

# PROJEKT ZALICZENIOWY SEMINARIUM

Kod przedmiotu: 1DM1103\_20Z

Prowadzący: dr Robert Szmurło

Autorzy projektu: Hubert Parada, Jakub Tomulewicz

Data: 28.01.2021

## Cel projektu:

Cały projekt polegał na wykorzystaniu wszystkich umiejętności, które poznaliśmy i utrwalaliśmy na zajęciach z programowania. Celem projektu było stworzenie programu, który poprzez wysyłanie zapytań do serwera API i interpretację ich odpowiedzi, pozwoli użytkownikowi odkryć całą mapę.

## Rozdziały programu:

W pierwszej kolejności podzieliliśmy nasz program na pliki nagłówkowe:

**Odpowiedz** – w tej części umieściliśmy funkcje odpowiedzialne za komunikację z serwerem. Na początku wysyłamy zapytanie do serwera przy pomocy biblioteki *curl*, następnie otrzymujemy odpowiedź, którą zapisujemy do pliku tymczasowego, która później jest interpretowana w pozostałych plikach nagłówkowych programu.

- *Zapisz\_odpowiedz*
- *Wyślij\_zapytanie*
- *info*
- *ruch\_do\_przodu*
- *lewo*
- *prawo*
- *reset*

**Logika** – w tej części programu umieściliśmy funkcje odpowiedzialne za interpretację odpowiedzi z serwera do odpowiedniej struktury. Obliczamy również przesunięcie w dwóch ostatnich funkcjach.

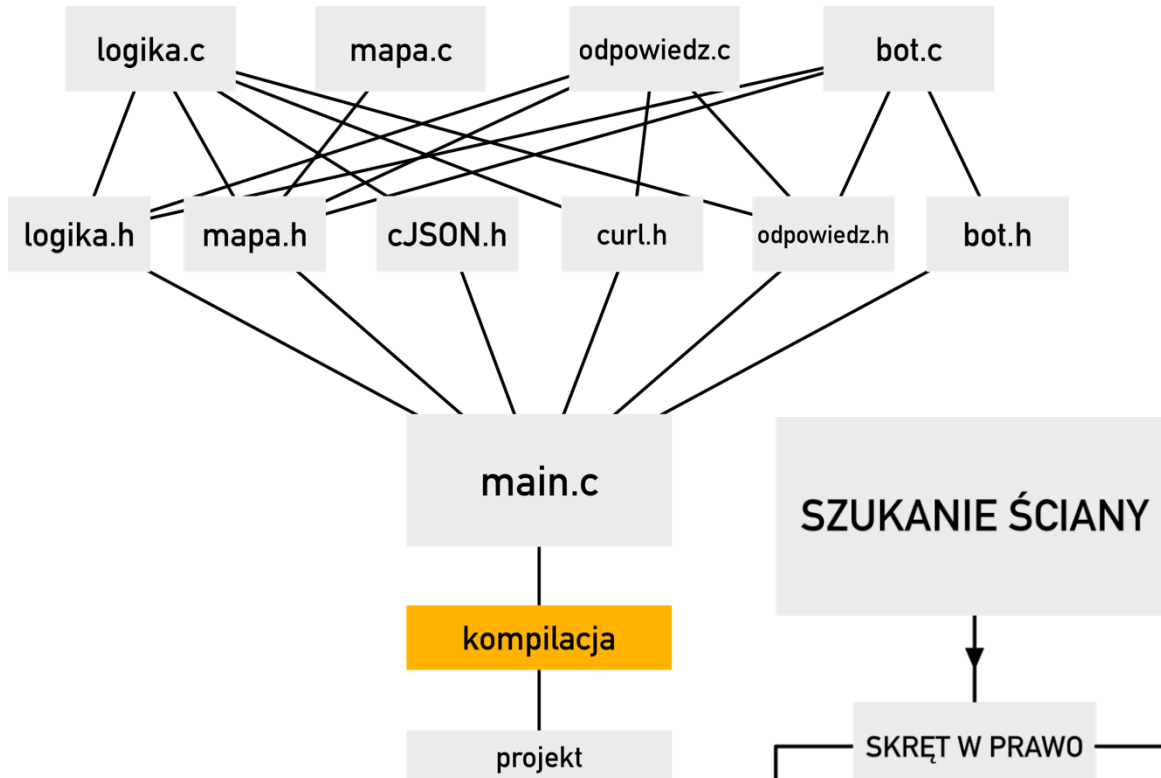
- *Parametry\_ruchu*
- *Odkrywanie*
- *Ruch*
- *Offsetx*
- *Offsety*

