

Tetriste

Projet Programmation C, L2 MIDO 2023-2024

Résumé

Le but de ce projet est de développer un petit jeu avec des pièces caractérisées par une forme et une couleur, où le joueur tente de créer des motifs de forme ou couleur répétés.

1 Dates importantes

- ▷ Deadline composition des groupes (Excel dans le Teams) : 22 décembre 2023 à 23h59.
- ▷ Rendu du projet : lundi 5 février 2024 à 23h (sur Moodle).
- ▷ Soutenances : à définir.

2 Versions

Ce sujet est susceptible d'être modifié.

- ▷ 9 décembre 2023 : première version du projet.

3 Description

3.1 Pièces et plateau

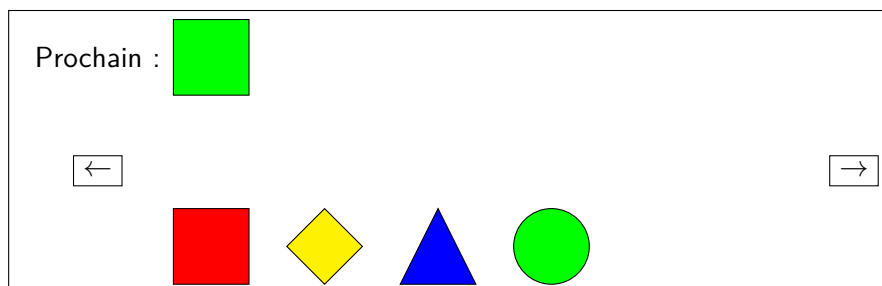
Une *pièce* du jeu sera caractérisée par une *couleur* et une *forme*. Les couleurs seront **Bleu**, **Jaune**, **Rouge** et **Vert**. Les formes seront Carré □, Losange ◇, Rond ○, Triangle △.

Le « plateau » de jeu sera simplement une ligne droite, extensible par la gauche et la droite.

3.2 Insertions des pièces

À chaque étape, le jeu génère aléatoirement une nouvelle pièce, que le joueur devra insérer sur le plateau de jeu.

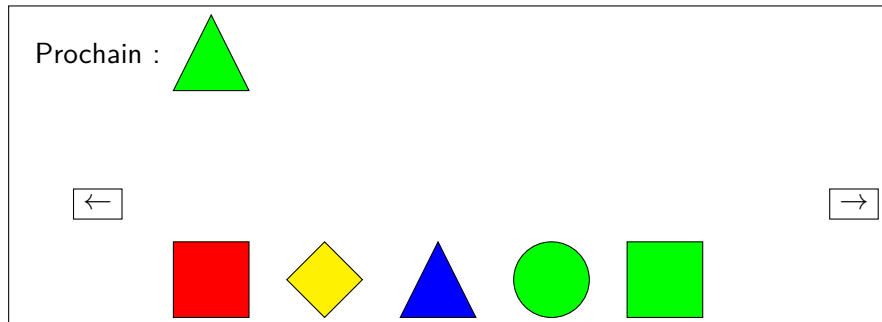
Par exemple, supposons la configuration suivante :



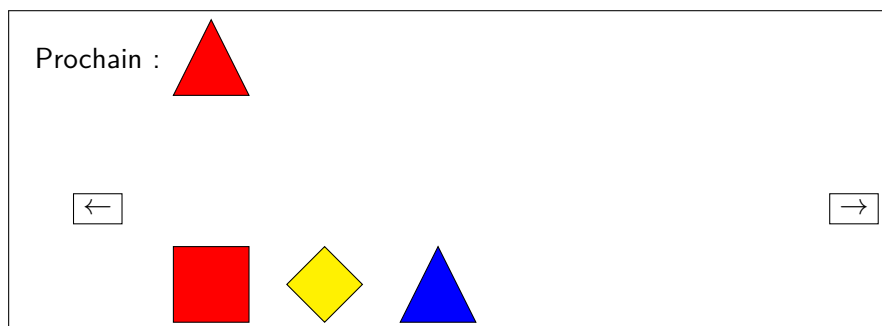
Dans cet exemple, les quatre pièces du bas sont déjà placées sur le plateau de jeu. Le joueur doit maintenant placer la nouvelle pièce, le carré vert. Il a deux possibilités : *soit à gauche*, avant le carré rouge, *soit à droite*, après le rond vert.

