Dokumentacja projektu PSZT

Autorzy: Łopusiński Paweł, Kaczor Bartosz, Domagała Bartosz

# Decyzje projektowe

## Treść zadania

Należy stworzyć prostą grę typu "orbit" (http://www.2dplay.com/orbit/orbit.swf) ,   
a następnie zastosować algorytm ewolucyjny znajdujące wartości wejściowe, które maksymalizują długość "życia" obiektu.

## Środowisko programistyczne

Projekt został przez nas napisany w języku C++ z użyciem bibliotek Qt.

## Algorytm rozwiązania zadania

Zdecydowaliśmy się skorzystać z algorytmu ewolucyjnego µ + λ. Daje on potencjalnie szansę na dość szybkie znalezienie rozwiązania, dzięki licznym krzyżowaniom i dodatkowym mutacjom. Ponadto najsilniejsi osobnicy danej populacji, przechodzą do populacji następnej i mogą dalej się rozmnażać, co zwiększa szansę na jeszcze silniejszych potomków. Z tego też względu odrzuciliśmy algorytm (µ, λ), który nie brałby pod uwagę populacji źródłowej, a samych potomków.

Algorytm 1 + 1 został odrzucony ze względu na brak krzyżowania oraz duże uproszczenie, które de facto prowadziło do tego, że była mała szansa na szybkie znalezienie zadowalającego wyniku. Implementacja równoległa ostatnio wymienionego algorytmu również byłaby mniej efektywna od algorytmu µ + λ.

# Instrukcja dla użytkownika

## Kompilacja

Do kompilacji wymagane są biblioteki Qt.

## Uruchomienie

Uruchomienie następuje po uprzedniej kompilacji.

## Obsługa

Do obsługi programu wystarczy mysz komputerowa. Po otwarciu okna programu możemy wygenerować poziom ustawiając odpowiednie parametry: docelowy czas „życia” komety, ilość planet, minimalna i maksymalna waga planety a następnie klikając przycisk „Generuj poziom”. Wagę każdej wygenerowanej planety możemy zobaczyć najeżdżając na nią kursorem. Następnie możemy wybrać tryb gry „Graj samemu” lub symulować rozwiązania klikając przycisk „Symuluj”. W drugim przypadku możemy dodatkowo ustalić ile populacji zostanie utworzonych w jednym kroku, a następnie kliknąć przycisk „Następna populacja”. Na ekranie zobaczymy punkty początkowe osobników oraz drogę jaką przebyli. Po każdym kroku możemy zobaczyć szczegółowe dane na temat każdego osobnika klikając „Wyświetl zestawienie”.  
Program wyłączamy klikając krzyżyk w prawym górnym rogu ekranu.

# Opis struktury programu

## Ogólny opis programu

# Wnioski dotyczące rezultatów

Działanie algorytmu jest zgodne z oczekiwaniami. Najsilniejsi osobnicy oraz najsilniejsi potomkowie zostają rodzicami w kolejnych populacjach. Zdarza się też, że niektórzy osobnicy są w stanie przetrwać kilkadziesiąt populacji, jeśli ich czas życia jest dostatecznie długi, tym samym nadal tworząc silnych potomków. W związku z tym wyniki są zadowalające. Dużo zależy od początkowej populacji, która generowana w całości pseudolosowo może znaleźć się blisko jak i daleko od planet. W przypadku rozwijania programu warto byłoby się zastanowić nad algorytmem, który wybierał by optymalne współrzędne oraz wektory prędkości dla początkowych osobników. Można by także pomyśleć o wyborze początkowej populacji przez użytkownika. Człowiek bowiem wybierałby najprawdopodobniej punkty przewidując początkowy ruch komety, co daje przewagę nad wyborem pseudolosowym. Powyższe zabiegi mogłyby jeszcze bardziej usprawnić działanie algorytmu i znajdować osobników, których czas „życia” byłby jeszcze dłuższy.

Użytkownik w grze kieruje się przede wszystkim intuicją i stara się przewidzieć początkowy ruch komety ze względu na odległość planet oraz ich przyciąganie. Daje to dużą przewagę nad programem, który wybiera stricte losowe miejsca i prędkości. Jak jednak było wyżej wspomniane jest możliwość usprawnienia tego zagadnienia poprzez dodatkowe algorytmy albo wybór początkowej populacji przez człowieka.