

---

## Module « COOL, QL »

Equipe DDMP

---

### Synthèse Projet

*A l'attention des enseignants du module « COOL, QL ».*

---

*Ce document est une synthèse de notre projet « Monkey Island ». Il est à destination des professeurs en charge du module.*

---

Auteurs	Contact
François de Broch d'Hotelans	francois.debrochdhotelans@reseau.eseo.fr
Cailyn Davies	cailyn.davies@reseau.eseo.fr
Guillaume Muret	guillaume.muret@reseau.eseo.fr
Clément Pabst	clement.pabst@reseau.eseo.fr

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Objectif . . . . .	3
1.2	Portée . . . . .	3
<b>2</b>	<b>Réalisations et Validation</b>	<b>4</b>
2.1	Premier incrément . . . . .	4
2.2	Deuxième incrément . . . . .	4
2.3	Troisième incrément . . . . .	4
2.4	Fonctions non réalisées . . . . .	5
<b>3</b>	<b>Retours &amp; Critiques // Rapports d'étonnement</b>	<b>6</b>
3.1	François de Broch d'Hotelans . . . . .	6
3.2	Cailyn Davies . . . . .	6
3.3	Guillaume Muret . . . . .	6
3.4	Clément Pabst . . . . .	6

## 1 Introduction

### 1.1 Objectif

L'objectif de ce document est, dans un premier temps, de présenter l'état du projet Monkey Island à la date de livraison. Dans un second temps, il permettra à chaque membre de l'équipe de faire un retour individuel sur le projet.

### 1.2 Portée

Ce document est à destination des professeurs en charge de l'évaluation du module « COOL, QL » de l'année 2017-2018. Il n'a pour vocation qu'une exploitation pédagogique, interne à l'ESEO.

**La partie "Retours & Critiques // Rapports d'étonnement"** est pour l'usage EXCLUSIF des professeurs en charge du projet « COOL, QL » de l'année 2017-2018 et est fourni avec pour seule fin une éventuelle amélioration du projet pour les promotions suivantes.

## 2 Réalisations et Validation

Lors de ce projet, il nous était demandé de concevoir, développer, tester un programme informatique basé sur le jeu Monkey Island.

Ce document présente les fonctionnalités qui ont été développées, testées et validées selon un certain nombre d'incrément.

### 2.1 Premier incrément

- Déplacement des singes erratiques  
Les singes erratiques se déplacent aléatoirement avec une probabilité de 25 % d'aller dans une des quatre directions haut, bas, droite ou gauche.
- Gestion des pirates  
Les pirates se déplacent sur la carte dans l'objectif de trouver le trésor avant de se faire manger par l'un des différents singes du jeu, le tout sans arriver à cours d'énergie.
- Gestion du trésor  
La présence du trésor sur la carte du jeu. Découvrir ce trésor permet à un pirate de mettre fin à la partie, et d'en sortir victorieux.
- Gestion du fichier de configuration  
Un fichier .json permettant de configurer les différents paramètres et données du jeu.

### 2.2 Deuxième incrément

- Communication Client – Serveur  
La communication entre le serveur du jeu et les différents clients (joueurs) est faite. Ainsi les joueurs peuvent jouer en temps réel sur la même carte de jeu.
- Gestion de l'énergie de chaque pirate  
L'énergie de chaque pirate, fixée à 20 (fichier de configuration), est fonctionnelle. Si un pirate n'a plus d'énergie, il meurt. De plus, s'il dépasse le seuil de son énergie maximum, il devient saoul.
- Gestion des bouteilles de rhum  
Les bouteilles de rhum peuvent être :
  - soit visible, alors les pirates peuvent les boire et ainsi augmenter leur énergie
  - soit invisible, les pirates ne peuvent pas les voir ni les boire.Un timer est mis en place afin de faire ré-apparaître les bouteilles de rhum.

### 2.3 Troisième incrément

- Gestion des parties  
La découverte du trésor par un pirate signifie la fin de la partie en cours.

- Gestion des singes chasseurs

Les singes chasseurs ont des déplacements « intelligent », c'est-à-dire qu'ils se dirigent vers le pirate le plus proche de leur position. Des tests de modèle de déplacement systématique ont été validés afin d'éviter toute faille dans la gestion des déplacements des singes chasseurs.

- Gestion de l'alcoolémie du pirate

L'alcoolémie du pirate est fonctionnelle. Si un pirate dépasse le seuil de son énergie maximum (20) , il devient saoul et se déplace comme un singe erratique avec une probabilité de mouvement de 25 % dans chaque direction (haut, bas, droite et gauche), quelle que soit la direction demandée par le joueur.

L'utilisation des différents outils à notre disposition (Checkstyle, Findbugs, Subversion) nous a permis d'avoir un code de qualité en préconisant une performance maximale.

## 2.4 Fonctions non réalisées

Le projet, ainsi que les trois incréments ont été testés et validés. Nous n'avons donc actuellement aucune fonctionnalité manquante.

## **3 Retours & Critiques // Rapports d'étonnement**

### **3.1 François de Broch d'Hotelans**

Le projet Monkey Island a été un bon projet mêlant un ensemble de d'aspects variés comme la conception, la programmation, la réalisation de tests et enfin une qualité du code à respecter en fonction des métriques définies par l'équipe en début de projet. Le fait de mélanger les options SE et LD est une bonne chose dans ce projet. Cela permet de partager les différentes connaissances acquises d'étudiants d'options différentes et ainsi de pouvoir monter en compétences.

Un bémol dans ce projet est l'apprentissage des différents design pattern qui nous ont été présentés alors que le projet avait déjà commencé. De plus, le support permettant de les programmer est pour ma part trop mince. J'aurais aimé un peu plus d'accompagnement dans les démarches à suivre pour réaliser chaque design pattern.

### **3.2 Cailyn Davies**

Dans l'ensemble le cours est cohérent : Les design patterns sont un nouveau concept pour la majorité d'entre nous et intéressant à apprendre, le projet nous permet d'étudier spec, conception, réalisation et tests avec une complexité variable et s'adapte donc bien aux différents étudiants.

Le bémol de ce cours fût son organisation, les derniers cours de design pattern ont eu lieu bien trop tard : Nos conceptions étaient trop avancées pour que ça soit valable de tout changer pour implémenter les design patterns qui nous étaient présentés. Je proposerais de scinder les deux parties (projet et design pattern) en deux parties bien séparées. En commençant par les cours de design pattern, puis en nous présentant le projet. Cela permettrait aux étudiants de connaître les outils avant le projet (ou au pire très tôt dans le projet), au lieu de les apprendre à mi-chemin quand des solutions moins viables sont déjà implémentées.

### **3.3 Guillaume Muret**

### **3.4 Clément Pabst**