

v1.2 **Projet CY-Connect**

CLASSE préING1 • 2022-2023

AUTEUR Romuald GRIGNON

E-MAIL romuald.grignon@cyu.fr

DESCRIPTION

Le but de ce projet est de réaliser un jeu d'alignement de pièces compétitif qui peut se jouer de 2 ou 3 joueurs.

Le plateau de jeu est constitué d'une surface rectangulaire programmable entre 8 et 10 cases de largeur et entre 6 et 8 cases de hauteur.

Chacun des joueurs à tour de rôle va placer une pièce en haut de la surface, et cette pièce va chuter vers le bas jusqu'à être bloquée par une autre pièce précédemment insérée, ou bloquée par le bas de la surface. Le but du jeu est d'aligner 5 pièces identiques. Jusque là toutes les règles sont similaires au jeu de type puissance 4 (disons plutôt puissance 5 ^^).

Lors du tour d'un joueur, après avoir choisi sa colonne, la pièce descend jusqu'à sa position finale. Ensuite le joueur sélectionne obligatoirement une zone carrée de la surface du jeu et va la faire pivoter de 90 degrés vers la gauche ou la droite, au choix. Une fois la rotation de la zone effectuée, la gravité va de nouveau faire effet et les pièces à l'intérieur de cette zone pivotée vont chuter vers le bas jusqu'à rejoindre leurs positions finales.

La zone qui va être pivotée, ne doit pas forcément contenir la nouvelle pièce qui vient d'être insérée, par contre elle doit contenir des cases non-vides : il est interdit de faire pivoter une zone totalement vide (ce qui reviendrait à ne pas faire de rotation de zone).

Après toutes ces opérations, si il y a 5 symboles identiques qui sont alignés horizontalement, verticalement ou en diagonale, le joueur associé est déclaré vainqueur. La condition de victoire est forcément déclenchée APRES la rotation de la zone de jeu.

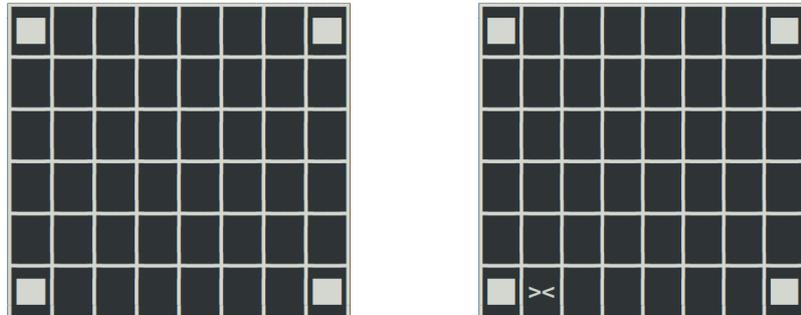
Il est possible que plusieurs joueurs soient déclarés vainqueur en même temps : dans ce cas une égalité sera prononcée. Sinon c'est au tour du joueur suivant d'insérer une pièce en haut de la zone de jeu et de faire pivoter une partie de la zone.

Dans la surface de jeu, les 4 cases des coins initiaux sont des cases remplies, indestructibles, infranchissables et insensibles à la gravité. Cela signifie que rien ne peut les traverser. Par conséquent, aucune pièce ne

peut être insérée sur les colonnes aux extrémités gauche et droite de la zone de jeu lors du tout premier tour. Ensuite cela dépend de la topologie de la zone.

Voici un exemple de la zone de jeu au début de la partie (zone de 8x6 cases).

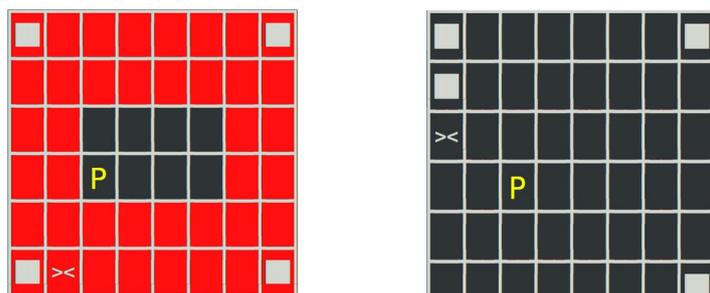
Le joueur 1 doit placer une de ses pièces (ici symbolisé par '><'). Il ne peut pas le faire sur les colonnes gauche et droite car il y a les cases indestructibles (symbolisées par '■'). Si il place sa pièce sur la 2ème colonne, le résultat après application de la gravité est le suivant :



Maintenant, le joueur 1 doit choisir une case qui sera le point de pivot de la zone à faire tourner. La largeur de la zone à pivoter est une valeur aléatoire impaire fixée par le programme entre les valeurs 3 ou 5. Il n'est pas possible de choisir une case pivot qui placerait la zone à tourner en dehors de la zone de jeu. Par exemple si la largeur vaut 3, les cases pivot ne peuvent pas être les cases des bords extérieurs.

