
Analyse

Robot joueur de billard

DESSAULX - CAPAR - RETTER - LATH

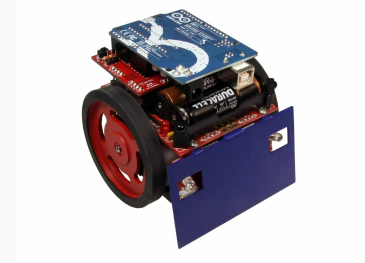
Tuteur : Pierre-André GUENEGO

Sommaire

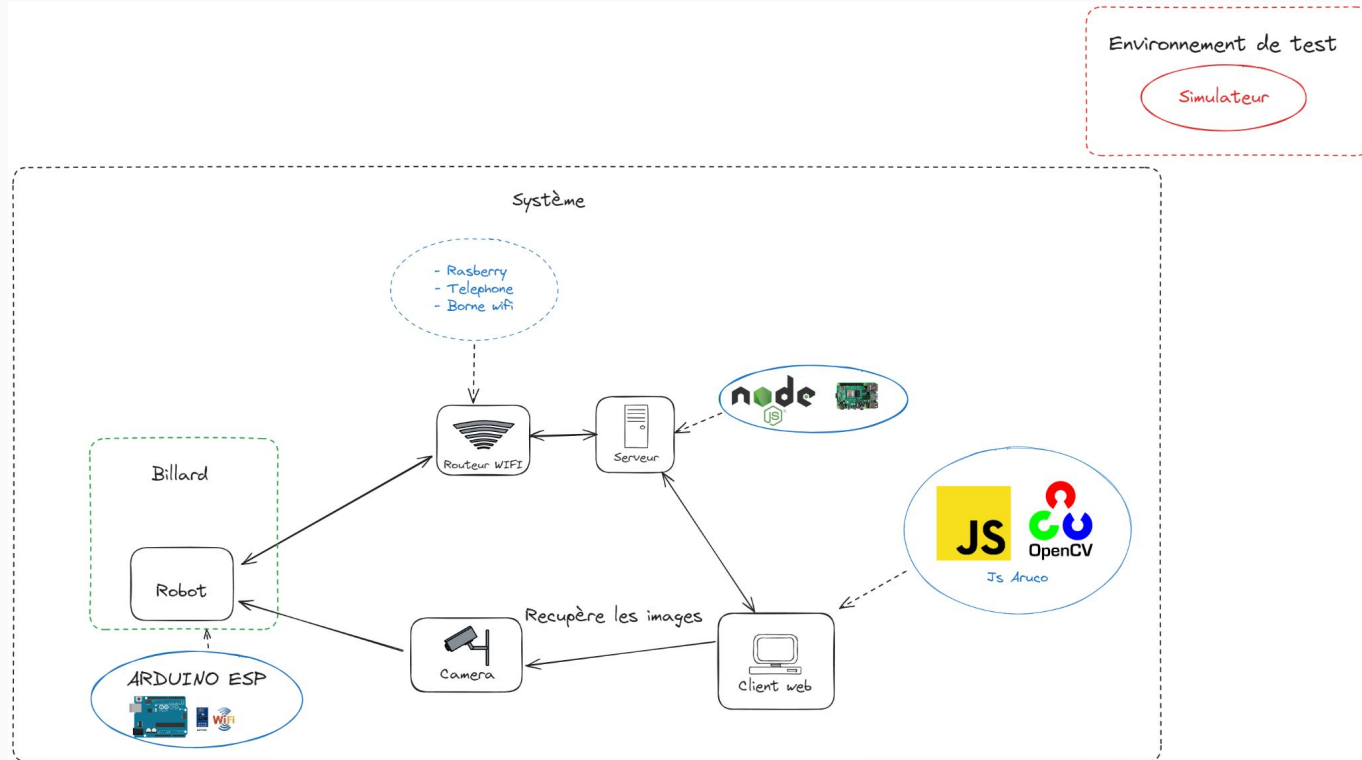
1. Présentation du projet
2. Architecture
3. Risques et solutions
4. Fonctionnalités
5. Diagrammes
6. Maquettes
7. Diagramme de classe simulateur
8. Stratégies
9. Planning

Présentation du projet

- Développer un robot autonome qui joue au billard.
- Un robot qui roule et sera équipé d'un marqueur (Aruco)
- Le robot sera guidé avec une caméra fixée au plafond.
- Un serveur web qui gère la communication entre tous les éléments pour pouvoir commander le robot.



Architecture du projet



Risques et solutions

Les risques et solutions

- L'ESP ne capte pas le signal (Message en retard)
=> Effectuer un mouvement pour un temps donnée.
- Le robot ne va pas dans la direction souhaitée.
=> Faire un calibrage au démarrage de la partie (autocalibrage).
- Une boule dérange le déplacement du robot.
=> Décaler celui qui dérange ou bien contourner la boule.
- Problème de luminosité : table trop ou pas assez éclairée. Le système de reconnaissance peut confondre les boules aussi.
=> Gérer l'éclairage / manipuler l'image pour accentuer certaines couleurs.
- Boule qui disparaît au cours du jeu.
=> Mettre les boules en tampon mémoire.

