ÉCOLE POLYTECHNIQUE DE L'UNIVERSITE 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

CAHIER DE SPECIFICATION

Projet :	Smart-billard	
Emetteur :	Maël Bervet Léo Legrand	**MOA : Maël Bervet, Léo Legrand**
Date d'émission :	11/02/2020	

Validation

Nom	Date	Valide (O/N)	Commentaires
Léo Legrand	05/04/2020	O	Version 04 incomplète
Maël Bervet	05/04/2020	O	Version 04 incomplète

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
03	03/04/2020	Ajout des spécifications non fonctionnelles
04	05/04/2020	Ajout spécifications fonctionnelles

TABLE DES MATIERES

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. Toutefois 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 constituée 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'environnent 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 continu lors d'une partie
- D'un Raspberry Pi 3 afin de récupérer, d'analyser et de transmettre le flux vidéo
  D'un projecteur 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 avancée 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

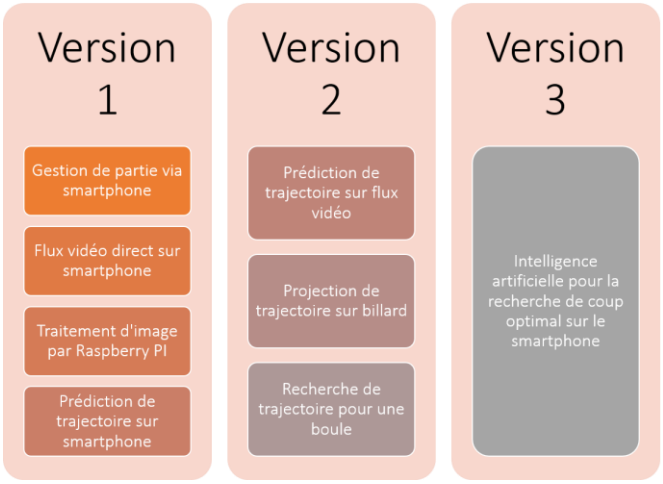


Figure 1 : Versionning du projet

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

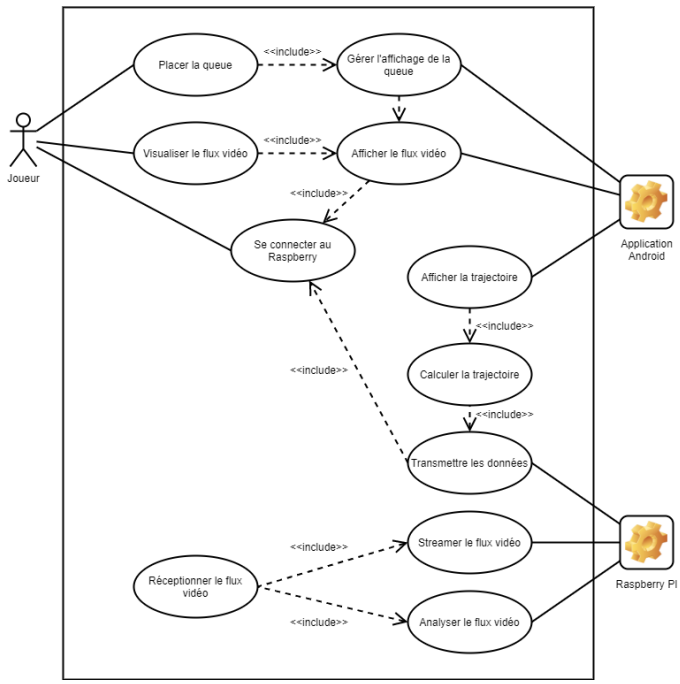


Figure 2 : Diagramme de cas d'utilisations de la version 1

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 reconnaitre 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

