

POLITECHNIKA WARSZAWSKA INSTYTUT TECHNIKI LOTNICZEJ I MECHANIKI  
STOSOWANEJ ZAKŁAD AERODYNAMIKI

**INFORMATYKA I**  
**PROJEKT DOMOWY – GRA W WĘŻA**

Wykonawca:

Dominika Ziółkiewicz

11 styczeń 2021

## 1. Wstęp

Przedmiotem mojej pracy jest opracowanie i realizacja popularnej gry w węża. Została ona wydana po raz pierwszy w październiku 1976 roku pod nazwą „Blockade” i spopularyzowana na przełomie XX i XXI wieku dzięki wersjom dla telefonów komórkowych firmy Nokia. Polega na tym, że gracz kontroluje stworzenie podobne do węża, poruszając się po obramowanej planszy i zbierając jedzenie. Cały czas musi uważać na to, aby głowa węża nie uderzyła o żadną ze ścian ani o pozostałą część ciała. Trudność gry tkwi w tym, że wąż jest w ciągłym ruchu i robiąc się coraz dłuższy, staje się cięższym do kontroli.

## 2. Opis działania

### a. Przyjęte metody

Sam wąż jest złożony z kół o promieniu  $r = 5$  w kolorze niebieskim. Każde z nich jest przytrzymywane w postaci współrzędnych w tablicach statycznych  $x[485]$  i  $y[485]$ . Rozmiar okna graficznego to  $220 \times 220$ . Maksymalny rozmiar węża to  $\lceil 220 / 2r \rceil^2 = 484$ . Poruszanie wężem odbywa się przez przyciskanie na klawiaturze odpowiednio „a” – lewo, „w” – góra, „s” – dół, „d” – prawo. W każdym ruchu głowa węża jest rysowana w odpowiednim miejscu, zależnie od obranego kierunku, współrzędnym części ciała przypisuje się współrzędne następnej części, natomiast ogon zostaje usuwany poprzez rysowanie w jego miejscu koła w kolorze tła.

Jedzenie, które się zdobywa, też jest kołem w kolorze jasnopurpurowym, a jego współrzędne są losowane w taki sposób, aby nigdy nie zdarzyła się sytuacja, że głowa najeżdża na np. pół jedzenia a pół zostaje. Dlatego losowana współrzędna jest z przedziału 1 - 20 (nie może być przedział 1 – 22, ponieważ okno graficzne nie jest prawidłowo wykalibrowane i jedzenie mogłoby pojawiać się w połowie w oknie a w połowie poza nim) i mnożona przez  $2r$ . Po zdobyciu jedzenia, losowane jest następne, zwiększa się liczba zdobytych punktów i liczba elementów ciała węża oraz dorysowywany jest ogon. Podczas rozgrywki aktualna liczba zdobytych punktów wyświetla się na bieżąco w konsoli.

Mechanizm rekordów działa następująco. Najpierw zostaje sprawdzone, czy jest to pierwsza gra użytkownika. Jeśli tak, to zostaje tworzony plik „wyniki.txt”. Jeśli jednak plik już istnieje, to zostaje wczytana z niego największa wartość, która oznacza aktualny rekord gracza.

Gra kończy się w sytuacji, gdy użytkownik najeździe na którąś z krawędzi, lub inną część ciała. W tym momencie zostaje wyświetlone „GAME OVER” oraz liczba zdobytych punktów.

Jeżeli jest ona większa, niż dotychczasowy największy wynik, to zostaje wpisana do pliku „wyniki.txt” i ponadto wyświetla się, że użytkownik pobił swój rekord.

#### b. Zastosowane narzędzia

Program był pisany w języku C w środowisku Visual Studio IDE. Wykorzystane zostały następujące biblioteki: „`stdio.h`”, „`stdlib.h`”, „`Windows.h`”, „`time.h`” oraz graficzna „`winbgi2.h`”

### 3. *Sposób użycia, specyfikacja wejścia*

Do obsługi programu niezbędne jest użycie wyżej wymienionych bibliotek. Aby móc grać, użytkownik musi przyciskać na klawiaturze przyciski „W A S D” oznaczające odpowiedni kierunek ruchu węża. Plik „wyniki.txt” zostanie stworzony automatycznie w folderze z programem, dlatego nie ma potrzeby tworzenia go manualnie.

