Une image contenant carré, Rectangle, Caractère coloré, shoji

Description générée automatiquementTetris

Pièce.js

const TETROMINOS :

initialise la forme et la couleur des pièces

1. draw() : Dessine la pièce sur la grille en fonction de sa forme et de sa couleur actuelles.

2. rotate() : Fait pivoter la pièce dans le sens des aiguilles d'une montre, en vérifiant les collisions avant d'appliquer la rotation.

3. moveDown() : Déplace la pièce d'une case vers le bas, en verrouillant la pièce si elle rencontre un obstacle.

4. moveHorizontal(direction) : Déplace la pièce horizontalement dans la direction spécifiée (gauche ou droite), en vérifiant les collisions.

Sketch.js

1. setup() : Initialise le canevas et la grille, puis génère la première pièce aléatoire.

2. draw() : Met à jour et dessine le canevas, gérant le mouvement automatique de la pièce vers le bas et le dessin de la pièce actuelle.

3. keyPressed() : Gère les entrées du clavier pour faire pivoter la pièce, la déplacer à gauche, à droite ou la faire descendre plus rapidement.

4. lockPiece() : Verrouille la pièce actuelle dans la grille et appelle la fonction pour vérifier et supprimer les lignes complètes.

5. spawnNewPiece() : Génère une nouvelle pièce aléatoire à partir des formes disponibles et l'assigne à `currentPiece`.

6. checkForFullLines() : Vérifie chaque ligne de la grille pour détecter les lignes pleines, les supprime si elles sont complètes et ajoute une nouvelle ligne vide en haut.

5. collision(xOffset, yOffset, testShape) : Vérifie si la pièce (ou une forme test) entre en collision avec les limites de la grille ou d'autres pièces, en tenant compte des décalages fournis.

Grid.js :

Voici un résumé de l'utilité de chaque fonction et variable dans le code que tu as fourni :

1. let rows = 20; : Définit le nombre de lignes dans la grille, ici 20 lignes.

2. let cols = 10; : Définit le nombre de colonnes dans la grille, ici 10 colonnes.

3. let resolution = 30; : Définit la taille (en pixels) de chaque cellule de la grille, ici 30 pixels.

4. let grid = []; : Initialise un tableau vide qui contiendra la grille, où chaque élément représentera une ligne.

5. createGrid() : Crée et initialise la grille en remplissant chaque cellule avec la valeur 0.

6. drawGrid() : Dessine la grille à l'écran en remplissant chaque cellule avec une couleur en fonction de sa valeur, utilisant le blanc pour les cellules vides et la couleur correspondante pour les cellules occupées.