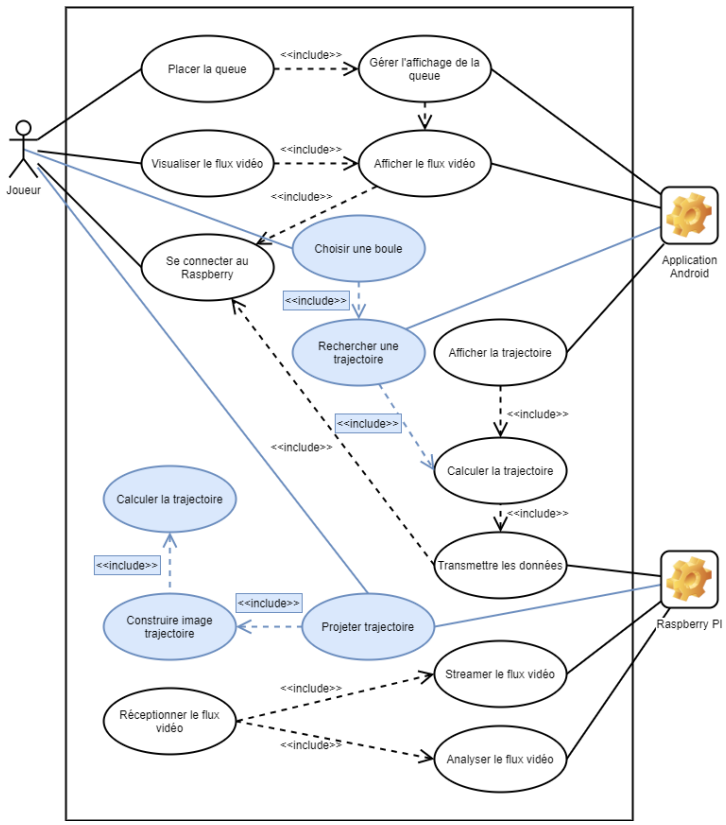


Figure 3 : Diagramme de cas d'utilisations de la version 2

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

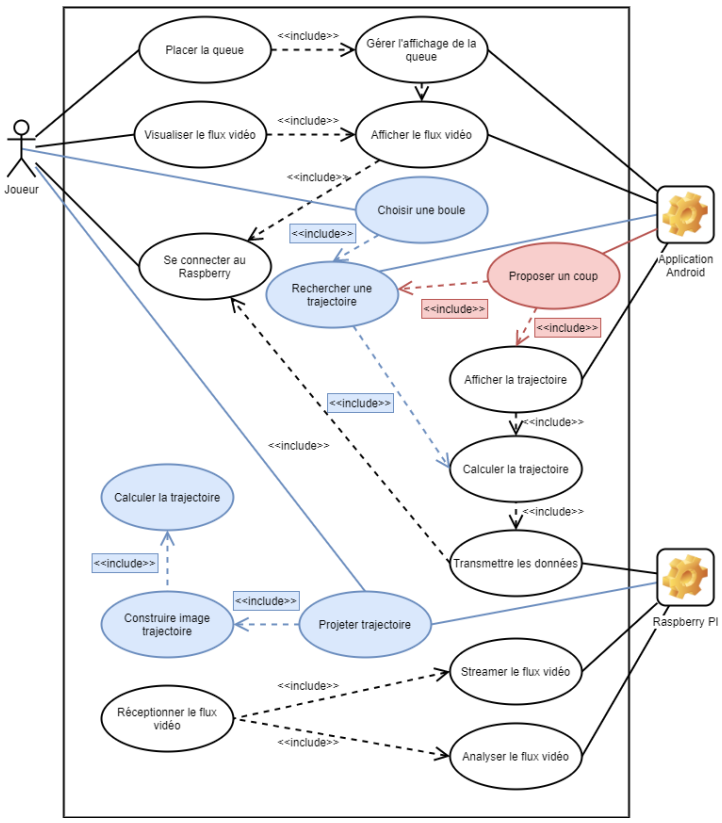


Figure 4 : Diagramme de cas d'utilisation de la version 3

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

7.1. Première version du projet

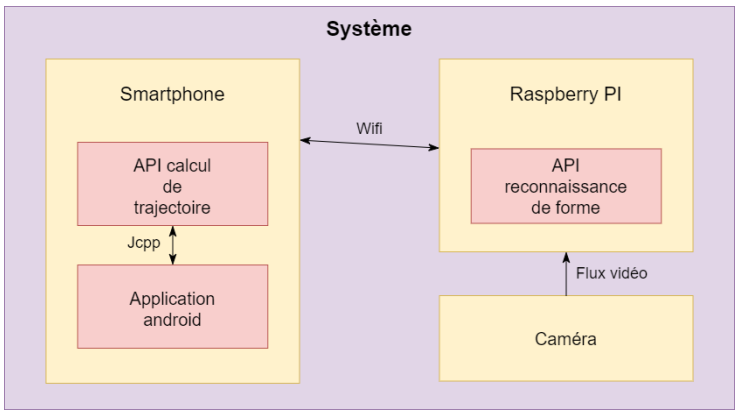


Figure 5 : Diagramme de déploiement de la version 1

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.

