PROJEKT KCK - DROGA

SEMESTR ZIMOWY 2014/2015

**WMI UAM**

**II rok / I stopień**

**Gr. 1CG**

Patrycja Stefańska,

Rafał Wojciechowicz,

Grzegorz Rombalski,

Jakub Gąsiorowski,

Marcin Jadwiszczak,

Paweł Dukarski

Spis treści

[**WSTĘP:** 1](#_Toc411597530)

[**Technologie i własności:** 2](#_Toc411597531)

[**PRZYKŁADOWA TRASA:** 3](#_Toc411597532)

[**KLASY UŻYTE W KODZIE:** 5](#_Toc411597533)

[**PRZEDSTAWIENIE LOGIKI KODU:** 6](#_Toc411597534)

[**SŁOWNIK:** 9](#_Toc411597535)

**WSTĘP:**

Projekt jest przedstawiony w bardzo przejrzysty i logiczny sposób na naszej stronie:

[http://kck.appsolution.pl.](http://kck.appsolution.pl/)

Dzięki dynamicznemu wykonywaniu skryptu po stronie klienta jesteśmy w stanie na bieżąco i z podglądem zmian modyfikować i eksportować drogę do trybu tekstowego. Ułatwiają Nam to konkretne opcje dostępne w projekcie.

Podczas kliknięcia opcji „Eksportuj do edytora” program pobiera wartości wygenerowanej wcześniej drogi z funkcji „opisz” i eksportuje je do edytora w prostym języku poleceń.

W przypadku wybrania opcji „Generuj drogę z tekstu”, program pobiera dane z formularza, które następnie wysyła do funkcji „Przerysuj”.

Zaś funkcja „Przerysuj” przedstawia nam konwersje tekstowego języka poleceń ponownie w formie wizualnej drogi.

**Technologie i własności:**

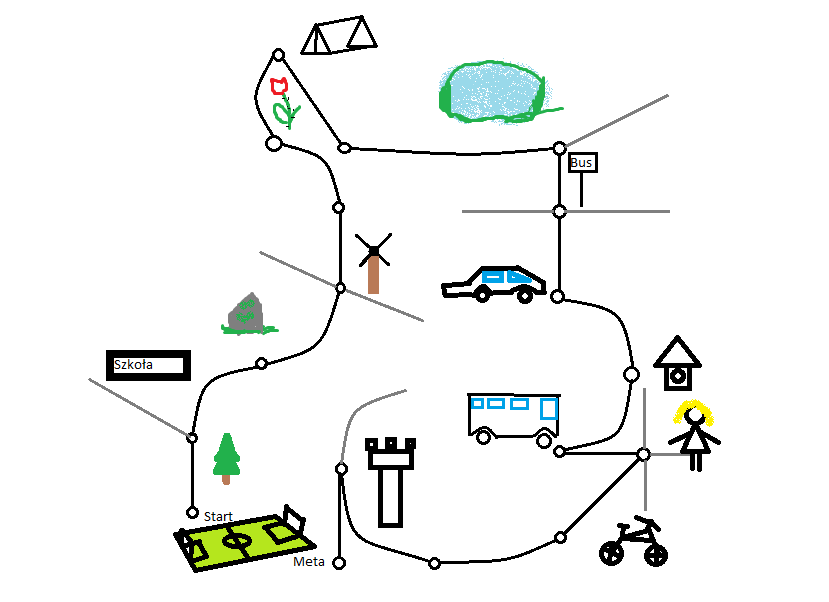
**HTML, Javascript**

Technologia internetowa HTML i język skryptowy JavaScript pozwala na bieżąco generować trasę przy wykorzystaniu tzw. canvasów. Jest to znacznik pozwalający na dynamiczne, skryptowe renderowanie kształtów i obrazów bitmapowych. Skrypt w języku JavaScript będzie rysował obszar, w obrębie którego będzie mógł poruszać się Agent 1 oraz generował trasę, po której będzie on się realnie poruszał.

**KRZYWE BEZIERA**

To parametryczne krzywe powszechnie wykorzystywane w projektowaniu grafiki komputerowej. W projekcie będą one odpowiadały za trasę, po której będzie się poruszał Agent 1. Podawane są 3 punkty (początkowy, końcowy i punkt zakrzywienia) i na ich podstawie powstaje krzywa. Krzywe łączą się ze sobą w całość, dzięki czemu powstaje trasa.

