

Politechnika Śląska

Wydział Matematyk Stosowanej

Kierunek Informatyka

Programowanie II

Projekt zaliczeniowy

"*Wonsz*"

Michał Pawełek

1. Opis projektu.

Projekt dotyczy prostej gry znanej powszechnie jako „Wąż”. Dla przypomnienia: wąż ma za zadanie zjadać coraz więcej owoców, każdy owoc go wydłuża, musi unikać ścian i własnego ogona. Gra działa w konsoli.

2. Wymagania

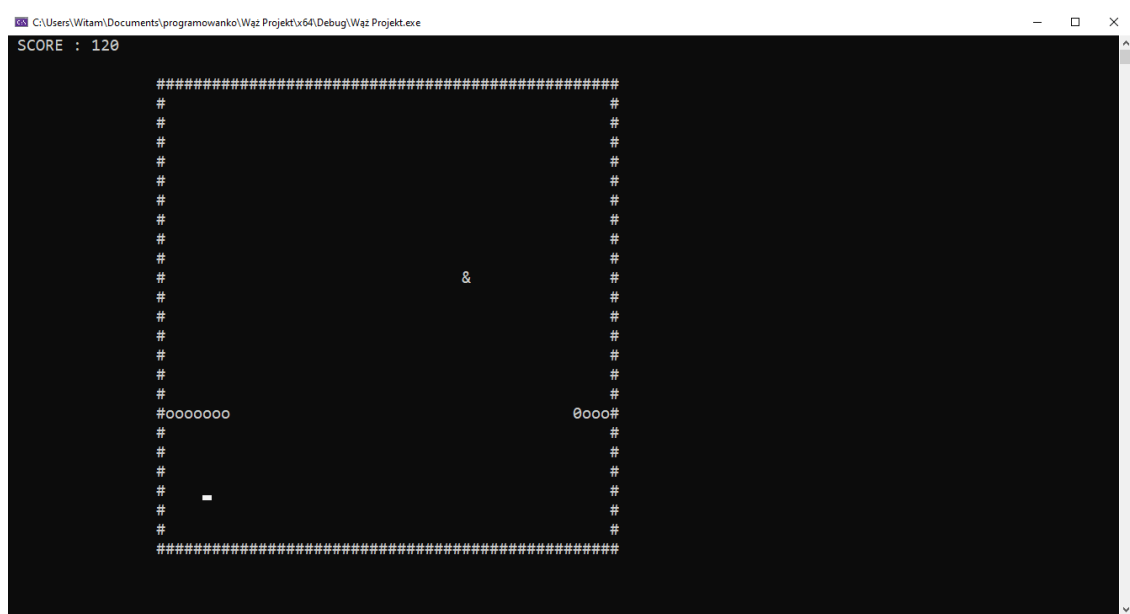
- Przejrzyste menu
- Poruszanie się za pomocą AWSD
- Interakcja z otoczeniem
- Pseudolosowe miejsca pojawiania się owoców
- Różne tryby gry
- Możliwość wyboru koloru tła
- Pokazywanie aktualnego wyniku

3. Przebieg realizacji

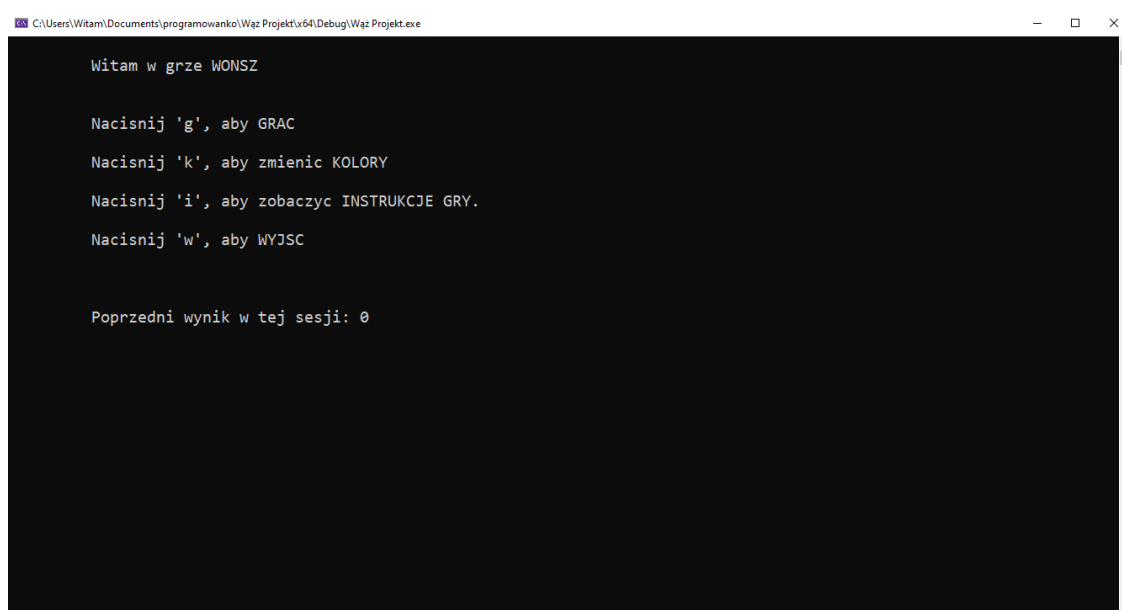
Z dodatkowych bibliotek użyłem conio.h, windows.h, cstdlib, ctime oraz vector do m.in. pobierania klawisza z buforu klawiatury, losowania miejsca pojawienia się owoców czy sposobu poruszania się węża między liniami. Każdą część próbowałem porozmieszczać tak aby było gołym okiem widać, która sekcja za co jest odpowiedzialna. Tworzenie mapy jak i ciała węża oraz owocu była dość prostym zadaniem, ponieważ wystarczyło użyć kilka pętli for oraz if. Kolejnym krokiem było dość trudniejsze sporządzenie poruszania się w grze poprzez klawisze AWSD (bez konieczności potwierdzania ENTER'em) a nie jak dotychczas to miało miejsce. Pomogła mi z tym funkcja kbhit oraz getch. Najtrudniejszym jak i najdłuższym procesem było stworzenie logiki węża w czego skład wchodziło jego poruszanie się po mapie jak również przez obiekty