7.2. Deuxième version du projet

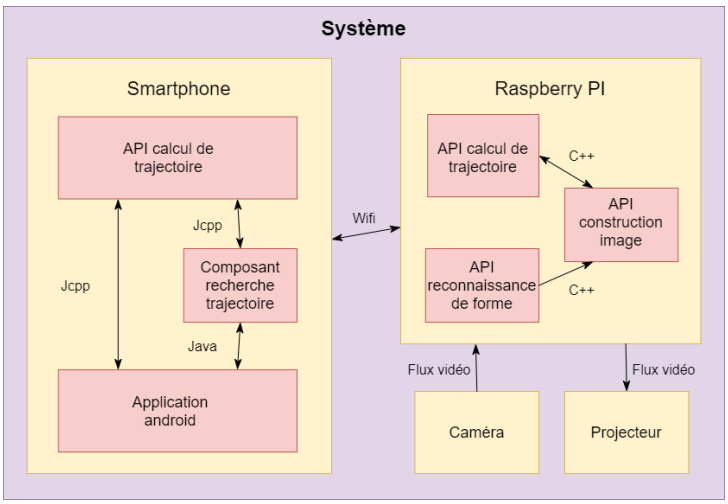


Figure 6 : Diagramme de déploiement de la version 2

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.

7.3. Troisième version du projet

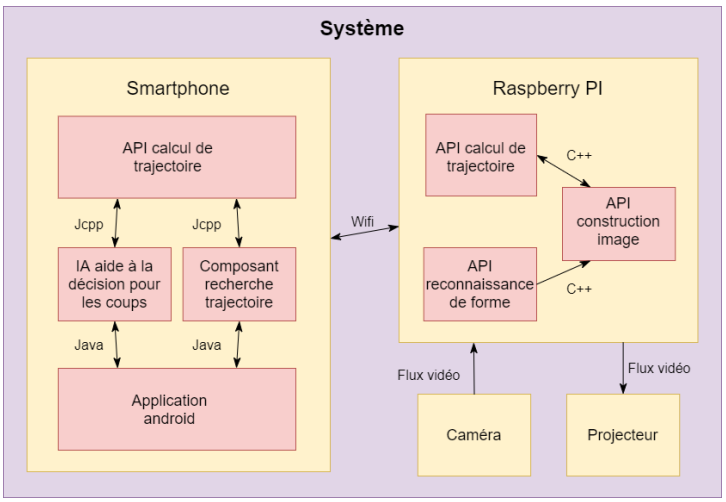


Figure 7 : Diagramme de déploiement de la version 3

La dernière version du système n'intègrera 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.

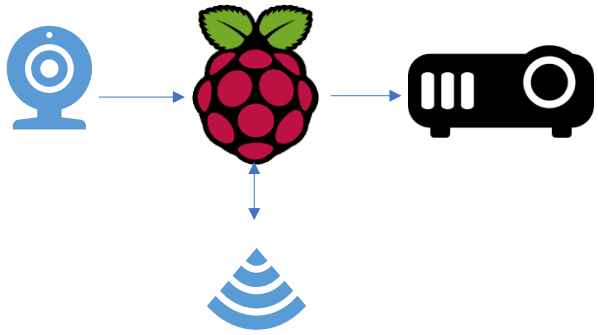
Plus précisément cette aide se fera sous la forme de proposition, sur le smartphone, d'un coup calculé comme étant un coup intéressant à faire pour mener le joueur à la victoire. Ainsi ce nouveau composant pourra utiliser l'api de calcul de trajectoire afin d'en évaluer pour ensuite retourner à l'application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

DESCRIPTION DES INTERFACES DU SYSTEME

8. Interface matériel/logiciel

Il y a plusieurs interface matériel/logiciel :

1. Raspberry
    a. Matériels
        i. Notre camera se branchera sur le port CSI de la Raspberry
        ii. Carte WIFI ou clef WIFI la création sur point d'accès WIFI (2,4GHz ou 5 GHz)
        iii. L'interface HDMI pour sortir la vidéo de la Raspberry
    b. Logiciel
        i. Raspbian



2. L'application côté Android
    a. Matériels
        i. WIFI intégré connecté au réseau de la Raspberry
        ii. Affichage sur l'écran
    b. Logiciel
        i. Android

