

# Rapport projet libre

Smart-billard



Maël Bervet - Léo Legrand

## Sommaire

<u>Avant-propos .....</u>	<u>3</u>
<u>Introduction .....</u>	<u>3</u>
<u>Equipe et encadrant.....</u>	<u>3</u>
<u>Présentation du projet.....</u>	<u>4</u>
<u>Gestion de projet .....</u>	<u>5</u>
<u>Lancement .....</u>	<u>5</u>
<u>Gestion .....</u>	<u>5</u>
<u>Spécifications .....</u>	<u>6</u>
<u>Générale.....</u>	<u>6</u>
<u>Version 1 .....</u>	<u>7</u>
<u>Version 2 .....</u>	<u>8</u>
<u>Version 3 .....</u>	<u>9</u>
<u>Structure du projet .....</u>	<u>10</u>
<u>Version 1 .....</u>	<u>10</u>
<u>Version 2 .....</u>	<u>11</u>
<u>Version 3 .....</u>	<u>12</u>
<u>Outils.....</u>	<u>13</u>
<u>Gestion de projet .....</u>	<u>13</u>
<u>Technologies .....</u>	<u>13</u>
<u>Réalisation.....</u>	<u>14</u>
<u>Point d'accès WIFI .....</u>	<u>14</u>
<u>Caméra .....</u>	<u>14</u>
<u>API reconnaissance .....</u>	<u>15</u>
<u>Détection de boule.....</u>	<u>15</u>
<u>Détection de la couleur.....</u>	<u>16</u>
<u>Résultats.....</u>	<u>16</u>
<u>Bilan .....</u>	<u>17</u>

## Avant-propos

Ce document est rédigé par Maël Bervet et Léo Legrand, étudiants à l'institut polytechnique de l'université de Tours et réalisateurs de ce projet, dans le cadre du projet libre demandé par Polytech Tours qui est encadré par Pascal Makris, enseignant chercheur dans le même institut.

## Introduction

Le sujet de ce projet, trouvé par ses deux réalisateurs, est la création d'un pack permettant de transformer un billard lambda en un smart-billard offrant ainsi prédiction et affichage des trajectoires des boules. Ainsi ce pack permettra aux plus jeunes et aux débutants d'apprendre correctement à jouer au billard en pouvant visualiser à l'avance les trajectoires et obtenir diverses aides supplémentaires disponible grâce au côté connecté de ce pack. Car oui ce pack sera connecté directement à une application Android afin de permettre de gérer une partie, avec différentes options, et d'offrir des fonctionnalités supplémentaires pour un apprentissage et une expérience de jeu optimale.

## Equipe et encadrant

Ce projet est donc encadré par Pascal Makris, enseignant chercheur à Polytech Tours, qui constitue donc la partie notation de ce projet pour son côté académique.

La réalisation et la gestion de ce projet son faite par Maël Bervet et Léo Legrand qui sont également les dépositaires de ce sujet. Ainsi ils constituent tous deux aussi bien la maitrise d'ouvrage, ou MOA, que la maitrise d'œuvre, ou MOE.

## Présentation du projet



Ce projet concerne donc la création d'un pack permettant à un billard classique, de type américain ou anglais, de devenir un smart-billard.

Mais qu'est-ce qu'un smart-billard concrètement ? Un smart-billard, ou billard intelligent, est un billard qui peut calculer et afficher la trajectoire qui sera prise non seulement par la boule blanche lorsque le joueur tapera dedans mais également par la première boule qu'elle percute.

Ainsi grâce à cette affichage un débutant pourra apprendre à visualiser correctement les trajectoires en fonction du positionnement de sa queue par rapport à la boule blanche.

In fine, ce pack sera constitué d'un Raspberry PI connecté à une caméra et à un projecteur le tout embarqué dans un conteneur désigné pour l'occasion. Le Raspberry PI, une fois installé, mettra à disposition un point d'accès wifi permettant à un téléphone de s'y connecter. Une fois connecté le téléphone, sous Android, permettra, grâce à l'application préalablement installée, de commencer une partie assistée. La partie commencée le joueur pourra visualiser le flux vidéo, capturé par le Raspberry PI, directement sur son téléphone et ainsi avoir accès aux différentes fonctionnalités offertes par l'application.