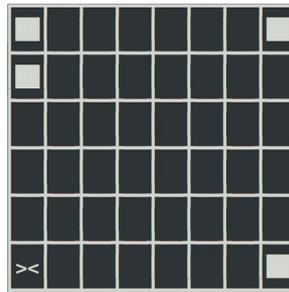
Pour l'exemple, disons que la largeur aléatoire de la matrice de rotation donnée par le programme vaut 5, donc seules les cases à 2 de distance du bord sont éligibles pour être un pivot. Le joueur choisit la case pivot symbolisée ici avec la lettre P. C'est autour de cette case que les pièces vont tourner. Cela signifie que la case symbolisée par P ne va pas changer de place.

Après rotation dans le sens horaire (le choix du sens de rotation est fait par le joueur également), la zone de jeu se trouve dans cet état :



On peut voir que malgré le fait que les cases indestructibles ne sont pas soumises à la gravité, elles sont quand même déplacées par les opérations de rotation. La case indestructible en bas gauche se retrouve en haut du plateau après rotation. On voit également que la pièce insérée par le joueur a été déplacée vers le haut.

La gravité va maintenant s'appliquer sur l'ensemble des pièces qui ont été tournées :

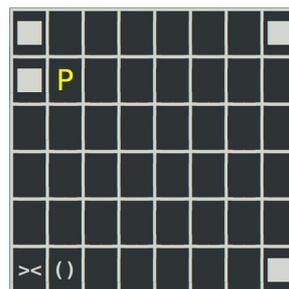


La pièce du joueur 1 tomber à nouveau et se retrouve tout en bas de la zone mais avec la rotation elle se retrouve dans la colonne 1.

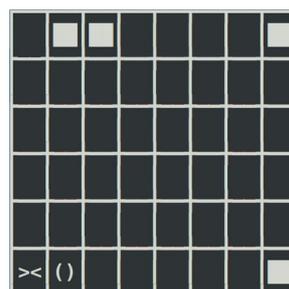
Le programme teste à ce moment si il y a un gagnant. Ce n'est pas le cas ici donc c'est au joueur suivant de jouer : c'est à dire choisir une colonne, un pivot et un sens de rotation (la largeur de la zone à tourner est donnée au joueur AVANT qu'il ne sélectionne la colonne pour qu'il puisse prévoir son coup).

Le joueur 2 possède des pièces symbolisées par les caractères '()'. Il a une largeur de rotation de 3 : il va jouer sur la 2ème colonne, et il va tourner la zone la plus en haut à gauche vers la droite (sens horaire).

Voici l'étape après insertion dans la colonne numéro 2 et application de la gravité.



ET voici le résultat après rotation et application de la gravité.



On voit que la zone que fait pivoter le joueur 2 ne contient aucune pièce de joueurs, mais il y a des cases non-vide (les 2 cases indestructibles en haut de la zone).

On rappelle qu'il est interdit de choisir une zone à faire pivoter qui ne contiendrait QUE des cases vides. La zone doit contenir au moins 1 case non vide (une pièce d'un joueur ou une case indestructible).

La partie continue comme cela jusqu'à ce qu'il y aie 1 ou 2 joueurs avec 5 cases identiques alignées (horizontalement, verticalement ou en diagonale).

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.

- Votre projet doit pouvoir montrer les différentes étapes lors de l'insertion d'une pièce. A minima, on doit voir l'affichage une fois que la pièce est tombée. Ensuite voir le résultat de la rotation de la zone sélectionnée (mais avant application de la gravité), et ensuite le résultat final sur la zone de jeu : donc 3 étapes clés pour pouvoir vérifier que le programme s'effectue correctement.
 - Mais il est préférable de se documenter sur la manière de rafraichir l'écran du terminal afin de redessiner chaque étape lorsqu'une pièce change de case. En mettant une pause légère entre 2 affichages, on aura une réelle impression de mouvement.
-

RESSOURCES UTILES

Github

- 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
- <http://sdz.tdct.org/sdz/des-couleurs-dans-la-console-linux.html>

Contrôler le curseur du terminal

- [https://en.wikipedia.org/wiki/ANSI_escape_code#CSI_\(Control_Sequence_Introducer\)_sequences](https://en.wikipedia.org/wiki/ANSI_escape_code#CSI_(Control_Sequence_Introducer)_sequences)
-