

# v1.1 **Projet CY-Colors**

CLASSE préING1 • 2022-2023

AUTEUR Romuald GRIGNON

E-MAIL [romuald.grignon@cyu.fr](mailto:romuald.grignon@cyu.fr)

## DESCRIPTION

Le but de ce projet est de réaliser un jeu de plateau qui se joue à 2 joueurs au tour par tour.

Le plateau de jeu est constitué de 64 cases colorées et chaque joueur dispose de 8 pièces à déplacer, toutes positionnées sur 2 lignes opposées du plateau, comme les pièces d'un jeu d'échecs.

Le but du jeu est d'amener une de ses pièces sur la ligne adverse, c'est à dire de traverser complètement le plateau. Le premier joueur à y parvenir est déclaré vainqueur et marque des points.

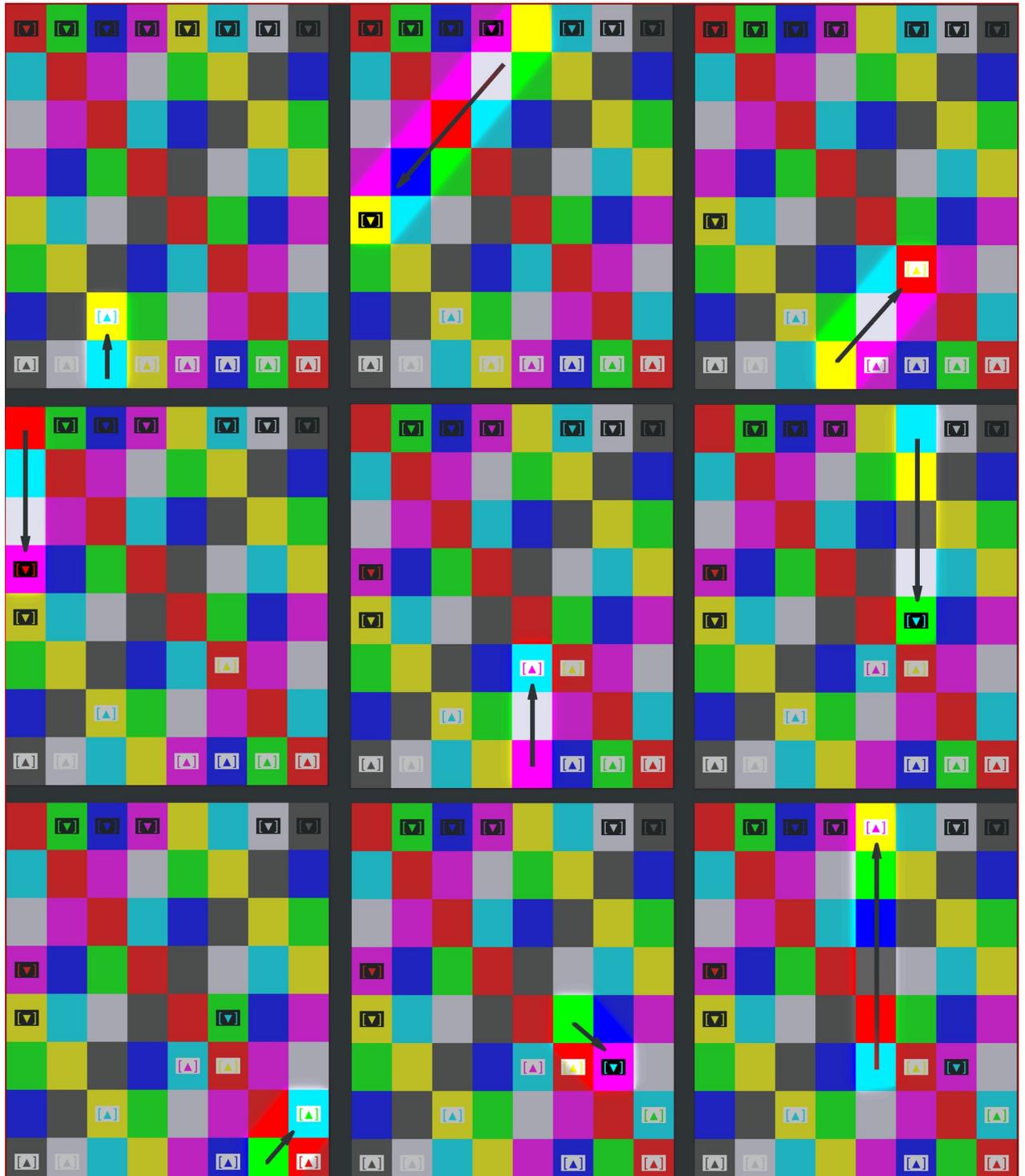
Les pièces peuvent se déplacer d'autant de cases que voulu et possible, toujours vers l'avant ou sur le côté mais jamais vers l'arrière. Il est possible de les déplacer également en diagonale vers l'avant. Il est impossible pour une pièce de traverser une autre pièce, ni de se positionner sur une case déjà occupée par une autre pièce. Les pièces ne peuvent pas sortir du plateau de jeu.

