

Return To The Snake

Filip Strózik

Gliwice, 30 stycznia 2017

1 Opis programu

"Return To The Snake" to gra, powracająca do koncepcji, legendarnego już, "Snake". W grze wcielamy się w głodnego węża, który w poszukiwaniu pokarmu przemierza planszę, omijając przeszkody, ściany, a także uważając, żeby nie zjeść samego siebie!

Wymagania:

1. System operacyjny z rodziny Windows: xp, vista, 7, 8, 8.1, 10
2. Klawiatura

2 Instrukcja obsługi

Aby rozpocząć niesamowitą przygodę z "Return to the Snake" wystarczy uruchomić plik "Return To The Snake.exe". Po uruchomieniu aplikacji, ukaże nam się menu wyboru, w którym, używając strzałki w górę i w dół na klawiaturze, możemy wybrać interesującą nas opcję spośród: "Stwórz poziom", "Gotowy poziom", "Wyjdź". Wybór zatwierdzić możemy enter'em. Po wybraniu opcji "Wyjdź", gra zakończy swoje działanie. "Gotowy poziom" pozwoli nam rozpocząć grę z gotowymi ustawieniami spośród dwóch dostępnych map. "Stwórz poziom" przeniesie nas do menu kreowania własnej rozgrywki, w którym możemy ustawić szerokość, wysokość, kolizje ze ścianami (przy wejściu w ścianę, wąż przechodzi na drugą stronę planszy), przeszkody na mapie. Każdą z tych czynności możemy anulować przyciskiem esc, oraz zatwierdzić enter'em. Z poziomem "Stwórz poziom", możemy rozpocząć grę, najeżdżając kursorem na "Graj" i zatwierdzając enter'em, lub wybrać opcję "Edytor mapy", gdzie gracz może postawić lub usunąć przeszkody z pola, przy użyciu przycisku enter.

3 Specyfikacja techniczna

Zalecane aplikacje:

1. System Windows 10
2. Microsoft Visual Studio 2015

Podział na pliki: Gra.h Gra.c
Konsola.h Konsola.c
Main.h Main.c
Menu.h Menu.c
resource.h
icon1.ico
Resource.rc

Gra - zawiera funkcje silnika rozgrywki oraz funkcje dostępu do tablicy pole

Konsola - zawiera funkcje z biblioteki Windows.h oraz conio.h

Main - zawiera funkcję main() oraz pętlę gry

Menu - zawiera funkcje menu głównego oraz pochodnych

resource.h - zawiera informacje wywołania ikony aplikacji

icon1.ico - zawiera ikonę aplikacji w formacie *.ico

Resource.rc - zawiera informacje na temat ikony programu

4 Szczegóły techniczne

W celu wyeliminowania migotania ekranu, spowodowanego użyciem funkcji system("CLS"), użyty został algorytm dorysowywania kolejnego, nowego, elementu węża i usuwania najstarszego, co imituje ruch węża, jednocześnie odświeżając tylko interesujące nas obszary ekranu.

```
// funkcja wyswietla glowe weza, dodawanie jej do pola i
historii
void WywietlGlowe(int *XWeza, int *YWeza, int *XMapy, int
    *YMapy, int *liczbaObrotowPetli, int *dlugoscWeza)
{
    *XMapy = *XWeza * 2 + 1;
    *YMapy = *YWeza + 1;
    IdzDo(&XMapy, &YMapy);
    printf(" "); // wyswietlanie wyswietlanieglowy weza

    historiaX[*liczbaObrotowPetli] = *XWeza;
    historiaY[*liczbaObrotowPetli] = *YWeza;
}
```

```

// funkcja odpowiada za czyszczenie glowy weza
void UsunOgon(int *XMapy, int *YMapy, int *liczbaObrotowPetli,
int *dlugoscWeza)
{
    if (*dlugoscWeza >= 4) //warunek pozwala nie usuwac
        ogona, zanim nie zostanie wygenerowany caly
    {
        if (*liczbaObrotowPetli < *dlugoscWeza)
        {
            *XMapy = 2 * historiaX[abs(100 -
            (*dlugoscWeza -
            *liczbaObrotowPetli))] + 1;
            *YMapy = historiaY[abs(100 -
            (*dlugoscWeza -
            *liczbaObrotowPetli))] + 1;
        }
        else
        {
            *XMapy = 2 *
            historiaX[abs(*liczbaObrotowPetli -
            *dlugoscWeza)] + 1;
            *YMapy =
            historiaY[abs(*liczbaObrotowPetli -
            *dlugoscWeza)] + 1;
        }

        if (*dlugoscWeza >= 4)
        {
            if (*liczbaObrotowPetli < *dlugoscWeza)
            {
                pole[historiaX[abs(100 -
                (*dlugoscWeza -
                *liczbaObrotowPetli))]] [historiaY[abs(100 -
                (*dlugoscWeza -
                *liczbaObrotowPetli))]] =
                '\0';
            }
            else
            {
                pole[historiaX[abs(*liczbaObrotowPetli
                -
                *dlugoscWeza))] [historiaY[abs(*liczbaObrotowPetli
                -
                *dlugoscWeza))] = '\0';
            }
        }

        IdzDo(&XMapy, &YMapy);
        printf(" ");
    }
}

