Le jeu du damier

420-C21-IN Programmation II

Évaluation **formative**

Directive : Vous devez remettre votre laboratoire sur le dépôt Moodle **Laboratoire #1** du cours

Objectif du laboratoire

Dans ce laboratoire, vous replongerez dans la dépendance au développement de logiciel. Vous devez développer le jeu du damier. Dans ce jeu, vous devez amasser tous les $$$$ caché au travers du dédale de pièces et de couloirs secrets. Pour y parvenir, vous devez :

- Utiliser les tableaux multidimensionnels
- Utiliser les types énumérés et les types de structure
- Isoler les différentes composantes du code.

Spécification

Votre jeu est composé d'un damier de 96 cases réparties sur 8 lignes et 12 colonnes. Le joueur ou la joueuse pourra déplacer un curseur jaune à l'aide des flèches du clavier ou du pad numérique sur les différentes cases du damier. Chaque case du jeu possède un des quatre états : **case bleue**, **case dollar**, **case jaune** ou **case noire**. Au début du jeu, toutes les cases sont soit une **case noire,** soit une **case bleue**.

Le curseur ne peut pas accéder à une **case noire**. Lorsque le curseur se pose sur une case bleue, celle-ci change en une **case jaune** après avoir quitté la case. Si le curseur quitte une **case rose**, celle-ci devient une **case noire**.

Toutefois, il y a une exception pour les **cases dollars**. Au départ, les cases dollars sont **bleues**. Une fois que le curseur quitté la case, celle-ci devient une **case dollar dévoilée**. Par la suite, lorsque le curseur passe sur la **case dollar dévoilée**, elle devient une **case jaune**. Le cheminement ne reste pas la suite le même. En résumé, il existe deux cheminements différents :

```
    CO -> CF -> CV
CS -> CD -> CF -> CV
```

Liste des chemins selon les différentes cases (voir la section sur l'énumération des cases)

Le jeu se termine à la première des deux situations suivantes :

- Toutes les cases dollars (16 au total) ont été ramassées
- Le curseur ne peut pas se déplacer.

Il est donc important de vérifier à chaque déplacement si le curseur peut se déplacer. À vous de trouver l'algorithme. Vous trouverez un exemple du jeu dans l'exécutable en pièce jointe.

Remarque importante

Au départ, le joueur est sur une case bleue. Après son déplacement, votre programme doit dévoiler la case suivante selon les chemins prévus.

Les déplacements

Le curseur peut se déplacer selon les huit directions : haut, bas, gauche, droite, diagonales.

Console et case

La console a une taille de 120 caractères par 45 caractères. De plus, les éléments de jeu (cases et textes) doivent être en couleur. Chaque case a une hauteur de 3 caractères et une largeur de 6 caractères.

Spécification technique

Les touches de déplacement

Pour se déplacer, le logiciel devra lire les flèches du clavier ou du pad numérique. Comme tous les caractères, ces touches possèdent un code ASCII. Toutefois les flèches ont la particularité d'être représentées par deux caractères successifs.

Précédées par les caractères 0 (pour le *numpad*) ou 224 (pour le *arrowpad*), les différentes directions sont représentées par des nombres. Ces nombres ont été introduits dans une énumération afin de faciliter la gestion du code :

```
enum class ArrowKeys
{
    UP_LEFT = 71,       UP = 72,      UP_RIGHT = 73,
    LEFT = 75,                        RIGHT = 77,
    DOWN_LEFT = 79, DOWN = 80, DOWN_RIGHT = 81
};
```

L'exemple « **fleche.cpp** » présente un fonctionnement des touches.

Les cases

Dans ce jeu, les cases sont représentées par deux éléments : un caractère et une couleur. Toutefois, pour décrire facilement ces cases, nous utiliserons une structure de base les représentant :

```
enum Case
{
   CS, CD, CO, CF, CV
   // CS --> Case surprise (bleu)
   // CD --> Case dollars (vert)
   // CO --> Case ordinaire (bleu)
   // CF --> Case fragile (jaune)
   // CV --> Case vide (noir)
};
```

Par la suite, nous utiliserons une structure de style simple pour lier la couleur et le caractère d'une case.

```
struct Style
{
      Color color; uint8_t c;
      // color --> Couleur de la case
      // c --> Caractère de la classe
};
```

Toutefois, pour être efficaces, nous utiliserons un vecteur pour décrire les différentes cases.

```
Style map[4] =
{
      { Color::blu, '\xB2' },
      { Color::grn, '\x24' },
      { Color::blu, '\xB2' },
      { Color::yel, '\xB0' },
      { Color::blk, '\x00' },
};
```

Le fichier « **cases.cpp** » présente un exemple pour l'affichage des cases.