### 4. *Specyfikacja wewnętrzna*

#### a. Używane zmienne, tablice i wskaźniki

`int` a = 0, b = 0, k = 1 zmienne pomocnicze

`int` r = 5 promień koła (części węża i jedzenia)

`int` n = 5 liczba elementów węża

`int` c = 68 domyślny kierunek ruchu (68 to "D" w ASCII)

`int` rec = 0 przechowuje aktualny rekord

`int` points = 0 przechowuje aktualną liczbę punktów w każdej rozgrywce

`int` x[485], y[485] tablice przechowujące współrzędne każdego elementu węża

`int*` w = &points

`int*` p = &a

`int*` g = &b

`int*` re = &rec;

`FILE*` f wskazuje na plik „wyniki.txt”

#### b. Używane funkcje

`void` init(`int*` x, `int*` y, `int` n, `int` r) inicjalizacja węża

`void` head(`int` a, `int` b, `int` r) rysowanie głowy w innym miejscu

`void` body(`int*` x, `int*` y, `int` n) przesuwanie ciała

`void tail(int a, int b, int r)` usuwanie starego ogona

`void move(int * x, int * y, int n, int r, int c)` poruszanie węzłem w zależności od podanego kierunku

`void losuj(int * a, int * b)` losowanie współrzędnych dla jedzenia

`void edges_check(int * x, int * y, int * w, int * re, FILE* f)` sprawdzenie czy węź wyszedł poza okno graficzne

`void body_check(int * x, int * y, int n, int * w, int * re, FILE * f)` sprawdzenie czy głowa węża nie zderza się z ciałem

`void end(int * re, int * w, FILE* f)` zakończenie rozgrywki

### c. Pseudokod algorytmu

#### INIT (X, Y, n, r)

Ustaw kolor -> Niebieski

X[0] <- 40

X[1] <- 30

X[2] <- 20

X[3] <- 10

X[4] <- 0

Y[0] = Y[1] = Y[2] = Y[3] = Y[4] <- 10

Dla i <- 0 do n

Rób Narysuj koło o współrzędnych (X[i],Y[i]) i promieniu r  
Wypełnij koło na niebiesko

Dla i <- n do 485

Rób X[i] <- -1

Y[i] <- -1

#### HEAD (a, b, r)

Ustaw kolor -> Niebieski

Narysuj koło o współrzędnych (a, b) i promieniu r

Wypełnij koło na niebiesko

#### BODY (X, Y, n)

Dla i <- n-1 do 1

Rób X[i] <- X[i-1]

Y[i] <- Y[i-1]

#### TAIL (a, b, r)

Ustaw kolor -> Zielony

Narysuj koło o współrzędnych (a, b) i promieniu r

Wypełnij koło na zielono

**MOVE** (X, Y, n, r, c)

Jeżeli  $c = 68$  lub  $c = 100$

To **TAIL** (X[n-1], Y[n-1], r)

**BODY** (X, Y, n)

$X[0] \leftarrow X[0] + 2 * r$

**HEAD** (X[0], Y[0], r)

Jeżeli  $c = 115$  lub  $c = 83$

To **TAIL** (X[n-1], Y[n-1], r)

**BODY** (X, Y, n)

$Y[0] \leftarrow Y[0] + 2 * r$

**HEAD** (X[0], Y[0], r)

Jeżeli  $c = 97$  lub  $c = 65$

To **TAIL** (X[n-1], Y[n-1], r)

**BODY** (X, Y, n)

$X[0] \leftarrow X[0] - 2 * r$

**HEAD** (X[0], Y[0], r)

Jeżeli  $c = 87$  lub  $c = 119$

To **TAIL** (X[n-1], Y[n-1], r)

**BODY** (X, Y, n)

$Y[0] \leftarrow Y[0] - 2 * r$

**HEAD** (X[0], Y[0], r)

**LOSUJ** (a, b)

\*a <- randomowa liczba z przedziału 1-20

\*b <- randomowa liczba z przedziału 1-20

\*a <- a \* 10;

