

ÉCOLE POLYTECHNIQUE DE L'UNIVERSITÉ FRANÇOIS RABELAIS DE TOURS  
Spécialité Informatique  
64 av. Jean Portalis  
37200 TOURS, FRANCE  
Tél +33 (0)2 47 36 14 31  
[www.polytech.univ-tours.fr](http://www.polytech.univ-tours.fr)

CAHIER DE SPECIFICATION			
<b>Projet :</b>	Smart-billard		
<b>Emetteur :</b>	Maël Bervet Léo Legrand	<b>MOA : Maël Bervet, Léo Legrand</b>	
<b>Date d'émission :</b>	11/02/2020		
Validation			
Nom	Date	Valide (O/N)	Commentaires
Léo Legrand			
Maël Bervet			
Historique des modifications			
Version	Date	Description de la modification	
01	11/02/2020	Plan du cahier et structure du projet	
02	17/02/2020	Ajout des interfaces	

## TABLE DES MATIERES

Introduction .....	3
1. Contexte du projet .....	3
2. Objectifs du projet .....	3
Description générale .....	3
3. Environnement du projet.....	3
4. Caractéristique des utilisateurs .....	3
5. Bases méthodologiques .....	4
6. Fonctionnalités du système .....	4
7. Structure générale du système .....	7
Description des interfaces du système .....	11
8. Interface matériel/logiciel.....	11
9. Interface homme/machine .....	11
10. Interface logiciel/logiciel .....	11
Spécifications fonctionnelles .....	12
11. Définition de la fonction i .....	12
Spécifications non fonctionnelles .....	12
12. Contraintes de développement et conception.....	12
13. Contraintes de fonctionnement et d'exploitation.....	13
14. Maintenance et évolution du système .....	14

## INTRODUCTION

### 1. Contexte du projet

Ce projet se situe dans le cadre d'un projet libre d'étudiant à l'institut polytechnique de l'université de Tours. Par conséquent la MOA ainsi que la MOE sont les participants à ce projet, soit Maël Bervet et Léo Legrand. Toute fois ce projet est encadré par Pascal Makris, enseignant chercheur à l'école polytechnique de Tours.

### 2. Objectifs du projet

L'objectif de ce projet est de mettre en place un pack d'évolution pour billard, de type anglais ou américain, afin de le transformer en smart-billard connecté.

Le principe, basique, du smart-billard est d'ajouter à un billard une aide à la visée en projetant la trajectoire prise par une boule sur le billard en temps réel et/ou sur un smartphone. Cette trajectoire est calculée grâce à la position des différentes boules et le positionnement de la queue. Cette partie est constitué d'une caméra et d'un Raspberry PI 3 afin de gérer les calculs et la transmission du flux vidéo.

Ce smart-billard sera connecté car la Raspberry communiquera en wifi avec un smartphone afin de pouvoir afficher le flux vidéo de la caméra mais également afin de pouvoir gérer la partie. L'ajout du smartphone permettra au système de proposer des options supplémentaires tel que la proposition d'une trajectoire afin de rentrer la boule choisie ou encore de proposer une aide à la décision de coup intéressant effectuer.

## DESCRIPTION GENERALE

### 3. Environnement du projet

Pour ce projet l'environnement définit est un billard de type 8 pool, américain ou anglais, supposé en bon état, auquel on pourra ajouter le pack afin d'en faire un smart-billard.

Ce pack est donc constitué :

- D'une caméra afin de filmer le billard en continue lors d'une partie
- D'un Raspberry Pi 3 afin de récupérer, d'analyser et de transmettre le flux vidéo
- D'une caméra afin de projeter la trajectoire calculée sur le billard
- D'un smartphone sous Android afin de gérer la partie et d'intégrer les aides

### 4. Caractéristique des utilisateurs

L'utilisateur sera un joueur sur le billard équipé du pack d'évolution. L'utilisateur n'a pas à avoir de compétences quelconques en informatique ou en tant que joueur de billard. Les prérequis sont d'avoir un téléphone Android avec l'application installée et de savoir utiliser le système Android. L'application devra être suffisamment intuitive afin de permettre à l'utilisateur de l'utiliser sans accompagnement ou aide quelconque.