L'application permettra de choisir si l'on souhaite projeter en direct la prédiction de trajectoire sur le billard ou se contenter de l'affichage sur smartphone. Elle offrira également des fonctionnalités supplémentaires, par rapport à un smart-billard classique, comme la possibilité de rechercher une trajectoire pour pouvoir rentrer la boule souhaitée ou encore de pouvoir recevoir une suggestion de coup intéressant à faire pour mener à la victoire.

Le côté connecté de ce pack permet donc d'offrir plus de fonctionnalités au joueur qu'un smart-billard classique. Et les possibilités ne se limitent pas à celles déjà évoquées. On pourrait, par exemple, ajouter à cette application un menu d'entraînement, virtuel ou réel, qui proposerait des situations, plus ou moins complexes, qu'il faudrait résoudre avec le ou les coups adéquats. Ainsi l'entraînement n'en serait que plus bénéfique et la montée en compétence que plus rapide.

## Gestion de projet

### Lancement

Afin de lancer ce projet et de définir noir sur blanc sa ligne directrice, une réunion, de type brainstorming, a été faite dès le début du projet. Ainsi les besoins et spécifications attendue ont rapidement pu être définies et mise à l'écrit, dans un document non formalisé, pour créer une « mind-map » qui aura servie de ligne directrice tout au long du projet.

### Gestion

Etant donné que la définition de ce projet était déjà très claire dès le début pour les deux protagonistes et que les ressources, aussi bien humaines que temporelles, était très limité il a été décidé unanimement de mettre en place une gestion de projet d'une simplicité extrême.

Pour ce faire chacun des protagonistes est partie sur deux parties différentes du projet afin de pouvoir travailler de manière indépendante. Bien évidemment une communication a été maintenue tout au long du projet via un serveur Discord, qui est l'outil choisi par l'équipe pour gérer les communications. Et afin de pouvoir gérer l'avancée du projet de manière optimale un système de gestion de versions a été mis en place, ainsi toute la gestion des fichiers de documentations et projets de développements n'était plus un point susceptible d'être bloquant. De plus la mise en place de ce système a permis aux deux protagonistes de pouvoir avoir accès au même fichier sans impliquer de communications inutiles.

## Spécifications

### Générale

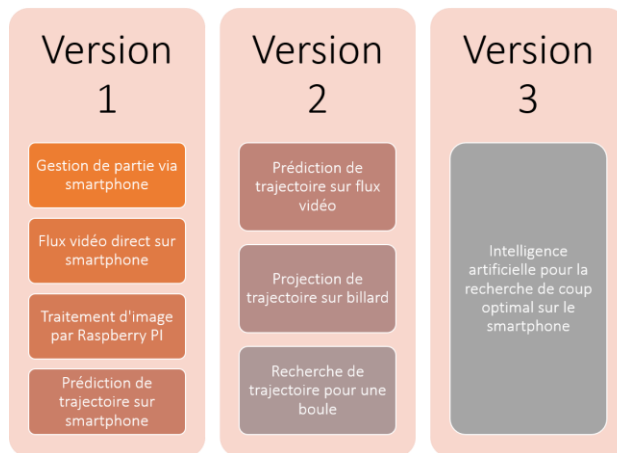


Figure 1 : Versionning du projet

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.

