

# Synthese du projet MonkeyIsland

## Avancement :

### Incrément 1 :

- Déplacement des singes erratiques
  - Déplacement vers terre/mer/hors carte OK
  - Déplacement vers singe/pirate OK
  - Equiprobabilité de déplacement OK
  - Non répétitivité KO (Non testé)
- Gestion de pirate
  - Déplacement vers terre/mer/hors carte OK
  - Déplacement vers singe/pirate OK
  - Déplacement vers Tresor OK
  - Gestion de la mort OK
- Gestion du trésor
  - Création OK
  - Victoire OK
- Gestion du fichier de configuration
  - Création de la carte OK
  - Création de singe OK (/!\ ils sont positionnés aléatoirement /!\)
  - Création de Rhum OK (/!\ ils sont positionnés aléatoirement /!\)
  - Création du trésor OK

### Incrément 2 :

- Communication client-server KO (En cours de conception et d'implémentation)
- Gestion de l'énergie de chaque pirates OK
- Gestion des bouteilles de rhum OK (Pas de gestion de l'attente avant réapparition)
- Gestion des singes chasseurs OK

### Incrément 3 :

- Gestion des parties KO
- Gestion de l'alcoolémie du pirate KO
- Gestion de changement du comportements des singes OK

Afin de clore ce projet a un état convenable, il faudrait finir, dans un premier temps, la lecture du fichier de configuration (créer les singe et rhums ou l'on veut et non de manière aléatoire), cela permettrait, dès la mise en place de la communication, de pouvoir effectuer des tests de validation et le tests JMeter. Cela pourrait être fait rapidement, au vu de ce que j'ai compris au protocole de communication. Par la suite, je souhaiterai avoir une plus grande couverture de test sur différentes classes du projet, car étant seul, je n'ai pas pu m'assurer du fonctionnement normal et limite de toutes les classes comme je le souhaiterais. Un refactoring du code pourrait aussi permettre d'alléger certaines classes ou de simplifier certaines méthodes, ObserverPirate.java pourrait très certainement être simplifiée.

Si je pouvais refaire ce projet, je souhaiterais essayer de le faire en écrivant les tests au préalable, comme dans la méthode extreme programming afin de pouvoir me rendre compte des inconvénients et avantages de cette méthode. Même si j'ai déjà fait ce projet, le manque de temps (arrivé en cours qu'en octobre) et des problèmes concernant les tests m'ont considérablement ralenti m'empêchant de pouvoir essayer la méthode extreme programming, ne serait-ce que pour certaines classes.

Je pense que des revues de code m'aurait été utile, même si je sais que cela peut être difficile à mettre en place au sein d'un projet à l'école. En revanche, encore dans le cas où ce projet aurait été réalisé à plusieurs, croiser l'implémentation et les tests, faire en sorte de ne jamais tester le code que on a implémenter, m'aurait permis de gagner beaucoup de temps sur des erreurs simples. Je pense que ces deux aspects du travail en équipe, ainsi que le savoir faire d'un élève de SE, qui m'ont le plus fait défaut durant ces trois mois de projet.

