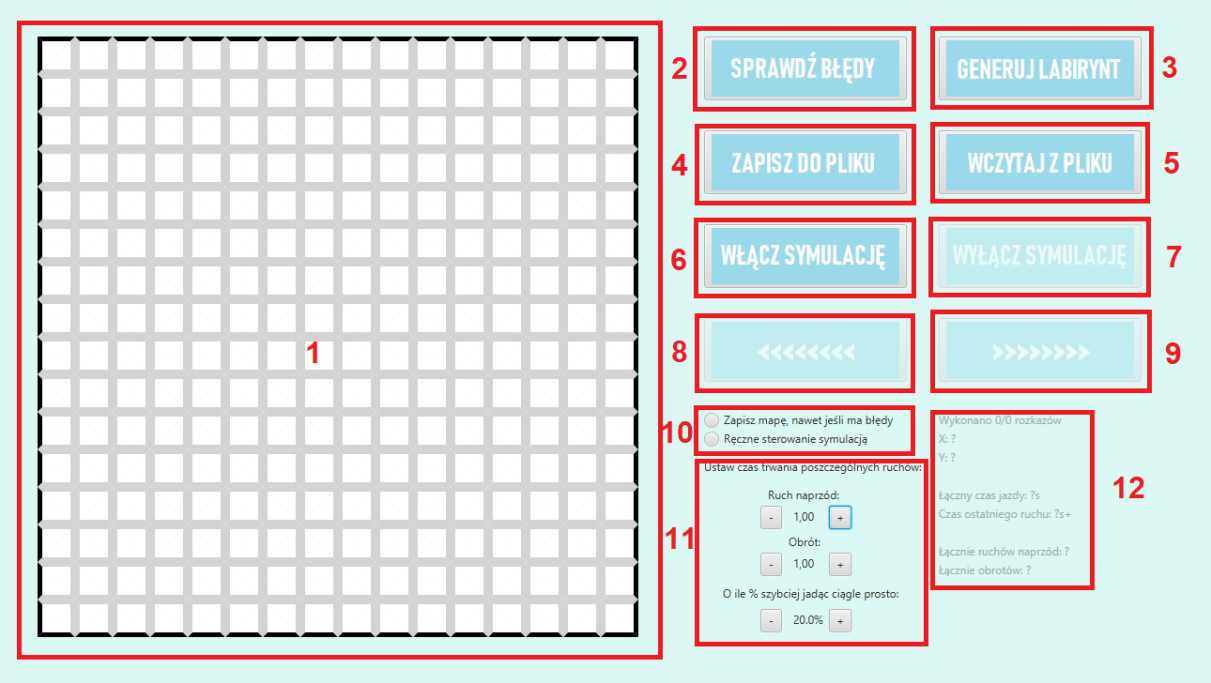
Kreator labiryntów – krótki poradnik i instrukcja

Witam serdecznie i przedstawiam króciutki poradnik jak używać mojego programu. Nie jest to zbyt skomplikowane, lecz idąc duchem bycia poważnym człowiekiem piszę ten mały kawałek tekstu. Prawie jak dokumentacja normalnie.

1. Ogólne znaczenie wszystkich przycisków

Program po włączeniu wygląda tak jak na obrazku:



1. Labirynt sam w sobie. Jest to miejsce gdzie budujemy labirynt klikając w odpowiednie ściany, lub części segmentu. Tutaj także będzie widoczna symulacja jeśli ją włączymy.
2. Sprawdzanie błędów – po kliknięciu sprawdza czy w zbudowanym lub wczytanym labiryncie są jakieś podstawowe błędy, tzn. czy istnieje tylko jedna meta, czy można dojechać do każdego fragmentu labiryntu itp.
3. Generowanie losowego labiryntu – dokładniej opisane dalej.
4. Zapisywanie do pliku – zapisuje labirynt do pliku tekstowego, aby można go było później wczytać lub użyć we własnym programie do generowania trasy.
5. Wczytywanie z pliku – wczytuje z pliku tekstowego wcześniej zapisany labirynt.
6. Włączanie symulacji – włącza symulację, dokładnie opisane dalej.
7. Wyłączanie symulacji – analogicznie wyłącza symulację, opisane dalej.
8. Poprzedni krok – cofa krok przy ręcznym sterowaniu symulacją.
9. Następny krok – wykonuje następny krok przy ręcznym sterowaniu symulacją.
10. Przyciski pomocnicze – służą do obsługi dodatkowych funkcji. Działanie ich jest takie jak napisane.