Les fonctionnalités

Les fonctionnalités

Fonctionnalités serveur :

- Lancer serveur
- Vérifier que tout est connecté
- Mettre à jour les coordonnées des boules
- Mettre à jour les coordonnées du robot
- Calibrer le robot
- Calculer la trajectoire du robot
- Donner les ordres pivoter au robot (avec date)
- Donner les ordres avancer au robot (avec date)
- Vérifier fin de jeu
- Manipuler robot manuellement

Les fonctionnalités

Fonctionnalités client web :

- Se connecter au serveur
- Détecter les boules
- Détecter le robot
- Reconnaître les boules
- Envoyer coordonnées boules
- Envoyer coordonnées robot

Fonctionnalités Arduino :

- Se connecter au serveur
- Vérifier heure du message
- Avancer
- Pivoter

Fonctionnalités Simulateur :

- Affichage des objets
- Affichage mouvements
- Calcul : trajectoire, rebond,..
- Avancer le robot
- Pivoter le robot
- Manipuler robot manuellement

Les diagrammes

Diagramme de séquence

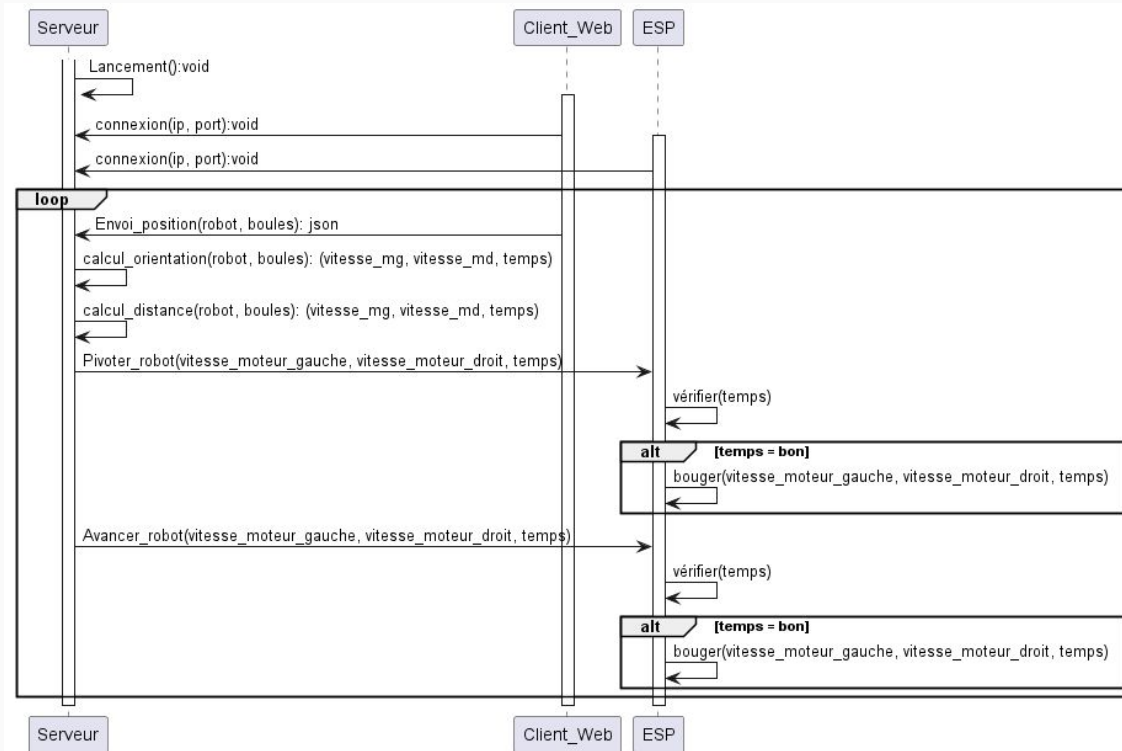


Diagramme de séquence

Diagramme de séquence : calibrer le robot

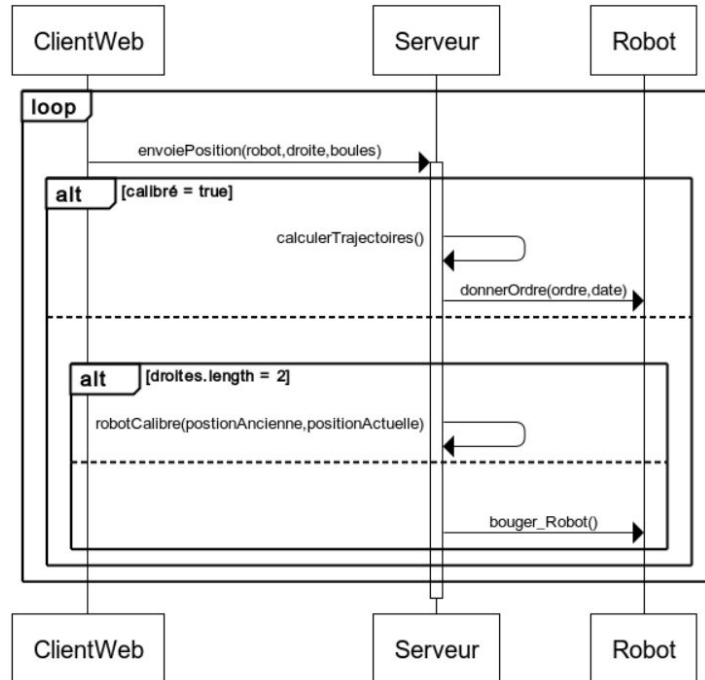


Diagramme Use Case du robot

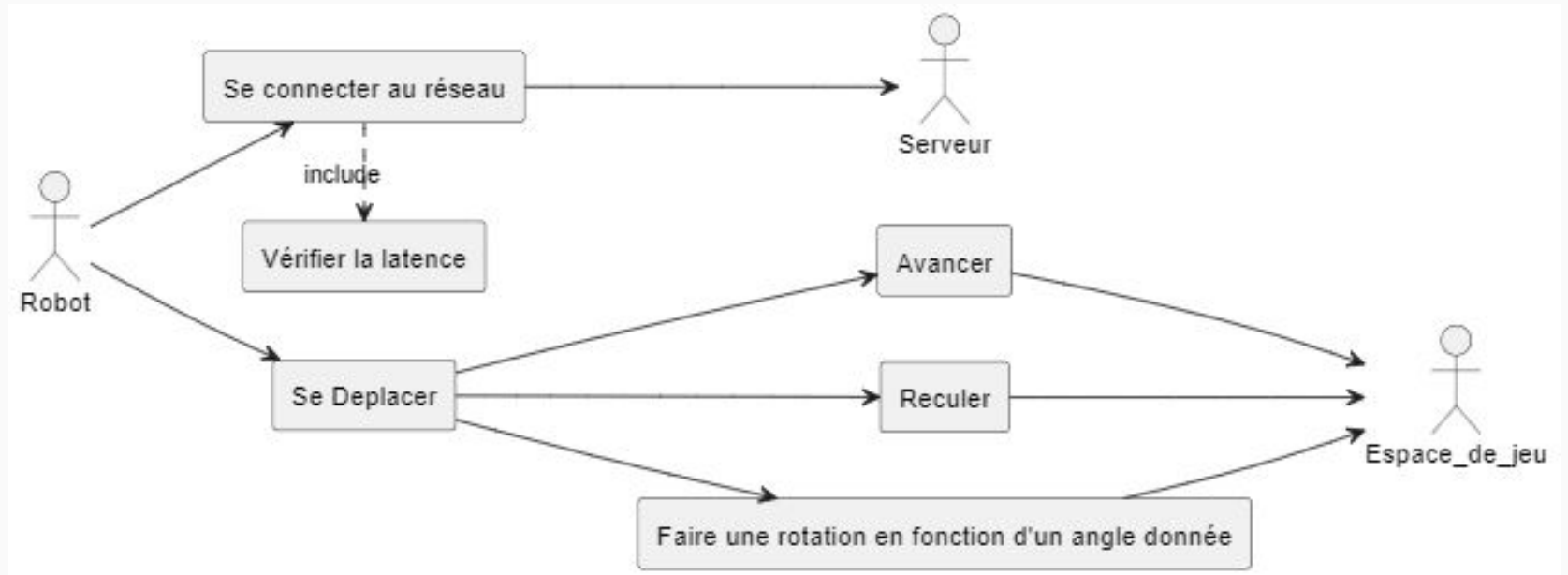


Diagramme d'état de l'ESP

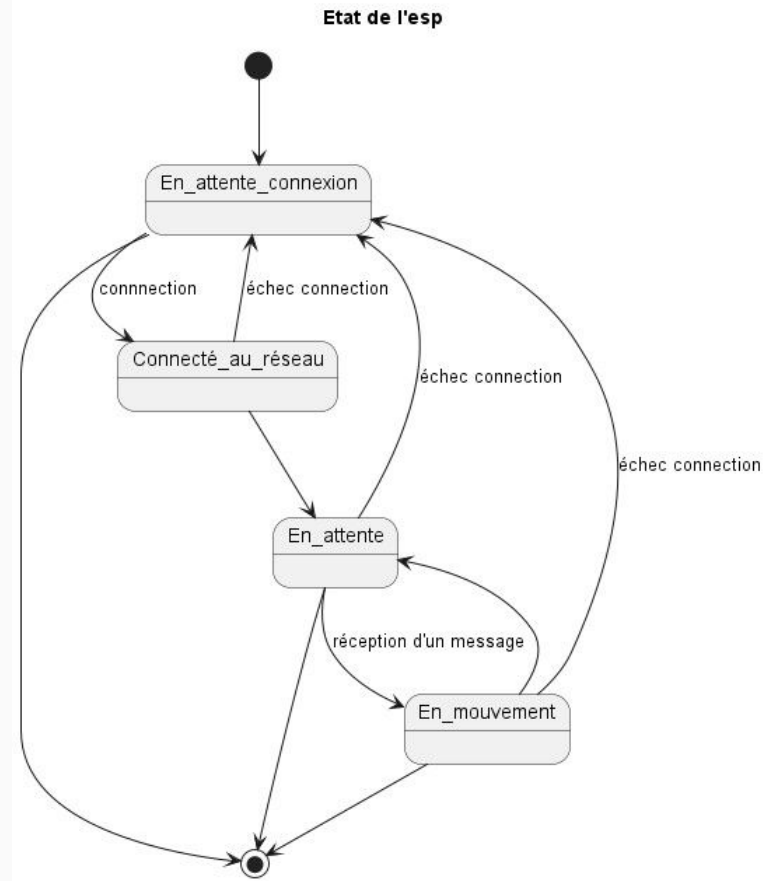
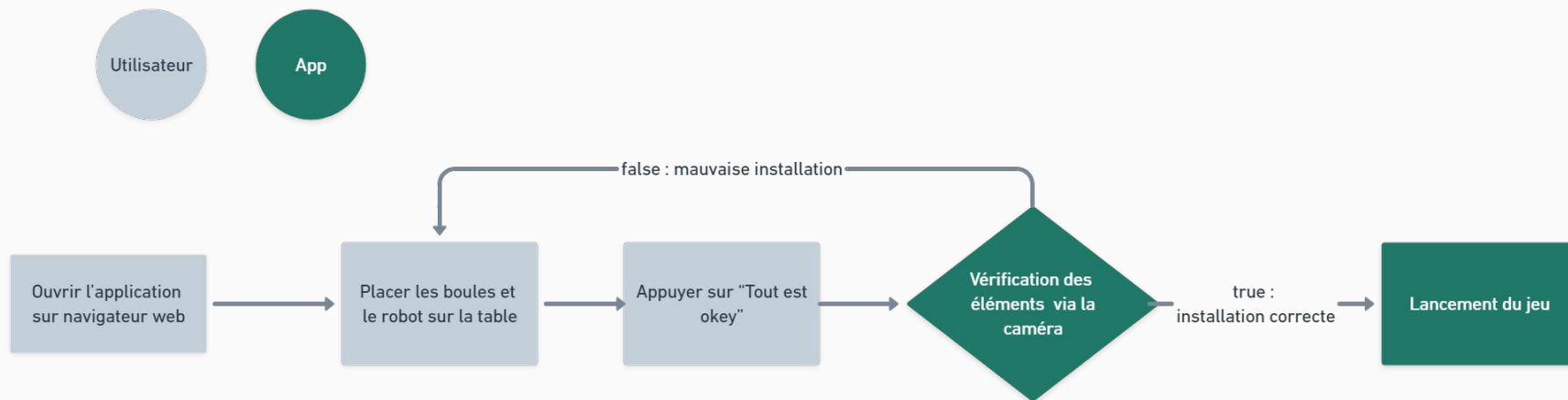


