

CSI2772

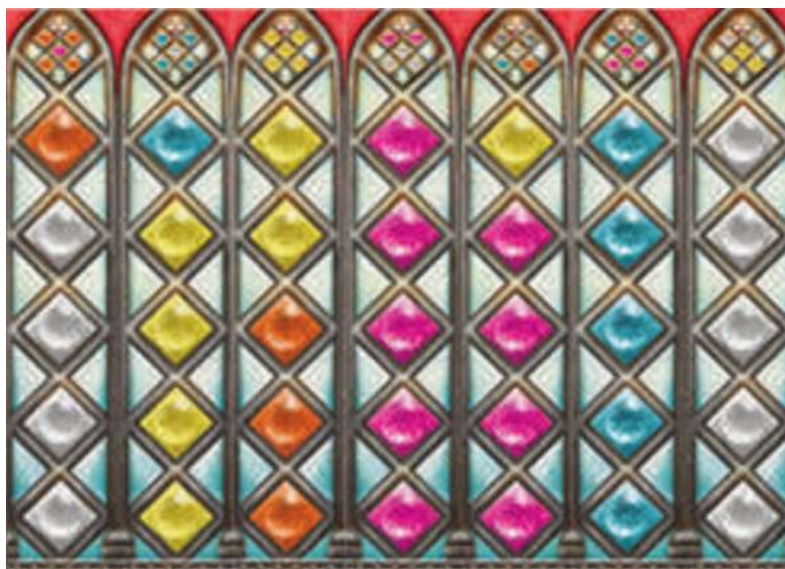
Projet

Automne 2019

Dans ce projet, on vous demande de programmer un jeu ou vous devrez construire les vitraux d'une église à l'aide de pièces de vitre de différentes couleurs. Ce jeu se joue à 2 joueurs et se déroule en 5 phases.

Règles du jeu :

Pour ce faire vous avez à votre disposition des vitres de 5 couleurs différentes : gris (G), jaune (J), bleu (B), rose (R) et orange (O). L'église à décorer est constituée d'une série de vitraux disposés en colonnes. Afin de concevoir vos vitraux vous devez vous baser sur un plan décrivant les couleurs à disposer sur les différentes colonnes de vitraux. Le plan ci-dessous montre le plan pour 7 vitraux à réaliser (numérotés de 0 à 6), chacun constitués de 5 vitres. La première colonne, par exemple, exige 5 vitres grises et 1 vitre orange.



6 5 4 3 2 1 0

Pour concevoir vos vitraux, un vitrier doit se promener d'un vitrail à l'autre. Il se déplace de gauche à droite. Sa position courante est donc un nombre entre 0 et le nombre de colonnes-1. Lorsque le vitrier se trouve à la position v , il peut travailler sur n'importe quel vitrail situé entre 0 et v (inclusivement).

Dans la réalisation de vos vitraux, vous devez vous procurer vos pièces de vitres à partir des lots disponibles. Au début de chaque phase, il y a 5 lots de 4 vitres disponibles plus un espace de surplus initialement vide. Lorsqu'un joueur désire s'approprier des vitres, il choisit un lot, puis une couleur de vitres (présente dans le lot). Il doit alors prendre toutes les vitres de cette couleur dans le lot. Les vitres restantes sont envoyées à l'espace de surplus. Aussi, au lieu de prendre les vitres dans un lot, il est aussi possible de prendre les vitres d'une couleur dans l'espace de surplus. Lorsqu'il n'y a plus de vitres dans les lots et dans le surplus, une phase de jeu est terminée.

A son tour, un joueur doit faire l'une des 2 actions suivantes.

- i) Prendre des vitres d'une couleur dans l'un des lots ou dans le surplus. Puis les placer dans la colonne de son choix située entre 0 et v (v étant la position courante du vitrier). Les vitres ne pouvant être installées que sur une seule colonne, toute pièce de vitre qui ne peut être placée fait perdre un point au joueur. La position du vitrier se retrouve au numéro de colonne choisie.
- ii) Replacer le vitrier à la position `nombre de colonnes - 1` (i.e. complètement à gauche). A noter que si le vitrier se trouve déjà sur la colonne la plus à gauche, le joueur ne peut effectuer cette action et doit donc prendre des pièces de vitres (et possiblement perdre des points si ces vitres ne peuvent être placées).

Chaque fois qu'un vitrail (une colonne) est complétée, le joueur marque des points de la façon suivante : 3 points pour avoir complété un vitrail plus 1 point pour chaque vitrail entamé (au moins une vitre a été placée) situé à droite de la colonne complétée (entre 0 et $v-1$).