## 5. Bases méthodologiques

Le projet sera géré via la méthode agile avec l'aide d'un « board » Trello définissant les tâches à effectuer, les ressources associées ainsi que les date due de chaque tâche. La documentation du projet sera gérée via un google drive et la communication entre les participants via un serveur discord. La communication avec l'encadrant se fera par email sur l'avancement du projet lorsqu'une avancé sera faite.

La modélisation du projet se fera à l'aide de diagramme UML fait sous google draw.io. Le projet, en pratique, se fera avec un Raspberry PI 3 en utilisant le langage de programmation C++ pour les différentes API à implémenter mais aussi avec un téléphone Android sur lequel sera développé une application à l'aide d'Android studio en utilisant le langage de programmation Java. Toute la programmation se fera donc en différents composants, soit C++ soit Java, dont le code suivra la norme Camel case.

Les différents tests du projet se feront principalement à un niveau fonctionnel, hors environnement dans un premier temps puis à l'aide d'un billard dans un deuxième temps. Chaque composant sera donc testé indépendamment et dans le système global.

## 6. Fonctionnalités du système

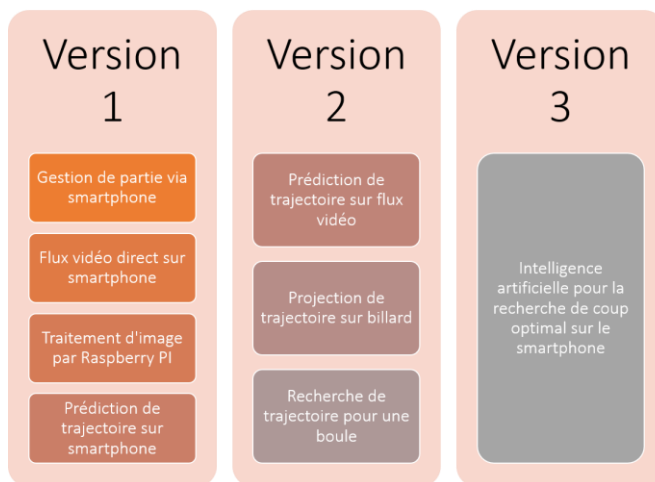


Figure 1 : Versionning du projet

Le projet est séparé en trois grandes étapes constituant trois versions différentes. Chaque version sera complètement fonctionnelle ce qui permettra d'avoir des versions plus rapides à implémenter que la version complète. Une version aura forcément les fonctionnalités de la précédente plus certaines qui seront nouvelles.

Le seul et unique utilisateur du système est donc le joueur mais le système utilisera également certaines fonctionnalités transparentes pour l'utilisateur humain.

### 6.1. Première version du projet



Figure 2 : Diagramme de cas d'utilisations de la version 1

Dans la première version le joueur pourra démarrer l'application afin de se connecter au Raspberry pour commencer une partie assistée. Une fois la partie commencée il pourra visionner le flux de la caméra placé au-dessus du billard et placer sur le même écran une queue virtuelle autour de la boule blanche afin d'observer la trajectoire incidente.

Du côté de la Raspberry le système pourra interpréter le flux vidéo de la caméra et l'analyser afin de reconnaître les bords du billard ainsi que les trous et les boules tout en les catégorisant (blanche, noir, équipe 1 et équipe 2). Le système pourra également transmettre le flux vidéo et les informations retournées par l'analyse, via le wifi, aux smartphones connectés.