Chaque pièce de chaque joueur dispose d'une marque de couleur, correspondant à l'une des 8 couleurs du plateau jeu. Ces pièces sont placées initialement sur des cases portant leurs couleurs respectives. Voici un exemple d'affichage dans le terminal :



Le premier joueur dont c'est le tour de jouer, peut déplacer n'importe quelle pièce. La couleur de la case sur laquelle vient d'arriver la pièce détermine la pièce à déplacer pour l'adversaire et ainsi de suite. C'est toujours la couleur de la case sur laquelle une pièce vient de se poser qui détermine la pièce suivante à déplacer (celle avec la marque de couleur correspondante).

Sur la page suivante, se trouve le schéma de plusieurs coups pour illustrer les règles de déplacement :



Si un joueur ne peut pas déplacer sa pièce avec la marque de couleur active, alors son tour est annulé et c'est à l'autre joueur de rejouer de nouveau. Il déplacera la pièce correspondant à la couleur de la dernière case recouverte comme c'est le cas à chaque tour (sauf qu'ici il joue 2 fois de suite).

Si après un déplacement, les 2 joueurs sont bloqués, le gagnant est le dernier à avoir déplacé une pièce.

Si un joueur déplace une de ses pièces sur la dernière ligne (ligne adverse) alors il est déclaré vainqueur.

Le jeu peut se dérouler en plusieurs manches (entre 1 et N à définir dans votre programme. Cela peut être un choix de l'utilisateur au lancement), et chaque victoire d'une manche rapporte 1 point. A la fin de toutes les manches, le joueur avec le plus grand score est déclaré vainqueur. Si il y a égalité entre les 2 joueurs, une manche supplémentaire est lancée pour départager les joueurs. Il ne peut donc pas y avoir d'égalité en fin de jeu. Une manche peut exceptionnellement rapporter 2 points : voir section "Promotion d'une pièce" ci-après.

### **Promotion d'une pièce :**

Dans le cas d'une partie à plusieurs manches, la pièce qui a donné la victoire à un joueur, est promue. Une pièce ne peut être promue qu'une seule fois dans toute la partie, même si elle permet de remporter plusieurs manches. Toutefois une pièce promue qui remporte une manche donne 2 points au joueur.

Une pièce promue possède une mécanique de déplacement différente des pièces communes. Une pièce promue peut se déplacer au maximum de 5 cases, tout droit vers l'avant, en diagonale vers l'avant, ou sur les côtés, mais jamais en arrière.

Une pièce promue, si elle se trouve juste devant une pièce adverse, peut pousser cette dernière vers l'arrière d'une case. La pièce promue avance donc d'une case vers l'avant, la pièce adverse recule d'une case. La couleur de la case où la pièce adverse a reculé devient la couleur active pour le prochain tour.

Une pièce promue ne peut pas pousser une pièce adverse en dehors du plateau de jeu. Une pièce promue ne peut pas pousser 2 pièces positionnées l'une derrière l'autre. Une pièce promue peut pousser uniquement les pièces adverses. Enfin une pièce promue ne peut pas pousser une autre pièce promue.

Graphiquement, une pièce promue doit être différenciée des autres, en utilisant un autre symbole par exemple. Elle conserve malgré tout sa marque de couleur.

---

## INFORMATIONS GENERALES

### Taille de l'équipe

Ce projet est un travail d'équipe. Il est autorisé de se réunir en groupe de 3 personnes.

