*Institut National des Sciences Appliquées de Rennes*

*Département Informatique*

*4eme année*

**Projet Canon Noir**

**Rapport de conception**



*Maxime HAVEZ*

*Gareth THIVEUX*

*4INFO – G2.2 2010-2011*

# Sommaire

[Sommaire 2](#_Toc278716808)

[A. Introduction 3](#_Toc278716809)

[B. Diagrammes comportementaux 4](#_Toc278716810)

[Cas d’utilisations 4](#_Toc278716811)

[Diagrammes d’activités 4](#_Toc278716812)

[Diagrammes d’états-transitions 5](#_Toc278716813)

[C. Diagrammes structurels 5](#_Toc278716814)

[Diagramme de classes 5](#_Toc278716815)

[Diagrammes de composants et de déploiement 5](#_Toc278716816)

[D. Diagrammes dynamiques 6](#_Toc278716817)

[Diagrammes d’interactions 6](#_Toc278716818)

[E. Conclusion 7](#_Toc278716819)

1. Introduction

Ce rapport a pour objectif de détailler notre analyse conceptuelle pour notre projet de réalisation d’un logiciel. Le logiciel en question correspond à l’adaptation d’un jeu de société : le jeu du Canon Noir.

Ce jeu de société, constitué d’un plateau, d’îles, de trésors, d’un canon (noir) et de bateaux, peut accueillir de deux à quatre joueurs. Le vainqueur est le joueur qui ramènera le premier trois trésors à son port. Bien entendu, les trésors doivent être ramassés sur des cases spécifiques, et pour pimenter le jeu, il existe tout un mécanisme de tirs et duels au canon afin de tenter de couler ou ralentir la progression de son/ses adversaire(s).

La réalisation et la réussite de ce projet passe avant tout par la mise en œuvre des connaissances acquises en cours de Programmation et Modélisation Orientées Objet et de Design Patterns.

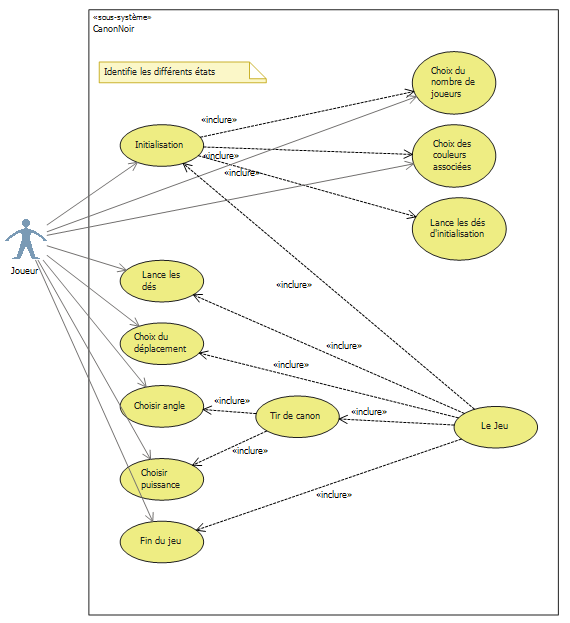
Nous allons dans un premier temps, et notamment par le biais de ce rapport de conception, aborder et réutiliser les outils UML qui nous ont été présentés lors de notre cursus. Ceux-ci nous ont permis de retranscrire les fonctionnalités de notre application ainsi que sa mise en œuvre et son fonctionnement interne. Ce modèle conceptuel est donc une représentation abstraite de notre système, et il permettra par la suite de faciliter l’étude ainsi que notre communication.

Nous détaillerons notre démarche, qui évidemment part d’une analyse globale du logiciel, pour qu’ensuite nous puissions approfondir les différentes étapes. Nous commencerons donc par vous présenter les diagrammes comportementaux, qui nous ont aidé à concevoir les diagrammes représentant la structure statique et nous vous donnerons enfin les diagrammes dynamiques.

1. Diagrammes comportementaux

## Cas d’utilisations

Simples et efficaces, les cas d’utilisation doivent nous permettre de retranscrire les principales fonctionnalités (pour l’utilisateur) de notre logiciel, mais également de montrer ses limites.



## Diagrammes d’activités

Le diagramme d’activité, comme son nom l’indique, se concentre sur l’activité d’un processus.

## Diagrammes d’états-transitions

Les objets interagissent pour implémenter des comportements qui peuvent être décrit par des diagrammes d’interactions ou des diagrammes d’états-transitions qui sont pour leur part d’avantage centrés sur l’objet. Le diagramme se concentre donc sur 1 unique objet. (représenter les états possibles pour les objets de la classe)

1. Diagrammes structurels

## Diagramme de classes

Ce sont les diagrammes les plus connus et les plus couramment utilisés. Ils retranscrivent la structure statique de notre application.

## Diagrammes de composants et de déploiement

1. Diagrammes dynamiques

## Diagrammes d’interactions

Ils illustrent les réalisations des cas d’utilisations, en décrivant un exemple de scénario. Il existe plusieurs types de diagramme d’interactions mais nous n’étudierons uniquement des diagrammes de séquences. Ils correspondent à l’échange de messages entre les différents objets de notre diagramme de classe. (1 diagramme = 1 CU).

1. Conclusion