



Génie Mécanique

Programmation pour ingénieurs

Rapport de projet : Billard

2025

TEO HALEVI 329561

NEVÒ MIRZAI HAMADANI 328344

Nom des fichiers et leur fonction ..	2	Description des algorithmes.....	5
Flux de données et fonctionnement général.....	4	Informations de compilation et versions.....	6
Gestion des erreurs	4		

Nom des fichiers et leur fonction

C			
ErrorFreadNbEOF	Checks for errors in reading a file using fread and exits with a stderror code if an error is detected.		
ErrorDim_width_height	Checks if the width and height parameters are within a valid range given as input. Exits the program with an error code if they are not valid.		
ErrorNbElements_commandline	Checks if the number of elements in the command line is valid. Exits with an error message if it is not.		
ErrorBounds_commandline	Checks if the input arguments for the dimensions of the table and RGB color limits are within the specified range. Exits with an error message if they are not.		
ErrorCalloc	Checks if a pointer returned by calloc is null. Exits with an error message if it is.		
ErrorFile_read_write	Checks if a file in reading or writing exists. Exits with an error message if it does not.		
ErrorNbElementsInPM	Checks if the number of pixels in an image is valid. Exits with an error (if the number of pixels is less than what should be) or warning message (if it is more) accordingly.		
WarnNoBallFound	Warns and returns a key code if a ball was not found based on the input score.		
CommandLine_ConvertToInt	Converts command line arguments from strings to integers, using strtol and returns an array of integers argv_int. Exits with error code if the function didn't manage to convert the string (error detected by errno).		
ReadPixmap	Reads aPixmap file and returns its content as a struct.		
pixel_to_rgb	Takes an integer pixel value and returns a struct containing the corresponding RGB values.		
is_a_pixel_in_color_range	Checks if a pixel is in range.		
group_pixel_in_color_range	Calculates the score of pixels within a specified color range in a given area of the pixmap.		
find_balls	Finds the positions and scores of red, yellow, and white balls in an image using color range detection.		
write_pos_score	Writes the color, the position and the score of each of the 3 balls to a file.		

MATLAB	
<code>GetBallMoveOrder</code>	Finds which ball moves and in which order.
<code>GetBallPathLength</code>	Returns the length of a path drawn by a ball, given the arrays of the coordinates of a ball X and Y.
<code>GetFirstMoveIdx</code>	Finds the index corresponding to the first move of the ball, i.e when the distance between the initial coordinate and the ball is bigger than MoveDistPx.
<code>GetFrame</code>	Defines the 4 values needed to create a frame around the paths of the 3 balls. It finds the absolute min and max of the X and the Y arrays.
<code>GetTouchIdx</code>	Finds the indexes of the shocks between the ball and the sides of the pool table (frame) using GetTouchPartialIdx.
<code>InterpolateNan</code>	Finds whether there are NaN value in X and Y and substitutes them with the next value.
<code>RemoveOutlier</code>	Finds the indexes that are outliers both in X and Y and replaces them with the previous value.
<code>Win</code>	Finds out if the player has won. Checks if there are 3 balls that moved, and if the first ball touched the pool table sides at least 3 times.

LabVIEW	
<code>Trouver_bornes</code>	Finds the bounds of the pool table by checking where are the smaller/bigger values that fall in the right RGB limits for the DarkBlue color using the SubVI_ChOOSE_left and SubVI_ChOOSE_right_bottom.
<code>Choose_left</code>	Chooses the right value for the left and top bounds. If the boolean is true (i.e if the RGB values are within bounds) and the X or Y value is smaller than the default value, it will output the new X or Y value.
<code>Choose_right_bottom</code>	Chooses the right value for the right and bottom bounds. If the boolean is true (i.e if the RGB values are within bounds) and the X or Y value is smaller than the default value, it will output the new X or Y value.
<code>display_image</code>	Opens and displays the image in the path given as input.
<code>open_image</code>	Opens the image in the path given as input, and outputs an array containing the width, the height and the pixmap of the image (in 1D).
<code>write_pixmap</code>	Takes an array (containing width, height and pixmap) to create pixmap.bin.
<code>Appel_programme_C</code>	Calls the C program using the parameters given as input. If the stderr is not empty, then it is merged with the error out of System Exec to create the final error out of the SubVI.
<code>lecture_pos</code>	Reads the text file of the position and score in the path given as input. Outputs the text file as a string.
<code>score_to_nan</code>	Checks if the score is equal to 0, if it is it changes the X and Y coordinates to NaN.
<code>coord_array_to_string</code>	Takes the Xr, Yr, Xy, Yy, Xw and Yw coordinates and converts their type into string to be put in the Matlab code.
<code>matlab_param_empty</code>	Verifies the parameter string given to Matlab, if it is not correct, it sets it a default value.
<code>matlab_script_strings</code>	Puts together all the strings (type of lines, markers, colors, coordinates, sequence name and script) that will make the final Matlab code.