## Version 1

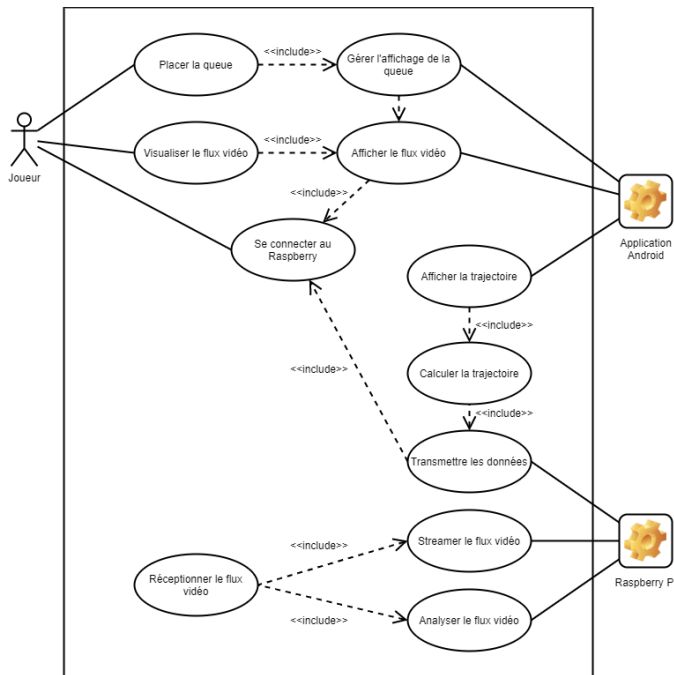


Figure 2 : Diagramme de cas d'utilisations de la version 1

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 reçu 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.

## Version 2

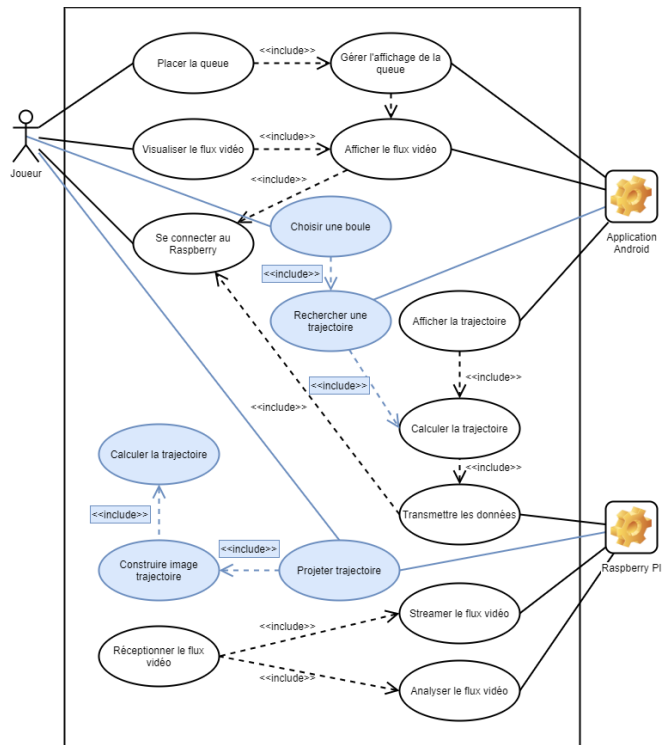


Figure 3 : Diagramme de cas d'utilisations de la version 2

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.



## Version 3

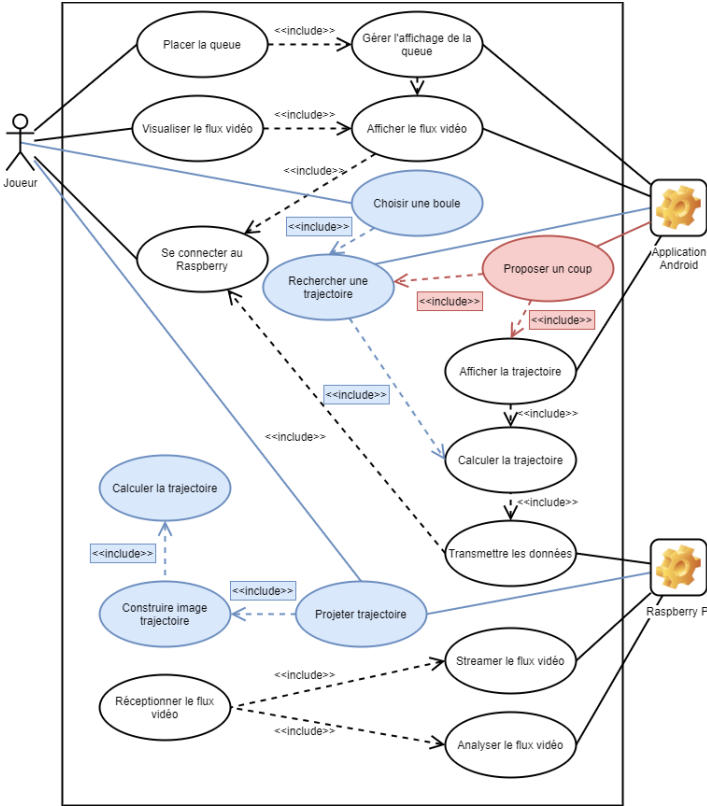


Figure 4 : Diagramme de cas d'utilisation de la version 3

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.