Du côté des smartphones, l'application devra pouvoir se connecter au Raspberry via au réseau wifi et interpréter les informations reçues. Elle pourra donc afficher l'image reçue et utiliser les informations sorties de l'analyse de l'image faite par le Raspberry. Enfin l'application devra pouvoir placer l'affichage de la queue par rapport au pointage fait par l'utilisateur puis calculer et afficher la trajectoire induite par ce positionnement.

## 6.2. Deuxième version du projet

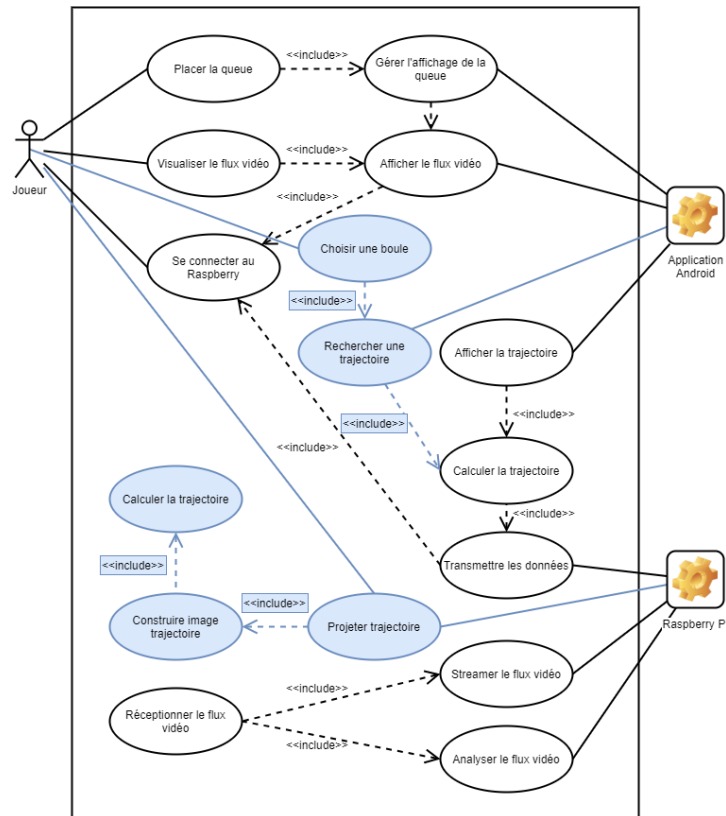


Figure 3 : Diagramme de cas d'utilisations de la version 2

La deuxième version, reprend bien évidemment tous les cas d'utilisations de la version 1, et donne en plus au joueur deux nouvelles possibilités.

La première est qu'il peut en plus de pouvoir visualiser la trajectoire prédite il peut demander au système de lui afficher la meilleure trajectoire pour rentrer une boule. Ainsi en cliquant sur la boule le joueur pourra visualiser le placement recommandé, par le composant de recherche de trajectoire du système, afin de la rentrer.

La seconde possibilité apportée par la version 2 est de pouvoir demander à la Raspberry Pi de projeter en direct sur le billard la trajectoire prédit par rapport au placement de la queue sur la blanche. Ce qui implique, du côté de la Raspberry, que celle-ci puisse calculer la trajectoire par rapport au retour vidéo et construire une image de celle-ci qu'elle projettera sur le billard.