Diagramme d'activité



Made with Whimsical

Diagramme d'activité

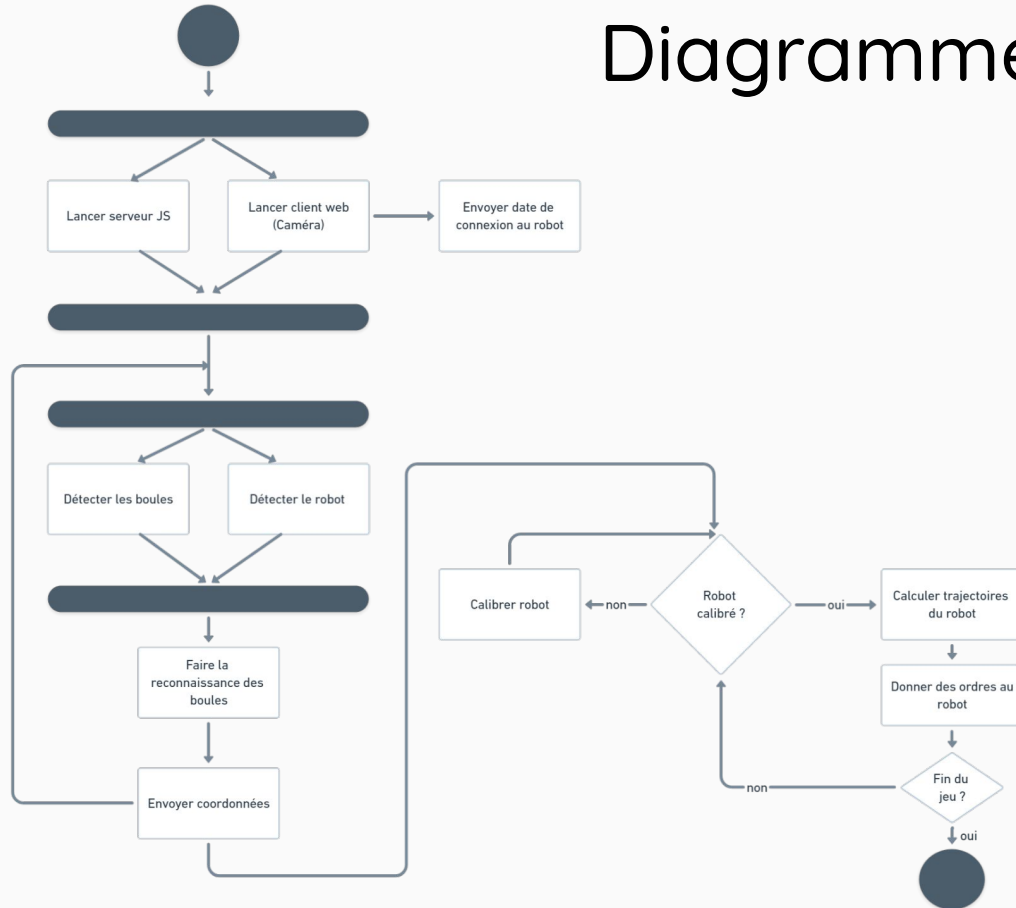
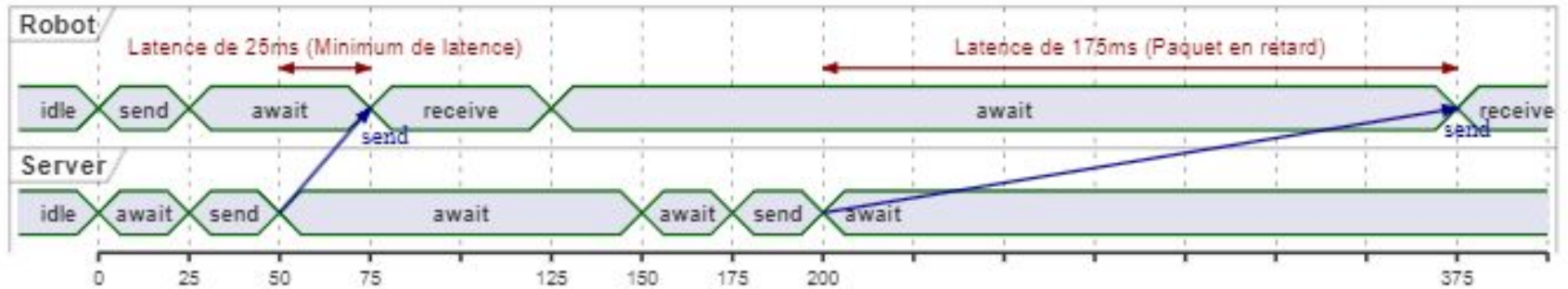
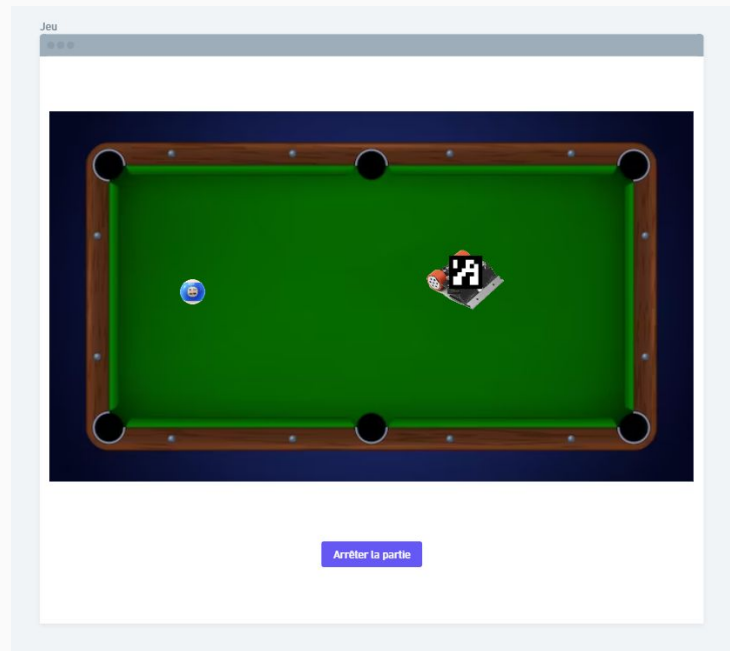
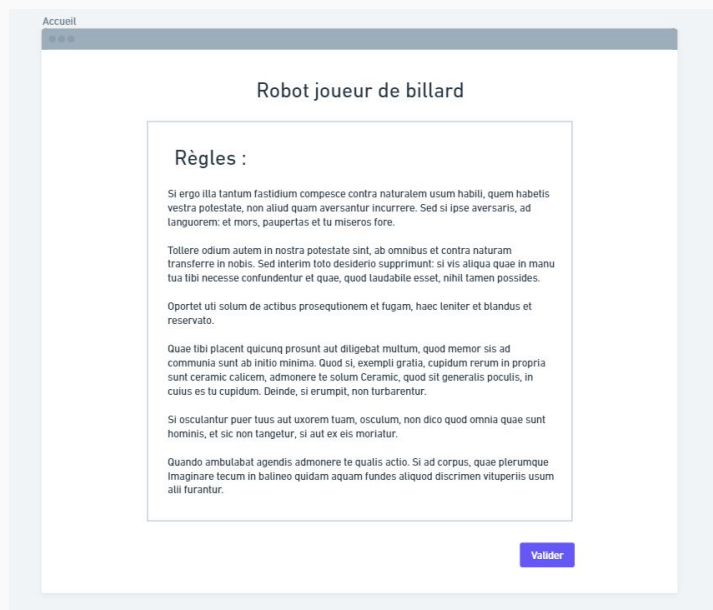