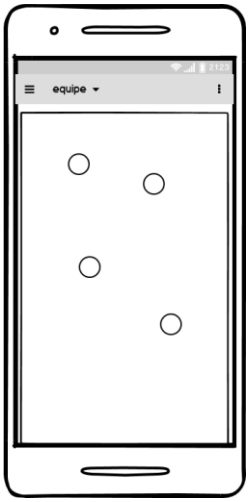
Commented [ll1]: L'interface matériel/logiciel décrit précisément le matériel informatique et les périphériques, les procédures d'échange d'informations mis en jeu entre eux… On notera donc ici les caractéristiques du matériel qui peuvent avoir une influence sur le logiciel, telles que :
- Les normes de communication : protocole d'échange et de raccordement (réseau local …) ;
- Type de liaison (série, parallèle, synchrone, asynchrone, …) ;
- etc.

9. Interface homme/machine

Commented [ll2]: 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 [ll3R2]:



Sur l'interface, on retrouve un menu pour paramétrer l'équipe dans laquelle on est. En dessous, il y aura la retransmission du plateau du billard. Il y aura apparition d'une canne artificielle sur la boule blanche et la trajectoire sera calculer en temps réel et afficher.

Erreur pouvant apparaitre :

Un popup si la connexion a été coupé ou si un time out a été dépassé.

10. Interface logiciel/logiciel

Raspberry > Android

1. Flux vidéo Raspberry vers Android (port vidéo)
    a. VLC pour la gestion du Stream
2. Flux Data (port Data)
    a. Données sur la position des boules
    b. Couleurs des boules

RASPVID vers VLC :

Elle permet de rediriger le flux du la caméra sur un port Ethernet du Raspberry.

Commented [MB4]: 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 interagie (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).

11. Version 1

11.1. Joueur

11.1.1. Se connecter au Raspberry PI

Le rôle de cette fonction est de permettre à un joueur de se connecter au Raspberry afin de lui permettre, via l'application de recevoir les informations et le flux vidéo transmis par le Raspberry. Sans cette fonction le système ne peut être utilisé, elle est donc de très haute priorité.

Préconditions	Le wifi est activé
Entrées	Un booléen disant si le smartphone est déjà connecté au Raspberry
Interactions	Système Android pour accès au gestionnaire de wifi
Traitements	Si le smartphone est déjà connecté : - Ne fais rien Si le smartphone n'est pas connecté : - Vérifie les autorisations d'accès au wifi et les demandent si nécessaire - Lance le gestionnaire de wifi afin que l'utilisateur se connecte au Raspberry
Retours	Un booléen disant si le smartphone est déjà connecté au Raspberry
Postconditions	Le smartphone est connecté au Raspberry

11.1.2. Placer la queue

Le rôle de cette fonction est de permettre au joueur de placer la queue de billard sur l'écran du smartphone pour que le joueur fasse connaître quelle trajectoire il veut visualiser. Cette fonctionnalité est nécessaire au fonctionnement de la version 1 elle est donc de très haute priorité.

Préconditions	L'application est démarrée
Entrées	Les coordonnées du doigt de l'utilisateur sur l'écran
Interactions	Ecran tactile pour réception des coordonnées Fonction d'affichage de la queue Fonction de calcul de trajectoire
Traitements	Récupère les coordonnées afin de pouvoir correctement placer l'image de la queue de billard et déclencher le calcul de trajectoire
Retours	Aucun
Postconditions	La queue a été correctement placée et la trajectoire est affichée

11.1.3. Visualiser le flux vidéo

Préconditions	
Entrées	
Interactions	
Traitements	
Retours	
Postconditions	

11.2. Smartphone

11.2.1. Gérer l'affichage de la queue

Préconditions	
Entrées	
Interactions	
Traitements	
Retours	
Postconditions	

11.2.2. Afficher la trajectoire

Préconditions	
Entrées	
Interactions	
Traitements	
Retours	
Postconditions	

11.2.3. Calculer la trajectoire

Préconditions	
Entrées	
Interactions	
Traitements	
Retours	
Postconditions	

11.3. Raspberry PI

11.3.1. Réceptionner le flux vidéo

Préconditions	
Entrées	
Interactions	
Traitements	
Retours	
Postconditions	

11.3.2. Analyser le flux vidéo

Préconditions	
Entrées	
Interactions	
Traitements	
Retours	
Postconditions	

11.3.3. Streamer le flux vidéo

Préconditions	
Entrées	
Interactions	
Traitements	
Retours	
Postconditions	

11.3.4. Transmettre les données

Préconditions	
Entrées	
Interactions	
Traitements	
Retours	
Postconditions	

12. Version 2

12.1. Joueur

12.1.1. Choisir une boule

Préconditions	
Entrées	
Interactions	
Traitements	
Retours	
Postconditions	

12.1.2. Projeter la trajectoire

Préconditions	
Entrées	
Interactions	
Traitements	
Retours	
Postconditions	

12.2. Smartphone

12.2.1. Rechercher une trajectoire

Préconditions	
Entrées	
Interactions	
Traitements	
Retours	
Postconditions	

12.3. Raspberry PI

12.3.1. Calculer la trajectoire

Reprend le même fonctionnement que pour le calcul de trajectoire de la version 1 car il s'agit de la même api.