### 6.3. Troisième version du projet

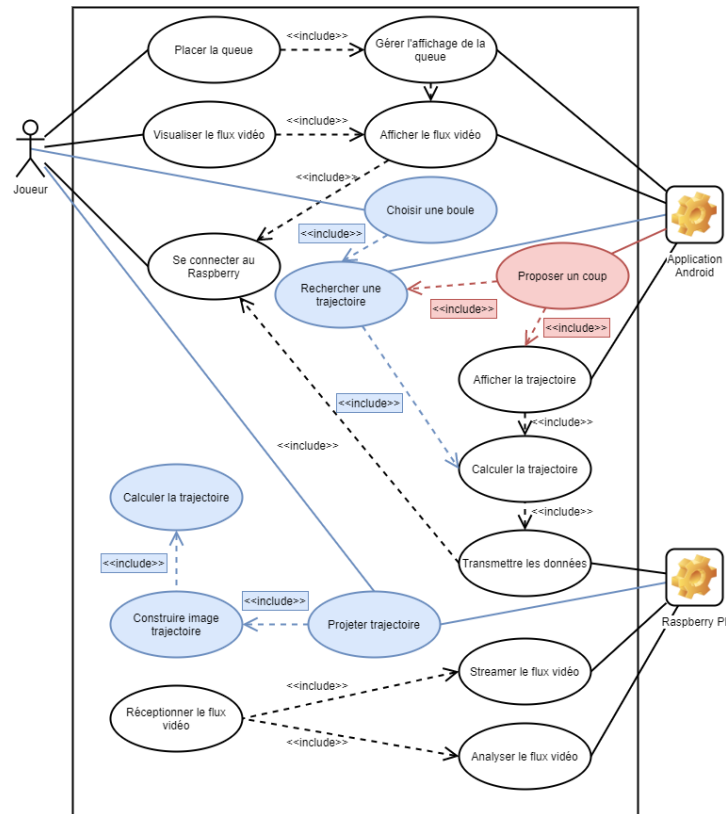


Figure 4 : Diagramme de cas d'utilisation de la version 3

La dernière version de ce projet viendra ajouter un unique composant au système, ceux-ci au niveau de l'application Android.

Ce composant sera une intelligence artificielle capable, par rapport aux placements actuels des boules, de proposer le coup qui semble le plus intéressant à jouer et de l'afficher à l'utilisateur.

## 7. Structure générale du système

Identifier les principaux composants/éléments du système ainsi que leurs relations. Cela peut être fait au moyen de diagrammes statiques (« objet » au sens large, *i.e.* classe, composant, déploiement, etc.).

Sans être une analyse à part entière, cette partie doit montrer que vous avez une première réflexion sur la structure interne.