**PRZYKŁADOWA TRASA:**



**Opis do przykładowej trasy:**

1. Z punktu S kierujemy się na N
2. Po prawej stronie widzimy drzewo.
3. Jesteśmy na rozwidleniu dróg. Po lewej stronie mamy szkołę i drogę, ale idziemy w prawo po łuku.
4. Po lewej mamy głaz.
5. Kierujemy się na NE.
6. Po prawej jest wiatrak.
7. Jesteśmy na skrzyżowaniu, możemy iść na NW, SE i N.
8. Idziemy na N.
9. Skręcamy po łuku na W.
10. Mijamy kwiaty po prawej stronie.
11. Idziemy na N po łuku, wygiętym w stronię W.
12. Po lewej mijamy namiot.
13. Idziemy na SE.
14. Po lewej widzimy jezioro.
15. Idziemy po łagodnym łuku wzdłuż S strony jeziora.
16. Jesteśmy na skrzyżowaniu, możemy iść na NE lub na S.
17. Idziemy na S.
18. Mijamy przystanek.
19. Stajemy na krzyżówce, możemy iść na S, E i W.
20. Idziemy na S.
21. Po prawej stronie mijamy samochód.
22. Idziemy po łuku na SE.
23. Po lewej widzimy psią budę.
24. Idziemy po łuku na SW.
25. Po prawej stronie widzimy autobus.
26. Idziemy na E.
27. Na przeciwko widzimy dziewczyny.
28. Możemy iść na N, E, S i SW.
29. Idziemy na SW.
30. Po lewej stronie mijamy rower.
31. Po łagodnym łuku idziemy na SW.
32. Idziemy na NW.
33. Po prawej stronie widzimy wieżę.
34. Stoimy na rozwidleniu.
35. Możemy iść na NE, albo na S.
36. Idziemy na S.
37. Po prawej stronie widzimy boisko.
38. Jesteśmy u celu, na mecie.

**KLASY UŻYTE W KODZIE:**

* **Punkt** - posiada informacje o swoich współrzędnych i wszystkich drogach i skrzyżowaniach, które przechodzą przez ten punkt oraz obiektach, które znajdują się w pobliżu tego punktu. Informacje o skrzyżowaniach, drogach i obiektach przechowywane są w postaci odpowiadających indeksów w globalnych tablicach skrzyżowań, dróg i obiektów.
* **Generuj\_kierunek** – generuje kierunki które później są przekazywane jako elementy do generowania drogi. Kierunki są generowane w sposób losowy, ale minimalizowana jest szansa na wystąpienie pętli w drogach.
* **Generuj\_drogę** – klasa ta zawiera informacje o typie drogi tzn. kierunku i zakrzywieniu. Odpowiada za rysowania drogi na canvas. Zawiera metodę ‘opisz\_wchodzace’ która generuje opis dla wszystkich dróg które wchodzą do tego punktu. ‘opisz\_ruchy’ dla wszystkich dróg które wychodzą z tego punktu i ‘wybierz’ dla drogi którą porusza się agent.
* **Generuj\_skrzyżowania** – w sposób losowy generuje drogi po których nie porusza się agent w losowych punktach. Posiada metodę ‘opisz\_skrzyzowania’ która działa podobnie do metody ‘opisz\_ruchy’ w klasie Generuj\_drogę.
* **Generuj\_obiekty**- tworzy obiekty w jednym z ośmiu kierunków wokół punktu od 0 do 8 obiektów. Posiada metodę ‘opisz\_obiekty’ która tworzy opis wszystkich obiektów znajdujących się wokół punktu.

**PRZEDSTAWIENIE LOGIKI KODU:**

1. **RYSUJ**

Na starcie losujemy punkt startowy oraz długość trasy z zakresu 15-20. W pętli generujemy kolejno kierunki, ruchy i kolejne punkty po punkcie startowym.

**Generowanie kierunków** przebiega w następujący sposób : w pętli *while* losujemy dowolną liczbę naturalną od 1 do 8 gdzie każda liczba odpowiada innemu kierunkowi następnie sprawdzamy czy poprzedni kierunek nie jest przeciwny do nowo wybranego kierunku(dzięki temu unikamy chodzenia cały czas w jednym kierunku). Obliczamy nowe współrzędne punktu i sprawdzamy czy ten punkt był już odwiedzany jeśli tak, to losowanie jest powtarzane. Dzieje się to maksymalnie 20 razy i ma to na celu zminimalizowanie ryzyka powstawania pętli w drodze. Kierunki są przechowywane w tablicy globalnej ‘Kierunki’.

**Generowanie ruchów** polega na pobieraniu kierunku z tablicy kierunków następnie wylosowaniu jednej z trzech możliwych krzywizn drogi. Na tej podstawie obliczane są nowe współrzędne punktu i współrzędne kontrolne będące argumentami funkcji rysującej krzywe. Dodajemy informację o tym, że punkt był już odwiedzany. Ruchy składujemy w tablicy globalnej ‘Ruchy’.

Na podstawie współrzędnych nowego punktu dodawany jest nowy obiekt klasy punkt do tablicy punkty. Do zmiennej punkt dodawane są i formacje o drogach, obiektach i skrzyżowaniach w otoczeniu tego punktu.

