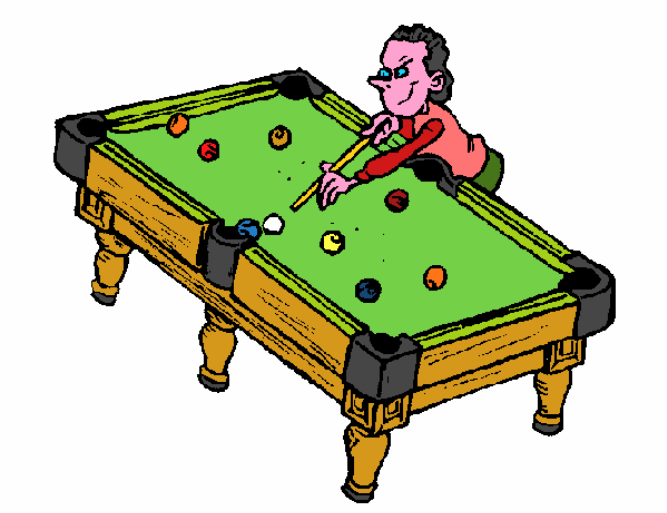
Programmation pour ingénieur

PROJET :

Analyse d'une partie de billard Français (Documentation)

Alexandre VALLET (341212)
Mahé VELAY (345882)

Plateforme : MacOs



I. Plateforme/Compilateurs Utilisées

- **Plateforme** : MacOs
- **Compilateur C** : Visual studio Code
- **Compilateur Matlab** : MATLAB_R2022b
- **Compilateur LabView** : LabView(21)

II. Nom des fichiers

- **Pix2Pos.c** : programme C
- **Pix2Pos** : executable du programme C
- **AnalyseTX.m** : fichier Matlab créer pour la séquence TX (X=1,2,…9)
- **ScoreSheetTX.pdf** : score sheet de la partie pour la séquence TX (X=1,2,…9)
- **Billard2023.vi** : vi principal du projet
- **GetBox.vi** : sub vi
- **CallC.vi** : sub vi
- **SaveML.vi** : sub vi
- **MP_LaunchMatlabScript3.vi** : sub vi fournit
- **pixmap.bin** ; **pos.txt** : fichier créer lors de l'execution du projet

III. Analyse du projet

1. Programme C

1.1 Description des étapes

On commence par récupérer les 30 arguments

Étape 1, 2 et 3
On crée le fichiers Pos.txt et on ouvre le fichier Pixmap.bin.
On récupère les deux premières données du fichier Pixmap.bin qui correspondent à la largeur et la hauteur. On crée un espace mémoire à l'aide de malloc, puis on récupère les données des pixels dont le nombre est égal à la largeur fois la longueur pour avoir le bon nombre.

Étape 4
On crée deux nouvelles variables ll et hh qui correspondent à la largeur et la hauteur de la table de billard et une structure Color qui permet de diviser une variable en 3 parties (.w pour le blanc, .y pour le jaune et .r pour le rouge).
On crée un tableau Pix en deux dimensions de type struct Color qui est initialisé à 0. On cible uniquement les pixels correspondant à la table de billard et à l'aide de la fonction findcolor on remplit chacune de ses cases correspondant aux coordonnées (x ; y) du pixel en question par 1 si il peut être la couleur. (Ex Pix[0][0].r = 1 si le premier pixel peut être rouge et Pix[0][0].r =0 si il ne peut pas être rouge).
Ensuite on cherche les 3 balles (là ou le score est le plus élevé) et leurs coordonnées dans notre tableau Pix. On parcourt ainsi tout le tableau en faisait attention aux bordures car nous parcourons un paterne de taille ballsize x ballsize en partant du pixel en haut à gauche. A l'aide de la fonction colorball nous avons le score de la balle avec le premier pixel, dès qu'elle est supérieur au meilleur score obtenu jusque-là nous remplaçons ainsi le score et notons les coordonnées.
On repasse en coordonnées de l'image et on finit par noter les scores et coordonnées des 3 boules dans le fichier Pos.txt.

1.2. Description des Fonctions

Findcolor : il s'agit d'une fonction void qui a pour arguments les coordonnées du tableau Pix, le tableau Pix et toutes les bornes de chaque couleur. Elle a pour but d'attribuer 1 à la case du tableau donnée si celle-ci pour être de la couleur indiquée. Pour cela on sépare les données

correspondant aux 3 couleurs avec le shifting puis on compare celles-ci avec les bornes, si elles sont valides on attribue 1 à la case du tableau.

Colorball : il s'agit d'une fonction de type struct Color qui a pour arguments les coordonnées d'un pixel (celle en haut à gauche), le tableau Pix et le diamètre de la balle. Elle a pour but de calculer le score d'une balle. Il suffit ainsi de parcourir toutes les cases du tableau Pix à droite et en bas jusqu'à atteindre la longueur du diamètre en partant de la coordonnée en haut à gauche. Et d'additionner tout les scores de ses cases puisque q'ils sont égaux à 1 s'ils peuvent être de la bonne couleur et 0 sinon. Enfin, elle revoit un struct Color qui contient les scores de la balle pour les 3 couleurs.

