Robot sprzątający 104.4

Skład zespołu:

Naumenko Kateryna

Nitkiewicz Jakub

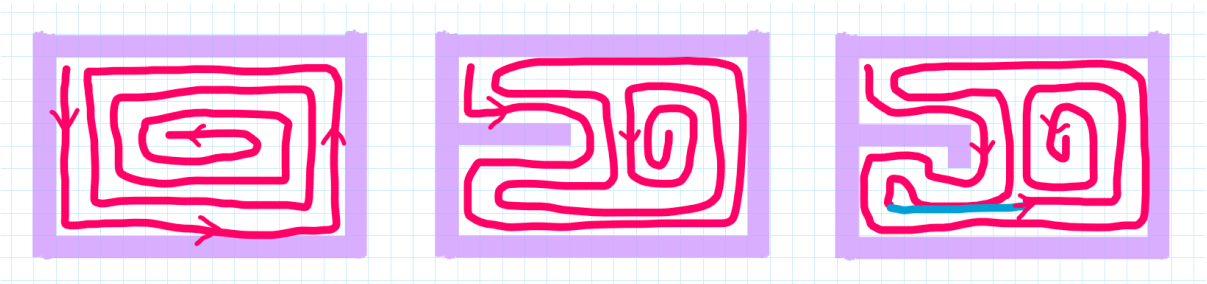
Świderska Ada

Robot sprzątający

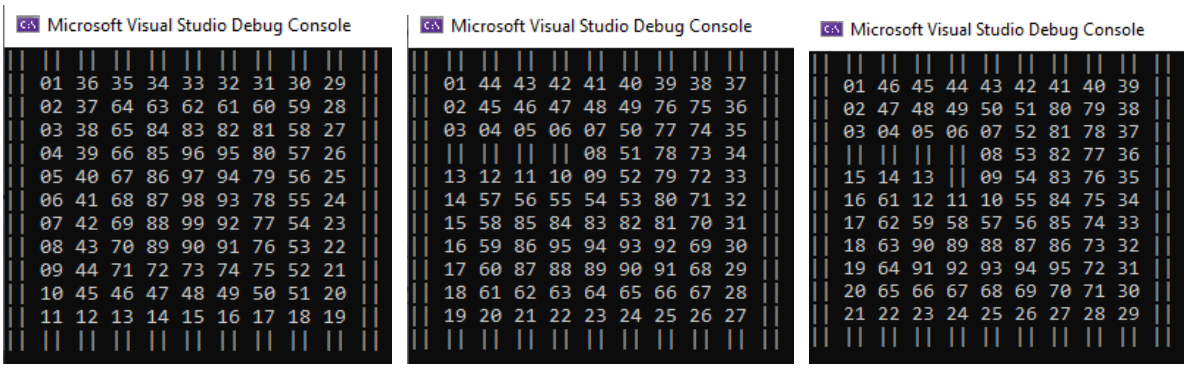
Projekt Robot Sprzątający został zrealizowany poprzez implementacje sześciu klas, pozwalających na wybranie rozmiaru pomieszczenia, dodanie ścian oraz przeszkód, a następnie „uruchomienie" robota, tak by przejechał całą powierzchnię wolną od przeszkód i ścian. Robot jedzie w taki sposób, że zawsze ma po swojej prawej stronie ścianę, przeszkodę lub miejsce już posprzątane, gdy dojedzie do miejsca w którym nie ma już obok siebie miejsca do posprzątania, ale jest ono jeszcze w innym miejscu mapy, wyznacza najbliższą odległość do tego miejsca i dojeżdża do niego a następnie kontynuuje standardową jazdę.

Przebieg pracy nad projektem

Najpierw zastanowiliśmy się nad drogą robota i dlatego skupiliśmy nad algorytmem jazdy:

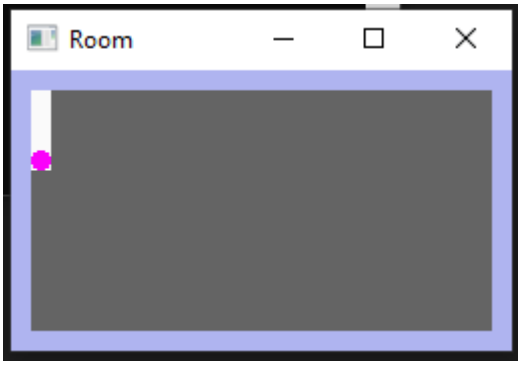


Potem zdecydowaliśmy się na aplikację terminalową:



Na zrzutach ekranu widać przeszkody oznaczone ||, a kolejne liczby - to kolejne położenia robota.

Wersja terminalowa była całkiem trudna do zrozumienia, więc zdecydowaliśmy się na wersją graficzną z wykorzystaniem SFML.