12.3.2. Construire une image de trajectoire

Préconditions	
Entrées	
Interactions	
Traitements	
Retours	
Postconditions	

13. Version 3

13.1. Smartphone

13.1.1. Proposer un coup

Préconditions	
Entrées	
Interactions	
Traitements	
Retours	
Postconditions	

SPECIFICATIONS NON FONCTIONNELLES

14. Contraintes de développement et conception

Commented [MB5]: 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.

La conception de notre système est assez simple. Elle consiste en la modélisation des besoins et de la structure du système via des diagrammes UML fait sous draw.io de Google et en l'écriture d'un cahier des spécifications sous Word.

La partie développement, quant à elle, nécessite plus de technologie afin de pouvoir développer tous les composants du système. Tout d'abord pour mettre en place les deux composants physiques principaux le système nécessite un smartphone sous Android et un Raspberry PI 3.

Pour le smartphone qui accueillera l'application Android il sera nécessaire d'avoir l'environnement de développement Android studio ainsi que l'émulateur Android natif. L'application Android sera développée avec la SDK 21 afin que la plupart des smartphones puisse l'utiliser et l'adaptateur JCPP sera inclus afin de pouvoir communiquer avec l'api de calcul de trajectoire. Cette api, hébergé sur le smartphone, sera développé en C++ 14 en utilisant l'environnement de développement de Microsoft, Visual studio 2017. Elle sera développée sous la forme d'une Dynamic Linked Librairy, ou DLL, afin de pouvoir être facilement intégré dans n'importe quel système. Pour les versions 2 et 3, l'api de recherche de trajectoire ainsi que l'intelligence artificielle seront développées en java sous JDK 12 (pour l'intelligence artificielle il est possible de changer avec le C++ en fonction des besoins).

Pour que la Raspberry PI 3 qui gèrera le flux vidéo, il sera également nécessaire d'avoir l'environnement de développement de Microsoft « Visual studio 2017 », afin de développer l'api de reconnaissance de forme en C++ 14, avec l'aide de la librairie OpenCV, sous la forme d'une seconde DLL. Pour la version 2, l'api de construction d'image sera également développée sous Visual studio 2017 en C++ 14 en tant que DLL. Le Raspberry PI 3 fonctionnera sous Raspbian, le système d'exploitation Linux, ce qui nécessitera de compiler les apis qui y seront hébergé avec le compositeur GCC intégré à toutes versions de Linux. Enfin pour pouvoir mettre en place le flux streaming du billard le logiciel VLC sera nécessaire sur le Raspberry PI. Une certaine configuration sera également nécessaire sur le Raspberry afin de le paramétrer en tant que point d'accès wifi à deux slots. Il faudra également le configurer afin qu'à son démarrage il lance automatiquement le streaming vidéo ainsi que les différentes apis.

15. Contraintes de fonctionnement et d'exploitation

15.1. Performances

En termes de performance l'api de calcul de trajectoire devra pouvoir de manière presque instantanée afin de pouvoir garantir la fluidité d'affichage de la trajectoire que ce soit sur le smartphone ou en projection. Cela implique un temps de réponse similaire de la part de l'api de construction d'image, pour la version 2, afin de garantir la fluidité de la projection de trajectoire.

L'api de reconnaissance de forme quant à elle peut se permettre un temps de réponse plus lent, de l'ordre de 100 millisecondes, bien qu'un temps de réponse plus rapide serait grandement apprécié par les joueurs, de l'ordre de 10 millisecondes.

L'api de recherche de trajectoire peut se permettre de répondre en 1 seconde car une fois la recherche activée par l'utilisateur un petit temps de latence n'impactera que très peu la qualité d'utilisation. Cependant de la même que pour l'api de reconnaissance de forme un temps de réponse plus rapide serait apprécié par les joueurs.

