



# Échecs, entre calcul et « calcul »

Bilal SEDDIKI et Alban LE JEUNE

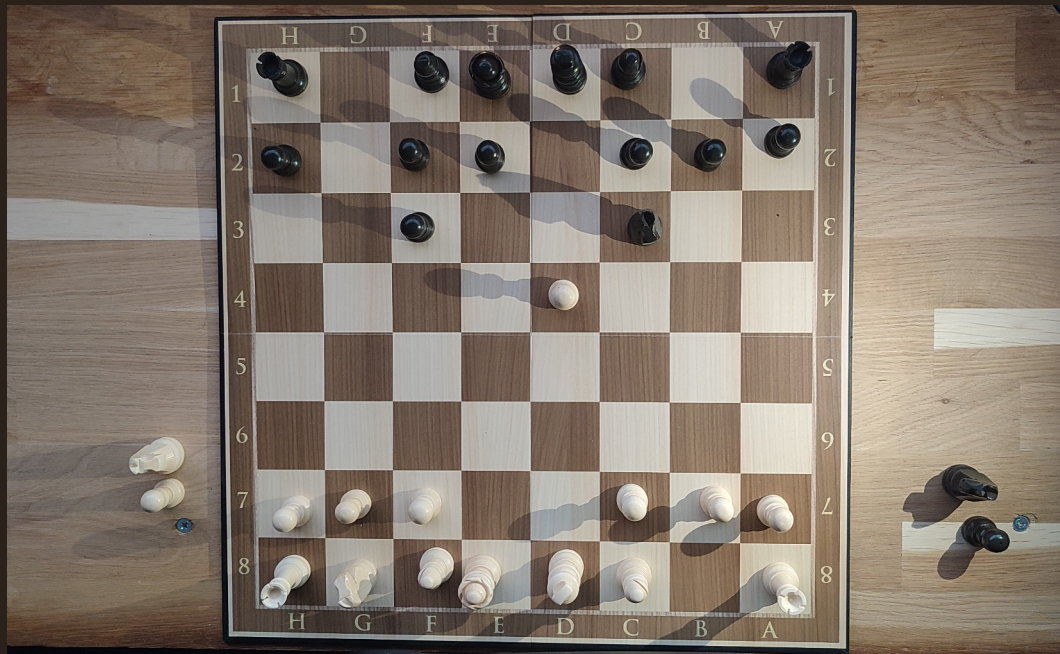
# Objectif Initial

- Rebondir sur l'actualité récente du monde des échecs
- Apprendre dans les domaines de l'IA
- Revisiter un domaine que l'on apprécie



Hans Niemann, Grand maître international accusé de tricherie

# Les problématiques



- Apprendre à entraîner une IA
- Trouver/Construire un Dataset
- Savoir dans quelle direction aller
- Pré-Traitement de l'échiquier
- Manipuler une IA d'échec

Exemple d'image du Dataset

# Ce que le projet fait concrètement

- Parse un échiquier à partir d'une photo
- Transmission au serveur + analyse
- Transmission du coup optimal au téléphone du joueur