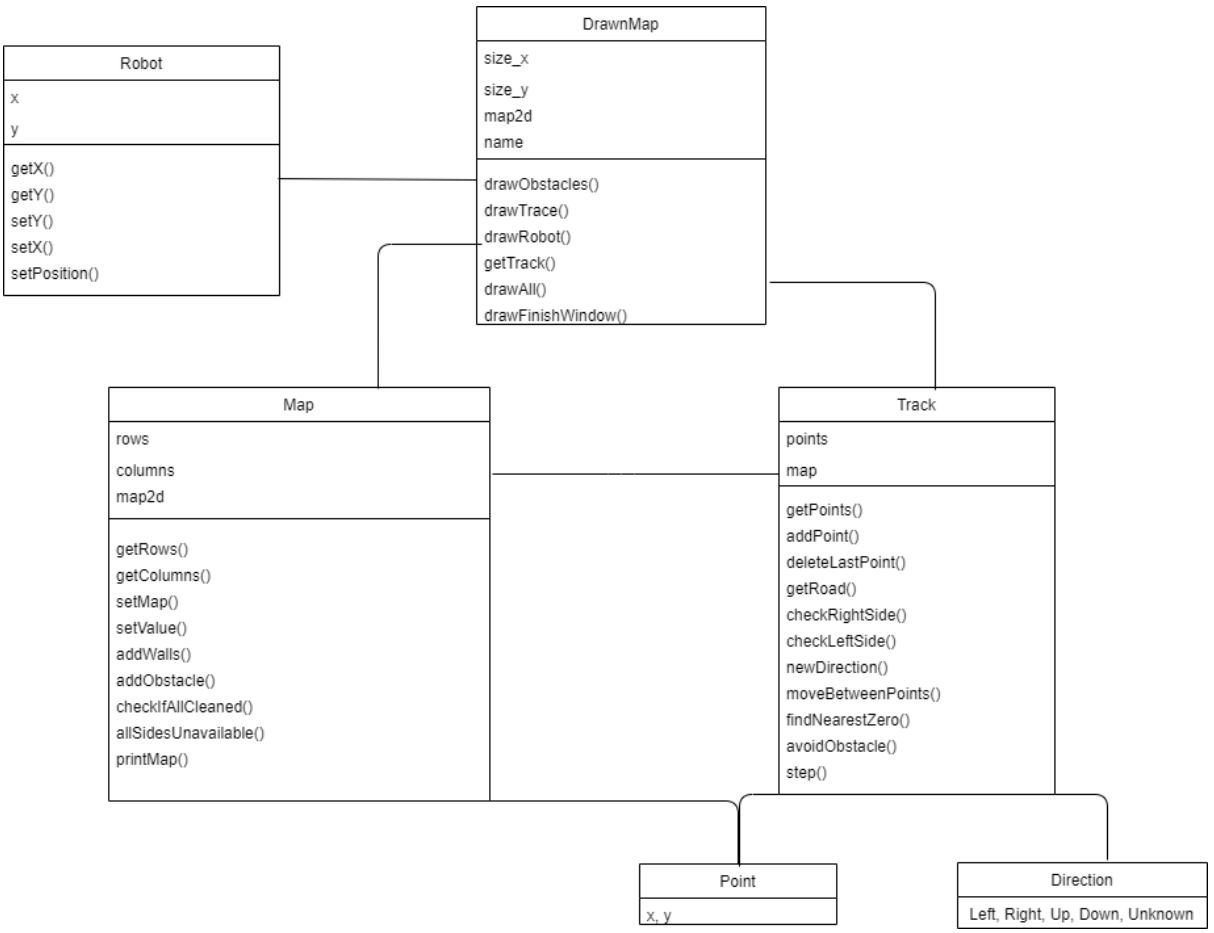


Na zrzutach ekranu widać przeszkody i ściany oznaczone fioletowym kolorem, robot - różowe kółko. Szare obszary są nadal brudną powierzchnią, a białe - już czystą.

Wykorzystane narzędzia

Interfejs graficzny został stworzony z wykorzystaniem biblioteki **Simple and Fast Multimedia Library**, która umożliwia nam obserwację jazdy robota w czasie rzeczywistym. Pomocniczą częścią pracy było wykorzystanie **TDD**(test-driven development).

Architektura projektu



1. Klasa Map – Zawiera metody pozwalające m.in. na dodawanie ścian, dodawanie przeszkód, sprawdzanie czy cały obszar mapy został już posprzątany.

2. Klasa Track – Jej metody służą implementacji algorytmu poruszania się robota. Jak zostało wcześniej nadmienione robot jedzie tak, by po prawej stronie mieć przeszkodę, ścianę lub element posprzątany, w zależności od tego co napotka na swojej drodze tak dopasowuje swój kierunek jazdy odnosi się tutaj do klasy wyliczeniowej Direction, w której przechowywane są kierunki jazdy: Left, Right, Up, Down, Unknown. Pierwsze 4 kierunki to kierunki standardowe, natomiast kierunek Unknown określa jak ma jechać robot w momencie, gdy nie ma wokół siebie już miejsc, które mógłby posprzątać. Wyszukuje wtedy najbliżej siebie leżące pole o wartości 0 (nieposprzątane), wyznacza prostą łączącą dwa pola i porusza się wzdłuż niej po drodze omijając napotkane przeszkody.

3. Klasa DrawnMap - Jest implementacją trasy i ruchu robota. Obiekt klasy DrawnMap jest tworzony dla obiektu klasy Map. Dzięki użyciu biblioteki Simple and Fast Multimedia Library mapę zapisaną jako wektor punktów 2d jesteśmy w stanie pokazać na ekranie jako obraz, co więcej jesteśmy w stanie na bieżąco śledzić ruch robota.

4. Klasa Robot - Jej metody pozwalają nam na ustawianie oraz pobieranie pozycji robota.

5. Klasa Point - Struktura do przechowywania współrzędnych.

6. Klasa Direction - klasa wyliczeniowa do przechowywania możliwych kierunków jazdy: Left, Right, Up, Down, Unknown.

Uruchamianie programu

Obowiązkowym jest instalowanie biblioteki SFML. W przypadku pominięcia tego etapu nie uda się uruchomić program, o co użytkownik będzie poinformowany przez komunikaty systemu.

Dla uruchomienia aplikacji niezbędnym jest: pobranie folderu z repozytorium https://gitlab-stud.elka.pw.edu.pl/proi.21l/104.4-robot-sprzatajacy.git i uruchomienie go.