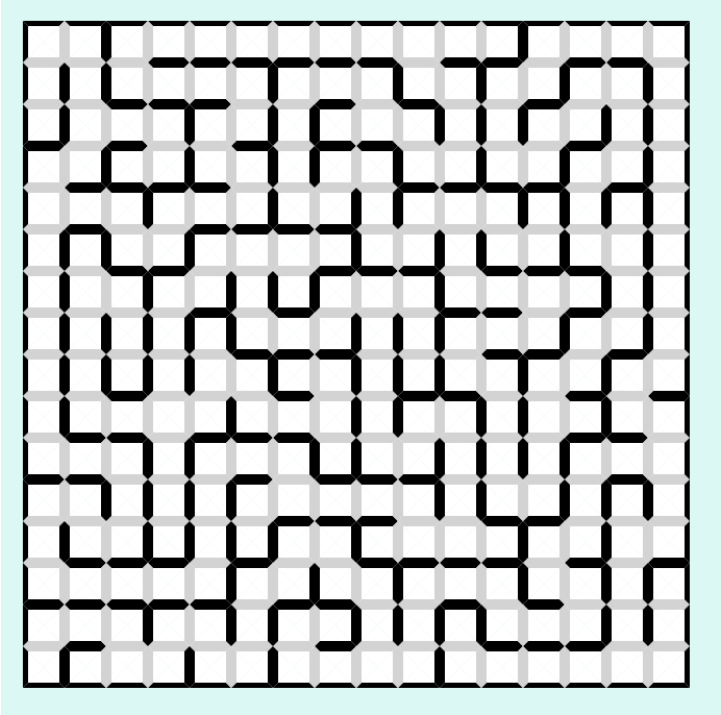
11. Ustawianie czasu trwania ruchów – umożliwia ustawienie czasu trwania (w sekundach) konkretnych ruchów robota oraz o ile % szybciej robot jedzie naprzód jeżeli jedzie parę segmentów z rzędu. Wpływ tych danych na szybkość animacji jest znikomy (chociaż wpływa), ale służą one głównie do danych statystycznych zbieranych podczas symulacji.
12. Dane statystyczne wyświetlane podczas symulacji. Pozwalają na zbiorcze opisanie działania algorytmu w danym labiryncie. Wyświetlana jest ilość wykonanych rozkazów, położenie robota, łączny czas jazdy, czas ostatniego ruchu oraz ilość konkretnych ruchów wykonanych podczas symulacji.

2. Losowe labirynty

Nie jest to zbyt ważny punkt, ale nie byłbym sobą gdybym o tym nie powiedział. Ogólny sposób generowania losowych labiryntów jest poprawny i działa jak trzeba. Mimo to, ma parę mankamentów. Dokładniej:

- Nie tworzy mety
- Sam labirynt jest dosyć „monotematyczny”

Co do tworzenia mety to wystarczy ją po prostu ręcznie zrobić (dla przypomnienia – meta to wolny kawałek labiryntu 2x2 z jednym wejściem). Śpieszę też z wyjaśnieniem o co chodzi z tą monotematycznością. Przykładowo weźmy pierwszy lepszy wygenerowany labirynt:



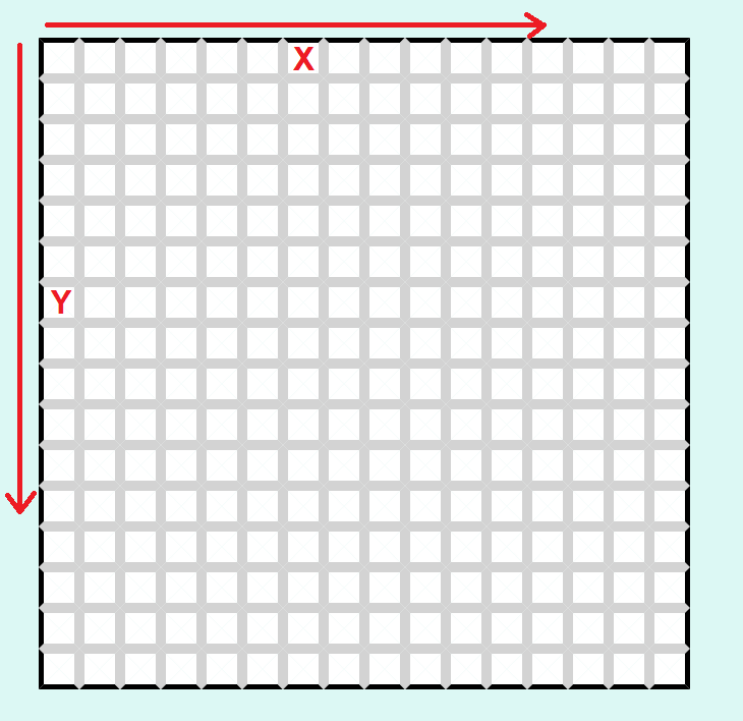
Ogólnie wygląda zadowalająco, ale jak się przyjrzeć to można zauważyć, że ma mało odnóg, przejść, rozjazdów itp. To już jest mniej zadowalające. Rozwiązanie jest mało eleganckie, lecz skuteczne – ręczne usunięcie paru ścian. Teoretycznie jest możliwość zmniejszenia tego efektu, ale już nie miałem na to siły.