Diagramme de temps : Latence



Maquettes

Maquette de l'interface user



Maquette de l'interface user

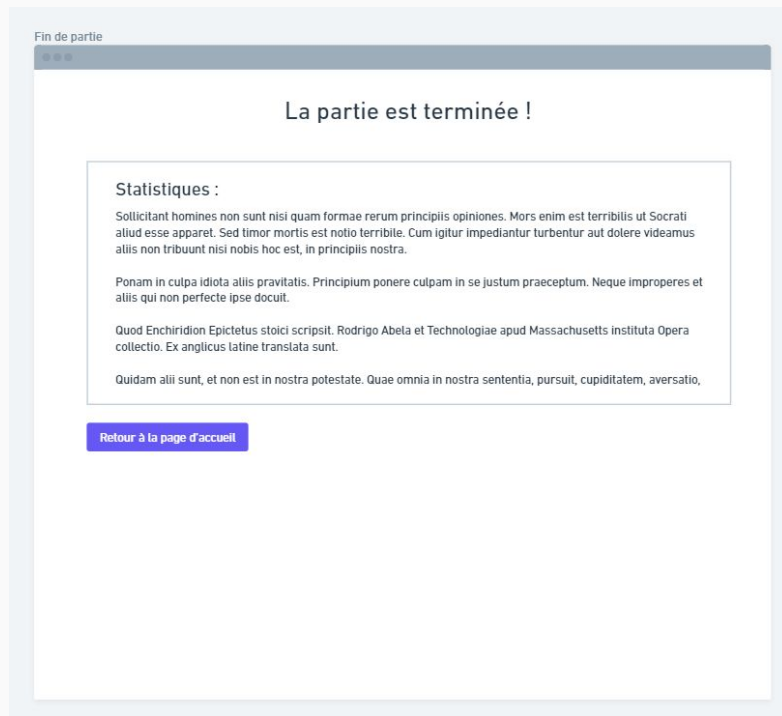
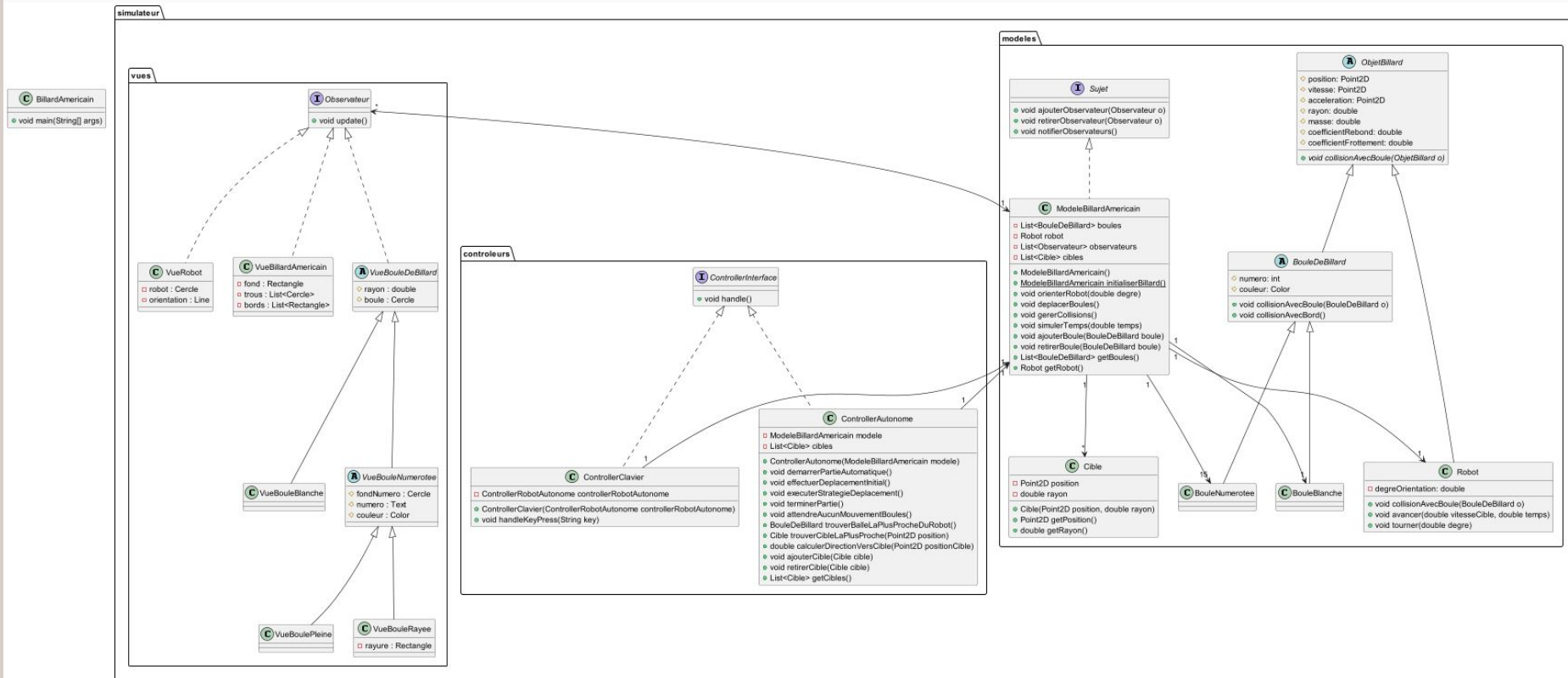


