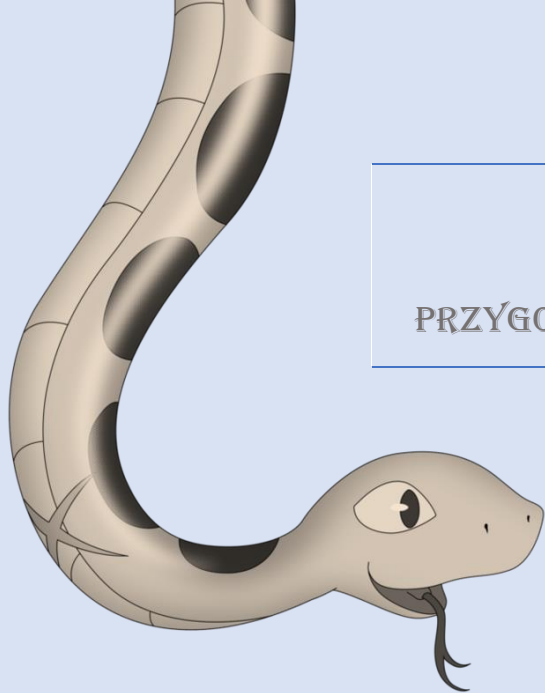




INYOKA

PRZYGODY WESOŁEGO WĘŻA



1. Założenia gry

Gra „Inyoka: Przygody wesołego węża” jest prostą grą, której wzorzec pochodzi od gier typu „snake”.

W grze gracz kontroluje długie i cienkie stworzenie, podobne do węża, które porusza się po obramowanej planszy zbierając jedzenie, próbując nie uderzyć własną głową o ściany otaczające planszę gry, a także o część własnego ciała. Kiedy wąż zje kawałek jedzenia, jego ogon robi się coraz dłuższy, co utrudnia grę. Gracz kontroluje kierunek ruchu węża za pomocą klawiszy WASD. Gracz nie może zatrzymać węża, gdy gra jest w toku.

2. Grafika

Gra została stworzona przy pomocy języka cpp i z założeń miała być grą konsolową, dlatego elementy grafiki zostały zastąpione znakami ASCII. I tak, plansza otoczona jest znakami „#”, jedzenie „F”, głowa węża „O” a jego tułów znakiem „o”.

W aspekcie graficznym zostało stworzone menu oraz łączące się z nim opcje gry.



3. Mechanika gry oraz kod

Gra nie zawiera olbrzymie ilości wierszy, jednak nawet w tak małym projekcie znalazło się miejsce na różnorakie funkcje.

- a) `clearScreen()` – funkcja pozwala na odświeżanie planszy bez użycia funkcji „`cls`” co pozwala na znaczące podniesienie wydajności gry oraz na brak ciągle widocznego kursora odświeżania

```
{  
    HANDLE output = GetStdHandle(STD_OUTPUT_HANDLE);  
    COORD pos = { 0,0 };  
    SetConsoleCursorPosition(output, pos);  
}
```

- b) `score_add()` – punkty nie są naliczane jednakowo. To od ustawień przyjętych przez gracza punkty będą się inaczej sumować. Zależy to m.in. od wielkości planszy, prędkości węża czy też możliwości przenikania przez ściany.

```
{  
    if (h*w < 150) a = 2;  
    else a = 1;  
    if (time_snake < 40) b = 2;  
    else b = 1;  
    if (przenikanie == 0) c = 3;  
    else c = 1;  
    score = score + 10 * a * b * c;  
}
```



- c) `drawingMap()` – w tej oraz w kilku innych funkcjach został dodany element, który nieco ubarwi nam rozgrywkę, mianowicie kolory. Dzięki temu „czarno na białym” widać jak od planszy kolorystycznie różni się nasz wąż. Poniżej fragment kodu, w którym zostało to dodane.

```
HANDLE hOut = GetStdHandle(STD_OUTPUT_HANDLE);
clearScreen();
for (int i = 0; i < w + 2; i++) {
    cout << "#";
}
cout << endl;
for (int i = 0; i < h; i++) {
    for (int j = 0; j < w; j++) {
        if (j == 0)
            cout << "#";
        if (i == y && j == x) {
            SetConsoleTextAttribute(hOut, FOREGROUND_GREEN);
            cout << "O";
            SetConsoleTextAttribute(hOut, FOREGROUND_RED);
        }
        else if (i == fruitY && j == fruitX)
            cout << "*";
        else {

            bool print = false;
            for (int k = 0; k < nTail; k++) {
                if (tailX[k] == j && tailY[k] == i) {
                    SetConsoleTextAttribute(hOut, FOREGROUND_GREEN);
                    cout << "o";
                    SetConsoleTextAttribute(hOut, FOREGROUND_RED);
                    print = true;
                }
            }
        }
    }
}
```

- d) `Input()` – w tej funkcji zostało zastosowane takie elementy jak „`_kbhit`”, „`switch(_getch)`” czy „`enum class`”.

```
if (_kbhit()){
    switch (_getch())
    {
        case 'a':
            movement = l;
            break;

        case 'd':
            movement = p;
            break;

        case 'w':
            movement = g;
            break;

        case 's':
            movement = d;
            break;

        case 'x':
            gameOver = true;
            break
    }
}
```

