

Programowanie II
projekt zaliczeniowy
"Ants Simulation"

Alicja Gruca gr. lab. 3E

Artur Iwański gr. lab. 3E

Wojciech Królik gr. Lab. 3E

1. Opis projektu.

Projekt „Ants Simulation” jest to aplikacja w języku angielskim, stworzona w oparciu o bibliotekę SFML (a dokładniej SFML-GRAPHICS). Językiem wiodącym jest tutaj obiektowy C++. Ants Simulation jest aplikacją wyświetlaną w okienku, samej konsoli natomiast używaliśmy tylko do wypisywania logów programu. Głównym trzonem programu jest klasa engine, która odpowiada za zarządzanie zasobami, przechodzeniem między stanami gry, kontrolowaniem framerate’u oraz ogólnym wyświetlaniem okna. Następnie na podstawie klasy state tworzone były wszystkie stany aplikacji takie jak powitalny initState, menuState czy chociażby simulationState. Sam simulationState jest również podzielony na dwa state’y, które są rozróżniane przez klasę enum - jest to START oraz PLANNER. W stanie symulacji po ustawieniu przeszkód, gniazda, mrówek i jedzenia mrówki poruszają się zgodnie ze swoim algorytmem zbierając po drodze jedzenie i zanosząc je do mrowiska. W projekcie znajdziemy także interaktywne menu oraz zasady gry.

2. Wymagania

- Program okienkowy
- Istnieje wspólne repozytorium na GitHubie
- Istnieje menu główne
- Wyświetlanie zasad działania programu
- Istnieje logika poruszania się mrówek
- Poruszanie się mrówek
- Możliwość ustawiania mrówkom przeszkód
- Możliwość pozycjonowania przez użytkownika mrowiska oraz jedzenia
- Licznik zdobytego jedzenia

3. Przebieg realizacji

Użyte biblioteki:

iostream- biblioteka strumieni standardowych;

Biblioteka SFML- biblioteka graficzna na której bazuje aplikacja

String- biblioteka operacji na klasie string

Memory, Map, Stack- biblioteki używane w silniku

Vector- biblioteka klasy vector, używana do przechowywania danych w wektorach

Sstream- umożliwia dynamiczną konwersję obiektów na napisy

Pliki programu:

Pliki w folderze Engine: Odpowiedzialne są za silnik aplikacji.

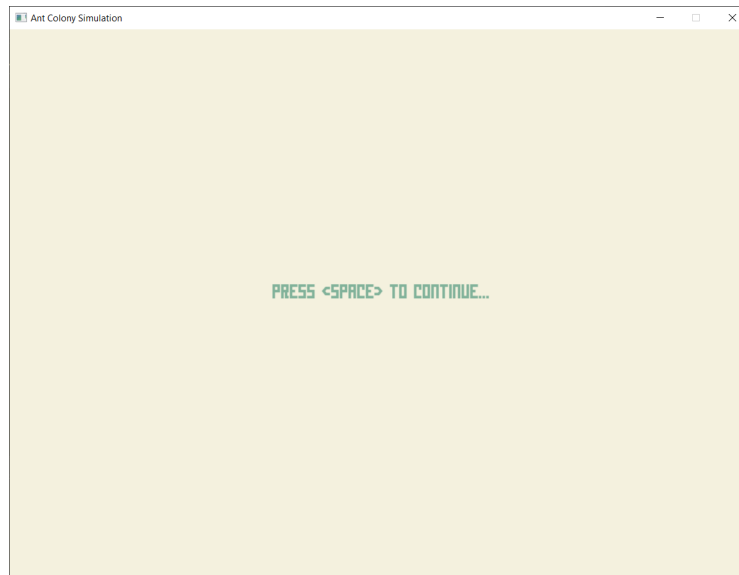
Pliki w folderze GameClasses: Są to pliki z klasami takimi jak klasa mrówki, klasa jedzenia, klasa przeszkody, klasa mrowiska

Pliki w folderze res: Są to pliki z teksturami używanymi przez aplikację.