Jak widać, mimo że generowanie losowych labiryntów brzmi jak bardzo szybkie i wygodnickie narzędzie, wymaga trochę pomocy od użytkownika. Ale zapewniam, warto tego używać.

3. Symulator

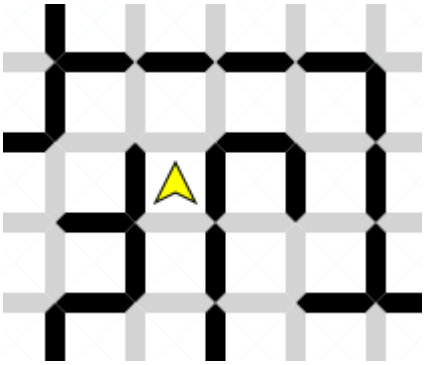
3.1 Ogólne działanie symulatora

Używanie symulatora jest na tyle proste, że zrozumiałby je nawet średnio ogarnięty absolwent przedszkola. Jedynym plot twistem może być ustawienie współrzędnych. Tutaj wygląda ono następująco:



Czyli wyjaśniając oś X rośnie w prawo, czyli normalnie. Oś Y rośnie w dół, czyli najniższe segmenty mają współrzędną Y równą 16, a najwyższe 1.

Reszta aspektów symulatora jest bardzo prosta. Używając przycisku ***WŁĄCZ SYMULACJĘ*** włączamy symulację, co otwiera okno wyboru pliku z rozkazami (jak takie pliki są zbudowane jest napisane dalej). Kiedy go wybierzemy, a on się poprawnie wczyta, przechodzimy do właściwej symulacji. Jeżeli wybraliśmy ręczne sterowanie, możemy nią sterować używając przycisków ***NASTĘPNY KROK*** oraz ***POPRZEDNI KROK***. Dzięki nim robot zaczyna się poruszać wykonując wczytane rozkazy. Przy okazji zbierane są statystyki dotyczące danego przejazdu.



Jeżeli chcemy zakończyć symulację, po prostu klikamy przycisk ***WYŁĄCZ SYMULACJĘ***. Ot co.

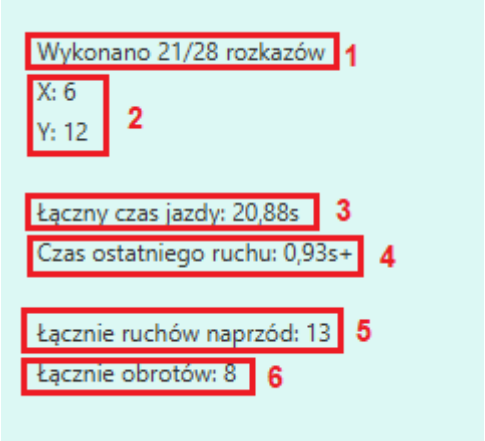
Gdybyśmy nie zaznaczyli opcji ręcznego sterowania, nasz robot by po prostu automatycznie przejechał całą trasę. O ile oczywiście w międzyczasie nie wjedzie w ścianę, ale to już by była wina algorytmu, nie robota.

3.2 Statystyki

Symulator w trakcie działanie symulacji zbiera podstawowe dane umożliwiające porównanie działania, czy ogólny ogląd sytuacji. Dane są wyliczane teoretycznie na podstawie liczb podanych w menu obok (dokładnie opisane w podpunkcie 3.3).

