Square Division

Nastasă Ioana-Raluca[1], Rebegea Irina[2]

[1, 2] Faculty of Computer Science, "Al. I Cuza" University of Iasi, Romania

Abstract. Game developed in C using procedural programming concepts.

1 Introduction

The task is to make a game in which the user must complete a square with *n* dots of each colour. The most special feature of the project is the display. For each button in the menu, for the instructions and for each setting, a custom image has been designed. Also, in addition to the original version of the game from the book "Mai în glumă, mai în serios" (Nicolae Oprișiu), the user can select the size of the game depending on the desired difficulty. Another advantage of the program is the possibility to choose the language, making it suitable for several categories of people. In addition to this option, from the settings the user can change the shape of the game piece and the background music.

A video presentation of the project (in Romanian) can be found at: https://youtu.be/suHiwXSDpU8

2 Implemented technologies

2.1 Borderland Graphics Interface (BGI)

For the GUI we used the BGI graphic library. BGI is less powerful than modern graphics libraries such as SDL or OpenGL since it was designed for presentation graphics instead of event-based 3D applications. The library loads graphic drivers (*.BGI) and vector fonts (*.CHR) from disk in order to provide device independent graphics support. It is possible for the programmer to embed the graphic driver into the executable file by linking the graphic driver as object code with the aid of a utility provided by the compiler (bgiobj.exe).

3 Application design

3.1 Functions description

Name	Description
choose_level()	Loads the level chosen by the user.
print_winning_message()	Prints the message "You won!" in either Romanian, English or French.

print_losing_message()	Prints the message "Try again!" in either Romanian, English or French.
draw_game_piece()	Helper function that draws a piece of a specified colour.
place_game_piece()	Places a piece where the user clicks, if they click inside the square.
draw_square()	Draws the square and the initial pieces for each level.
check_solution()	Checks if the solution provided by the user is correct.
print_language()	Prints the menu in the selected language.

3.2 Application diagram

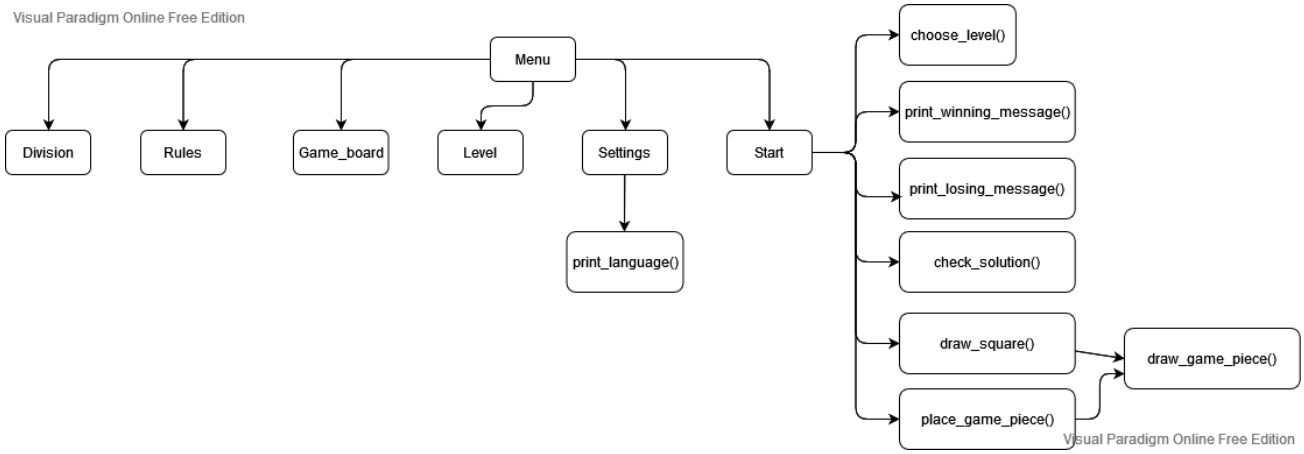

Figure 3.1: Application diagram

4 Implementation details

4.1 Menu description

4.2 The menu contains six labels (division, rules, game_board, level, settings, start), translated into five buttons. The first one is the instructions button, which shows directions on how to play the game. Next comes the game board button, where the user can choose the size of the square according to the desired difficulty (4x4, 6x6 or 8x8). The level button allows the user to select one of the four possible levels for each difficulty. The settings allow changing the language of the game (English, French or Romanian), the shape of the pieces (circle or square) and the background sound (choosing between two songs and no sound). Once the game is started, the user can check themselves along the way, receiving feedback on the uploaded solution. They can also return to the main menu at any time pressing the back button, at which point their progress resets.

4.3 Relevant pieces of code

The following code section implements **check_solution**() function:

```
int check_solution(int a[MAX_DIMENSION][MAX_DIMENSION])
{
   /*
   * Checks if the solution provided by the user is correct.
   */
   int correct = 1;
   int fr[MAX_DIMENSION] = {0};
   for(int i = 1; i <= square_size; i++)
      for(int j =1; j <= square_size; j++) {
         if(a[i][j] == 0)
            /* if there is an empty cell */
            correct = 0;

         if(a[i][j] > 0
           && a[i][j] != abs(a[i][j + 1])
           && a[i][j] != abs(a[i + 1][j])
           && a[i][j] != abs(a[i][j - 1])
           && a[i][j] != abs(a[i - 1][j]))
           /* the color of a circle must be the same as the color of its neighbors */
            correct = 0;

         fr[a[i][j]]++;
      }
```

```
    for(int i = 1; i <= square_size; i++)
      if(fr[i] != square_size - 1)
        correct = 0;

    return correct;
  }
```

The following code section implements **place_game_piece()** function:

```
  void place_game_piece()
  {
    /*
    * Places a piece where the user clicks, if they click inside the square.
    */
    int line, column;
    int x, y;
    if(ismouseclick(WM_LBUTTONDOWN)) {
      clearmouseclick(WM_LBUTTONDOWN);

      x = mousex();
      y = mousey();

      if (!(x >= LEFT && x <= LEFT + WIDTH && y >= UP && y <= UP +
HEIGHT))
        click_outside_square = true;

      else {
        line = (y - UP) / square_side + 1;
        column = (x - LEFT) / square_side + 1;
        if (game_board[line][column] >= 0) {
          /*
          == 0 means the cell is empty,
          > 0 means the cell already contains a piece,
          which can be of a different color than the selected one
          */
          game_board[line][column] = color_index;
          a[line][column] = color_index;
          draw_game_piece(line, column, color_index);
        }

        else if (game_board[line][column] < 0) {
          /* takes the color of a piece which was on the board from the beginning */
          color_index = - game_board[line][column];
        }
      }
    }
```

```
    if(ismouseclick(WM_RBUTTONDOWN)) {
        clearmouseclick(WM_RBUTTONDOWN);
        x = mousex();
        y = mousey();
        line = (y - UP) / square_side + 1;
        column = (x - LEFT) / square_side + 1;
        if (game_board[line][column] > 0) {
            game_board[line][column] = 0;
            a[line][column] = 0;
            draw_game_piece(line, column, 0);
        }
    }
}
```

5 Conclusions

The application can be improved by displaying the correct solutions of the square upon user request. Also, a function could be implemented to randomize the position of the pieces on the board to create more levels.

6 References

https://profs.info.uaic.ro/~introp/index.html
https://www.edusoft.ro/fisiere/Nicolae%20Oprisiu%20-%20Mai%20in%20gluma,%20mai%20in%20serios%20(1981).pdf