Si trois pièces consécutives ont la même couleur ou la même forme, alors les trois pièces disparaissent du plateau de jeu.

Par exemple, si le joueur a choisi de placer le carré vert à droite, et choisit ensuite de placer la pièce suivante, le triangle vert, à droite :



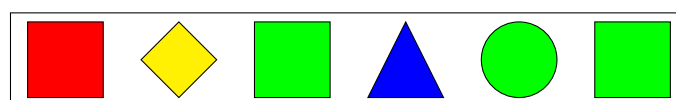
Le plateau de jeu passera dans la configuration suivante (car les trois pièces vertes consécutives présentes ont disparu) :



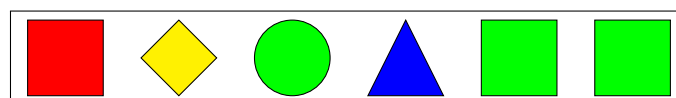
3.3 Décalages

En plus des insertions à gauche ou à droite, le joueur peut choisir d'effectuer des décalages dans le but d'effectuer des suppressions non atteignables autrement. En effet, on propose la possibilité de faire un *décalage à gauche de toutes les pièces ayant une certaine forme*, ou un *décalage à gauche de toutes les pièces ayant une certaine couleur*.

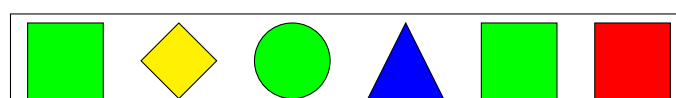
Par exemple, si le plateau de jeu est dans cette configuration :



faire un décalage pour les verts consiste à décaler circulairement par la gauche toutes les formes de pièces vertes, ce qui donnera la configuration suivante (où le carré le plus à droite a remplacé le cercle vert, le cercle vert a remplacé le carré vert le plus à gauche et le carré vert le plus à gauche a « remplacé » carré vert le plus à droite) :



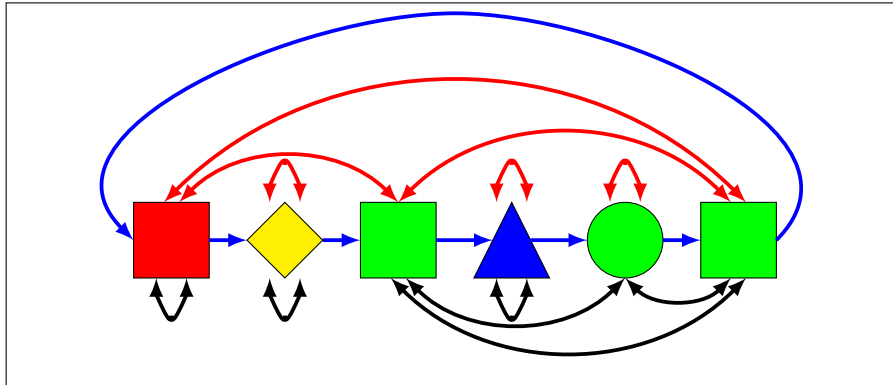
Si à présent le joueur décide de faire un décalage pour les carrés, alors on décale circulairement par la gauche toutes les couleurs des pièces carrés, ce qui donnera la configuration suivante :



Note : si le joueur choisit d'effectuer un décalage pour les autres formes ou les autres couleurs, rien ne se passera car il n'y a qu'une seule pièce triangle, rond ou losange, et une seule pièce rouge, bleue ou jaune.

3.4 Chainage des pièces

Les pièces seront liées entre elles par plusieurs listes (doublement) chaînées circulaires. Reprenons un exemple :



Les pièces sont liées entre elles par une liste chaînée simple circulaire constituées des insertions multiples du joueur (liste bleue). En gardant un pointeur sur le dernier, on facilite les insertions à gauche ou à droite.