**Generowanie obiektów** polega na losowaniu kierunku w którym może pojawić się obiekt następnie sprawdzenia czy dane miejsce nie jest już zajęte, jeśli jest już zajęte to powtarzamy losowanie. Następnie losowany jest typ i podtyp dotyczących szczegółów o obiekcie. Obiekt zostaje dodany do punktu i narysowany na canvas.

1. **Opisz**

Generowanie opisu podaje informacje o współrzędnych punktu startowego następnie iterujemy po tablicy punktów i generujemy opis.

W pierwszej kolejności sprawdzamy którymi drogami z danego punktu już wychodziliśmy następnie usuwamy skrzyżowania których kierunki pokrywają się ze skrzyżowaniami kolejnych dróg. W drugiej kolejności wyszukujemy drogę którą będzie szedł agent, potem opisujemy wszystkie dostępne dla agenta drogi z danego punktu.

Forma zdania wygląda następująco:

**„Możemy iść na „ +**

**<lista kierunków dostępnych z tablicy ‘Ruchy’ > +**

**„oraz” +**

**<lista kierunków dostępnych z tablicy ‘Skrzyżowania’ >.**

W następnej kolejności opisujemy obiekty.

Forma zdania wygląda następująco:

**„Na” + <kierunek> +**

**„widzimy ” +**

**<typ obiektu> + <podtyp>.**

Następnie opisujemy drogę którą poszliśmy. Zdanie wygląda następująco:

**„Idziemy na ” + <kierunek> +**

**<typ łuku> (‘prosto’ lub ‘po łuku zakrzywionym’ na) +**

**<kierunek> (prostopadły do kierunku pierwszego).**

Po skończonym opisie program wypisuje „Jesteśmy u celu.”

1. **PRZERYSUJ**

Za przerysowywanie odpowiedzialna jest **funkcja parse** która pobiera z opisu wszystkie zdania. Każde zdanie dzielone jest na słowa i zapisywane w tablicy, następnie sprawdzamy słowa kluczowe zdania. Już po pierwszym słowie znamy typ zdania, które decyduje co robimy dalej.

Jeżeli pierwszym słowem jest „Punkt”, to wiemy że jesteśmy na początku. Odszukujemy w zdaniu współrzędne i rysujemy punkt w tych współrzędnych.

Jeśli pierwszym słowem jest „Na ” to mamy do czynienia z opisem obiektu. Sprawdzamy ile słów zajmuje określenie kierunku pobieramy typ i podtyp obiektu (gdzie typ zajmuje pozycję zależną od długości kierunku, a podtyp jest ostatnim słowem w zdaniu) Kiedy pobraliśmy kierunek możemy obliczyć gdzie znajduje się obiekt w stosunku do punktu a na podstawie typu i podtypu określamy jaki to obiekt. Rysujemy go na canvas.

Jeśli pierwszym słowem jest „Możemy” tzn. że jesteśmy na skrzyżowaniu lub w punkcie są drogi po których agent może się poruszać. Szukamy w zdaniu słowa kluczowego „oraz” które oddziela część listy kierunków tych dróg i losowych skrzyżowań. Pobieramy kierunki wszystkich dostępnych dróg i tymczasowo je ignorujemy oraz zaznaczamy na canvas wszystkie losowe skrzyżowania w punkcie.

Zdanie zaczynające się słowem „Idziemy” mamy do czynienia z wyborem drogi przez agenta. Ze zdania pobieramy informacje o kierunku oraz typie łuku. Na podstawie tych danych obliczamy współrzędne i punkt według którego zakrzywiona jest droga i rysujemy tą drogę. Przechodzimy do kolejnego punktu i powtarzamy procedurę.

**SŁOWNIK:**

|  |  |
| --- | --- |
| **KIERUNKI** | |
| Północ | + Odpowiednia forma w  Miejscowniku |
| Południe |
| Wschód |
| Zachód |
| Północny Wschód |
| Południowy Wschód |
| Północny Zachód |
| Południowy Zachód |
| **OBIEKTY** | |
| **TYPY** | **PODTYPY** |
| Drzewo | Buk |
| Brzoza |
| Świerk |
| Dom | Biały |
| Żółty |
| Zielony |
| Flaga | Czerwona |
| Niebieska |
| Zielona |
| Skrzynia | Czarna |
| Brązowa |
| Wiatrak | Brązowy |
| Niebieski |
| Pomarańczowy |
| Znak | Niebieski |
| Czerwony |
| Żółty |
| **Słowa pomocnicze** | |
| Możemy | |
| Iść | |
| Na | |
| Punkt | |
| Startowy | |
| Widzimy | |
| To | |
| Po | |
| Łuku | |
| Wygięty | |
| Koloru | |
| Gatunku | |
| Prosto | |
| Jesteśmy u celu | |
| W | |
| Idziemy | |
| Oraz | |

**Dziękujemy za uwagę !**