### Démarrage du projet

Vous obtiendrez de plus amples informations quant aux dates précises de rendu, de soutenance, les critères d'évaluation, le contenu du livrable, ..., quand le projet démarrera officiellement.

### Dépôt de code

Vous devrez déposer la totalité des fichiers de votre projet sur un dépôt central Git. Il en existe plusieurs disponibles gratuitement sur des sites comme github ou gitlab.

### Démonstration

Le jour de la présentation de votre projet, la version finale sur votre dépôt sera celle utilisée, même si vous avez ajouté des fonctionnalités ou corrigé des bugs entre temps. La démonstration se fera sur une machine de l'enseignant chargé de suivre votre groupe. C'est la date de commit sur votre dépôt qui fera foi.

Vous ferez votre démonstration, en fonction des exigences du cahier des charges de votre projet, et vous aurez à modifier légèrement votre application en direct en fonction de la requête de votre chargé de TD.

De plus des questions supplémentaires pourront être posées afin d'évaluer votre connaissance de l'implémentation de votre projet.

## FONCTIONNALITES PRINCIPALES DU PROJET

- Le but principal du projet est de fournir une application **fonctionnelle**.
- Tous les **éléments de votre code** seront écrits en langue **anglaise** (structure, types, fonctions, fichiers, ...).
- Votre code sera généreusement **commenté** (langue française autorisée).
- Votre application ne doit **jamais** s'interrompre de manière intempestive (crash), quelle que soit la raison. Toutes les erreurs doivent être gérées correctement. Il est préférable de d'avoir une application stable avec moins de fonctionnalités qu'une application contenant toutes les exigences du cahier des charges mais qui plante trop souvent. Une application qui crée des fautes de segmentation par exemple sera très pénalisée.
- Votre application devra être **modulée** afin de ne pas avoir l'ensemble du code dans le même fichier.
- Votre **livrable** sera une **URL** d'un dépôt **Git public** envoyée à votre chargé de TD.
  
- Votre projet doit pouvoir respecter toutes les règles du jeu et afficher le résultat de la partie quand elle est terminée.
- Votre projet doit pouvoir intégrer une fonctionnalité pour sauvegarder et restaurer une partie (au moins une partie, à vous de voir si vous laissez la possibilité aux joueurs de sauvegarder plusieurs parties). Au début du programme, il y aura donc un choix pour lancer une nouvelle partie ou reprendre une précédente partie sauvegardée. Une partie peut être sauvegardée à n'importe quel moment, quel que soit le joueur dont c'est le tour.
- Les couleurs/symboles utilisés pour votre plateau de jeu sont totalement libres, mais la position sur le plateau doit respecter le

cahier des charges (cf. première image de la description pour voir la position des différentes couleurs).

- Le nombre de manches de la partie, le numéro de la manche en cours, ainsi que le score de chaque joueur doivent être affichés à l'écran pendant le déroulement du jeu.
- La durée de chaque manche doit être au moins mesurée et affichée à la fin de la manche. Si possible vous pouvez afficher constamment le temps de la manche en cours.
- Une difficulté pour ce projet est d'avoir une représentation claire du jeu : en effet les cases sont colorées mais les pièces portent aussi une marque de couleur. De plus il faut pouvoir distinguer les pièces de chaque joueur (avec du noir et du blanc par exemple, ou des symboles encadrant les pièces comme des parenthèses ou des crochets). Il est également possible d'utiliser des symboles plutôt que des couleurs pour l'affichage sur le terminal mais la lisibilité risque d'être réduite : il est conseillé d'utiliser la modification des couleurs du terminal pour ce projet (cf. lien). Quelle que soit la solution retenue, portez une attention particulière à la lisibilité globale du plateau de jeu.

## RESSOURCES UTILES

### Github

- [www.github.com](https://www.github.com)
- <https://docs.github.com/en/get-started/quickstart/hello-world>

### Modifier les couleurs du terminal Linux

- [https://en.wikipedia.org/wiki/ANSI\\_escape\\_code#Colors](https://en.wikipedia.org/wiki/ANSI_escape_code#Colors)
- <http://sdz.tdct.org/sdz/des-couleurs-dans-la-console-linux.html>