## Structure du projet

### Version 1

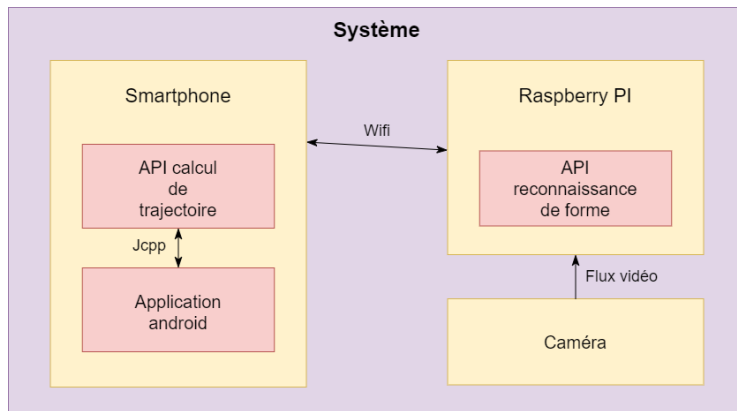


Figure 5 : Diagramme de déploiement de la version 1

Le système, dans toutes ses versions, sera constitué d'au moins un Raspberry PI, d'une caméra qui lui communiquera le flux vidéo et d'un smartphone Android connecté en wifi au Raspberry PI.

Dès la première version le Raspberry PI sera donc capable de recevoir le flux vidéo et de créer un flux streaming de cette vidéo auquel pourront se connecter les joueurs via leurs smartphones. En utilisant les images de cette vidéo le Raspberry pourra également appliquer une reconnaissance de forme afin de pouvoir délimiter le billard et reconnaître les différentes boules tout en s'adaptant aux deux types de billard pris en compte, soit le billard américain et le billard anglais.

Le smartphone, via le wifi, recevra donc le flux vidéo du streaming ainsi que les informations de reconnaissance de forme qui auront été retournées par l'api C++ sur la Raspberry. Depuis ces informations le smartphone pourra calculer les trajectoires incidentes en fonction du placement virtuel de la queue faite sur l'écran du smartphone. Les calculs de trajectoires seront faits par l'api C++ qui sera mise en place sur le smartphone. Cette api communiquera les informations de trajectoires calculées à une application Android, via l'adaptateur JCPP, qui pourra ainsi afficher ladite trajectoires.

## Version 2

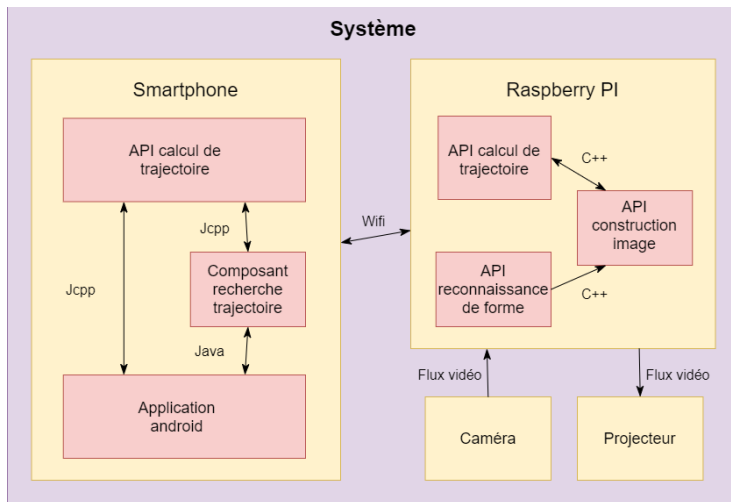


Figure 6 : Diagramme de déploiement de la version 2

La deuxième version du système, reprenant tous les composants de la première, sera constitué d'un composant physique supplémentaires qui sera un projecteur. Ce projecteur permettra d'ajouter la fonctionnalité de projection de trajectoires calculées sur le billard. Cela implique que l'api de reconnaissance de forme pourra reconnaître la queue sur le billard et qu'une version de l'api de calcul de trajectoires sera intégrée au Raspberry PI. L'association des informations retournées par ces api permettra à la nouvelle api de construction d'image de créer une image de la trajectoire calculée afin de la projeter sur le billard.