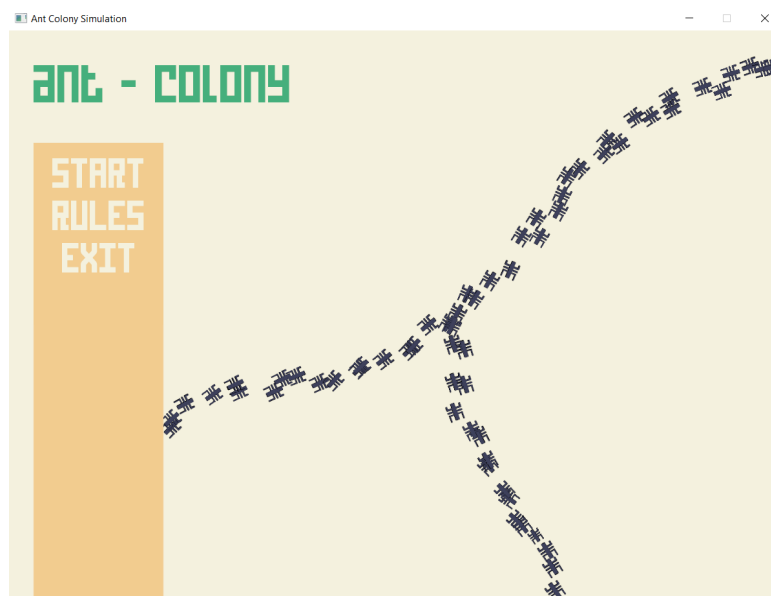
Pliki w folderze States: Pliki odpowiedzialne za stany, w jakich znajduje się gra.

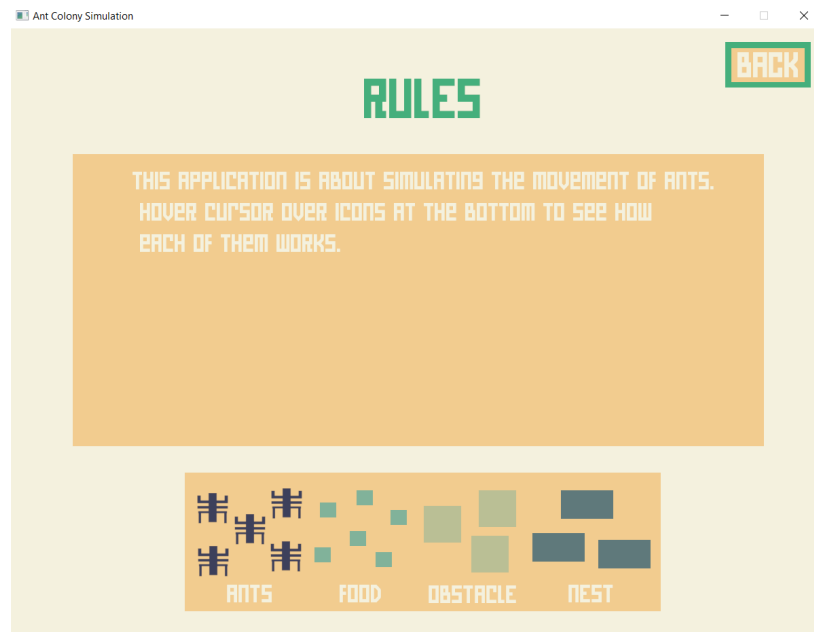
4. Instrukcja użytkownika

Po uruchomieniu programu naszym oczom ukazuje się początkowe okienko. By przejść do interaktywnego menu musimy wcisnąć *Spację*. Od teraz by poruszać się po programie należy używać myszy.