Le damier

Pour ce projet, nous représenterons la grille de jeu dans une matrice. Pour cette fois, la valeur du tableau sera statique :

```
Case damier[LIG][COL] =
{
        { CO, CO, CO, CO, CO, CO, CO, CO, CV, CO, CO, CS },
        { CO, CO, CV, CV, CO, CO, CO, CO, CO, CV, CO, CV },
        { CO, CO, CV, CS, CV, CO, CO, CO, CO, CO, CV, CS },
        { CO, CO, CV, CS, CV, CO, CO, CV, CV, CO, CV, CS },
        { CS, CO, CV, CV, CV, CS, CV, CO, CV, CO, CV, CO },
        { CS, CO, CS, CS, CO, CS, CV, CS, CV, CO, CV, CO },
        { CS, CO, CS, CO, CO, CO, CV, CV, CV, CO, CV, CO },
        { CS, CS, CO, CO, CO, CO, CO, CO, CO, CO, CO, CO }
};
```

Cette matrice servira de base à l'affichage et à la manipulation du tableau.

Déplacement

Pour vous aider, créer, une structure **Move** comportant deux variables de type **Coordonne** (structure à créer). Cette structure vous permettra de connaître la position d'origine au début du tour (**from**) et la position où le curseur doit se déplacer (**to**).

```
struct Move
{
    Coordonne from, to;
};
```

Au commencement du jeu, le curseur doit être dans le coin supérieur gauche (m.from = {0, 0]).

Curseur du joueur

Le curseur doit être en gris et il est représenté, dans un format de 3x3 cases, par :

```
char cursor[3][3] =
{
        { '\xC9', '\xCD', '\xBB' },
        { '\xBA', '\x00', '\xBA' },
        { '\xC8', '\xCD', '\xBC' }
};
```

Vous devez l'agrandir, sans le déformer, pour qu'il entre dans la grandeur d'une case.

Espace entre les lignes

Vous devez toujours avoir deux espaces vides entre les colonnes et un espace vide entre les lignes.

Rafraîchir le damier

À chaque déplacement, vous devrez rafraîchir uniquement les deux cases modifiées.

Détecter la fin du jeu

Vous devez détecter la fin du jeu à chaque tour.

Utilisation de constante

Vous devez utiliser, en tout temps, des constantes pour les termes immuables.

Remise

Vous devez remettre deux fichiers sur le dépôt Moodle du cours :

- Votre code avec le nom *C21-TP1-Nom-Prenom.cpp*
  - Tout votre programme doit être contenu dans un seul fichier.
  - Utiliser le fichier **C21-TP1-Base.cpp** et renommez-le.
  - Vous devez absolument utiliser les définitions (enum, struct, const et variable) du fichier de base.
- L'exécutable de votre programme *C21-TP1-Nom-Prenom.exe*.

Critères d'évaluation

Ce travail est formatif. Il n'est pas évalué pour la note finale du cours. Toutefois, il est fortement recommandé d'avoir complété les laboratoires afin de réussir le cours. De plus, comme indiqué dans le plan de cours, la remise des laboratoires peut être tenue en compte lors d'une demande de révision de note ou d'atteinte de la compétence.

Plan de réalisation (ébauche)

Réalisez le projet étape par étape. Ceci vous permettra de valider vos sources et vous idées.

1. **Afficher les trois cases et le curseur**

Affichez simplement les trois cases et le curseur dans les bonnes dimensions et les bonnes couleurs.

2. **Afficher le damier**

Afficher simplement un damier. Pour simplifier le tout, je vous conseille d'afficher un damier de cases bleues. Ceci vous permettra de comprendre la logique d'affichage.

3. **Afficher le damier initial**

Prenez le damier initial et affichez-le avec les cases noires.

4. **Afficher le curseur**

Afficher le curseur à sa position initiale. Petite astuce : Le curseur peut s'afficher après le damier.

5. **Déplacer le curseur**

Lisez les touches du clavier et vérifiez si le joueur peut se déplacer à la case désirée. Regardez si les trois conditions suivantes sont respectées :

- Le curseur ne sort pas du damier
- Le déplacement est valide (bonne touche sélectionnée)
- La prochaine case n'est pas une case noire

6. **Et la suite…**

Je vous laisse trouver les étapes suivantes. Ceci vous placera dans un contexte de développement authentique.