**Mapa** – w tej części programu zawarliśmy funkcje obsługujące przydzielanie pamięci na nowe tablice, warunkującą poszerzanie tablic, poszerzającą tablice, interpretującą informacje z odpowiedzi z serwera do wartości liczbowych oraz zwalnianie pamięci dla uprzednio zaalokowanych tablic. Dodatkowo, umieściliśmy tu również struktury: *Luneta*, *Parametry* oraz *Mapa*. Struktura *Mapa* zawiera w sobie odwołania do struktur 2 pozostałych, dlatego umieściliśmy je w jednym miejscu. Wykorzystujemy je później w każdym pliku nagłówkowym, natomiast pisząc program uznaliśmy, że ich obecność w tej części programu będzie dla nas najwygodniejsza, a dla osoby analizującej działanie programu intuicyjna do znalezienia.

- *Zapisz\_mape*
- *Wypisz\_mape*
- *Tłumacz\_podloze*
- *Wypisz\_ruch*
- *Wypisz\_eksplorowanie*
- *Buduj\_mape*
- *Granica*
- *Puzzle* – *puzzle\_prawo*, *puzzle\_lewo*, *puzzle\_gora*, *puzzle\_dol*
- *Zwolnij\_mape*

**Bot** – w tej części programu możemy znaleźć funkcje odpowiadające za wykonywanie przez nasz czołg ruchów w sposób autonomiczny.

- *Szukaj\_ściany*
- *Lewo\_ruch*
- *Prawo\_ruch*
- *Bot*



## Scenariusz

W głównym pliku *main*:

1. Rozpoczynamy od zresetowania ustawienia czołgu na mapie za pomocą funkcji *reset*.
2. Tworzymy początkową tablicę dwuelementową o wymiarach 5x5, która posłuży nam za pole startowy obszar przeszukiwań mapy dla czołgu.
3. Obliczamy przesunięcia za pomocą funkcji *offsetx* oraz *offsety*, które później będziemy wykorzystywać przy wpisywaniu odkrytych pól do poszczególnych elementów tablic.
4. Umieszczamy czołgu na środku początkowej, dwuelementowej tablicy.
5. Wywołujemy funkcję *bot*. Sekwencję poruszania się czołgu spróbujemy przedstawić na schemacie po prawej:
6. Na końcu, jeśli czołg wróci do punktu startowego, zwraca strukturę mapę, inicjalizuje ją w terminalu oraz zapisuje do pliku „*zapis.txt*”.



## Instrukcja uruchamiania programu:

Dla trybu autonomicznego:

W głównym folderze znajduje się plik `makefile`, którym posłużymy się, niebezpośrednio, do sprawnego kompilowania i uruchamiania głównego programu.

1. Wpisujemy w linii komend „*make boł*”. W tym czasie skompiluje nam się oraz uruchomi program, który pozwoli przejść czołgowi mapę, wracając do punktu początkowego oraz zapisując ostatni obraz mapy do pliku „*zapismapy.txt*”.
2. Po zakończeniu działania programu możemy wpisać „*make pamiec*”, wówczas wykona się poprzedni podpunkt, a na końcu zostanie wypisana stracona pamięć (niestety nie udało nam się zwolnić wszystkiego).
3. Po zakończeniu korzystania z programu możemy skorzystać z funkcji czyszczenia wpisując „*make czyszczenie*”. Zostaną wówczas usunięte pliki o rozszerzeniu `.o`, skompilowany program oraz ostatnia wiadomość z serwera.

Do testowania:

Stworzyliśmy pod folder, w którym znajduje się osobny plik `main_trybmanualny` wraz z osobnym `makefilem`.

1. Na początku możemy zacząć od skalibrowania kierunku czołgu przez wpisanie „*make test-prawo*”. Robimy tak do momentu aż w oknie terminala pojawi się komunikat „*Pozycja czołgu została skalibrowana!*”
2. Teraz możemy wpisać w oknie terminalu „*make trybmanualny*”. Czołg przejedzie i odkryje całą mapę, ustawiając się na koniec w tej samej pozycji.
3. Opcjonalnie możemy użyć pozostałych funkcji: „*make test-info*”, „*make test-prawo*”, „*make test-lewo*”, „*make test-ruch*”, „*make test-reset*”.
4. Po zakończeniu korzystania z programu możemy skorzystać z funkcji czyszczenia wpisując „*make czyszczenie*”. Zostaną wówczas usunięte pliki o rozszerzeniu `.o`, skompilowany program oraz ostatnia wiadomość z serwera.