\*b <- b \* 10;

**EDGES\_CHECK** (X, Y, \*w, \*re, \*f)

Jeżeli  $x[0] > 210$

To **END** (re, w, f)

Jeżeli  $x[0] \leq 0$

To **END** (re, w, f)

Jeżeli  $y[0] \geq 210$

To **END** (re, w, f)

Jeżeli  $y[0] < 10$

To **END** (re, w, f)

**BODY\_CHECK** (X, Y, n, \*w, \*re, \*f)

Zadeklaruj zmienną całkowitą:  $k \leftarrow 1$

Dopóki  $k < n$

Jeżeli  $x[0] = x[k]$

To Jeżeli  $y[0] = y[k]$

To **END** (re, w, f)

$k \leftarrow k + 1$

```

END (*re, *w, *f)
    Jeżeli *w = 1 //liczba punktów
        To Wyświetl „GAME OVER”
        Przejdź do następnej linijki
        Wyświetl „You scored <points> point”
        Przejdź do następnej linijki x2
    Jeżeli nie
        To Wyświetl „GAME OVER”
        Przejdź do następnej linijki
        Wyświetl „You scored <points> points”
        Przejdź do następnej linijki x2
    Jeżeli *w > *re
        To Wyświetl „Congratulations!!! You have a new record!!”
        Przejdź do następnej linijki
        Otwórz plik „wyniki.txt” w trybie dodawania
        Dodaj do pliku „wyniki.txt” *w //wartość zmiennej points
        Zamknij plik „wyniki.txt”

```

Zakończ działanie programu

## MAIN()

Zadeklaruj zmienne całkowite: a = 0, b = 0, k = 1, r = 5, n = 5, c = 68, rec = 0, points = 0  
 Zadeklaruj tablice o wartościach całkowitych: x[485], y[485]  
 Zadeklaruj wskaźniki: int\* p = &a, int\* g = &b, int\* w = &points, int\* re = &rec, FILE\* f

Otwórz plik „wyniki.txt” w trybie czytania

Jeżeli f=NULL

To Otwórz plik „wyniki.txt” w trybie pisania //utworzy pusty plik

Jeżeli nie to:

Dopóki plik się nie skończy

Wczytaj b

Jeżeli b > rec

To rec <- b

Zamknij plik „wyniki.txt”

Utwórz okno graficzne o wymiarach (220,220)

Ustaw kolor -> Zielony

Wypełnij okno zielonym kolorem

INIT (X, Y, n, r)

LOSUI (p, g)

Dopóki k<n

Jeżeli x[k] = a i y[k] = b

To LOSUI (p, g)

k <- -1

k <- k + 1

Ustaw kolor -> Czerwony

Narysuj koło o współrzędnych (a, b) i promieniu r

Ustaw kolor -> Jasnopurpurowy

Wypełnij koło na jasnopurpurowo

Zaczekaj 1000ms

Dopóki 1

Jeżeli został wciśnięty przycisk  
To c <- znak przycisku

MOVE (X, Y, n, r, c)

Jeżeli x[0] = a i y[0] = b

To n <- n + 1

points <- points + 1

X[n - 1] <- X[n - 2]

Y[n - 1] <- Y[n - 2]

Narysuj koło o współrzędnych (X[n-1], Y[n-1]) i promieniu r

Ustaw kolor -> Niebieski

Wypełnij koło na niebiesko

k <- 0

LOSUJ (p, g)

Dopóki k < n

Jeżeli x[k] = a i y[k] = b

To LOSUJ (p, g)

k <- -1

k <- k + 1

Ustaw kolor -> Czerwony

Narysuj koło o współrzędnych (a, b) i promieniu r

Ustaw kolor -> Jasnopurpurowy

Wypełnij koło na jasnopurpurowo

Wyświetl aktualną liczbę punktów

EDGES\_CHECK (X, Y, \*w, \*re, \*f)

BODY\_CHECK (X, Y, n, \*w, \*re, \*f)