Dokładnie zbierane są dane:

1. Liczba dotychczas wykonanych rozkazów oraz ich łączna ilość. Jeżeli robot by się rozbił na jakiejś ścianie, to można sprawdzić w którym konkretnie kroku to się dzieje.
2. Współrzędne robota w labiryncie, wyliczane według sposobu pokazanego wcześniej.
3. Łączny teoretyczny czas przejazdu robota.
4. Ile trwał ostatni wykonany ruch.
5. Łączna wykonana dotychczas ilość ruchów naprzód.
6. Analogicznie, tylko dotyczy obrotów.

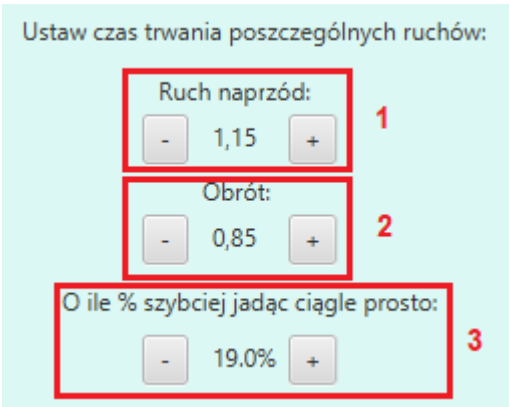


3.3 Przyciski wyboru czasu trwania poszczególnych ruchów

Dane zbierane w oknie statystyk są ustalane na podstawie czasu trwania poszczególnych ruchów ustawionych za pomocą tych przycisków. Należy je ustawić tak jak jest w warunkach rzeczywistych albo tak jak nam się wydaje.

Dokładne znaczenia:

1. Czas jazdy robota naprzód o jeden segment w sekundach, można zmieniać ten czas za pomocą przycisków o 0,05 sekundy.
2. Analogicznie jak wyżej tylko dla obrotów robota.
3. Jak wiadomo (lub nie), robot jeździ szybciej jeśli jedzie parę segmentów w linii prostej, ponieważ nie musi się cały czas zatrzymywać. Oczywiście zakładamy że znakomici programiści i konstruktorzy robota uwzględnili tę możliwość w trakcie pisania programu sterującego. Ta opcja pozwala na dostosowanie tej zmiany, można ją zmieniać o 0,5% klikając w odpowiednie przyciski.