Étant donné la complexité de son travail l'intelligence artificielle pour elle se permettre de répondre en quelques secondes lors de la demande de suggestion de coup par l'utilisateur. Mais ce temps ne devra pas dépasser les 5 secondes pour ne pas trop interrompre la partie. Une nouvelle fois plus ce temps de réponse sera petit plus son utilisation sera appréciée par les joueurs.

Pour finir la transmission du flux streaming via le wifi avec le smartphone devra se faire avec le moins de latence possible pour garantir la qualité de l'assistance pour la partie.

Commented [MB6]: 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.

15.2. Capacités

Etant donné qu'une partie de billard ce joue forcément avec deux équipes, ni plus ni moins, le point d'accès du Raspberry ne pourra accueillir que deux smartphones au maximum en même temps. De plus au vu de la nature séquentielle d'une partie de billard, au tour par tour, la mise en place d'une parallèlisation des requêtes faites au Raspberry n'est pas nécessaire.

Commented [MB7]: 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.

15.3. Modes de fonctionnement

Le démarrage de notre système est des plus simples car celui-ci sera configuré pour ce lancer dès le démarrage du Raspberry PI. Donc lors de la mise sous tension du Raspberry celui-ci démarrera automatiquement car une fois branché un Raspberry tourne en continue. Ainsi notre système sera continuellement sous tension et le flux streaming sera constamment disponible afin de pouvoir jouer.

Etant en fonctionnement constant notre système pourra continuellement accueillir les joueurs qui se connecteront à l'aide de leurs smartphones pour pouvoir visualiser le flux direct. Bien évidemment le système ne pourra fonctionner que si le billard est correctement placé dans le champ de vision de la caméra. Lorsque cela ne sera pas le cas, une alerte sera envoyée aux joueurs connectés via l'application.

Si lors de sa connexion au système un joueur s'aperçoit que celui-ci dysfonctionne, soit car il ne répond pas soit via les fichiers de log, il lui suffira de mettre le système hors tension en débranchant le Raspberry puis de le rebrancher afin de relancer le système.

Si l'un des joueurs perd la connexion avec le système il lui suffira de se reconnecter via le wifi pour retrouver toutes ses fonctionnalités.

Commented [MB8]: Décrire les modes d'exploitation du système tels que :
- La mise sous tension ;
- L'arrêt ;
- La reprise de secours ;
- Les modes dégradés ;
- etc.

15.4. Contrôlabilité

Dans les cas où le billard est mal cadré ou que l'exécution d'une commande dépasse le timeout qui lui est fixé un message d'alerte sera retourné au joueur via l'application.

En dehors de ces cas le suivi de l'exécution de système ce fera via des fichiers de log continuellement misent à jour afin de permettre aux joueurs, via l'application, de visualiser l'exécution de commande et les éventuelles erreurs qui pourrai remonter.

Commented [MB9]: 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.)

15.5. Sécurité

Le wifi du Raspberry sera constamment ouvert sur deux slots différents. Le premier permettra aux joueurs de se connecter afin d'accéder à l'assistance de jeu. Celui-ci sera constamment ouvert et ne communiquera qu'avec l'application Android. Il n'écoutera aucune commande autre que celles que peut renvoyer l'application afin de protéger le système.

Le second slot, qui sera non visible, permettra à toute personne connaissant l'adresse et le mot de passe de se connecter au Raspberry afin de pouvoir administrer le système si besoin.

Commented [MB10]: 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.**).

15.6. Intégrité

Dans le cas de déconnexion imprévu de l'application Android il suffit simplement de se reconnecter via le gestionnaire de wifi du smartphone.

Et dans le cas où le système dysfonctionnerait deux possibilités seront en place :

- Connexion via le wifi d'administration du système afin de pouvoir effectuer les corrections nécessaires selon les erreurs des fichiers de logs ;
- Relance complète du système en débranchant le Raspberry puis en le rebranchant.

Commented [MB11]: 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 ?

16. Maintenance et évolution du système

Toutes maintenances au niveau du Raspberry devront être faites via l'accès wifi d'administration du système en modifiant les fichiers de code nécessaire puis en les recompilant sous GCC pour les remplacer sur le Raspberry. Celles au niveau de l'application Android devront se faire sous forme de mise à jour de l'application via le Play Store de Google.

Commented [MB12]: Préciser les contraintes liées aux procédures de maintenance :
- Curative ou corrective ;
- Adaptative ;
- Évolutive du système ;
- Perfective