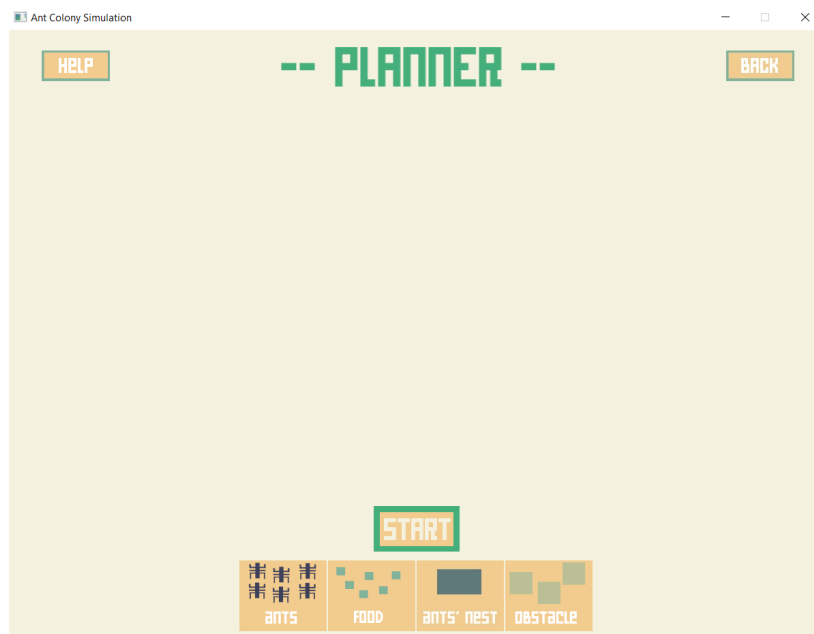


Po przejściu do menu głównego, mamy trzy możliwości działania. Po kliknięciu na przycisk "*Exit*" następuje wyjście z programu. Po wybraniu możliwości "*Rules*", naszym oczom ukaże się interaktywna instrukcja opisująca na czym polega symulacja. Z kolei wybierając opcję "*Start*" zostaniemy przeniesieni do planera.

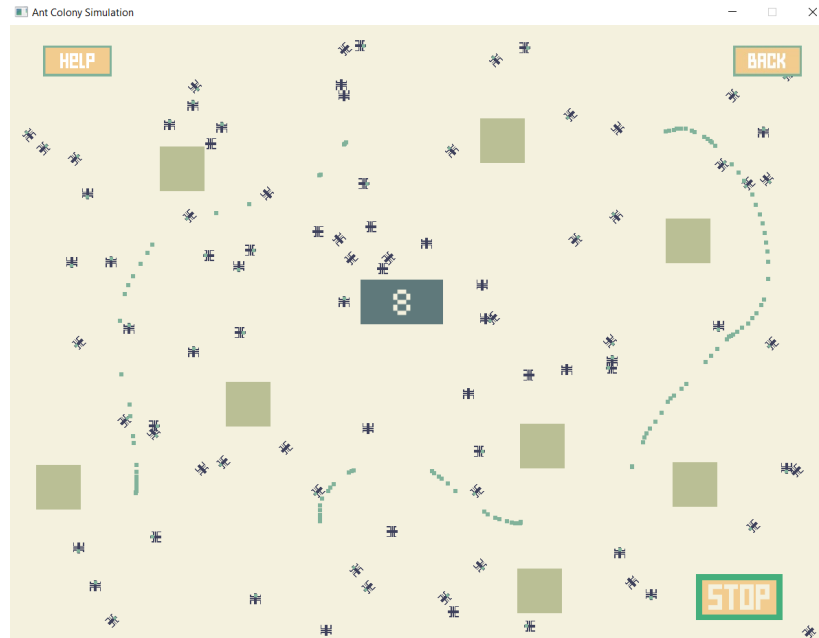




W planerze mamy możliwość ustawienia przeszkód, mrowiska oraz jedzenia. Istnieje także możliwość przejścia do zasad gry oraz powrotu do menu.



Po rozłożeniu obiektów i kliknięciu przycisku “Start” rozpoczyna się symulacja. Na mrowisku możemy zobaczyć licznik, którego zadaniem jest zliczanie ilości jedzenia przyniesionej do mrowiska.



5. Podsumowanie i wnioski.

Zrealizowane zostały wszystkie założone funkcjonalności projektu, aczkolwiek nie wszystko okazało się tak oczywiste do zrobienia jak myśleliśmy, że będzie. Zdecydowanie jednym z trudniejszych problemów było tutaj rozstawianie obiektów na ekranie, musieliśmy w tym celu użyć paru dodatkowych zmiennych (flag) by móc je sukcesywnie rozstawiać w ostatnim położeniu myszki w momencie kliknięcia. Kolejnym z problemów było losowe poruszanie się mrówki na ekranie w określonej prędkości, problem polegał głównie na zbyt częstym zmienianiu kierunku oraz sprawieniu, by wyglądało to wszystko w miarę naturalnie. Sam algorytm działa na zasadzie losowania między 0-7 kierunkami, a następnie losowanie co 2 sekundy jednego z sąsiednich kierunków, bądź podtrzymanie wybranego kierunku. Oczywiście jest to tylko ułamek problemów z którymi musieliśmy się zmierzyć, aczkolwiek końcowy efekt

zdecydowanie przyniósł nam satysfakcję. Dodatkowo podczas projektu nabraliśmy doświadczenia w obrębie projektowania aplikacji oraz pisania jej obiektowo.