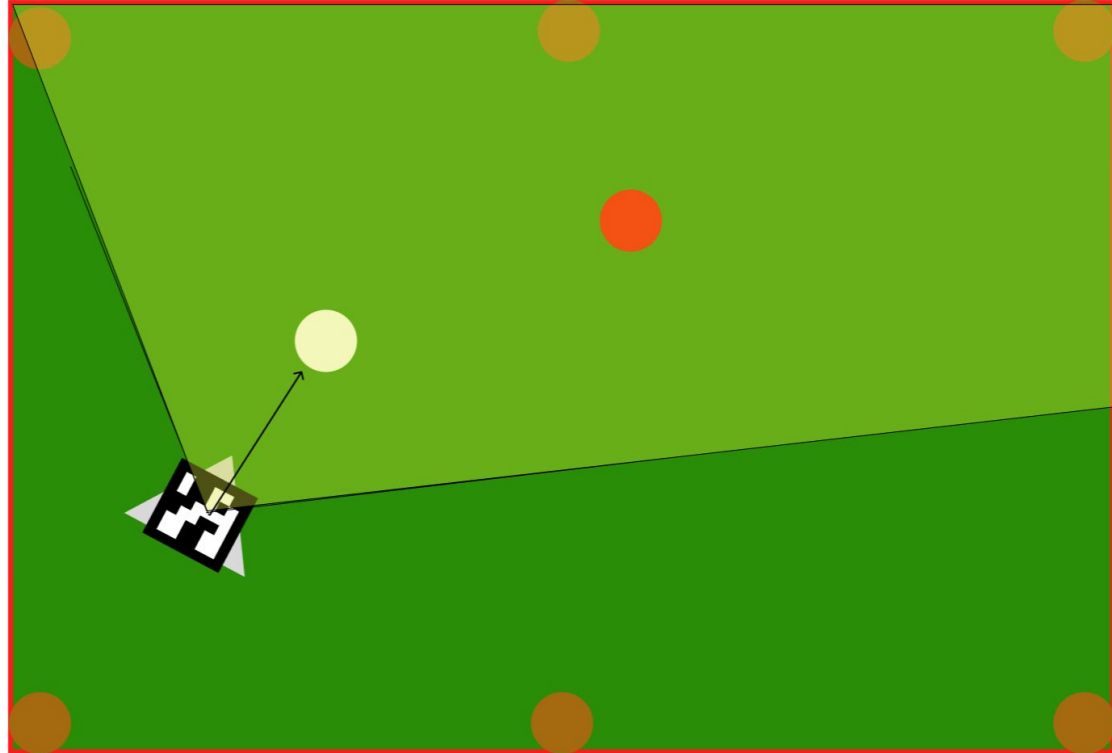
Diagramme de classe simulateur

Diagramme de classe



Stratégies

Stratégie pour gagner une partie de billard



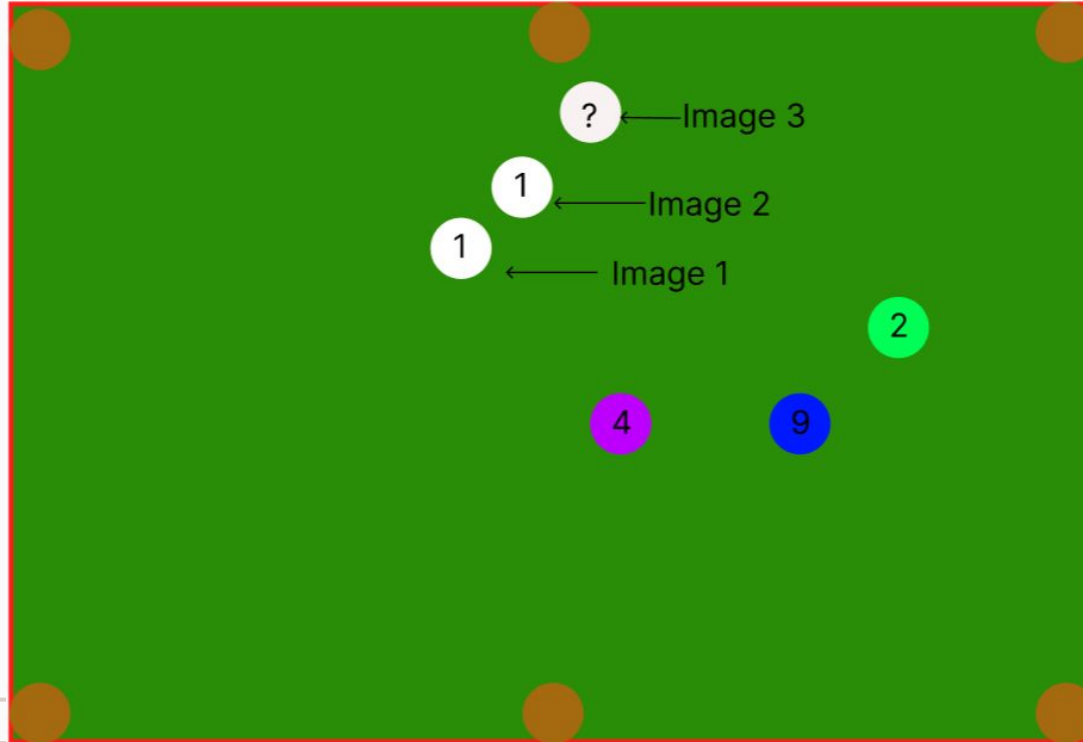


Stratégie pour gagner une partie de billard

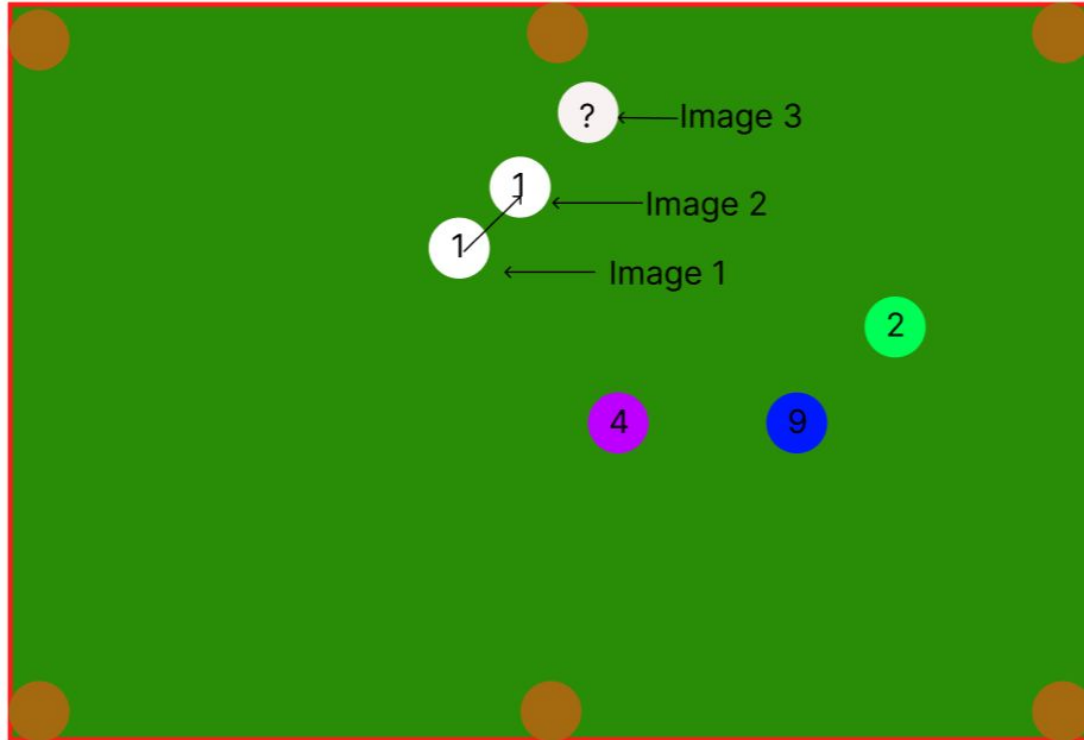
Algorithme naïf pour gagner une partie de billard :

```
meilleur_trajectoire <- trajectoire (angle_initial,vitesse_initial)
nombre de boules rentrées <- 0
pour chaque angle du robot
  pour chaque vitesse entre n et p
    nb_boules <- trajectoire(angle,vitesse).resultat()
    si nombre de boules rentrées > nb_boules
      meilleur_trajectoire <- trajectoire (angle,vitesse)
      nombre de boules rentrées <- nb_boules
retourne meilleur_trajectoire
```

Stratégie pour la reconnaissance des boules et du robot



Stratégie pour la reconnaissance des boules et du robot



Stratégie pour la reconnaissance des boules et du robot

Si le tracking n'est pas suffisant :

- TensorflowJs pour créer et entraîner un modèle de réseaux de neurones convolutifs
- Objectif : faciliter la reconnaissance des boules
- Permet d'identifier les boules avec leur numéros



Les itérations

Les Itérations

Itération 1

- Création du serveur
- Création du Client JS
- Mise en place de la connexion : client / robot via les websockets.
- Détection des boules et l'aruco
- Pilotage du robot à l'aide de l'ESP
- Développement du simulateur

Itération 2

- Algorithme de calcul de mouvement
- Calibrage du robot
- Différencier les boules (tracking ou/et réseau de neurone via tensorflow.js)

Itération 3

- Gestion de la latence
- Création de l'interface utilisateur
- Robot autonome
- Ajout de la gestion de la friction dans le simulateur
- Automatisation du démarrage du système

Itération 4

- Prévisualisation des trajectoires (dans l'interface du client js)
- Test & optimisations des algorithmes
- Déploiement du serveur sur un Raspberry pi
- Création d'autres modes de jeux.