Flux de données et fonctionnement général

Instructions d'exécution : avoir dans le même dossier **Billard2023.vi**, ses SubVIs et l'exécutable **Pix2Pos.exe**, ouvrir **Billard2023.vi**, choisir une séquence dans l'interface, choisir (avec le booléen "Create Matlab score sheet") si MATLAB doit être lancé pour analyser le jeu, choisir les paramètres d'affichage MATLAB si besoin, lancer le script. Si la case est cochée, le PDF **ScoreSheetTx.pdf** va se créer automatiquement dans le même dossier.

Billard2023.vi lit un dossier contenant des images *.png, et itère sur chaque image dans une boucle pour créer un binaire correspondant à chaque image. Il appelle le programme C **Pix2Pos.c** avec les paramètres de ligne de commande appropriés (couleurs, bords du billard).

Pour chaque image ouverte par LabVIEW :

1. **WritePixmap.vi** crée **pixmap.bin**, binaire correspondant à l'image.
2. **Appel_programme_C.vi** lance **Pix2Pos.exe**, qui ouvre le fichier **pixmap.bin** pour analyser les positions des boules.
3. Lorsque l'image est analysée sans erreur, **Pix2Pos.exe** crée **Pos.txt** contenant les coordonnées et score de chaque boule.
4. **lecture_pos.vi** récupère les valeurs de **Pos.txt**

Si l'utilisateur décide de lancer MATLAB, la liste de coordonnées de toutes les images du dossier est donnée à **coord_array_to_string.vi** qui transforme les coordonnées en strings. Les strings sont données à **matlab_script_strings.vi** qui crée le fichier **AnalyseTx.m** (où x est le numéro de séquence).

AnalyseTx.m crée le fichier pdf **ScoreSheetTx.pdf**.

Gestion des erreurs

Erreurs C

Nous avons suivi la convention donnée dans le tableau suivant pour les codes d'erreurs C (convention donnée dans le fichier **key_ErrWarn.h**) :

Reading	$1x$
Writing	$2x$
Pixmap	$3x$
Bounds	$4x$
System	$10x$

Erreurs LabVIEW

Le fil d'erreur définit l'ordre d'exécution du programme et lie les erreurs des subVIs. En cas d'erreur dans une étape du programme (image corrompue, fichier manquant, ...), LabVIEW s'arrête en donnant une erreur.

Description des algorithmes

Chaque programme (C, Matlab, SubVI) est documenté indépendamment de l'explication qui suit.

Programme Matlab

Le programme Matlab ([AnalyseTx.m](#)) est créé par LabVIEW en fonction des choix d'utilisateur pour l'affichage. L'utilisateur a le choix des couleurs des chocs balle-bande, ainsi que le type de ligne et de symbole.

Le programme Matlab va analyser les coordonnées de chaque boule pour définir si la partie a été gagnée ou non. Une partie est gagnée si les 3 boules ont bougée, et si au moins 3 bandes ont été touchées entre 2 chocs.

A partir des coordonées de chaque balle, un cadre va être dessiné représentant l'espace intérieur du billard. Les traces des 3 boules seront affichées dans ce cadre : la boule blanche en bleu, la boule jaune en jaune, et la boule rouge en rouge. Nous affichons également les distances parcourues par les boules, le nombre de boules ayant bougées, et le nombre de bandes touchées par la boule étudiée. Pour ce dernier paramètre, nous affichons uniquement le nombre de bandes touchées entre 2 chocs si la partie est gagnée, et le nombre total de bandes touchées si la partie est perdue.

La fonction [GetBallPathLength](#) donne la distance parcourue par chaque boule (en pixels). Pour les 3 boules, cette distance est affiché dans le pdf final.

Programme C

Le programme C [Pix2Pos.c](#) est en charge de trouver à partir d'un pixmap binaire (représentation d'une image), la position de chaque boule, ainsi que leur score. Le pixmap est enregistré dans un tableau à mémoire dynamique, allouée par `calloc()`. Toutes les allocations dynamiques de mémoire ont été libérées, Valgrind renvoie : `All heap blocks were freed - no leaks are possible.`

Pour effectuer la recherche d'une boule, on assume qu'elle est représentée par un carré de 13 pixels côté. Le score d'une boule est obtenu en itérant l'opération de recherche de la balle pour chercher le carré qui contient le plus de pixels de la couleur recherchée.