Du côté du smartphone la nouveauté de cette version sera l'arrivée d'un composant de recherche de trajectoire. Il sera utilisé afin d'offrir à l'utilisateur la possibilité de choisir une boule sur l'écran du smartphone pour en rechercher la trajectoire optimale pour la rentrer. Ce nouveau composant sera fait en java et communiquera avec l'api de calcul de trajectoires pour trouver la bonne trajectoire et la renvoyer à l'application Android.

### Version 3

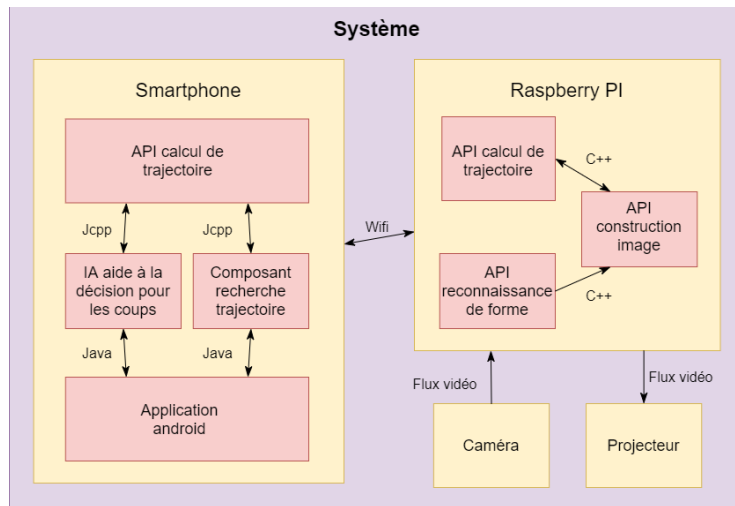


Figure 7 : Diagramme de déploiement de la version 3

La dernière version du système n'intégrera qu'un seul nouveau composant mais pas des moindres. Ce nouveau composant ne sera rien de moins qu'une intelligence artificielle qui permettra d'apporter à l'utilisateur une aide à la décision.

Plus précisément cette aide se fera sous la forme de proposition, sur le smartphone, d'un coup calculé comme étant un coup intéressant à faire pour mener le joueur à la victoire. Ainsi ce nouveau composant pourra utiliser l'api de calcul de trajectoire afin d'en évaluer pour ensuite retourner à l'application la plus prometteuse.

## Outils

Pour ce projet, comme pour la plupart des projets, nous avons utilisé beaucoup d'outils différents afin de pouvoir aussi bien gérer le projet, réaliser la documentation, stocker le projet que pour programmer les composants du système.

### Gestion de projet

Pour la gestion du projet nous avons donc utilisé Google doc afin d'écrire la « mind map » directrice sur laquelle nous nous sommes basés afin de réaliser ce projet. Nous avons bien évidemment utilisé des systèmes de stockage cloud afin de ne pas perdre notre travail et de pouvoir avancer simultanément sur le projet. Ces systèmes sont Google Drive pour la totalité du projet, OneDrive pour la partie documentation du projet via le pack office et bien évidemment la totalité du projet est hébergé avec un système de versionning qui n'est nul autre que GitHub.

Pour ce qui est des revues, compte rendu de progression ou encore problème survenu nous avons utilisé Discord par le biais d'un serveur créé spécialement pour le projet nous permettant ainsi de communiquer en toute liberté et de manière instantanée sur le projet.

### Technologies

Pour réaliser ce projet plusieurs technologies ont été nécessaires. Nous avons tout d'abord, utilisé pour la partie Android du projet l'environnement de développement Android studio pour lequel nous avons choisi de mettre en place la SDK 21 afin de permettre une compatibilité de l'application avec la plupart des smartphones Android. Bien évidemment l'application est en JAVA et intègre l'adaptateur JCPP afin de pouvoir communiquer avec les différentes apis nécessaires au fonctionnement du système.