### 7.1. Première version du projet

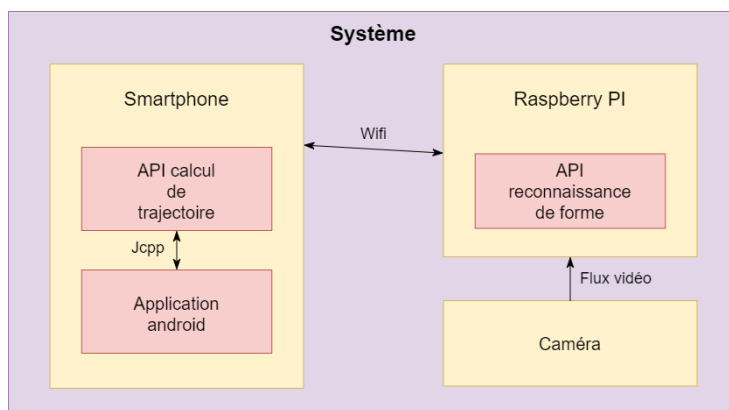


Figure 5 : Diagramme de déploiement de la version 1



7.2. Deuxième version du projet

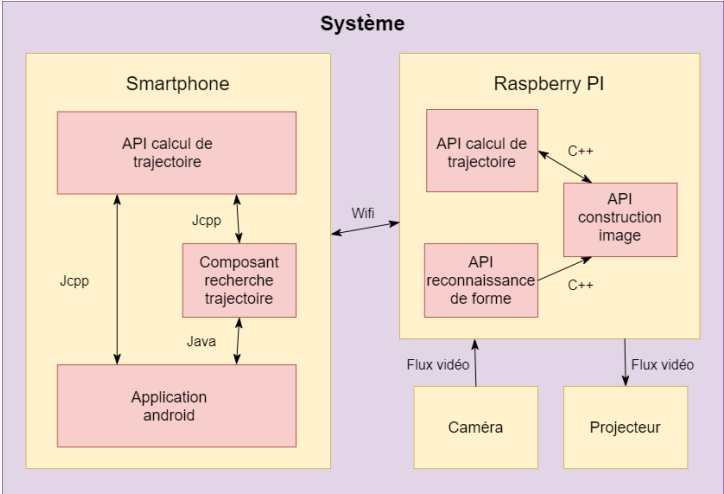


Figure 6 : Diagramme de déploiement de la version 2

7.3. Troisième version du projet

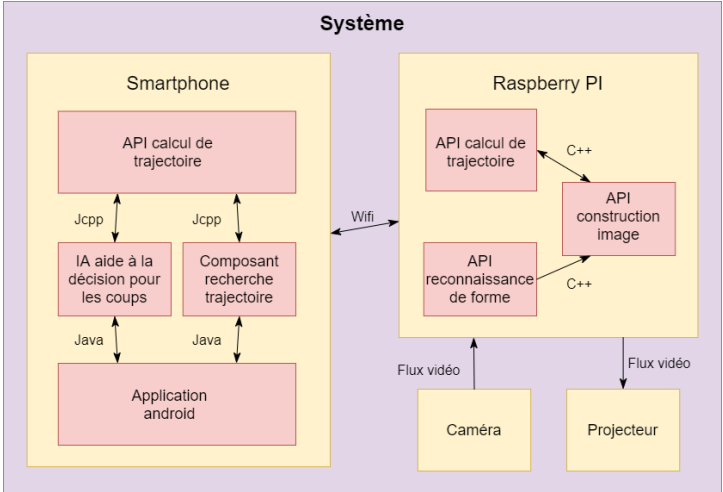


Figure 7 : Diagramme de déploiement de la version 3

## DESCRIPTION DES INTERFACES DU SYSTEME

### 8. Interface matériel/logiciel

Il y a plusieurs interface matériel/logiciel :

1. Raspberry
  - a. Notre camera se branchera sur le port CSI de la raspberry et pourra utiliser les librairies intégrées dans l'OS par default « raspbian »
  - b. Carte WIFI ou clef WIFI la création sur point d'accès WIFI (2,4GHz ou 5 GHz)
2. L'application côté Android
  - a. WIFI intégré connecté au réseau de la raspberry
  - b. Affichage sur l'écran

### 9. Interface homme/machine

### 10. Interface logiciel/logiciel

API Android – API Raspberry

1. Flux vidéo raspberry vers Android (port vidéo)
2. Flux Data (port Data)
  - a. Données sur la position des boules
  - b. Couleurs des boules

**Commented [I11]:** L'interface matériel/logiciel décrit précisément le matériel informatique et les périphériques, les procédures d'échange d'informations mis en jeu entre eux... On notera donc ici les caractéristiques du matériel qui peuvent avoir une influence sur le logiciel, telles que :

- les normes de communication : protocole d'échange et de raccordement (réseau local ...);
- type de liaison (série, parallèle, synchrone, asynchrone, ...);
- etc.

**Commented [I12]:** Il faut spécifier les points suivants :

- ergonomie du système : caractéristiques des messages d'erreur, type de navigation dans le logiciel, etc. ;
- description des formes des éditions sur papier et écrans ;
- mode d'apprentissage de l'interface éventuellement ;
- niveau d'intelligence des interfaces H/M ;
- etc.

Des maquettes ou schémas décrivant ces interfaces ainsi que la charte graphique pourront être présentés ici.

**Commented [I13R2]:**

## SPECIFICATIONS FONCTIONNELLES

Il s'agit de l'expression des besoins fonctionnels. Cette partie a donc comme objectif de décrire l'ensemble des fonctions du système en précisant avec quels composants elles interagissent. Des diagrammes de cas d'utilisation plus détaillés, ainsi que l'arbre hiérarchique des fonctionnalités pourra être fourni ici pour donner une vision plus globale. En outre, chaque fonction sera décrite précisément (cf. ci-dessous). Là encore, il s'agit d'une pré-analyse indispensable à l'évaluation de la complexité de votre projet et à la planification de sa réalisation.

