SIMULATEUR PSEUDO ALÉATOIRE DU JEU DE LA VIE

Introduction

Ce code est une implémentation du célèbre Game of Life de John Conway.

Il utilise la bibliothèque SDL2 pour créer une interface graphique et afficher l'évolution des cellules en temps réel.
Le Jeu de la Vie est une simulation basée sur des règles simples permettant de simuler des phénomènes complexes.

Chaque cellule peut être dans l'un des deux états possibles (vivante ou morte), et son état futur dépend des états de ses voisines selon des règles préétablies. Le code utilise une taille de grille de 40x60 et une taille de cellule de 10 pixels.

La configuration initiale de la cellule est générée aléatoirement. Le joueur peut interagir avec la grille en cliquant sur les cellules, en les faisant passer de vivantes à mortes et vice versa, et peut également appuyer sur la barre d'espace pour faire évoluer la grille selon les règles du jeu.

```c
#include <SDL2/SDL.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define ROWS 40
#define COLS 60
#define CELL_SIZE 10

void drawGrid(SDL_Renderer *renderer, int grid[][COLS]) {
    SDL_Rect rect;
    rect.w = CELL_SIZE;
    rect.h = CELL_SIZE;
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
            if (grid[i][j] == 1) {
                SDL_SetRenderDrawColor(renderer, 255, 255, 255, 255);
            } else {
                SDL_SetRenderDrawColor(renderer, 0, 0, 0, 255);
            }
            rect.x = j * CELL_SIZE;
            rect.y = i * CELL_SIZE;
            SDL_RenderFillRect(renderer, &rect);
        }
    }
}

int countNeighbors(int row, int col, int grid[][COLS]) {
    int count = 0;
    for (int i = row - 1; i <= row + 1; i++) {
        for (int j = col - 1; j <= col + 1; j++) {
            if (i >= 0 && i < ROWS && j >= 0 && j < COLS) {
                if (grid[i][j] == 1) {
                    count++;
                }
            }
        }
```

```
    }
    count -= grid[row][col];
    return count;
}

void update(int grid[][COLS]) {
    int newGrid[ROWS][COLS];
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
            int neighbors = countNeighbors(i, j, grid);
            if (grid[i][j] == 1) {
                if (neighbors < 2 || neighbors > 3) {
                    newGrid[i][j] = 0;
                } else {
                    newGrid[i][j] = 1;
                }
            } else {
                if (neighbors == 3) {
                    newGrid[i][j] = 1;
                } else {
                    newGrid[i][j] = 0;
                }
            }
        }
    }
    for (int i = 0; i < ROWS; i++) {
        for (int j = 0; j < COLS; j++) {
            grid[i][j] = newGrid[i][j];
        }
    }
}

void handleInput(SDL_Event *event, int grid[][COLS]) {
    int mouseX, mouseY;
    switch (event->type) {
        case SDL_QUIT:
            exit(0);
            break;
        case SDL_MOUSEBUTTONDOWN:
```

```
 77             mouseX = event->button.x / CELL_SIZE;
 78             mouseY = event->button.y / CELL_SIZE;
 79             grid[mouseY][mouseX] = !grid[mouseY][
    mouseX];
 80             break;
 81         case SDL_KEYDOWN:
 82             if (event->key.keysym.sym ==
    SDLK_SPACE) {
 83                 update(grid);
 84             } else if (event->key.keysym.sym ==
    SDLK_c) {
 85                 for (int i = 0; i < ROWS; i++) {
 86                     for (int j = 0; j < COLS; j
    ++) {
 87                         grid[i][j] = 0;
 88                     }
 89                 }
 90             }
 91     }
 92 }
 93
 94             int main(int argc, char *argv[]) {
 95                 srand(time(NULL));
 96                 SDL_Init(SDL_INIT_VIDEO);
 97
 98                 SDL_Window *window =
    SDL_CreateWindow("Game of Life",
 99
        SDL_WINDOWPOS_UNDEFINED,
100
        SDL_WINDOWPOS_UNDEFINED,
101
        COLS * CELL_SIZE,
102
        ROWS * CELL_SIZE,
103
        SDL_WINDOW_SHOWN);
104
105                 SDL_Renderer *renderer =
    SDL_CreateRenderer(window, -1,
    SDL_RENDERER_ACCELERATED);
106
```

```
                int grid[ROWS][COLS] = {0};

// Initial random configuration
                for (int i = 0; i < ROWS; i++) {
                    for (int j = 0; j < COLS; j
    ++) {
                        grid[i][j] = rand() % 2;
                    }
                }

                while (1) {
                    SDL_Event event;
                    while (SDL_PollEvent(&event
    )) {
                        handleInput(&event, grid);
                    }

                    SDL_SetRenderDrawColor(
    renderer, 0, 0, 0, 255);
                    SDL_RenderClear(renderer);

                    drawGrid(renderer, grid);

                    SDL_RenderPresent(renderer);
                    SDL_Delay(10);
                }

                SDL_DestroyRenderer(renderer);
                SDL_DestroyWindow(window);
                SDL_Quit();

                return 0;
            }
```