Pistes d’amélioration du script billard

# Choix de la caméra

camera = cv2.VideoCapture(n), n n’est pas le même sur tous les PC : il faudrait un script pour sélectionner les caméras à utiliser.

# Positionnement de la table

Le rectangle pour positionner la table a été calibré pour faire tenir notre table de billard, avec une position du vidéoprojecteur définit et la table de billard à une hauteur définie. La calibration faite ne permet pas de s’adapter à toute les tables. D’autant que pour beaucoup, elles sont fixes et le projecteur l’est aussi.

L’idée est donc de changer radicalement l’algorithme de positionnement de la table. Voici à quoi il devrait ressembler : on affiche un rectangle dont les dimensions sont lues depuis le fichier « AllData.json ». Si il ne correspond pas à la taille de la table, l’utilisateur pourra, après avoir pressé « r », cliquer sur l’écran pour sélectionner les deux angles extrêmes du billard et redimensionner le rectangle. Comme précédemment il pourra quitter cette étape lorsqu’il sera satisfait avec « q ». Les dimensions du nouveau rectangle seront alors enregistrées dans le fichier « AllData.json ».

# Ajout d’une fonction de visualisation des coups

En parallèle de la possibilité de rejouer des coups, on pourrait créer un mode de visualisation des coups. On commence par choisir un coup avec le mode 1 mais on est ramené à un mode de visualisation seulement. Possibilité de revoir le coup avec « r », de changer de fichier avec les flèches directionnelles ou en appuyant sur « 1 » (retour au mode 1, choix du fichier par son numéro), ou de revenir au mode 0 avec « 0 » .

# Possibilité de supprimer des coups enregistrés

La liste des coups enregistrés a une taille maximale défini et prend de la place en mémoire. On pourrait rajouter la possibilité de supprimer un des coups enregistré. Dans le mode visualisation ?

# Prévisualisation au moment de choisir un coup

Choisir un coup par son numéro n’est pas très orienté utilisateur. Il serait intéressant de modifier le mode 1 ainsi. Choix d’un coup par son numéro, prévisualisation, validation avec « y » ou choix d’un autre coup avec « n ».

# Possibilité de redéfinir les bornes de recherche de couleur dans l’espace HSV

La détection des billes fonctionne par recherche de couleur, et les valeurs choisies ont été déterminées dans les condition de la salle Amigo. Il serait bon de laisser la possibilité à l’utilisateur de redéfinir au besoin l’espace de recherche des couleurs. Pour cela il faut utiliser le même code que le fichier range\_detector\_video.py et sauvegarder les valeurs dans un fichier json.

# Algorithme fonctionnant avec plusieurs boules

La grosse amélioration à faire : gérer plusieurs billes en même temps. Cela suppose de redéfinir les conditions de début et de fin de trajectoire (au moins une bille bouge ou toute à l’arrêt), la manière dont les trajectoires sont stockées mais aussi l’algorithme de tracking. Hypothèse nécessaire sur le type de bille : boules numérotées, pratique pour le plaçage des billes en mode rejeu par ailleurs mais difficile de définir une plage de valeur de recherche (bille non uniforme), ou boules uniforme (jaunes ou rouges) mais comment les distinguer (mémoire des positions ?).