

Le Contrebandier

La Base :

A partir du programme de départ, permettant de déplacer deux pions à tour de rôle et de se déplacer sur un damier de taille définie dans quatre directions : haut, bas, droite et gauche.

Nous nous sommes servis de cette base pour confectionner notre jeu

Les Améliorations :

Tout d'abord voici les bibliothèques que nous avons utilisées :

```
#include <iostream>
#include <istream>
#include <vector>
#include <sys/ioctl.h>
#include <unistd.h>
#include <termios.h>
#include <time.h>
#include <dirent.h>
#include <stdio.h>
#include <random>
#include <fstream>
#include <iomanip>
#include <string>
#include <map>
#include <algorithm>
```

Nous avons rajouté plusieurs constantes :

```
const vector <string> VControls {"J1Up", "J1Down", "J1Left", "J1Right", "J2Up",
"J2Down", "J2Left", "J2Right", "pause"};
```

//! Liste des options des différentes commandes des 2 joueurs lors de la partie

```
const vector <string> VColors {"ColorPlayer1", "ColorPlayer2", "ColorFood"};
```

//! Liste des options de couleur disponible de l'ensemble des objets.

```
const string languageSetting = "langue";
```

//! Option des différentes langues disponible

```
const string foodSetting = "food";
```

//! Option du trésor a "manger" permettent de gagner un maximum de point

```
const char KPlayerUp ('^');
```

//! Joueur allant en haut

```
const char KPlayerDown ('v');
```

//! Joueur allant en bas

```
const char KPlayerLeft ('<');
```

//! Joueur allant a gauche

```
const char KPlayerRight ('>');  
//! Joueur allant à droite
```

```
const char KEmpty (' ');  
//! Espace vide
```

```
typedef pair <unsigned, unsigned> CTerminalSize;  
// Un type qui représente la taille de la fenêtre
```

Nous avons rajouté plusieurs Variables :

```
CMyParam Settings;  
//! Réglages du jeu
```

```
CMatrix Mat ;  
//! Grille de jeu
```

```
CPosition Player1;  
//! Position du joueur 1
```

```
CPosition Player2;  
//! Position du joueur 2
```

```
unsigned ScoreJ1;  
//! Score du joueur 1
```

```
unsigned ScoreJ2;  
//! Score du joueur 2
```

```
unsigned long Time;  
//! Temps au début du chronomètre
```

```
unsigned randomCounter = 0;  
//! Compteur du générateur aléatoire
```

Nous avons codé différentes fonctions :

```
struct CMyParam {};  
// Cette classe permet dans le programme de regrouper les différents  
paramètres du jeu
```

```
void playMusic() ;  
//Fonction pour jouer la musique
```

```
void stopMusic() ;  
//Fonction pour arreter la musique
```

```
vector<string> getLanguages() ;  
//Fonction pour avoir la liste des langues disponibles (vecteur contenant les  
langues)
```

```
void setLanguage(string & LanguageFile) ;
```

//Fonction permettent de mettre la langue définit par le joueur (les paramètres de langue)

void LoadParams(CMyParam & MyParams) ;

// Fonction qui charge tous les paramètres de configuration du jeu à partir du fichier « config.yml »

void setDefaults(CMyParam & MyParams) ;

//Fonction qui définit tous les paramètres par défaut du jeu

string getColor(string & color) ;

// Fonction permettent de donner une couleur pour chaque objet du jeu

void setDifficulty() ;

// Fonction qui affiche le menu de sélection des différents modes de difficulté disponible

void languageMenu() ;

// Fonction qui affiche le menu de sélection des différentes langues disponible

void showHelp() ;

// Fonction qui affiche le fichier d'aide du jeu

bool setOption(int & Choix) ;

//fonction qui mets en paramètre les différentes options sélectionner

bool mainMenu() ;

// Fonction qui affiche le menu principal avec les différentes rubrique (choix difficultés, choix de langues, ...)

void startTimer() ;

// Fonction qui démarre un chronomètre lors du commencement de la partie

unsigned getSecondsElapsed() ;

// Fonction qui donne le nombre de secondes écoulées depuis le début du chronomètre

int getTimeLeft() ;

// Fonction qui donne le temps restant aux joueurs

void gamePause() ;

//Fonction permettent de mettre le jeu en pause

void popFood(CMatrix & Mat) ;

// Cette fonction permet lors de la partie de faire apparaître le « trésor » afin de gagner un maximum de points

bool isFirstPlayer(CPosition & Pos) ;

// Fonction pour déterminer le joueur en fonction d'une position

bool checkEat(char & Object, CPosition & PosObject, char & Player, CPosition & PosPlayer, char & Input);

//Fonction de détections avant un mouvement qui va analyser ce qui se trouve à l'emplacement ou le pion veut se déplacer et dire si le déplacement est autorisé.

Par exemple le pion ne peut pas avancer s'il y a un mur ou si il est face à un autre pion, mais il peut si déplacer sur des case vide ou manger un pion si il se trouve sur son coté.

```
void readInput(char & Input) ;
```

```
// Fonction permettant de lire la saisit du clavier des différentes touches de  
chaque joueur lors de la partie
```

```
void getWindowSize(CTerminalSize & Size);
```

```
//Fonction permettant d'adapter la fenêtre selon l'écran du terminal
```

```
bool win(unsigned Player, unsigned & Score1, unsigned & Score2);
```

```
// Lors de la fin de la partie, cette fonction permet d'afficher le gagnant ( joueur  
1 ou joueur 2)
```