

WYDZIAŁ ARCHITEKTURY, BUDOWNICTWA   
I SZTUK STOSOWANYCH

**OPRACOWANIE GRY KOMPUTEROWEJ WYKORZYSTUJĄCEJ ALGORYTMY GENETYCZNE**

PRACA INŻYNIERSKA

PROMOTOR: AUTOR:

**dr inż. ŁUKASZ WALUSIAK ARTUR PILICHOWSKI**

KIERUNEK: informatyka

TRYB: studia niestacjonarne

NUMER ALBUMU:10256

KATOWICE, 2022

1. WSTĘP

1.1 Gry komputerowe

Gry pełnią przede wszystkim funkcję ludyczną (zapewniając rozrywkę zarówno uczestnikom jak i obserwatorom) i prezentacyjną (ukazując sprawność fizyczną lub intelektualną uczestników). Choć stwierdzenie to wymagałoby dowodu (ang. proof), co mogłoby wiązać się z osobną pracą dyplomową, to nawet pobieżna obserwacja różnego rodzaju gier (od planszowych po różnorakie dyscypliny sportu) dostarczyłaby wielu dowodów (ang. evidence) zdających się je potwierdzać.

Nie inaczej jest w przypadku gier komputerowych, nawet u zarania ich istnienia, tj. w latach 40. i 50. XX wieku. Powstałe wówczas gry takie jak „Tennis for Two” z 1958 roku czy późniejszy „Spacewar” zdecydowanie nie były projektami komercyjnymi, jednak nadawały się do prezentacji działania ówczesnych komputerów publiczności niezaznajomionej z budową i działaniem komputerów ani skomplikowanymi obliczeniami przez nie wykonywanymi.

Współcześnie prezentacyjna funkcja gier komputerowych może być kojarzona z dążeniem do fotorealizmu grafiki 3D renderowanej w czasie rzeczywistym w wysokobudżetowych produkcjach, jednak nie wyczerpuje to dostępnych możliwości.

1.2 Algorytmy

Algorytm definiuje się jako ‘skończony ciąg jednoznacznych działań prowadzących do rozwiązania problemu należącego do danej klasy’ bądź ‘jednoznacznie zdefiniowaną procedurę obliczeniową, która dla otrzymanych danych wejściowych produkuje odpowiednie dane wyjściowe’ lub po prostu przepis na rozwiązanie danego problemu.

Algorytm zaimplementowany w programie komputerowym musi rozwiązywać problem w skończonym czasie i przy wykorzystaniu skończonych zasobów (głównie pamięci). Miarą kosztów jego działania jest złożoność obliczeniowa, odpowiednio: czasowa i pamięciowa, którą opisuje się jako funkcję rozmiaru danych wejściowych.

Dążenie do minimalizacji złożoności obliczeniowej przyczyniło się do opracowania różnorakich paradygmatów tworzenia algorytmów, takich jak metoda „dziel i zwyciężaj”, algorytmy zachłanne lub algorytmy genetyczne.

1.3 Algorytmy genetyczne

Algorytmy genetyczne są rodzajem heurystyki (lub metaheurystyką), inspirowanej biologicznymi procesami ewolucji i doboru naturalnego.

Ogólne działanie algorytmów genetycznych polega na wygenerowaniu zbioru potencjalnych rozwiązań (tzw. populacji) będącego podzbiorem wszystkich możliwych rozwiązań danego problemu, a następnie cyklicznemu przetwarzaniu go za pomocą operacji krzyżowania, mutacji i selekcji (operacje te są zwykle przynajmniej w pewnym stopniu losowe, stąd algorytmy genetyczne klasyfikuje się także jako algorytmy stochastyczne). Kolejne iteracje tych operacji (tzw. pokolenia) powinny powodować polepszenie średnich wyników w populacji.

Algorytmy genetyczne są stosowane w dziedzinach takich jak optymalizacja i przeszukiwanie, gdzie wykazują istotną przewagę nad tradycyjnymi algorytmami analitycznymi, dzięki wysokiej odporności, rozumianej jako równowaga między wydajnością a skutecznością, łatwości zastosowania do szerokiego spektrum problemów oraz niewrażliwości na pułapki stanowione przez rozwiązania pozorne (np. ekstrema lokalne w zadaniach optymalizacyjnych).

2. ZAŁOŻENIA I CEL PRACY

Celem niniejszej pracy jest implementacja algorytmu genetycznego wraz z funkcjonalnościami mającymi umożliwić śledzenie jego działania w trakcie wykonywania programu oraz gromadzić dane pozwalające na późniejszą analizę tegoż działania. Analiza ta powinna przynieść odpowiedź na pytanie: na ile zasadne jest użycie mechanizmu algorytmów genetycznych w taki sposób, jak w opisywanej tutaj aplikacji. Sama analiza jak i odpowiedź na wspomniane pytanie nie należą do zakresu niniejszej pracy.

2.1 Założenia

2.1.1 Mechanizm algorytmów genetycznych

Mechanizm algorytmów genetycznych, rozumianych jako rodzina algorytmów oparta o ideę doboru naturalnego i dziedziczności, posiada szereg elementów stałych, na czele z kluczowymi operacjami selekcji, krzyżowania i mutacji. Sposoby realizacji tych operacji (a także ich kolejność) mogą się jednak znacznie różnić, np. na skutek wymagań narzucanych przez rozwiązywany problem.

