

# Spécification des exigences

## Plan :

1. Introduction
2. Description générale
3. Exigences fonctionnelles
4. Exigences non fonctionnelles
5. Annexes

## 1. Introduction

---

### Objectif du document:

Ce document vise à décrire les exigences fonctionnelles et non fonctionnelles pour le développement du jeu de pirate, un jeu de société basé sur le jeu de l'oie à deux joueurs.

---

### Règles :

Le jeu consiste en un plateau de 30 cases, avec des pions représentant des pirates qui progressent en fonction des résultats de deux dés. Le jeu se termine quand un joueur atteint la dernière case ou que l'adversaire perd tous ses points de vie.

## 2. Description générale

---

### Contexte:

Le jeu est conçu pour être joué sur ordinateur, offrant une interface utilisateur graphique avec des animations pour les dés et des indicateurs visuels pour les points de vie et la progression sur le plateau.

---

### Fonctionnalités de l'utilisateur:

Les joueurs peuvent lancer des dés, déplacer des pions, et suivre visuellement leurs points de vie et leur position sur le plateau.

---

### Contraintes de conception:

Le logiciel doit suivre l'architecture ECB.

Les interactions doivent être claires et intuitives, avec des animations fluides pour le lancer des dés et le mouvement des pions.

### 3. Exigences fonctionnelles

---

#### Déplacement des pirates:

- EX1 : Les pions représentant les pirates doivent être de couleurs différentes.
  - EX2 : Les jetons doivent être débloqués uniquement quand c'est au tour du pirate concerné de jouer.
  - EX3 : Le bouton lancer les dés doit permettre de lancer les deux dés.
  - EX4 : Le nombre de cases à avancer doit être déterminé par le lancé des deux dés.
  - EX5 : Les déplacements doivent être contraints par les résultats des lancer de dés.
  - EX6 : Les déplacements doivent se faire manuellement par le joueur.
  - EX7 : Si le joueur pose le pion sur la mauvaise case, le tour est bloqué tant que le pion n'est pas bien placé.
  - EX8 : Si le joueur pose le pion sur la bonne case, le déplacement doit être effectif.
  - EX9 : La case doit signifier que le déplacement du pirate a bien été pris en compte.
  - EX10 : Le joueur doit reculer si il arrive à la fin et que il lui reste des déplacements.
- 

#### Gestion des points de vie:

- EX11 : Chaque pirate commence avec 5 points de vie.
  - EX12 : Le visuel des points de vie doit être compréhensible.
  - EX13 : Les interactions peuvent réduire ou augmenter les points de vie.
- 

#### Fin de jeu :

- EX14 : Le jeu doit se terminer si un des pirates atteint la dernière case.
  - EX15 : Le jeu se termine si un des pirates perd tous ses points de vie.
- 

#### Interface utilisateur:

- EX16 : Afficher des animations pour les dés.
  - EX17 : Valider et mettre à jour l'état du plateau après chaque mouvement.
  - EX18 : Distinction claire des cases de départ et d'arrivée, numérotation des cases.
  - EX19 : Améliorer le visuel des points de vie.
  - EX20 : Clarifier l'attribution des pions aux joueurs.
  - EX21 : Ajouter des éléments visuels pour indiquer quel pion est actif.
- 

#### Cases spéciales :

- EX22 : Le jeu doit proposer des cases spéciales malus :
- EX23 : Falaise glissante : La case doit faire retourner le joueur à son dernier coup avec 2 pdv en moins.
- EX24 :Avalanche de pierres : La case doit bloquer le joueur pendant un tour.
- EX25 : Pris dans les lianes : La case donne trois essais au joueur pour faire un double (les mêmes chiffres sur les deux dés) et bloque le joueur tant qu'il ne réussit pas.
- EX26 : Retour à la case départ : La case renvoie le joueur au début avec tout ses pdv.
- EX27 : Boue : La case ralentira le joueur de -3 sur le prochain coup.

- EX28 : Le jeu doit proposer des cases spéciales bonus :
- EX29 : Trouver de la nourriture : La case donne au joueur un pdv.
- EX30 : Passage secret : Renvoie le joueur à la case qui suit le passage secret.
  
- EX31 : Le jeu doit proposer des cases spéciales bonus/malus :
- EX32 : Dragon de komodo : Bonus : La case doit faire rejouer le joueur.
- EX33 : Dragon de komodo : Malus : Si le joueur fait moins de 8, il perd trois points de vie et à un changement de -3 pour son prochain tour.

## 4. Exigences non fonctionnelles

---

### Performance:

Le jeu doit charger et répondre aux interactions des utilisateurs sans délai perceptible.

---

### Sécurité:

Les données du jeu comme les scores et l'état des parties doivent être protégées contre des modifications non autorisées.

---

### Extensibilité | Modularité | Adaptabilité

Le code doit être structuré de manière à faciliter l'ajout de nouvelles fonctionnalités, telles que de nouvelles cases spéciales.

---

### Contrainte ECB :

Le code doit suivre l'architecture ECB

## 5. Annexes

---

### Glossaire:

- Pdv : point de vie

---

### Références:

- Slides de la présentation du projet.
- Cours de Java suivis durant les 4 derniers modules d'ILU.
- Javadoc