Zaczekaj 150ms

## 5. Kod programu

```
#define _CRT_SECURE_NO_WARNINGS
```

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
#include <Windows.h>
```

```
#include <time.h>
```

```
#include "winbgi2.h"
```

```
void init(int* x, int* y, int n, int r); //inicjalizacja węża
```

```
void head(int a, int b, int r); //rysowanie głowy w innym miejscu
```

```
void body(int* x, int* y, int n); //przesuwanie ciała
```

```
void tail(int a, int b, int r); //usuwanie starego ogona
```

```
void move(int* x, int* y, int n, int r, int c); //poruszanie wężem
```

```
void losuj(int* a, int* b); //losowanie współrzędnych dla jedzenia
```

```
void edges_check(int* x, int* y, int* w, int* re, FILE* f); //sprawdzenie czy węź wyszedł  
poza okno graficzne
```

```

void body_check(int* x, int* y, int n, int* w, int* re, FILE* f); //sprawdzenie czy
głowa węża nie zderza się z ciałem
void end(int* re, int* w, FILE* f); //zakończenie rozgrywki

void main()
{
    MoveWindow(GetConsoleWindow(), 300, 100, 800, 500, TRUE);
    int a = 0, b = 0, k = 1; //zmiennne pomocnicze
    int r = 5; //promień koła (części węża i jedzenia)
    int n = 5; //liczba elementów węża
    int c = 68; //domyślny kierunek ruchu (68 to "D" w ASCII)
    int x[485], y[485]; //tablice przechowujące współrzędne każdego elementu węża
    int* p = &a;
    int* g = &b;
    int rec = 0; //przechowuje aktualny rekord
    int points = 0; //przechowuje aktualną liczbę punktów w każdej rozgrywce
    int* w = &points;

    FILE* f = fopen("wynik.txt", "r");
    if (f == NULL)
    {
        f = fopen("wynik.txt", "w"); //utworzenie pliku wyniki.txt
        fclose(f);
    }
    else
    {
        while (!feof(f)) //wczytanie największego rekordu z pliku z wynikami
        {
            fscanf(f, "%d", &b);
            if (b > rec)
                rec = b;
        }
        fclose(f);
    }
    int* re = &rec;

    graphics(220, 220); //rysowanie okna graficznego i kolorowanie go na zielono
    setfillstyle(SOLID_FILL, GREEN);
    floodfill(1, 1, WHITE);

    init(x, y, n, r);
    srand(time(NULL));

    losuj(p, g); //rysowanie jedzenia w losowym miejscu
    k = 0;
    while (k < n)
    {
        if (x[k] == a && y[k] == b)
        {
            losuj(p, g);
            k = -1;
        }
        k++;
    }
    setcolor(RED);
    circle(a, b, r);
    setfillstyle(SOLID_FILL, LIGHTMAGENTA);
    floodfill(a, b, BLUE);

    Sleep(1000);
}

```



```

while (1)
{
    if (kbhit())          //sprawdzenie czy użytkownik wcisnął przycisk
    {
        c = getch(); //wczytanie jaki to przycisk
    }                  //obsługiwane są W A S D

    move(x, y, n, r, c);

    if (x[0] == a && y[0] == b) //sprawdzenie czy głowa najeżdża na jedzenie
    {
        n++;
        points++;
        x[n - 1] = x[n - 2];
        y[n - 1] = y[n - 2];
        circle(x[n - 1], y[n - 1], r);
        setfillstyle(SOLID_FILL, BLUE);
        floodfill(x[n - 1], y[n - 1], BLUE);

        losuj(p, g);
        k = 0;
        while (k < n)
        {
            if (x[k] == a && y[k] == b)
            {
                losuj(p, g);
                k = -1;
            }
            k++;
        }
        setcolor(RED);          //rysowanie nowego jedzenia
        circle(a, b, r);
        setfillstyle(SOLID_FILL, LIGHTMAGENTA);
        floodfill(a, b, BLUE);
    }
    printf("%d\n", points);      //wyświetlanie aktualnego wyniku
    edges_check(x, y, w, re, f);
    body_check(x, y, n, w, re, f);