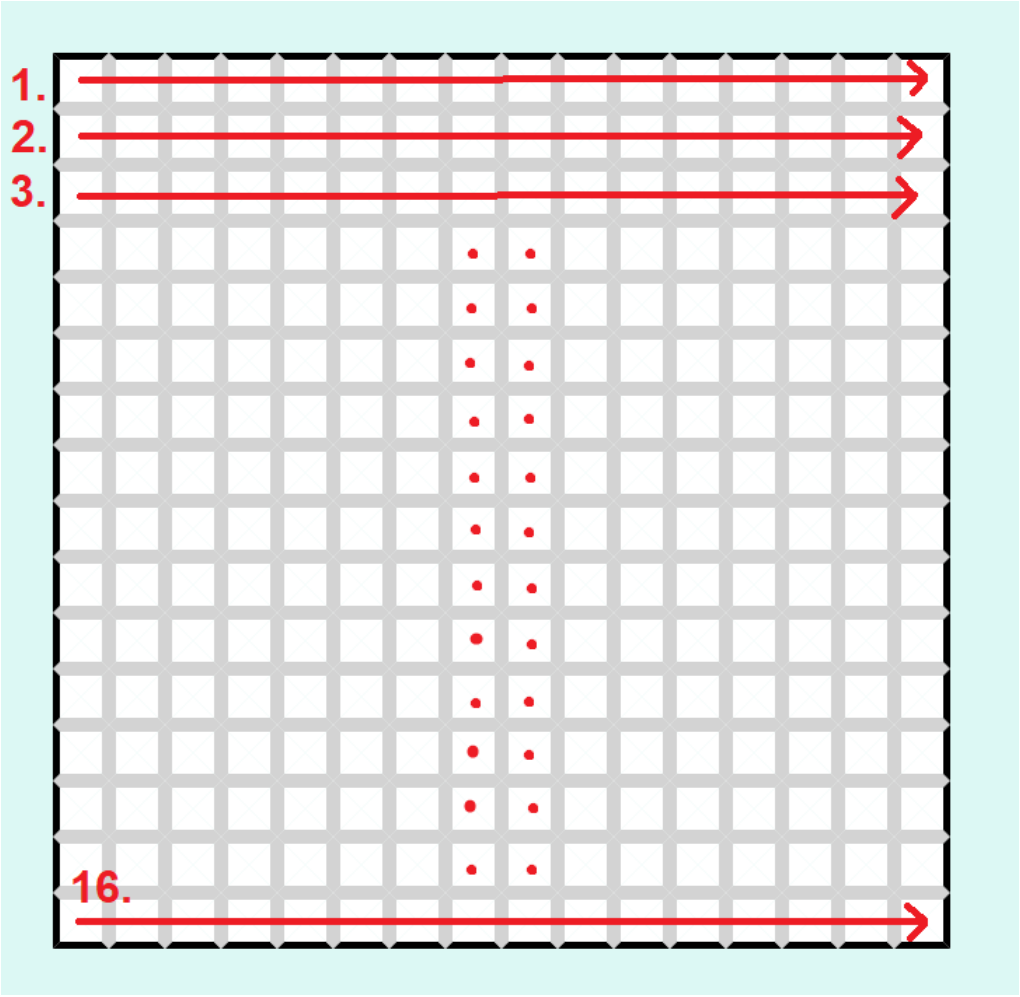


4. Konwencja zapisu labiryntu oraz rozkazów symulatora

4.1 Zapis labiryntu

O ile wcześniejszych informacji można się domyślić po przeczytaniu nazw odpowiednich funkcji, bez tego raczej nie uda nam się z tego symulatora skorzystać. Mimo to, informacje tu napisane też nie są skomplikowane. Ot, kwestia wyjaśnienia.

Labirynt jest zapisywany w pliku tekstowym txt za pomocą 256 kolejno zapisanych w nowych liniach liczb. Każda z nich oznacza stan kolejnych segmentów labiryntu. Liczby odwzorowują stan segmentu idąc od lewej do prawej, od góry do dołu, tzn. tak jak na rysunku.



Każdy segment jest zapisany za pomocą 4-bitowej liczby gdzie każdy bit idąc od lewej oznacza kolejne ściany idąc wskazówkami zegara – góra, prawo, dół, lewo. Czyli przykładowo:

- 0 = 0000 - pusty segment, brak ścian
- 16 = 1111 - zamknięty segment, takich akurat nie może być, zapis takich się nie uda
- 12 = 1100 – ściana z góry oraz oraz prawej, reszta pusta
- 10 = 1010 – ściana z góry oraz dołu
- 5 = 0101 – ściana z prawej oraz lewej

I tak dalej. Odczyt takich danych trzeba zrobić za pomocą masek bitowych (było na wykładzie!).

4.2 Zapis rozkazów symulatora

Akurat zapis rozkazów jest prosty, łatwy i przyjemny, jako że są tylko trzy możliwości ruchu robota. Rozkazy są tak samo zapisane za pomocą liczb od góry do dołu w oddzielnych liniach, gdzie konkretne liczby oznaczają:

- 1 – jedź naprzód
- 2 – obróć się w lewo
- 3 – obróć się w prawo

Dodatkowo każdy plik z rozkazami MUSI się kończyć zerem, bez tego plik nie zostanie przyjęty. I to tyle w sumie na ten temat, średnio ogarnięty przedszkolak by ogarnął.

5. Ogólny i zalecany sposób użytkowania

Wydaje mi się, że ten programik jest całkiem przydatnym narzędziem do testowania algorytmów wyznaczania trasy kiedy nie możemy tego jeszcze robić na robocie. Ogólną ideę tego jak używać programu zawarłem w tych punktach.

1. Wygeneruj losowy labirynt i odpowiednio go dostosuj według wymagań i zachcianek (można oczywiście cały od podstaw zrobić, nikomu nie bronię).
2. Zapisz go do folderu z programem zawierającym twój algorytm.
3. Otwórz go w swoim programie i daj mu wygenerować odpowiedni ciąg rozkazów.
4. Wczytaj ten ciąg rozkazów w symulatorze.
5. Obserwuj jeżdżącego robota, ewentualnie steruj nim ręcznie.
6. Miej nadzieję, że nie wjedzie w ścianę.

To by było chyba na tyle, dzięki za przeczytanie tych wypocin i życzę samych sukcesów z waszymi algorytmami. Pozdro 600.

PK

PS. Jeżeli mój opis konwencji zapisu labiryntu oraz rozkazów mało rozjaśnia to w folderze są przykładowe pliczki.