

# Dokumentacja wstępna

- Mikołaj Garbowski
- Michał Łuszczek

## Temat projektu

Oprogramowanie przeprowadzające ewolucję sztucznych pojazdów w 2D. Chodzi o wyewoluowanie pojazdu, który dotrze jak najszybciej do końca. Coś podobnego dostępne jest pod adresem: [http://rednuht.org/genetic\\_cars\\_2/](http://rednuht.org/genetic_cars_2/). Przydatna może być biblioteka do symulacji fizyki, np. Box2D. Stopień skomplikowania trasy powinien rosnąć wraz z odległością od początku. Trasa powinna jednak być przejezdna dla domyślnego pojazdu. Ewolucja powinna odbywać się na różnych trasach, aby możliwe było wyewoluowanie jak najlepszego pojazdu, a nie tylko dopasowanie się do konkretnej losowej trasy, tak aby przeskoczyć konkretną dolinkę. Przed rozpoczęciem realizacji projektu proszę zapoznać się z zawartością <http://staff.elka.pw.edu.pl/~rbiedrzy/ZPR/index.html>.

## Opis i koncepcja rozwiązania

Projekt będzie składać się z aplikacji desktopowej przedstawiającej wizualizację przebiegu symulacji ewolucji pojazdów 2D oraz interfejsu użytkownika do konfigurowania parametrów symulacji.

Ewolucja pojazdów będzie przeprowadzona, korzystając z algorytmu ewolucyjnego. Wizualizowane będą przebiegi kolejnych iteracji. Użytkownik aplikacji będzie mógł ustawić parametry algorytmu (np. siłę mutacji) i wybrać warianty składowych algorytmu ewolucyjnego (schemat reprodukcji, schemat selekcji).

Funkcja celu będzie wyliczana na podstawie przejechanej odległości i czasu. Samochód będzie składać się z kadłuba zbudowanego z trójkątów o wspólnym centralnym wierzchołku oraz 2 kół o osiach w zewnętrznych wierzchołkach trójkątów kadłuba, kręcących się ze stałą prędkością kątową. Samochody będą zdefiniowane przez genom, gdzie genami będą: położenie zewnętrznych wierzchołków trójkątów kadłuba, rozmiar kół, parametry fizyczne elementów składowych (np. gęstość) i położenie kół (wybór wierzchołka). Populacja będzie się kończyć, kiedy wszystkie samochody dotrą do końca trasy lub zmiana ich położenia będzie dostatecznie mała albo zostanie wyczerpany limit obliczeniowy.

Samochody będą poruszać się po losowo generowanej trasie, której poziom trudności rośnie wraz z odległością od startu. Co pokolenie generowana jest nowa trasa. Do generowania terenu planujemy użyć algorytmu szumu Perlina.

## Funkcjonalności

- Przeprowadzenie symulacji przejazdu populacji pojazdów dla iteracji algorytmu ewolucyjnego
- Wizualizacja przebiegu symulacji (kolejnych iteracji)
  - podgląd pojazdów na trasie
  - kamera podążająca za najdalej położonym pojazdem
- Przeprowadzenie ewolucji samochodów pomiędzy iteracjami
- Ustawianie parametrów ewolucji w GUI
  - wybór wariantów selekcji, reprodukcji, krzyżowania, mutacji i ich parametrów
- Ustawienie parametrów symulacji w GUI
  - długość trasy, parametry generatora trasy
- Wystartowanie ewolucji o ustawionych parametrach
- Przedstawienie wyników po zakończonej ewolucji
  - wyświetlenie modelu najlepszego pojazdu i jego parametrów (genomu)

## Lista zadań

| Grupa                              | Zadanie  | Czas<br>[h] |
|------------------------------------|--|-------------|
| Dobór bibliotek                    | silnik fizyczny  | 1           |
|                                    | renderowanie grafiki   | 1           |
|                                    | gui  | 1           |
|                                    | testy  | 1           |
| Zamodelowanie hierarchii klas      | podział programu na moduły   | 3           |
| Przygotowanie minimalnej aplikacji | przygotowanie diagramów UML  | 3           |
|                                    | zapoznanie się z możliwościami silnika fizycznego  | 3           |
|                                    | zapoznanie się z biblioteką do GUI   | 3           |
|                                    | zapoznanie się z biblioteką do renderowania grafiki  | 4           |
|                                    | uproszczony model ewolucji   | 2           |
|                                    | uproszczony model samochodu  | 2           |
|                                    | uproszczony model trasy  | 2           |
|                                    | uproszczone sterowanie symulacją z GUI   | 2           |
|                                    | integracja modelu ewolucji z GUI   | 3           |
|                                    | integracja silnika fizycznego z GUI  | 5           |
|                                    | integracja silnika fizycznego z uproszczonym modelem ewolucji                                | 4           |
| Zestawienie potoku CI              | dla Ubuntu   | 2           |
|                                    | dla Windowsa   | 3           |
| Generowanie trasy                  | zestawienie narzędzi do analizy statycznej, formatowania kodu                                | 2           |
|                                    | wybór i implementacja algorytmu  | 3           |
| Oprogramowanie ewolucji            | wygenerowanie modelu trasy w silniku fizycznym   | 3           |
|                                    | zamodelowanie genomu   | 2           |
|                                    | implementacja algorytmu ewolucyjnego z możliwością wymiany wariantów algorytmu               | 3           |
|                                    | implementacja funkcji celu   | 1           |
|                                    | implementacja wariantów krzyżowania  | 3           |
|                                    | implementacja wariantów mutacji  | 3           |
|                                    | implementacja wariantów reprodukcji  | 3           |
|                                    | implementacja wariantów selekcji   | 3           |
| Symulacja przejazdu                | generowanie modelu fizycznego samochodu na podstawie genomu                                  | 5           |
|                                    | symulowanie oddzielnych przejazdów po jednej trasie  | 10          |
|                                    | pobranie parametrów dla wszystkich osobników: dystans, czas                                  | 2           |
| Wizualizacja iteracji ewolucji     | wyświetlanie trasy   | 2           |
|                                    | wyświetlanie pojazdów  | 4           |
| GUI                                | podążanie kamerą za najdalej położonym pojazdem  | 3           |
|                                    | wyświetlanie parametrów liczbowych symulacji w czasie rzeczywistym: numer pokolenia, dystans | 2           |
|                                    | kontrolki do wyboru parametrów dla każdego wariantu składowych algorytmu ewolucyjnego        | 4           |
|                                    | kontrolki do wyboru parametrów symulacji: parametry generatora trasy, grawitacja             | 2           |
|                                    | sterowanie przebiegiem symulacji: start, stop  | 4           |
|                                    | wyświetlanie wyników po zakończeniu ewolucji   | 3           |
| <b>SUMA</b>                        |  | <b>112</b>  |