czy resetowanie go przy następnych podejściach. W tym punkcie po części wzorowałem się różnymi projektami z Internetu i złożyłem to co mi się udało znaleźć w jedną, dość spójną całość. Ważne też było stworzenie różnych trybów np. z kolizją ze ścianami lub też bez kolizji, gdzie przy tym drugim trzeba było zastosować rozwiązanie, które pozwalałoby przechodzić węžu z górnej ściany w dolną albo też z lewej do prawej.

[obrazek poniżej]



Reszta była czystą przyjemnością, czyli tworzenie menu i otoczki wizualnej. Niby nic a jednak bierze dość dużo czasu patrząc na to, żeby była dopracowana i ładnie wyglądała oraz by nie posiadała bugów tj. gdy się wpisze coś innego niż program wymaga. Ostatnią opcją, którą zaimplementowałem to możliwość zmiany koloru tła konsoli (polecam używać czarnego lub białego).

A screenshot of a Windows command prompt window. The title bar at the top shows the file path: C:\Users\Witam\Documents\programowanko\Waz Projekt\Debug\Waz Projekt.exe. The window has standard Windows window controls (minimize, maximize, close) on the right. The command prompt itself has a black background with white text. The text displayed is as follows:
Witam w grze WONSZ

Nacisnij 'g', aby GRAC
Nacisnij 'k', aby zmienic KOLORY
Nacisnij 'i', aby zobaczyc INSTRUKCJE GRY.
Nacisnij 'w', aby WYJSC

Poprzedni wynik w tej sesji: 0

4. Instrukcja użytkownika

Z racji tego, iż raczej każdy kojarzy klasycznego snake'a, przypomnę tylko podstawowe zasady: poruszasz się węzem (tutaj „O” to głowa a „o” ciało) poprzez AWSO, zjadasz owoce (tutaj „&”), nie możesz zjeść własnego ogona, (w zależności od wybranej rozgrywki) unikasz dotknięcia ze ścianą. Reszta związana z nawigacją po menu jest prosto wyjaśniona i raczej niepodatna na błędy.

5. Problemy podczas tworzenia programu

Cały czas, gdy pisałem program napotykałem pewne błędy w większym czy mniejszym stopniu przeszkadzające poprawnemu działaniu. W większości było to złe użycie zmiennej czy funkcji, lecz szybko to naprawiałem. W niektórych przypadkach było tak trudno coś zaimplementować, że nawet Internet w tym nie pomagał i w ten sposób kilka dodatkowych pomysłów na rozszerzenie programu musiało wylecieć z kodu. Dotychczas jestem świadom i nadal opracowuje dwa błędy w programie: jeden dotyczący braku resetowania długości ciała węża

(chodzi o to co widać w grze – sama wartość jest resetowana *[już naprawione]*) oraz znikaniu raz za czas owoców z mapy (domyślam się, że owe owoce są tworzone i umieszczane na planszy, ale mogą być w ścianach lub poza nimi). Jeszcze udało się znaleźć jeden drobiazg, który wg mnie nie podchodzi pod błąd tylko bardziej o niezaimplementowanie pewnej funkcji gry i został opisany w instrukcji gry.

6. Podsumowanie i wnioski

Wszystkie punkty w założeniach udało mi się w pewnym stopniu spełnić z mniejszymi lub większymi problemami. Na pewno największą trudnością dla mnie było stworzenie logiki gry i ruchu węża. W przyszłości mam zamiar podjąć się kontynuowania i rozszerzania programu w czego skład wchodziłby: więcej trybów, bardziej rozszerzone menu, nowe mapy, tabela wyników i co najważniejsze – stworzenie tej gry za pomocą SFML.