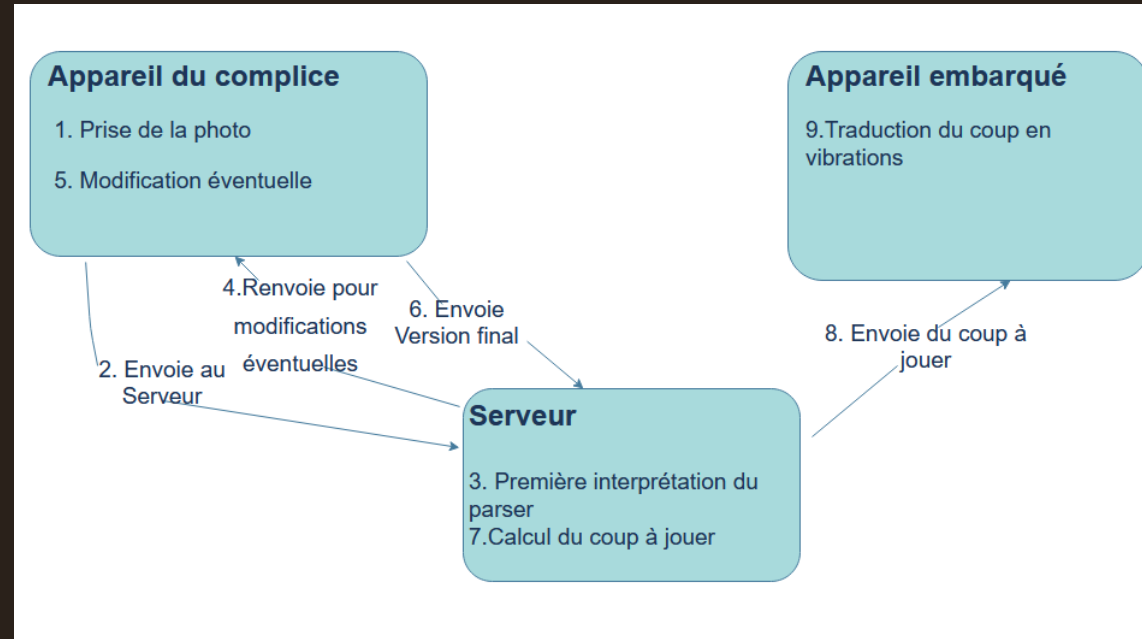


Diagramme de fonctionnement du projet

# Principaux scenarii du projet

## Scenarii favorables

- Parsing = Position réel
- Pas de déconnexion
- Pas de fouille

## Scenarii défavorables

- Perte de connexion
- Besoin de fixer les imprécisions du parser
- Objet embarqué découvert



# Architecture et concept du projet

# Choix de conception/Décomposition

- Python, un choix évident
  - Développement mobile, moins un choix
- I Pré-traitement des images d'échiquier
  - II Apprentissage de l'IA
  - III Communication entre les différents appareils



# Modules principaux

- Module Parsing
- Module IA
- Module Réseau
- Module Mobile

# Compétences techniques

- Langages :  
Python et Kotlin
- Réseau avec  
réception et  
envoi de packets
- Tensorflow



# Principales difficultés

- La pluralité des solutions
- Choix d'un modèle + construction dataset
- Multiples étapes et donc multiples problèmes
- Retouche d'image nécessaire au pré-traitement
- Pré-traiter ou machine learning ?

# Répartition du travail/Difficultés

Bilal

- Prétraitement
- Machine-learning
- Serveur

Alban

- Echeecs
- Manipulation IA
- Outil dissimulé



# Programmation

# Algorithme de construction de dataset

**Entrée :** fichier .pgn, dossier d'image

```
Pour chaque image dans le dossier d'image,  
  Pré-traiter l'image  
  Si le pré-traitement a réussi,  
    Couper l'image en 64 images pour chaque case  
    Pour chacune des 64 images,  
      Récupérer le type de la pièce via le fichier .pgn  
      Sauvegarder l'image dans le dossier corresp.  
  Passer à l'état suivant du fichier .pgn
```

- A partir de n photos d'une partie, permet d'obtenir  $n * 64$  images pour notre dataset.

# Algorithme de pré-traitement

**Entrée :** Une Image

**Algorithme :** Récupération des 64 \* 4 coordonnées des coins des cases de l'échiquier.  
Distorsion de l'image pour ne contenir que l'échiquier grâce aux coordonnées des coins.  
Reconnaissance des cases noirs et blanches  
Création de **l'output** des cases  
Boucle for pour toutes les coordonnées (donc 64)  
Association d'une case avec une couleur  
**Si** le résultat == 64,  
|retourner **l'output**;  
}

Exemple d'algorithme présent dans le projet

- Travail important de prétraitement
- Nécessité d'être précis pour aider au maximum l'IA de reconnaissance

# Testabilité

- Batterie de tests sur le modèle
- Modèle éprouvé et donc performant
- Utilisation de StockFish pour garantir l'envoi du meilleur coup (pas d'erreur possible)

# Limites de la testabilité

- Angles de vue limités
- Limites sur l'échiquier
- Obligation d'ajout de correction



Exemple d'image ne pouvant pas être parsé



# Conclusion

- Les leçons retenus
- Une Version 2.0 ?
- Des choses à refaire ? À faire différemment



Une idée pour la V2...

Fin de la présentation

Merci de l'avoir suivi