Vous trouverez en annexe un exemple de déroulement d'une partie.

Aussitôt que les lots et le surplus est vide, le joueur termine son action et la phase est terminée. Une nouvelle phase démarre alors avec les lots remplis à nouveau.

Programmation :

Votre projet consiste donc à programmer ce jeu. Dans votre réalisation, vous devez définir les classes suivantes :

`class Joueur` qui doit comprendre un nom (`string`), un nombre de points (`int`), la position du vitrier et un pointeur à un vitrail.

La classe doit avoir un opérateur de pré-incrémentation `--` ainsi qu'un opérateur `+=(int)` permettant de déplacer le vitrier vers la droite. Si vous essayez de déplacer le

vitrier en dehors des colonnes, une exception de type `std::out_of_range` doit être lancée. Elle doit aussi avoir l'opérateur `~` permettant de remettre le vitrier à l'extrême gauche.

Cette classe doit aussi surcharger l'opérateur d'insertion `<<` permettant l'affichage à la console de l'état d'un joueur comme suit :

```
Joueur A : 0 points
```

```
Vitrier à la position 6.
```

```
Plan :
```

O	B	J	R	J	B	G
G	J	J	R	R	B	G
G	J	O	R	R	B	G
G	J	O	R	R	G	G
G	J	O	R	R	G	G
6	5	4	3	2	1	0

class Vitrail qui se construit en spécifiant un nombre de colonnes et un nombre de vitres par colonne. La description du vitrail se fait donc à l'aide d'un tableau 2D de caractères (`char**`) qui doit être alloué dynamiquement. Chaque colonne du vitrail est initialisée de la façon suivante : 2 couleurs sont choisies au hasard; puis un nombre aléatoire `H` entre 0 et le nombre de cases dans la colonne est tiré. Les `H` premières cases de la colonne seront de la première couleur et le reste des cases seront de la seconde couleur (donc il ne peut y avoir que 1 ou 2 couleurs dans une colonne).

Cette classe doit aussi avoir les méthodes suivantes :

```
int construireVitrail(std::vector<char> vitres, int colonne);
```

Les vitres contenues dans le vecteur sont ajoutées à la colonne spécifiée. Les caractères correspondants sont donc remplacés par des `x` afin de signifier que cette vitre a été installée. La méthode retourne le nombre de vitres qui n'ont pas pu être placées. Si la colonne spécifiée est déjà complétée, cette méthode doit lancer une exception de type `std::invalid_argument`.

```
bool estComplete(int colonne);
```

qui retourne vrai si le vitrail de cette colonne a été complétée.

```
bool estEnConstruction(int colonne);
```

qui retourne vrai si au moins une vitre de ce vitrail a été installée.

Cette classe doit aussi surcharger l'opérateur d'insertion << permettant l'affichage à la console de l'état des vitraux comme suit :

O	B	J	R	J	x	G
G	J	J	R	R	x	G
G	J	O	R	R	x	G
G	J	O	R	R	G	G
G	J	O	R	R	G	G
6	5	4	3	2	1	0

class Lots permettant de contenir les lots de vitres disponible. Elle contient un tableau de 4 par 5 caractères représentant les 5 lots ainsi qu'un `std::vector<char>` pour le surplus.

La classe doit avoir la méthode permettant de ramasser les vitres d'un lot :

```
vector<char> ramasseVitre(char couleur, int numeroLot);
```

Si le numéro de lot est négatif, cela signifie que les vitres sont ramassées dans le surplus. La méthode retourne toutes les vitres de la couleur spécifiée. Les vitres restantes sont transférées dans le surplus.

Il doit aussi y avoir une méthode permettant de recharger les lots en tirant au hasard des couleurs pour chaque lot (avec un surplus vide) :

```
void reset();
```

Cette classe doit aussi surcharger l'opérateur d'insertion << permettant l'affichage à la console des lots comme suit :

Lots :

J	O	.	.	R
G	G	.	.	R
G	B	.	.	O
G	J	.	.	O

4 3 2 1 0

Surplus:

[G R B]

On vous demande donc de programmer ce jeu de façon à ce que 2 joueurs puissent y jouer en interagissant avec la console.

Les deux instances de la classe joueur sont créées.

Deux vitraux de 7 colonnes par 5 cases sont créées et associés à chacun des joueurs.

Les lots sont créés

Votre programme alterne alors entre les 2 joueurs.