    Sleep(150);
}

wait();
}
void init(int* x, int* y, int n, int r)
{
    setcolor(BLACK);
    x[0] = 40;                  //przypisanie współrzędnych pierwszemu węzowi
    x[1] = 30;
    x[2] = 20;
    x[3] = 10;
    x[4] = 0;
    y[0] = y[1] = y[2] = y[3] = y[4] = 10;

    for (int i = 0; i < n; i++) //rysowanie początkowego węzła
    {
        circle(x[i], y[i], r);
        setfillstyle(SOLID_FILL, BLUE);
        floodfill(x[i], y[i], BLUE);
    }
    for (int i = n; i < 485; i++) //pozostałe współrzędne przyjmują wartość "-1"
    {

```

```

        x[i] = -1;
        y[i] = -1;
    }
}

void head(int a, int b, int r)
{
    setcolor(BLACK);                //rysowanie nowego koła (głowy)
    circle(a, b, r);
    setfillstyle(SOLID_FILL, BLUE);
    floodfill(a, b, BLUE);
}

void body(int* x, int* y, int n)
{
    for (int i = n - 1; i > 0; i--) //przepisywanie współrzędnych
    {
        x[i] = x[i - 1];
        y[i] = y[i - 1];
    }
}

void tail(int a, int b, int r)
{
    setcolor(0.5); //usunięcie ogona, rysowanie zielonego koła w jego miejsce
    circle(a, b, r);
    floodfill(a, b, BLUE);
}

void move(int* x, int* y, int n, int r, int c)
{
    if (c == 68 || c == 100) //ruch w prawo, "d" lub "D"
    {
        tail(x[n - 1], y[n - 1], r);
        body(x, y, n);
        x[0] += 2 * r;
        head(x[0], y[0], r);
    }
    if (c == 115 || c == 83) //ruch w dół, "s" lub "S"
    {
        tail(x[n - 1], y[n - 1], r);
        body(x, y, n);
        y[0] += 2 * r;
        head(x[0], y[0], r);
    }
    if (c == 97 || c == 65) //ruch w lewo, "a" lub "A"
    {
        tail(x[n - 1], y[n - 1], r);
        body(x, y, n);
        x[0] -= 2 * r;
        head(x[0], y[0], r);
    }
    if (c == 87 || c == 119) //ruch w górę, "w" lub "W"
    {
        tail(x[n - 1], y[n - 1], r);
        body(x, y, n);
        y[0] -= 2 * r;
        head(x[0], y[0], r);
    }
}

void losuj(int* a, int* b)

```

```

{
    *a = 1 + rand() % 20; //współrzędne mają być z przedziału 1 - 20
    *b = 1 + rand() % 20;
    *a *= 10;           //mnożenie *2r żeby jedzenie zostało "zjedzone w całości"
    *b *= 10;
}

void edges_check(int* x, int* y, int* w, int* re, FILE* f)
{
    if (x[0] > 210) //wyjście poza prawą krawędź
    {
        end(re, w, f);
    }
    if (x[0] <= 0) //wyjście poza lewą krawędź
    {
        end(re, w, f);
    }
    if (y[0] >= 210) //wyjście poza dolną krawędź
    {
        end(re, w, f);
    }
    if (y[0] < 10) //wyjście poza górną krawędź
    {
        end(re, w, f);
    }
}

void body_check(int* x, int* y, int n, int* w, int* re, FILE* f)
{
    int k = 1;
    while (k < n)
    {
        if (x[0] == x[k]) //jeśli głowa jest w tym samym miejscu co część ciała
            if (y[0] == y[k])
            {
                end(re, w, f);
            }
        k++;
    }
}

void end(int* re, int* w, FILE* f)
{
    if (*w == 1)
    {
        printf("GAME OVER\nYou scored %d point\n\n", *w);
    }
    else printf("GAME OVER\nYou scored %d points\n\n", *w);
    if (*w > *re) //oznacza że rekord został właśnie pobity i zapisany do pliku
    {
        printf("Congratulations!!! You have a new record!!\n");
        f = fopen("wynik.txt", "a");
        fprintf(f, "%d\n", *w);
        fclose(f);
    }
    exit(1);
}

```