy et le Brigand, pouvant négocier avec un l’Indien pour augmenter un peu leur force. Des variantes existent aussi si le cowboy négocie avec le brigand pour en devenir un, ou si le brigand essaie avec le shérif il va en prison.

- Le Barman, qui va avoir un vrai stock de boissons avec quantité et prix va pouvoir en proposer et servir les clients qui lui en demandent via *demanderBoisson (Civil, Boisson).* C’est grâce au système d’argent que nous avons mit en place pour les personnages de type Civil, nous allons le détailler par la suite.

- Un système de « crimes » que va pouvoir commettre le brigand qui vont pouvoir être déjoués/contrés par le cowboy ou le shérif, comme exemple, la Dame va se faire enlever, le cowboy va pouvoir la libérer et le shérif emprisonner le brigand.