

PROJET FINAL : €-ROBOTS

Répartition des parties

Partie 1 : Reconnaissance de pièce

Arman – Alexandre – Sébastien

Partie 2 : Optimisation

Loann – Romain – Pierre-Alexandre

Partie 3 : Tests et site web

Guillaume – Irvin

Responsables d'équipe en gras

Travail à effectuer

Equipe 1 : Reconnaissance de pièce

Base de données :

- Alimenter la base de données créée par l'équipe web (Guillaume)

Arman – Alexandre – Sébastien

Arman : Photo : format ? jpg, png, ... => demande 720p

Seules consignes : couleur, prise de vue verticale, pile/face, distance : pas trop près (voir avec Arman)

Test de saturation plutôt que la valeur en N&B pour reconnaissance de pièces, le contour s'en détache bien. A voir avec les Algo.

Alex :

Creation Class pièces + vecteur de piece + générateur d'un vecteur de pièce

Sébastien : poursuit les recherches avec son algo et prog de stage

Base de saturation : Arman

Temps de calcul : long ? A voir.

-> ca nous permet d'avoir une marge de manœuvre

=> « **C'est l'idée, bon chemin** »

Chaque point du cercle est le centre du cercle, les cercles vont avoir un point commun : centre du cercle : en trois points (Proposition de M. Kervazo, gain de temps probable)

Seb : veut trouver un cercle sur l'image filtrée. Image de couleur, transformation en N&B puis détection : se recoupe avec la méthode proposée par Arman. Seb va se pencher sur la

reconnaissance de la valeur des pièces avec un principe de comparaison d'images

Arman : Bruitage supprimée par blur et threshold. le floutage et calcul pour reduire le nombre de points.

Bibliothèque de connexion PDO plutôt que MySQLi (gain en sécurité)

Se forment sur la reconnaissance

Alex a commencé la classe **Piece**, différentes valeurs **value** et **currency** pour la devise.

Vitesse à prendre en compte

Travail sur la couleur par Seb.

Transmettent un vecteur de pièces à l'équipe 2.

Première solution : noir et blanc

Haar Cascade :

Sebastien part sur une cascade d'images, reconnaissance d'yeux, base de 2000 photos en open source. Adaptation pour les pièces nécessaire.

Algorithme de Machine Learning pour la détection de pièces, à voir si HC intègre la couleur.

Le gros du travail porte sur la constitution de la BDD d'images positif et négatif, et le ML.

Test sur le serveur d'Alex.

Arman : débuge en cours.

Alex : algos de tracking commencés. Doit voir avec Irvin pour enregistrement du projet sur serveur, et test sur serveur ??

Equipe 2 : Optimisation

Romain, PA : Algo « voyageur de commerce », Algo heuristiques

Loann : Algo valeur (glouton)

Optimisation valeur.

Meilleure possibilité en fonction des valeurs données pour constituer la somme

Voir algorithmes gloutons... par exemple 😊

Implémentation, test de tris par valeur :

PA : Algorithme de shell

Romain : algorithme de Heap Sort

Loann : Algorithme Merge

Optimisation du trajet

Trajet le plus court pour constituer la somme

Réfléchir aux limites des solutions : mettre aussi les parcours de Loann

Documenter les algo heuristiques -> **envoi au prof**

Travelling salesman solution

Trajet : Exponentiel

Equivalent du Dijkstra : Algorithme le plus fiable, mais assez lourd.

Trouver un algo moins lourd (mais moins fiable) à étudier en 2^{nde} solution.

3 algo : OK

Sélection de pièce qui marche (en fonction d'une somme donnée), temps

Objectif : partir sur un trajet

Loann part sur l'insertion de trajet. ($n^2 \log n$).

Autre algo sur voisin le plus proche : PA

(3^e algo : à voir)

Affichage du résultat sur openCV, évolution du chemin et choix des pièces et du parcours.

Equipe 3 : Tests et site web

Site web :

Symfony : Guillaume travaille dessus ce week-end, point lundi avec PA, on voit si on tente le développement du site en Symfony.



- Création d'une page d'accueil : à faire dans un second temps, pages séparées
 - intro du projet -> PA
- page de connexion
 - page inscription : pas de travail graphique, pas de temps à passer sur cette page
- page d'upload d'images et d'informations sur la pièce
 - Voir pour le Drag&Drop
 - Espace User, connexion par mot de passe
- page perso
- Page de comptes-rendus

Base de données :

- Table pièce :
 - Chemin image (url interne)
 - ID pièce
 - Diamètre
- Table Devise :
 - ID devise
 - Nom

Irvin : doit faire le site en Mockup sur Balsamic puis export en pdf avec les liens

Guillaume : point lundi sur Symfony page d'upload

- Récupèrent les photos
 - Photos saturées plus ou moins
 - > faire des tests faciles, les prendre toutes côté pile pour commencer.

5 tables :

- 1 devise
- 1Pièces
- 1image de pieces
- 1 utilisateurs
- 1 fichiers

Comptes admin et guest -> fait

Upload : une base de données -> ?

Conseils de M. Kervazo:

- Plusieurs luminosités
- Symfony : option validée par le prof

Pour tous :

- télécharger la librairie/bibliothèque OpenCV
- Prendre des photos, à envoyer @Irvin et @Guillaume pour remplissage de la base de données et réalisation des essais de reconnaissance de pièces (Equipe 1)

Seules consignes : couleur, prise de vue verticale, pile/face, distance : pas trop près (voir avec Arman)

- Rester MOTIVÉ, sinon, aller chercher la motivation auprès des membres de SON ÉQUIPE, on ne peut avoir confiance qu'en ses coéquipiers... sauf un.
- Guetter **Slack** et les échanges internes
- Poser des questions sans hésiter, tout temps d'hésitation est un temps perdu, et il ne nous restera pas une minute à la fin, autant la prendre au début.
Alex et moi saurons être dispo à toute heure du jour et de la nuit ! Bon, pour la nuit, vous pouvez hésiter pour les questions... la nuit porte conseil, attendez le matin !

Informations générales

- Pour la partie programmation :
 - les uploads / downloads se feront via Git,
 - Alexandre se charge du tuto et des règles d'upload (formatage du commit)
- On se cantonne à la monnaie européenne
- On part du principe que le robot gère la vitesse du tapis, qu'il n'y a donc pas à la prendre en compte.

Certaines informations peuvent ne pas vous sembler exhaustives, pour les demandes complémentaires.... → SLACK ! ;-)

CR du 30 avril 2018.