```

Podczas podawania przez użytkownika zmiennych szerokości, wysokości i kolizji, została ograniczona możliwość wpisywania dowolnych znaków, do znaków wymaganych. Tak samo bufor został wielkościowo dostosowany do potrzeb pobieranej informacji.

```
int PodajZmienna(int *XMenu, int *YMenu)
{
    int liczba[2] = { -1,-1 }; //tablica zawiera liczbe
    podawana przez gracza
    int klawisz = 0; //ascii wczytanego klawisza
    int iloscRuchow = 0; //okresla pozycje, ktora aktualnie
    wskazujemy (dziesiatki, jednosci)

    *XMenu = 59; *YMenu = 21; IdzDo(&XMenu, &YMenu);

    while (1)
    {
        OdkryjKursor();

        klawisz = Pobierz();

        if (klawisz == 27) //esc
        {
            //czyszczenie pola bufora i pola bledu
            *XMenu = 47, *YMenu = 16, IdzDo(&XMenu,
                &YMenu);
            printf(" ");
            *XMenu = 59; *YMenu = 21; IdzDo(&XMenu,
                &YMenu);
            printf(" ");

            //czyszczenie celownika
            *XMenu = 49; *YMenu = 8; IdzDo(&XMenu,
                &YMenu);
            printf(" ");
            *XMenu = 69; *YMenu = 8; IdzDo(&XMenu,
                &YMenu);
            printf(" ");
            *XMenu = 0; *YMenu = 0; IdzDo(&XMenu,
                &YMenu);

            UkryjKursor();

            return -88732; // "kod" ESC
        }
        else if (klawisz == 13) //enter
        {
            //czyszczenie pola bufora i pola bledu
            *XMenu = 47, *YMenu = 16, IdzDo(&XMenu,
                &YMenu);
            printf(" ");
            *XMenu = 59; *YMenu = 21; IdzDo(&XMenu,
                &YMenu);
            printf(" ");
        }
    }
}
```

```

        UkryjKursor();

        if (liczba[1] == -1)
            return liczba[0];
        else
            return (10 * liczba[0]) +
                liczba[1];
    }
    else if (klawisz == 8) //backspace
    {
        if (iloscRuchow == 1)
        {
            *XMenu = 60, IdzDo(&XMenu,
                &YMenu);
            printf(" ");

            liczba[iloscRuchow] = -1;
            iloscRuchow--;

            *XMenu = 59, IdzDo(&XMenu,
                &YMenu);
        }
        else if (iloscRuchow == 0)
        {
            *XMenu = 59, IdzDo(&XMenu,
                &YMenu);
            printf(" ");

            liczba[iloscRuchow] = -1;
            *XMenu = 59, IdzDo(&XMenu,
                &YMenu);
        }
    }
    else if (klawisz >= 48 && klawisz <= 57) // 0-9
    {
        printf("%d", klawisz - 48);
        liczba[iloscRuchow] = klawisz - 48;

        if (iloscRuchow == 1)
            *XMenu = 60, IdzDo(&XMenu,
                &YMenu);
        else
            iloscRuchow++;
    }
}

```

```

// funckja odpowiada za odebranie od uzytkownika znaku
int PodajZnak(int *XMenu, int *YMenu)
{
    int znak = 0; // przechowuje ASCII
    int klawisz = 0; //ascii wczytanego klawisza

    OdkryjKursor();

    while (1)
    {
        *XMenu = 59; *YMenu = 21; IdzDo(&XMenu, &YMenu);

        klawisz = Pobierz();

        if (klawisz == 27) //esc
        {
            //czyszczenie pola bufora i pola bledu
            *XMenu = 47, *YMenu = 16, IdzDo(&XMenu,
                &YMenu);
            printf(" ");
            *XMenu = 59; *YMenu = 21; IdzDo(&XMenu,
                &YMenu);
            printf(" ");

            //czyszczenie celownika
            *XMenu = 49; *YMenu = 6; IdzDo(&XMenu,
                &YMenu);
            printf(" ");
            *XMenu = 69; *YMenu = 6; IdzDo(&XMenu,
                &YMenu);
            printf(" ");
            *XMenu = 0; *YMenu = 0; IdzDo(&XMenu,
                &YMenu);

            UkryjKursor();

            return -88732; // "kod" ESC
        }

        else if (klawisz == 13) //enter
        {
            //czyszczenie pola bufora i pola bledu
            *XMenu = 47, *YMenu = 16, IdzDo(&XMenu,
                &YMenu);
            printf(" ");
            *XMenu = 59; *YMenu = 21; IdzDo(&XMenu,
                &YMenu);
            printf(" ");

            UkryjKursor();

            return znak;
        }
    }
}

```

```

    }
    }

    else if (klawisz == 8) //backspace
    {
        znak = 0;
        printf(" ");
    }

    else if (klawisz >= 32 && klawisz <= 126)
    {
        znak = klawisz;
        printf("%c", klawisz);
    }
}

```