Chaque pièce connaît les pièces précédente et suivante ayant la même forme qu'elle. Ainsi, les pièces carrés (resp. losanges, triangles, ronds) sont liées entre elles par une liste doublement chaînée circulaire (en rouge).

Chaque pièce connaît les pièces précédente et suivante ayant la même couleur qu'elle. Ainsi, les pièces rouges (resp. jaunes, verts, bleus) sont liées entre elles par une liste doublement chaînée circulaire (en noir).

Ces deux derniers chainages facilitent les décalages par la gauche.

3.5 Interface

Le plateau de jeu sera affiché dans le terminal grâce aux **caractères d'échappement ASCII**.¹

Par exemple, le code suivant efface tout le terminal et affiche **LCTR**.

```
printf("\033[0;0H"); // on se place en haut à gauche
printf("\033[2J"); // on efface tout
printf("\033[31mL"); // L rouge (Losange)
printf("\033[34mC"); // C bleu (carré)
printf("\033[33mT"); // T jaune (triangle)
printf("\033[32mR"); // R vert (rond)
```

Le joueur commandera le jeu au moyen de commandes simples (par exemple g pour mettre la pièce suivante à gauche).

1. Le terminal Unix ou MacOS accepte nativement ces codes. Pour les utiliser sous Windows, vous pouvez ajouter quelques lignes de code.

Voir <https://solarianprogrammer.com/2019/04/08/c-programming-ansi-escape-codes-windows-macos-linux-terminals/> ou <https://superuser.com/questions/413073/windows-console-with-ansi-colors-handling>

4 Niveaux de difficultés

4.1 Tétriste

- ▷ Affichage du jeu dans le terminal (ASCII art)
- ▷ Les insertions à gauche et à droite fonctionnent
- ▷ La suppression de trois pièces consécutives ayant couleur ou forme commune fonctionne
- ▷ Allocation et libération de la mémoire parfaite (vérifiée avec valgrind)

4.2 Tépatrice

- ▷ Tout Tétriste
- ▷ Les décalages à gauche fonctionnent via les listes de couleurs et formes

4.3 Técontent

- ▷ Tout Tépatrice
- ▷ Système de score affiché (on marque des points à chaque suppression de triplets, avec plus de points si obtenu via décalage car plus difficile à voir)
- ▷ La partie s'arrête après plus de 15 pièces présentes sur le plateau de jeu
- ▷ Sauvegarde/chargement d'une partie (on stocke dans un fichier texte l'état du jeu)
- ▷ Système de meilleurs scores : on sauvegarde dans un fichier texte les 10 meilleurs scores réalisés avec le nom de l'auteur, stockés dans l'ordre décroissant

4.4 Tavalidé

- ▷ Tout Técontent
- ▷ On affiche les cinq prochaines pièces à venir dans le jeu plutôt qu'une seule
- ▷ Une interface graphique simple pour jouer (par exemple [Ncurses](#), [MLVlib](#), [SDL](#)...)

4.5 Bonus

Pour les plus motivés, voici quelques pistes de bonus possibles.

- ▷ Calculer un score plus réaliste qui donne plus de points pour les disparitions en cascade de triplets (combo), ou bien la création de 4-uplet ou 5-uplet
- ▷ Des niveaux de difficulté où le joueur peut choisir le nombre de couleurs ou de formes
- ▷ Une « IA » indique au joueur le meilleur coup à jouer
- ▷ ...

5 Conditions de rendu

Le projet est à effectuer en **binôme**. En cas de nombre impair d'étudiants, **un seul** groupe sera autorisé à effectuer le projet en **trinôme** (notation plus sévère). Pour former les groupes, utiliser le fichier Excel dans le canal Teams correspondant. **Les binômes doivent être différents de ceux utilisés pour le projet d'Archi !**

Toute question sur le projet devra être posée dans l'équipe Teams (et non par mail ou en individuel) afin de limiter les doublons de questions.