### 11. Définition de la fonction i

Identification de la fonction i

Présenter la fonction :

- nom de la fonction ;
- rôle, présentation générale ;
- priorité associée à la réalisation de la fonction (primordiale, secondaire, facultative).

Description de la fonction i

Décrire précisément :

- les entrées et les sorties ainsi que les préconditions et postconditions déjà connues, uniquement sous forme textuelle et en langue naturelle (pas dans un pseudo langage algorithmique). Si ces E/S sont connectées à d'autres fonction ou interfaces, le préciser également ;
- les composants avec lesquels cette fonction interagit (données/composants utilisés/modifiés, etc. ;
- le traitement associé à la fonction et à ses interfaces. Il peut s'agir d'une explication ou d'un pseudo-algorithme général précisant les différentes étapes du traitement. Lors de l'analyse, ce dernier pourra être précisément représenté par un diagramme d'activité ; faire référence à un document/article précisant le fonctionnement le cas échéant ;
- si une gestion des erreurs spécifique (hors format des E/S) est prévue et comment celle-ci doit être mise en place si cela est déjà connu (notamment pour les fonctions sensibles).

## SPECIFICATIONS NON FONCTIONNELLES

### 12. Contraintes de développement et conception

Préciser les contraintes liées aux :

- matériels : quelles sont les particularités du matériel qui vont contraindre le développement logiciel;
- langages de programmation imposés ou adoptés ;
- logiciels et bibliothèques à utiliser pour le développement;
- environnements nécessaires : simulateurs, outils logiciels ;
- bibliothèques de programmes imposées ;
- protocoles de communication imposés : si nécessaire mettre en annexe une présentation de ces protocoles de communication;

- etc.

### **13. Contraintes de fonctionnement et d'exploitation**

Il faut dans ce paragraphe décrire les dispositions qu'il est nécessaire de prendre en compte pour les différentes conditions de fonctionnement du système.

#### **13.1. Performances**

Préciser en termes mesurables, les spécifications temps réel liées à l'utilisation du système :

- du point de vue de l'utilisateur : temps de réponse souhaité, fréquence d'utilisation, temps d'indisponibilité acceptable, etc. ;
- du point de vue de l'environnement : fréquence moyenne d'acquisition d'états ou de mesures, fréquence maximale d'E/S, etc.

#### **13.2. Capacités**

Décrire les limites des problèmes traitables par le système et les limites des éventuelles extensions comme par exemple :

- nombre max de terminaux ;
- nombre max de points d'acquisition ;
- nombre max de transactions simultanées de tel type, etc. ;
- capacité max de stockage ;
- taille max des données traitées ;
- etc.

#### **13.3. Modes de fonctionnement**

Décrire les modes d'exploitation du système tels que :

- la mise sous tension ;
- l'arrêt ;
- la reprise de secours ;
- les modes dégradés ;
- etc.

#### **13.4. Contrôlabilité**

Il faut décrire, si elles existent, les spécifications particulières permettant de suivre l'exécution d'un traitement (fichier de log, niveaux d'affichages en mode debug, etc.)

#### **13.5. Sécurité**

Indiquer le niveau de confidentialité du système (contrôle d'accès des utilisateurs, mots clefs, mots de passe, etc.). Ceci est directement lié aux différents types d'utilisateurs (cf. **Error! Reference source not found.**).

#### **13.6. Intégrité**

Préciser les protections contre la déconnexion imprévue, les pertes d'information, etc. et quelles sont les procédures à suivre pour restaurer les données du système. Y-a-t-il des situations non protégées ?

#### **14. Maintenance et évolution du système**

Préciser les contraintes liées aux procédures de maintenance :

- curative ou corrective ;
- adaptative ;
- évolutive du système ;
- perfectiv