1.3. Gestion des erreurs

Type d'erreur	Return code	Justification	Conséquences
ERREUR	1	Erreur lors de l'ouverture ou création des fichiers	Le programme affiche "Erreur lors de l'ouverture du fichier de lecture/écriture“ dans stderr.
ERREUR	2	la largeur ou la hauteur est faux (inférieur à 100 ou supérieur à 1000)	le programme affiche "erreur la largeur/hauteur n'est pas valide” dans stderr.
ERREUR	3	nombre d'argument n'est pas bon	Le programme affiche "il n'y a pas le nombre de paramètre dans la ligne de commande”dans stderr.
ERREUR	5	le diamètre de la balle est faux (inférieur à 5 ou supérieur à 20)	le programme affiche "Erreur le diamètre de la balle est faux” dans stderr.
ERREUR	6	le nombre de pixel lu est inférieur à la taille de l'image	le programme affiche "il y a pas assez de pixel” dans stderr.
ERREUR	11	Erreur lors de la création du tableau malloc	le programme affiche "erreur de lors de la création du tableau malloc” dans stderr.
WARNING	0 (Default)	il y a trop de pixels	Le programme affiche“Warning: trop de pixel.” dans stderr.
Warning	0 (Default)	le meilleur score d'une boule est inférieur à 15	Son score passe à 0 et ses coordonnées (-1;-1) et le message “il manque la boule jaune/rouge/blanche” est écrit dans le fichier stderr .

2. Programme Matlab

2.1. Description fonction

Description du main
On commence par changer les coordonnées en Y des 3 boules. Ensuite, on trouve les coordonnées du bord du billard avec la fonction “GetFrame”.
On traite directement le cas où une boule est manquante sur la table du billard tout au long de la partie (les coordonnées sont que des NaN). Si en effet la balle n'apparait pas, on remplace toute ses coordonnées par des 0 en X et en Y. Cela nous permettras d'avoir une condition sur le premier indice de cette balle (0). A l'aide de “InterpolateNan”, on remplace les cordonnées NaN par la valeur suivante et avec “RemoveOutlier”on remplace les coordonnées absurdes par interpolation (PS : si une balle est manquant sur toute la sequence on n'appelle pas interpolate NaN). On continue par calculer la distance en pixels parcourue par les 3 boules (stockés respectivement dans Ly, Lr et Lw).
On détermine l'ordre du premier mouvement des boules et le nombre de celle(s) qui ont bougé avec “GetBallMoveOrder” (1 correspond à la boule rouge, 2 la jaune et 3 la blanche).
On affiche le cadre du billard, la trajectoire des boules et les chocs avec les couleurs et polices demandées ainsi que le titre de l'image.

Ensuite, on sépare en 3 cas (suivant qu'elle boule a été tapé par le joueur) pour avoir dans "IdxTouch" les indices où la première balle touche le bord, dans "Ball" la couleur de la boule jouée et dans "nbrebond" le nombre de bord touché entre le choc de la deuxième et troisième boule. Enfin, nous vérifions si le joueur a gagné (si le nombre de ces chocs est supérieur ou égal à 3 et que les trois boules ont bougé). Puis, on affiche en bas de l'image toutes les informations demandées. Nous avons également rajouté la condition que si le premier indice en X est 0, c'est à dire que la balle n'est pas la au cours de la partie, on n'affiche alors ni la position initiale, ni le tracé de la boule.

3. LabView

3.1. Descriptions des vi / sub vi

Billard2023.vi
Ce vi est le coeur de notre Labview, c'est lui qui orchestre tout le reste. Il se déroule en plusieurs étapes :

(i) Un path menant a un dossier d'une sequence est rentré.
(ii) En appellant list Folder, nous avons une liste de nom de fichier .png correspondant a toutes les images de la séquence
(iii) Nous indexons une photo de cette liste pour l'injecter dans notre vi get box.
(iv) Nous rentrons alors dans une boucle for parcourant toutes les images de la liste afin d'appeler callC et d'en extraire les coordonnées des boules ainsi que d'afficher les images une par une.
(v) Nous rassemblons ces coordonnées dans un tableau de cluster a l'extérieur de la boucle for.
(vi) Nous entrons alors dans un case structure (dépendant de l'envie ou non de l'utilisateur de créer un score_sheet) et nous appelons alors script ML pour créer le fichier AnalyseTX.m et ensuite l'appeler.

Front Panel (Billard2023.vi)
Le front panel du vi projet LV est composé de 3 parties.

- Une interface permettant de voir l'evolution de la partie image par image, un path a entrer menant jusqu'au dossier d'une sequence « TX et la possibilité de choisir la couleur de la marque laissé par les chocs bords-boules dans le cas gagné ou perdu, le style du tracé des bales ainsi que si l'on veux créer le Matlab Score_Sheet.
- Une partie avec tout les arguments correspondant au couleurs des bales, du font bleu, du bord, du diamètre de la balle. Ces informations sont réglés pour ces valeurs par défaut mais il est possible de les changer avant l'execution du projet.
- Une partie comportant les messages d'erreur afficher dans les « standard error » en C et en Matlab ainsi que « l'error out » ressorti par le programme Labview. Composé d'un code, d'un status et d'un message d'erreur (s'il y a une erreur).