Le joueur choisit un lot, une couleur et une colonne. Ce qui retire les vitres du lot (méthode `ramasseVitre`), déplace le vitrier (opérateur `--`), et place les vitres sur la colonne (méthode `construireVitrail`). Les points rapportés par cette action sont automatique comptés.

Alternativement, le joueur peut simplement demander de replacer son vitrier sur la colonne de gauche (opérateur `~`).

Si un joueur tente un coup illégal, une exception est lancée et il doit proposer une autre action.

En plus de tous vos fichiers `.cpp` et `.h` vous devez remettre un bref document donnant la description de toutes les classes créées. A noter que votre fonction `main` doit se trouver dans un fichier appelé `projetCSI2772.cpp`; ce fichier ne doit contenir que la fonction `main`, aucune autre définition de classes ou fonctions.

Annexe A : Déroulement du jeu

L'exemple suivant montre le déroulement du jeu. La partie débute, les 2 joueurs ont 0 point et les lots sont remplis.

Joueur A : 0 points

Vitrier à la position 6.

Plan :

O	B	J	R	J	B	G
G	J	J	R	R	B	G
G	J	O	R	R	B	G
G	J	O	R	R	G	G
G	J	O	R	R	G	G
6	5	4	3	2	1	0

Lots :

J	O	G	R	R
G	G	B	O	R
G	B	B	O	O
G	J	B	B	O
4	3	2	1	0

Surplus:

[]

Le joueur décide de prendre les pièces bleues du lot 2 et de les placer sur la colonne 1.

Vitrier à la position 1.

Plan :

O	B	J	R	J	x	G
G	J	J	R	R	x	G
G	J	O	R	R	x	G
G	J	O	R	R	G	G
G	J	O	R	R	G	G
6	5	4	3	2	1	0

Lots :

J	O	.	R	R
---	---	---	---	---

G	G	.	O	R
G	B	.	O	O
G	J	.	B	O
4	3	2	1	0

Surplus:

[G]

Au tour du second joueur.

Joueur B : 0 points

Vitrier à la position 6.

Plan :

B	B	R	J	R	B	G
B	B	R	J	R	O	G
G	O	R	R	R	O	G
G	O	R	R	R	O	G
G	O	O	R	R	O	J
6	5	4	3	2	1	0

Lots :

J	O	.	R	R
G	G	.	O	R
G	B	.	O	O
G	J	.	B	O
4	3	2	1	0

Surplus:

[G]

Le joueur décide de prendre les pièces oranges du lot 1 et de les placer sur la colonne 5.

Vitrier à la position 5.

Plan :

B	B	R	J	R	B	G
B	B	R	J	R	O	G
G	x	R	R	R	O	G
G	x	R	R	R	O	G
G	O	O	R	R	O	J
6	5	4	3	2	1	0

Lots :

J	O	.	.	R
---	---	---	---	---

G	G	.	.	R
G	B	.	.	O
G	J	.	.	O
4	3	2	1	0

Surplus:

[G R B]

Au tour du premier joueur :

Joueur A : 0 points

Vitrier à la position 1.

Plan :

O	B	J	R	J	x	G
G	J	J	R	R	x	G
G	J	O	R	R	x	G
G	J	O	R	R	G	G
G	J	O	R	R	G	G
6	5	4	3	2	1	0

Lots :

J	O	.	.	R
G	G	.	.	R
G	B	.	.	O
G	J	.	.	O
4	3	2	1	0

Surplus:

[G R B]

Le joueur décide de prendre les pièces grises du lot 4 et de les placer sur la colonne 1.

Vitrier à la position 1.

Plan :

O	B	J	R	J	x	G
G	J	J	R	R	x	G
G	J	O	R	R	x	G
G	J	O	R	R	x	G
G	J	O	R	R	x	G
6	5	4	3	2	*	0

Lots :

.	O	.	.	R
---	---	---	---	---

.	G	.	.	R
.	B	.	.	O
.	J	.	.	O
4	3	2	1	0

Surplus:

[G R B J]

Puisqu'il ne peut placer que 2 des 3 pièces qu'il a dû ramasser, il perd 1 points.
Puisqu'il a complété la colonne 1, il marque 3 points; il n'a pas entamé la colonne 0, alors il ne marque pas de points pour celle-ci : total 2 points. Si il avait placé au moins une pièce (grise) sur la colonne 0, alors il aurait marqué un point de plus.

Référence : Azul: Stained Glass of Sintra by Michael Kiesling, Next Move Games (2018)