

student: Bartosz Krasuski 1T4
B.Krasuski@stud.elka.pw.edu.pl
prowadzący: mgr inż. Krzysztof Dowalla
mgr inż. Monika Drabik

**Podstawy Programowania (PRM) – projekt
dokumentacja projektu**

Temat: Implementacja gry w życie.

Wykorzystane biblioteki niestandardowe: w programie została użyta biblioteka graficzna CSFML.

Możliwe modyfikacje programu w nagłówkowym pliku konfiguracyjnym config.h:

- rozdzielczość okna programu – dyrektywy WIDTH, HEIGHT,
- wielkość komórek – dyrektywy BWIDTH, BHEIGHT,
- odległość pomiędzy komórkami – dyrektywa SPACING,
- kolor żywych komórek – RED1, GREEN1, BLUE1,
- kolor martwych komórek – RED0, GREEN0, BLUE0,
- wielkość marginesów (z założenia gra w życie ma nieskończenie dużą planszę, margines określa jak wiele komórek generowanych jest poza widokiem okna programu) – dyrektywa OFFSET.

Napotkane problemy: niepoprawne wykrywanie naciśnięć klawiszy na klawiaturę przez bibliotekę graficzną.

Rozwiązanie: naciśnięcie danego klawisza jest sprawdzane na bieżąco, powtórzenie akcji klawisza zostało opóźnione przez funkcję usleep z biblioteki unistd.

Algorytm rozwiązania zadania:

procedura tworzenie nowej generacji

dane wejściowe wskaźnik **cells** na zmienną strukturalną z dwiema tablicami dwuwymiarowymi stanów komórek

Generation[COLS][ROWS], **parentGeneration[COLS][ROWS]**, gdzie COLS i ROWS to definicje preprocesora oznaczające wyliczoną liczbę kolumn i wierszy, **alive** jest zmienną informującą ile żywych sąsiadów ma komórka.

```
{
    dla każdego i ∈ [1, COLS-1)
        dla każdego j ∈ [1, ROWS-1)
            alive = cells->parentGeneration[i-1][j-1] +
                    cells->parentGeneration[i][j-1] +
                    cells->parentGeneration[i+1][j-1] +
                    cells->parentGeneration[i-1][j] +
                    cells->parentGeneration[i+1][j] +
                    cells->parentGeneration[i-1][j+1] +
                    cells->parentGeneration[i][j+1] +
                    cells->parentGeneration[i+1][j+1]

            jeśli cells->parentGeneration[i][j] == 0
                jeśli alive == 3
                    cells->Generation[i][j] = 1

            w przeciwnym przypadku
                jeśli alive < 2
                    cells->Generation[i][j] = 0
                w przeciwnym wypadku
                    jeśli alive > 3
                        cells->Generation[i][j] = 0

                    w przeciwnym wypadku
                        cells->Generation[i][j] = 1;

    dla każdego i ∈ [0, COLS)
        dla każdego j ∈ [0, ROWS)
            cells->parentGeneration[i][j] = cells->Generation[i][j];

    disp();
}
```

Struktury danych:

Program będzie się posługiwał wskaźnikiem na zmienną strukturalną zawierającą dwiema głównymi dwuwymiarowymi tablicami ze stanami komórek. Żywa komórka, w odpowiedniej generacji, będzie symbolizowana przez wartość 1, a martwa przez wartość 0.

Podział na funkcje:

- **int init()** – funkcja rezerwująca w pamięci miejsce na zmienną strukturalną, wywołuje funkcję `initgraphics()`, daje ziarno dla generatora liczb pseudolosowych,
- **void randomize()** – funkcja generująca losową matrycę komórek,
- **void create()** – funkcja tworząca żywą komórkę w miejscu wskazanym przez mysz,
- **void destroy()** – funkcja niszcząca komórkę w miejscu wskazanym przez mysz,
- **void clear()** – funkcja resetująca matrycę i licznik generacji,
- **void generale()** – funkcja realizująca mechanizm gry w życie,
- **int counter()** – funkcja zwraca numer generacji,
- **int returnflag()** – funkcja zwraca stan flagi tworzenia,
- **void switchflag()** – funkcja zmienia stan flagi tworzenia na przeciwny,
- **int initgraphics()** – funkcja inicjalizuje główne okno programu, kształt za pomocą którego będzie rysowana matryca komórek, wywołuje funkcję `textconfig()`,
- **int textconfig()** – funkcja inicjalizująca wyświetlanie licznika generacji w oknie programu,
- **void textdraw()** – funkcja rysująca numer generacji w oknie programu,
- **void clean()** – funkcja usuwająca zmienne stworzone na potrzeby biblioteki graficznej, dealokuje zmienną strukturalną `cells`.
- **void disp(struct Cells* cells)** – funkcja rysująca i wyświetlająca matrycę komórek.
- **int main()** – główna funkcja programu, wywołuje funkcję `init()`, realizuje obsługę zdarzeń:
 - naciśnięcia klawiszy klawiatury:
 - R – generuje losową matrycę komórek, funkcja `randomize()`,
 - C – zeruje matrycę komórek, funkcja `clear()`,
 - spacja – zmienia stan flagi tworzenia, funkcja `switchflag()`,
 - naciśnięcia przycisków myszy:
 - lewy przycisk myszy – tworzy żywą komórkę we wskazanym miejscu, funkcja `create()`,
 - lewy przycisk myszy – niszczy żywą komórkę we wskazanym miejscu, funkcja `destroy()`, jeśli flaga tworzenia ma wartość 1, wywołuje funkcję `generale()`.

Sposób testowania

Wygenerowane matrycę były porównywane z innymi implementacjami gry w życie dostępnymi w sieci Internet.

Źródła:

1. Wikipedia, https://pl.wikipedia.org/wiki/Gra_w_życie
2. <https://www.mimuw.edu.pl/~ajank/zycie>