GetBox
Ce VI permet d'obtenir les positions minimales et maximales en X et Y du bord de la table de billard. Sachant que ces valeurs sont les mêmes pour toutes les photos d'une meme séquence, nous pouvons donc exécuter ce VI sur n'importe quelle photo.
Ce VI est très important sachant que sans lui nous ne pouvons pas appeler le programme C.
Ce vi a été réaliser en analysant une image a l'aide de son tableau pixmap en 24 bit. Pour ce faire Nous avons rentrez ce tableau dans une double boucle for permettant d'accéder a toute les positions X et Y. Nous nous sommes servis de la fonction colors permettant de séparer les pixels en les 3 couleurs pour les comparés avec les bornes pour laquelle la couleur est considéré comme appartenant au bord. Nous avons ensuite, si la couleur était la bonne, comparer l'indice i en X et Y avec les currents max et min enregistrés dans des shifts registers initialiser a 0 pour les mass et a la valeur de la largeur et de la longueur pour le min.

CallC
Ce VI permet d'appeler le code C permettant de ressorti les coordonnées des boules pour 1 image. Il se réalise en plusieurs étapes :

(i) il prend en arguments un tableau de pixels et le transforme en tableau 1D comportant au début la largeur et la longueur de ce dernier.
(ii) il crée le fichier Pixmap.bin qui sera ensuite appelé par le C.
(iii) il prend également en arguments les valeurs des bords de la table trouver à l'aide de GetBox.

(iv) il crée un string comportant les indices nécessaires pour appeler le C.
(v) Une fois le C appelé, il ouvre et scan les valeurs de Pos.txt afin d'en extraire les coordonnées.
(vi) avant de retourner les coordonnées, il remplace les valeurs "-1" par des « Nan »

SaveML
Ce VI permet de créer le texte Matlab afin de créer un fichier AnalyseTX.m qui sera lu par la suite. Il est composé d'une série de "string constant" restituant tout le code et de fonction permettant d'insérer les informations d'affichage que l'utilisateur contrôle.

3.2. Gestion d'erreur

- Si une erreur arrive lors de l'appel de toutes les images dans la boucle for une condition d'arrêt est mise pour arrêter cette boucle.
- Nous avons essayer de passez le fil d'erreur dans le maximum de vi utilisé pour pouvoir permettre de detecter la moindre erreur.
- Pour la gestion des erreurs du C dans le LabView, Nous nous sommes contentés d'injecter le code que renvoie le C ainsi que le standard error dans le fil d'erreur.
- Les erreurs de Matlab sont automatiquement inséré dans le fil d'erreur.
- Les erreurs si un vi ou une quelconques partie du projet n'appartiennent pas au dossier sera automatiquement gérer puisque nous utilisons un chemin absolu pour tout le LabView.
- Lors de la presence d'une image invalide dans le fichier d'une séquence est automatiquement géré par le C. En effet Puisque la largeur et la longueur ne sont pas respectée puisque l'image n'est pas la bonne, le C r'enverras donc une erreur qui stopperas automatiquement le programme.

III. Remarques et observations

- Si l'indice où la première boule tape la seconde ou dernière boule est le même que l'indice d'un rebond sur le bord alors on le comptabilise comme un rebond permettant au joueur de gagner (un rebond entre le choque avec la deuxième et troisième boules), Ceci intervient principalement pour le fichier T9.
- **ATTENTION** : Lors de l'execution de notre projet une erreur apparait du à un manque d'installation de police sur notre ordinateur.
- Le code C renvoie 8 warnings lors de sa compilation. Ces warnings ne sont pas pris en compte comme une erreur puisqu'il s'agit simplement de variables non utilisés.
- Les 2 warnings décrit dans la gestion d'erreur du programme C seront affichés dans l'indicator standard error C. Seulement puisque le code C est appelé autant de fois qu'il y a d'images dans la sequence, cet indicator ne sera pas visible a la fin de l'execution. Une situation possible aurait été de conserver toutes les valeurs de cet indicator dans un tableau de string mais cette solution n'aurait que encombré le front panel. De plus, puisque le programme marche même avec ces erreurs, nous avons jugé non nécessaire de le faire.
- Le cas concernant le fait qu'il y est 0 balles pendant toute la partie est géré en affichant rien sur le score sheet.
- Le cas d'une boule manquante tout au long de la partie impacte aussi la fonction GetFirstmoveIdx qui gère donc le cas ou le premier indice vaut 0 et initialise donc le vecteur de distance a 0.

IV. Répartition du travail

Nous nous sommes répartis le travail équitablement en permettant à chacun de toucher à toutes les différentes parties du projet. Ainsi les fonctions Matlab et C ainsi que les sub vi on était presque tous réalisés ensemble. Nous avons essayé au maximum de les faire au même moment pour que aucun de nous deux soit perdu pour la suite du projet. Nous avons également guidé Archibald LECOINTRE et Ines CHARDONNEL pour certaines parties du projet comme le C et le LabView.