Pour la seconde version il a été prévu d'utiliser le langage JAVA également pour développer l'api de recherche de trajectoire. Pour la troisième version l'intelligence artificielle sera développée soit en JAVA soit en C++ 14 pour des soucis de communication avec l'application.

Pour finir le développement des apis de calcul de trajectoire et de reconnaissance de forme ont nécessité la mise en place de l'environnement, proposé par Microsoft, Visual studio 2017 afin de pouvoir les développer avec le langage C++ version 14. L'api de reconnaissance de forme à quant à elle nécessité l'apport d'une librairie supplémentaire qui est OpenCV.

## Réalisation

### Point d'accès WIFI

Pour que le joueur puisse se connecter au jeu, nous avons décidé que la Raspberry serait le centre des connections pour faciliter le développement de nos applications par la suite. Comme la Raspberry tourne sous le système d'exploitation « Raspbian », nous avons dû rechercher une solution sur linux pour permettre de créer un Point d'accès wifi.

**Hostapd** est un utilitaire sous linux qui permet la gestion de points d'accès wifi avec une carte wifi, il permet aussi l'authentification des clients qui se connecte sur le Raspberry.

**Dnsmasq** est un autre utilitaire que nous allons utiliser dans notre projet et celui-ci nous permettra de faciliter l'attribution des adresse IP au client qui se connecte au Raspberry.

Pour plus d'information sur la procédure à suivre, nous avons suivi le tutoriel sur le site suivant :

<https://www.raspberrypi.org/documentation/configuration/wireless/access-point.md>

### Caméra

Lorsque nous avons recherché les possibilités d'acquisition vidéo pour notre projet, nous avons trouvé une solution viable et qui allait de pair avec notre Raspberry. Il s'agit de la [caméra pour Raspberry](#) qui se connecte sur l'un des ports de celle-ci. Nous avons donc ensuite configuré l'accès à cette caméra sur le Raspberry en passant dans par le menu « rasp-config » de Raspbian.

Afin de mettre en place le flux streaming, de la vidéo retournée par la caméra, nous avons décidé d'utiliser le logiciel VLC. Nous avons donc eu à installer VLC en utilisant la commande : « sudo apt install vlc » puis il nous a fallu lancer le Stream via une seconde commande :

```
raspivid -t 0 -n --width 1280 --height 720 -o - | cvlc stream:///dev/stdin --sout '#standard{access=http,mux=ts,dst=:8090}' :demux=h264
```

A l'aide de cette commande nous pouvons gérer différents paramètres sur la diffusion de la vidéo via le réseau local du Raspberry. Une fois cette commande exécutée, la transmission de la vidéo est accessible sur l'IP du Raspberry lorsque l'on se connectera sur son réseau local.

## API reconnaissance

Pour le développement de la reconnaissance de forme, nous avons opter pour un code en C++ pour pouvoir l'utiliser sur le serveur et le client Android. Pour faciliter et accélérer le développement, nous avons utilisé une bibliothèque externe « OpenCV ».

Cette librairie nous a permis de faire abstraction de pas mal d'allocation mémoire et de faciliter le traitement des images ou d'un flux vidéo venant du réseau.

## Détection de boule

Notre première utilisation de cette librairie a été de détecter les boules du billard. Pour cela nous utilisons une méthode pour détecter des formes circulaires sur une image « HoughCircles ». Notre méthode va prendre une image en forme de matrice, la transformer en une image en nuance de gris pour y appliquer un filtre et faciliter la détection des différences de contraste. Dans la méthode certains paramètres auraient dû être modifiés pour qu'elle s'applique au tapis du billard, cependant nous n'avons pas pu réaliser ces modifications car nous n'avions plus accès à au billard dû aux circonstances de l'épidémie de COVID-19. Notre méthode va alors retourner un vecteur avec les informations de chaque cercle détecter.

```
/*
Input: image matrice cv::Mat
Output: Vecteur
*/
vector<Vec3f> API_detection::position_ball(Mat image)
{
    if (image.empty())
    {
        throw new Exception();
    }

    Mat gray;
    cvtColor(image, gray, COLOR_BGR2GRAY);
    medianBlur(gray, gray, 5);
    imshow("origin", image);
    vector<Vec3f> circles;
    HoughCircles(gray, circles,
        HOUGH_GRADIENT,
        1,
        gray.rows / 64, //matrix for minimum distance between circle
        30,
        13,
        12, //maximum radius of circle
        8 // minimum radius of circle
    );
    return circles;
}
```

