**CAHIER DES CHARGES**

Version : 2

Date : 25/09/2024

| Client | Prestataire |
| --- | --- |
| Toute personne disposant du projet. | Nom 1 : BOUNIOT Yanis  Nom 2 : WONG Marcel  Nom 3 : HERMIER Maxime  Nom 4 : DRÉAN Maxime |
| Cahier des charges approuvé dans sa version le   ... /… / ........ par | |

| zone réservée |
| --- |
|  |

# Introduction

## Objet du document

Ce document décrit **tous** les services que doivent rendre le produit et ses livrables et toutes les exigences qu'ils doivent satisfaire.

## Portée du document

Ce document est destiné à formaliser le besoin du client Mr Tchenegnon dans le cadre du projet Mille Bornes.

## Terminologie

| **Terme** | **Description** |
| --- | --- |
| Produit | Terme générique désignant l’objet de la demande du client. Il recouvre aussi bien un système qu’un service, sans préjuger de la part de logiciel et de matériel intervenant dans la réalisation. |
| JVM (Java Virtual Machine) | Environnement d'exécution qui permet d'exécuter des programmes Java. |
| Git | Système de contrôle de version décentralisé utilisé pour suivre les modifications apportées aux fichiers, principalement dans le cadre du développement de logiciels. |
| Open source | Modèle de développement de logiciels où le code source est rendu accessible au public. |
| CPU | Entité contrôlée par l'ordinateur qui remplace un joueur humain, capable de prendre des décisions et d'agir en fonction de la logique de jeu. |
| Borne | Représente une distance parcourue mais aussi le score du joueur. |
| Cartes Attaques | Vise à saboter l’adversaire en l’immobilisant ou en le ralentissant. |
| Cartes Parades | Vise à contrer les attaques, permettant au jour de continuer à jouer. |
| Cartes Bottes | Permet de se prémunir contre un type d’attaque. |
| Cartes étapes | Permet au joueur d’avancer de (25, 50, 75, 100, 200) bornes |
| Test unitaire | Pratique visant à valider le comportement d'unités individuelles de code, souvent des fonctions ou des méthodes, pour s'assurer qu'elles fonctionnent comme prévu. |
| Test d’intégration | Étape visant à vérifier l'interaction entre différents modules ou composants d'une application. |

## Abréviations

| **Abréviations** | **Signification** | **Libellé** |
| --- | --- | --- |
| *VIT* | *Vitale* | *Exigences fonctionnelles ou non fonctionnelles indispensables* |
| *IMP* | *Importante* | *Exigences souhaitées mais non exigées* |
| *MIN* | *Mineure* | *Exigences non exigées immédiatement, mais qui devront être prises en compte ultérieurement par le produit (impact sur l'évolutivité)* |
| *CPU* | *Central Processing Unit* | *Se référer à “1.3 Terminologie – CPU".* |

# Les objectifs du produit

## Définition du produit

Le projet consiste à développer une version numérique du jeu de société **Mille Bornes** en Java. Ce jeu offrira une expérience ludique immersive avec une interface graphique intuitive. Le joueur pourra se confronter à un ou deux robots (appelés CPU).

L'application comprendra une documentation détaillée, incluant un manuel utilisateur et des instructions techniques. Le projet est à but éducatif, mais pourra être partagé avec un public plus large (open source).

## Contexte économique du produit

Le produit se positionne sur le marché des jeux de société numériques. La clientèle cible comprend les amateurs de jeux de société et les familles principalement.

Le produit se concentrera initialement sur la France, mais une extension vers d'autres pays, notamment ceux avec des communautés francophones, est envisageable.

## Contexte d’exploitation du produit

Le développement du projet impliquera plusieurs acteurs. L'équipe de développement sera responsable de la création du code, de l'interface graphique et des tests. Les membres de l'équipe joueront également le rôle d'utilisateurs, tout en ayant la possibilité d'intégrer d'autres utilisateurs pour recueillir des retours sur l'expérience de jeu.

Le produit se mesurera à d'autres jeux de société numériques bien établis, tels que ***Papyrus 2000***et ***Androsace 2005***, qui offrent également des expériences ludiques similaires. L'analyse de ces concurrents permettra de mieux positionner **Mille Bornes** sur le marché.

# Exigences sur le produit

## Capacités Fonctionnelles

### Description des fonctionnalités

Le projet comprendra plusieurs fonctionnalités clés. L'application proposera une interface graphique intuitive, permettant à un joueur humain de s'affronter contre un ou deux robots. L'application inclura une gestion des scores (Bornes), qui sera enregistrée dans une base de données pour assurer un suivi des performances des joueurs.

Un récapitulatif des parties précédentes sera disponible, offrant une vue détaillée du déroulement des jeux, tour par tour. Cette fonctionnalité permettra aux joueurs d'analyser leurs stratégies et d'améliorer leur jeu au fil du temps.

### Interopérabilité

Le projet sera conçu pour fonctionner uniquement en local, avec un seul joueur s'affrontant contre un à deux robots. La base de données utilisée pour la gestion des scores et des récapitulatifs des parties sera également locale, garantissant ainsi des performances optimales et une expérience fluide.

