



---

# Table de Billard Connectée

---

Cahier des spécifications

September 23, 2022

Par :

Thibault Meynier,  
Florian Derlique

Tuteurs :

Xavier Redon,  
Thomas Vantroys,  
Alexandre Boé

## Contents

<b>1</b>	<b>Préambule</b>	<b>3</b>
<b>2</b>	<b>Utilisateur</b>	<b>3</b>
<b>3</b>	<b>Simulation et traitement d'images</b>	<b>3</b>
<b>4</b>	<b>Queue de billard</b>	<b>4</b>
<b>5</b>	<b>Application mobile</b>	<b>4</b>
<b>6</b>	<b>Tâches à effectuer (Diagramme de Gantt)</b>	<b>5</b>

## 1 Préambule

Le projet comportera les parties suivante. Chaque élément sera détaillé dans sa propre partie dans ce document.

- Une simulation de billard,
- Une queue de billard adaptée pour le projet,
- Une application mobile pour suivre en direct la partie de billard,

## 2 Utilisateur

### A la fin du projet, les utilisateurs "joueurs" pourront :

- Jouer une partie de billard sur un PC,
- Jouer une partie sur l'écran tactile intégré à la table,
- Utiliser une queue de billard spécialement créée pour cette application, dont nous détaillerons les spécifications dans la suite de ce document,
- Utiliser un menu avant la partie pour configurer différents réglages;

### Les utilisateurs "spectateurs" pourront :

- Suivre la partie en direct grâce à l'application,
- Définir un pseudonyme,
- Discuter en temps réel sur la partie en direct par cette même plate-forme;

## 3 Simulation et traitement d'images

La simulation est faite intégralement en langage Java, ce langage a été choisi pour différentes raisons :

- Ce langage est orienté objet ce qui est un paradigme plutôt adapté pour faire des jeux comme celui-ci,
- Ce langage est très rapide en termes de temps d'exécution comparé au Python par exemple,
- Nous avons déjà utilisé ce langage dans plusieurs projets, et nous sommes donc assez à l'aise.;

Le choix le plus adapté pour réaliser ce genre de projet aurait été le C++, cependant, l'équipe projet étant plus familière avec le langage Java, c'est celui-ci qui a été retenu.

Nous intégrons les fonctions mathématiques et physiques pour recréer :

- Les collisions entre deux boules,
- Les collisions entre une boule et un mur,
- Les collisions entre la queue et une boule,
- La vitesse et l'accélération d'une boule;

Le jeu est mis à jour (mouvement des boules, vérification de l'état du jeu, etc) à chaque itération de la boucle principale. Chaque itération de celle-ci passe par plusieurs étapes :

- Un calcul du mouvement des objets,
- Une vérification de l'état du jeu (on s'assure qu'un joueur n'a pas gagné),

- L'attente d'un coup par le prochain joueur;

L'utilisation ou non d'une programmation multi-threadé reste à définir.

Pour l'interface graphique, nous utilisons pour l'instant le package Java Swing, ce dernier offre la possibilité de créer facilement une interface graphique, cependant la qualité peut laisser à désirer avec ce framework. Le choix reste à définir si nous restons sur ce framework ou non, il se peut que nous basculions sur le framework JavaFx. Ce framework comporte certaines fonctionnalités qui ne sont pas disponibles avec Swing.

Ainsi il sera possible d'intégrer les animations des boules lorsqu'elles sont en mouvement, ou jouer des sons lorsque deux billes s'entrechoquent.

Pour la partie de traitement d'images, nous allons utiliser le framework OpenCV. Cette bibliothèque permet de faire du traitement d'image, elle a initialement été développée en C/C++ mais est aussi disponible pour Java.

## 4 Queue de billard

Les queues de billard sont généralement fabriquées avec du frêne, un bois confortable offrant de bonnes sensations de jeu. Cependant, pour ce projet, ce facteur n'est pas le plus important. En effet, étant donné que la partie se passe sur un écran et que celui-ci est plutôt petit comparé à une table de billard réelle, il paraît plus sage de réaliser une queue plus petite et dans un matériau qui ne rayera pas l'écran tactile.

Nous avons regardé les bois les plus tendres, le facteur qui décidera notre choix sera le prix, une étude de prix sera nécessaire pour faire notre choix. Nous nous projetons sur le choix du balsa pour notre queue, même si ce choix reste à définir.

Nous ajouterons des pastilles colorées (par exemple une couleur rouge vif) pour que la caméra puisse détecter la queue.

## 5 Application mobile

L'EDI<sup>1</sup> retenu pour développer l'application mobile est Android Studio et le langage retenu est Java. Les deux options les plus utilisées actuellement sont Java et Kotlin (un autre langage pouvant tourner sur la JVM<sup>2</sup> et plus intéressant en termes de fonctionnalités), mais toujours pour des raisons de familiarité avec Java, c'est ce langage qui a été retenu.

Cette application permettra de suivre la partie en se connectant à l'ordinateur de la table en bluetooth.

L'intégration d'une discussion en temps réel est une fonctionnalité qui sera intégrée en fonction de la vitesse d'avancement du projet.

---

<sup>1</sup>Environnement de Développement Intégré

<sup>2</sup>Java Virtual Machine

## 6 Tâches à effectuer (Diagramme de Gantt)

Voici les tâches à effectuer :

- Etude des besoins,
- Simulation de billard : mathématiques et physique liée au billard,
- Simulation de billard : interface graphique,
- Intégration sur écran tactile,
- Fabrication de la queue de billard,
- Traitement d'image,
- Création de l'application mobile : interface utilisateur,
- Création de l'application mobile : partie communication,
- Améliorations et fonctionnalités futures;