Le projet est à rendre avant la date et l'heure indiquées plus haut sur l'espace Moodle dédié. **Chaque heure de retard sera pénalisée d'un point sur la note finale** (une heure entamée étant due). Le format de rendu est une archive au format ZIP contenant :

- ▷ Le code-source de votre projet (éventuellement organisé en sous-dossiers).
- ▷ Un fichier README à la racine, indiquant comment compiler et exécuter votre projet.
- ▷ Un document dev.pdf, devant justifier les choix effectués, les avantages et inconvénients de vos choix, expliquer les algorithmes et leur complexité, indiquer quelles ont été les difficultés rencontrées au cours du projet ainsi que la répartition du travail entre les membres du binôme. Un programmeur averti devra être capable de faire évoluer facilement votre code grâce à sa lecture. En aucun cas on ne doit y trouver un copier/coller de votre code source. Ce rapport doit faire le point sur les fonctionnalités apportées, celles qui n'ont pas été faites (et expliquer pourquoi). Il ne doit pas paraphraser le code, mais doit rendre explicite ce que ne montre pas le code. Il doit montrer que le code produit a fait l'objet d'un travail réfléchi et minutieux (comment un bug a été résolu, comment la redondance dans le code a été évitée, comment telle difficulté technique a été contournée, quels ont été les choix, les pistes examinées ou abandonnées...). Ce rapport est le témoin de vos qualités scientifiques mais aussi littéraires (style, grammaire, orthographe, présentation).
- ▷ Optionnellement un makefile.

L'archive aura pour nom Nom1Nom2.zip, où Nom1 et Nom2 sont les noms des membres du binôme par ordre alphabétique. L'extraction de l'archive devra créer un dossier Nom1Nom2 contenant les éléments précisés ci-dessus.

Il va sans dire que les différents points suivants doivent être pris en compte :

- ▷ Projet compilant sans erreur ni warning avec l'option -Wall de gcc.
- ▷ Uniformité de la langue utilisée dans le code et des conventions de nommage et de code.
- ▷ Les sources doivent être commentées, dans une unique langue, de manière pertinente (pas de commentaire « fait un test » avant un if).
- ▷ Le code doit être propre et correctement indenté.
- ▷ Bonne gestion de la libération de la mémoire (utilisez valgrind pour vérifier que le nombre de free est égal au nombre de malloc : il faut d'abord compiler avec l'option -g, puis lancer valgrind avec en argument le nom du programme. Par exemple :

```
gcc -g main.c
valgrind ./a.out
```

Valgrind peut également vous indiquer la ligne où une *segmentation fault* a eu lieu, pratique ! – vous pouvez aussi compiler avec l'option -fsanitize=address pour savoir cela.

- ▷ Le projet doit évidemment être propre à chaque binôme. **Un détecteur automatique de plagiat sera utilisé.** Si du texte ou une petite portion de code a été emprunté (sur internet, chez un autre binôme,...), il faudra l'indiquer dans le rapport, ce qui n'empêchera pas l'application éventuelle d'une pénalité. Nous rappelons l'importance d'avoir du code original écrit par vous. L'utilisation de générateur automatique de code basé sur l'IA est à proscrire. Tout manque de sincérité sera lourdement sanctionné (conseil de discipline) – c'est déjà arrivé.

La documentation (rapports, commentaires...) compte pour un quart de la note finale.

Soutenance

Une soutenance d'une dizaine de minutes aura lieu pour chaque binôme, à la date et à l'heure indiquées plus haut, sur les machines de l'université. Elle doit être préparée et menée par le binôme (*i.e.* fonctionnant parfaitement du premier coup, avoir préparé des illustrations des qualités de votre projet, etc.).

Pendant la soutenance, ne perdez pas de temps à nous expliquer le sujet : nous le connaissons puisque nous l'avons écrit. Essayez de montrer ce qui fonctionne et de nous convaincre que vous avez fait du bon travail.

Bon courage !