### Conformité réglementaire

Étant donné qu'il s'agit d'un jeu conçu pour un usage local et éducatif, il n'y a pas de réglementations strictes liées à la distribution en ligne ou à la protection des données des utilisateurs.

Il sera essentiel de respecter les droits d'auteur et les licences associés au jeu de société original, ainsi que de s'assurer que toutes les ressources utilisées dans le développement du produit soient légalement acquises ou créées.

## Exigences non fonctionnelles

### Fiabilité

Des tests approfondis seront menés tout au long du développement, incluant des tests unitaires, des tests d'intégration et des tests de performance. Cela permettra d'identifier et de corriger les bugs. L'application sera également conçue pour être robuste face aux entrées inattendues, minimisant les risques de plantage ou de dysfonctionnement.

### Sécurité

Le projet sera développé en suivant les meilleures pratiques de programmation pour minimiser les vulnérabilités potentielles. Cela inclut la validation des entrées utilisateur afin de prévenir les erreurs ou comportements imprévus.

### Facilité d’utilisation

L’application présentera une interface graphique unique et intuitive, permettant aux utilisateurs de se plonger directement dans le jeu sans avoir à naviguer à travers des menus complexes.

Pour aider les nouveaux joueurs, un manuel d'utilisation détaillé sera fourni, expliquant les règles du jeu et les mécanismes de manière claire. Ce manuel servira de référence pour guider les utilisateurs à travers les différentes étapes de la partie.

### Rendement

L'application sera développée pour fonctionner efficacement sur des systèmes locaux, minimisant le temps de chargement et assurant une fluidité durant les parties.

### Maintenabilité

Le projet sera conçu avec une approche de maintenabilité, facilitant les ajouts et correctifs. Le code sera structuré de manière claire et documentée, ce qui permettra à de futurs développeurs de comprendre facilement les différentes composantes de l’application.

Des conventions de codage cohérentes seront appliquées pour assurer la lisibilité et la modularité du code, rendant ainsi les corrections de bugs plus simples et rapides.

Un système de gestion des versions (Git) sera mis en place pour suivre les modifications apportées au code, facilitant le retour à des versions antérieures si nécessaire.

### Portabilité

Développée en Java, l'application bénéficiera de la portabilité inhérente à ce langage, qui permet une exécution sur toute plateforme disposant d'une machine virtuelle Java (JVM). L'architecture du code sera également optimisée pour minimiser les dépendances spécifiques à un environnement.

## Exigences concernant le développement du produit

### Objectifs de délais

*Année 2024*

* **13/09** : Introduction du Projet et Analyse des besoins
* **20/09** : Poursuite de l'Analyse des besoins et Recette
* **27/09** : Élaboration des Spécifications fonctionnelles et de la Conception architecturale
* **04/10** : Finalisation de la Conception architecturale et Définition des Tests d’intégration
* **11/10** : Conception détaillée
* **18/10** : Poursuite de la Conception détaillée et Définition des Tests unitaires
* **25/10** : Réalisation et Tests unitaires (Première livraison)
* **08/11** : Poursuite de la Réalisation et Tests unitaires
* **15/11** : Poursuite de la Réalisation et Tests unitaires
* **22/11** : Poursuite de la Réalisation et Tests unitaires
* **29/11** : Réalisation et Tests d’intégration
* **06/12** : Poursuite des Tests d’intégration
* **13/12** : Poursuite des Tests d’intégration
* **20/12** : Validation (Recette) et Deuxième livraison

### Objectifs de coûts

Le projet étant de nature éducative, les objectifs de coûts se concentrent sur le temps et les efforts de l'équipe. Il n'y aura pas de coûts supplémentaires engagés, car le projet s'appuie sur les ressources déjà disponibles, notamment les ordinateurs et les logiciels utilisés dans le cadre de la formation.

### Exigences de réalisation

Les exigences de réalisation pour le projet stipulent que l'application sera développée en Java, avec une architecture modulaire et une interface graphique intuitive pour une utilisation facile. Un manuel d'utilisation détaillé sera fourni, et des tests unitaires et d'intégration garantiront la fiabilité du produit. Enfin, le projet respectera les droits d'auteur et les meilleures pratiques de développement.

# Synthèse des Exigences

## Hiérarchisation des exigences fonctionnelles

|  |  |
| --- | --- |
| **Nom fonction** | **Importance** |
| Logique de jeu principale | VIT |
| Gestion des scores | VIT |
| Interaction avec les robots (CPU) | VIT |
| Récapitulatif des parties précédentes | IMP |
| Personnalisation des paramètres (nombre de joueurs, etc.) | IMP |
| Statistiques avancées | MIN |
| Options de personnalisation avancées | MIN |

## Hiérarchisation des exigences non fonctionnelles

|  |  |
| --- | --- |
| **Exigence** | **Importance** |
| Fiabilité | VIT |
| Facilité d’utilisation | VIT |
| Performance | VIT |
| Sécurité | IMP \* |
| Maintenabilité | IMP |
| Portabilité | IMP |
| Documentation | MIN |

\* Passage à VIT si le projet est voué à une autre utilisation que localement.