Dans chaque carré de la taille d'une boule, on vérifie la couleur de chaque pixel. On incrémente le score d'une couleur si le pixel lu est de cette couleur. Une optimisation utilisée pour réduire le temps d'exécution du programme est de vérifier le milieu du carré. Si le pixel au milieu du carré est bleu, cela veut dire que la boule n'est pas dans ce carré, et on passe au suivant.

Après avoir scanné toute l'image, le programme prend les coordonnées de chaque boule trouvée (correspondant au score maximale pour chaque couleur) et les écrit dans le fichier `Pos.txt`, sous la forme suivante :

```
Red: Xr, Yr, scorer\n
Yellow: Xy, Yy, scorey\n
White: Xw, Yw, scorew\n
```

où X_i , Y_i , $score_i$ (pour $i=\{r, y, w\}$) sont des `int` représentant respectivement la coordonnée horizontale, verticale, et le score des boules.

Pour que le score soit considéré par valide, il faut qu'il soit au dessus de la constante `BallMinScore` définie à 15. Si le score maximale trouvé pour une boule est inférieur au score minimale, les coordonnées horizontales et verticales deviennent -1 et le score 0.

Précision importante : chaque exécution du programme `Pix2Pos.exe` n'analyse qu'un seul pixmap. Pour analyser une séquence entière d'image, on fait appel à LabVIEW qui appellera le programme C pour chaque image d'un dossier.

Programme LabVIEW

`Billard2023.vi` est le chef d'orchestre de tout le programme. En premier lieu, il recherche les bornes intérieures de la première image du billard par `Trouver_bornes.vi`, en détectant les pixels dans un intervalle de couleur Bleu sombre correspondant au bord. LabVIEW va ensuite itérer sur chaque image de la séquence choisie pour afficher l'image (`display_image.vi`) et la convertir en pixmap binaire par `writePixmap.vi`. Le format utilisé est little-endian, format le plus courant sur les systèmes d'exploitation actuels. Le fichier `Pixmap.bin` contient les valeur de largeur et hauteur de l'image complète, ainsi que les valeurs de chaque pixel de l'image.

La ligne de commande pour le programme C est créée à partir des informations données pour les constantes (intervalles de couleurs) et des informations trouvées par LabVIEW pour les bornes intérieures du billard. Dans `Appel_programme_C.vi`, on utilise le subVI `System Exec` pour lancer l'exécutable `Pix2Pos.exe`. Si tout s'est bien passé lors de l'exécution de `Pix2Pos.exe`, un fichier `Pos.txt` est créé contenant les coordonnées et le score de chaque boule dans le billard. Ce fichier est ouvert en lecture seule pour être lu entièrement par `lecture_pos.vi`.

Les coordonnées et scores trouvés dans `Pos.txt` sont passées à `pos_to_nan.vi` qui va analyser les scores. Si pour une boule, le score est nul, cela signifie qu'elle n'a pas été trouvée par le programme C. Ses coordonnées sont donc passées à `Nan`. En utilisant l'auto-index à la sortie de la boucle, les tableaux de coordonnées sont créés pour chaque boule en X et Y.

La dernière partie de `Billard2023.vi` consiste à envoyer les informations à Matlab. Pour ce faire, les tableaux de coordonnées sont transformés en vecteur ligne, puis en chaîne de caractère pour Matlab. Les options d'affichage et couleurs sont vérifiées (en cas d'erreur, la ligne par défaut est -.), et `matlab_script_strings.vi` concatène toutes les chaînes de caractère et le script Matlab pour être exécuté.

Informations de compilation et versions

Le programme `Pix2Pos.c` a été compilé par `gcc`. Pour accélérer l'exécution du programme, nous recommandons d'optimiser la compilation au 3e niveau par `-O3`.

Pour le développement, le programme a été compilé sous Ubuntu 20.04 :

```
gcc (Ubuntu 11.1.0-1ubuntu1 20.04) 11.1.0.
```

Pour produire l'exécutable Windows `Pix2Pos.exe` fourni à LabVIEW, le programme a été compilé sous Windows 11 :

```
gcc.exe (Rev1, Built by MSYS2 project) 12.1.0.
```

La version de LabVIEW utilisée est 2022 Q3.

La version de MATLAB est R2022b.