Figure 8 : Code détection des boules

### Détection de la couleur

Afin de reconnaître l'appartenance d'une boule à une équipe nous allons simplement détecter la couleur moyenne d'une boule. Pour cela, nous allons récupérer les coordonnées d'une boule détectée par la méthode précédente puis extraire une sous-image de l'image d'origine pour n'avoir que l'image de la boule concernée. Avec cette sous-image, il nous reste plus qu'à évaluer la couleur dans celle-ci et à la retourner.

```
/*
Function to return mean value of a circle
Input: Mat original image , Circle point
Output: Scalar of mean color detected
*/
Scalar API_detection::Color_detector(Mat image, Vec3i c)
{
    //roi subMatrix of Matrix image
    Mat roi = image(Range(c[1] - c[2], c[1] + c[2]), Range(c[0] - c[2], c[0] + c[2]));

    Mat1b mask(roi.rows, roi.cols);

    Scalar mean = cv::mean(roi, mask);

    return mean;
}
```

Figure 9 : Code pour détecter la couleur d'une boule

### Résultats

Ainsi grâce à ces deux méthodes nous pouvons déjà détecter les boules et en connaître la couleur. Nous avons donc pu obtenir les résultats suivants en appliquant ces méthodes sur la photo d'un billard prise à Polytech Tours. La photo de gauche montrant l'image d'origine et celle de droite l'image avec l'affichage du résultat de la détection et de l'identification des boules.

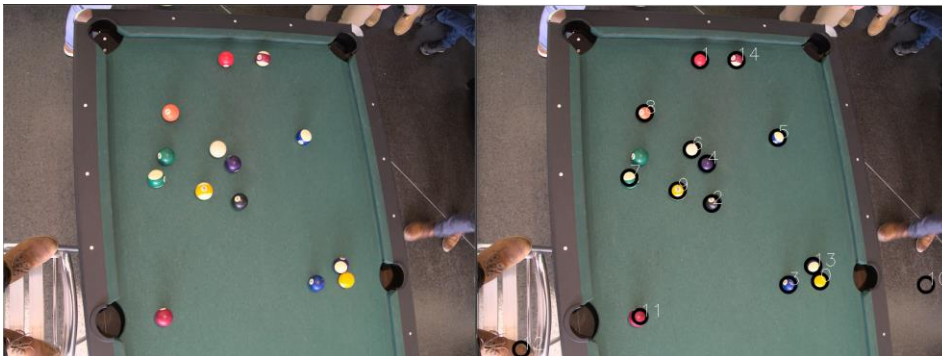


Figure 10 : capture d'un affichage de détection

Comme vous pouvez le constater sur la figure 10, l'api détecte autre chose que les boules. Afin de régler ce problème il serait nécessaire de modifier les paramètres sur le terrain et de limiter la possibilité de détecter autre part que sur le tapis du billard.



## Bilan

Par manque cruel de temps et dû aux circonstances exceptionnelles qui ont entouré cette fin de semestre, parmi les différents composants à implémenter seul l'api de reconnaissance de forme a été développé. Cependant la configuration du Raspberry PI a été mise en place, soit la possibilité de pouvoir se connecter à deux joueurs sur le système via au point d'accès du Raspberry et la mise en place du flux streaming en direct du billard pour retransmission au smartphone. De plus l'api de reconnaissance de forme a pu être testé et serait fonctionnelle avec un paramétrage sur le terrain. Au niveau documentation le cahier des spécifications n'a pas pu être fini. Cependant celui-ci est suffisamment avancé pour pouvoir mettre en place la totalité du système.

Par conséquent il reste à faire, pour la première version ce projet, le développement de l'application Android permettant l'accès au Raspberry et également le développement de l'api de calcul de trajectoire. De plus l'écriture d'un cahier de recette, nécessaire à la qualité d'un projet n'a pas pu être réalisé.

Pour conclure le bilan de ce projet n'est malheureusement pas bien brillant cependant il nous a permis de voir certaines circonstances capables de faire échouer un projet en informatique tout en nous faisant revoir les bases de la gestion de projet.