2.1.1.1 Tworzenie populacji

Pierwszym krokiem jest stworzenie populacji, czyli zbioru elementów zwanych osobnikami, z których każdy reprezentuje potencjalne rozwiązanie. Każdy osobnik posiada określony genom, czyli jeden lub więcej ciągów wartości odpowiadających (bezpośrednio lub w postaci zakodowanej) danym, z których składa się rozwiązanie. Dla przykładu: w przypadku problemu komiwojażera populacja zawierałaby różne cykle Hamiltona przedstawione jako ciągi wierzchołków lub krawędzi składających się na dany cykl, natomiast w przypadku problemu, którego rozwiązaniem jest pojedyncza wartość liczbowa, populacja może być zbiorem liczb przedstawionych w postaci binarnej, jako ciągi zero-jedynkowe. Niezależnie od przyjętej formy i postaci, populacja powinna być losowym podzbiorem przestrzeni rozwiązań danego problemu. Rozmiar populacji jest ustalany arbitralnie.

2.1.1.2 Selekcja

Selekcja, zwana także reprodukcją, jest procesem mającym na celu wyłonienie z aktualnej populacji najbardziej wartościowych osobników (najlepszych rozwiązań) przy jednoczesnym zapewnieniu różnorodności genetycznej populacji w następnym pokoleniu.

W pierwszej kolejności należy określić jakość każdego z aktualnie rozpatrywanych rozwiązań. Dla każdego z nich obliczana jest jego wartość za pomocą tzw. funkcji celu (zwanej też, w nawiązaniu do teorii ewolucji, funkcją przystosowania). Obliczenia te zazwyczaj uwzględniają wartości zapisane w genomie danego osobnika, ale postać funkcji celu jest zależna od rozwiązywanego problemu i formy, w jakiej zapisane są rozwiązania.

Następnie dokonywana jest selekcja na podstawie wartości osobników. Najprostszym sposobem jej realizacji jest metoda ruletki, czyli losowy wybór, w którym prawdopodobieństwo wybrania danego osobnika jest wprost proporcjonalne do jego wartości. Metoda ta jest wystarczająca do prawidłowego działania algorytmu, choć można ją wzbogacić o mechanizmy przeciwdziałające dominacji jednego z osobników lub zapewniające przetrwanie najlepszego z nich. Bez względu na wybraną metodę, wybrany osobnik zostaje dodany do nowego zbioru, który będzie stanowił następne pokolenie populacji lub podstawę dla jego utworzenia, jednocześnie pozostając w dotychczasowej populacji, co umożliwia jego ponowny wybór. Proces ten jest powtarzany do momentu osiągnięcia założonej liczebności nowego zbioru.

2.1.1.3 Krzyżowanie

W procesie krzyżowania dokonywana jest wymiana informacji genetycznej pomiędzy parami osobników z dotychczasowej populacji. Wymianie mogą podlegać losowe fragmenty genomu lub jego pojedyncze elementy (tzw. geny). Osobniki, które mają zostać skrzyżowanie, są również wybierane losowo, a po zakończeniu wymiany nie biorą udziału w dalszych iteracjach tego procesu. Operacja jest powtarzana dopóki w populacji występuje choć jedna para osobników nie poddanych krzyżowaniu.

2.1.1.4 Mutacja

Operacja mutacji polega na zmianie wartości losowych genów. Podobnie jak w naturze, prawdopodobieństwo zajścia takiej zmiany powinno być bardzo niskie.

2.1.2 Zastosowanie algorytmów genetycznych w grach komputerowych

Gry komputerowe nie są wymieniane wśród czołowych obszarów zastosowań algorytmów genetycznych, co może budzić zdziwienie, zważywszy na istnienie szeregu zagadnień w ramach tego medium, w których ukierunkowana losowość i adaptacyjność, charakterystyczne dla algorytmów genetycznych, wydają się być pożądane. Dotyczą one przede wszystkim dostosowania mechanik gry do działań graczy – czy to w celu zwiększenia realizmu interakcji, czy dla urozmaicenia rozgrywki – czego przykładem może być mechanika dynamicznego poziomu trudności.

Jednakże, nie można stwierdzić z całą pewnością, że algorytmy genetyczne mogą zawsze zapewnić odpowiednie rozwiązanie w we wszystkich kwestiach dotyczących mechaniki gier. Ma na to wpływ wiele czynników kształtujących tego typu algorytmy.

2.1.2.1 [Zastosowanie – do jakich problemów się nadają?]

Nawet pobieżna analiza problemów, do rozwiązywania których używane są algorytmy genetyczne, pozwala na wyróżnienie kilku cech wspólnych, które należałoby uznać za sprzyjające a może nawet konieczne.

Pierwszą z nich jest obecność funkcji matematycznej zawartej w opisie problemu, która może posłużyć za funkcję celu. Za przykład mogą posłużyć zadania optymalizacyjne polegające na poszukiwaniu ekstremum wspomnianej funkcji. Odpowiednio zdefiniowana funkcja celu pozwala na jednoznaczne przyporządkowanie wartości osobnika danej kombinacji jego genów. Jest to istotne dla procesu znajdowania najlepszego rozwiązania – wprowadza element stałości i stabilizacji do tego silnie losowego procesu, ukierunkowując poszukiwania.

Wspomniana jednoznaczność wynika ze swoistej „statyczności” problemu, rozumianej jako niezmienność opisujących go danych i warunków determinujących rozwiązanie. Można to też określić jako poszukiwanie jak najlepszego rozwiązania dla jednego, ściśle określonego problemu. Przykładem dobrze ilustrującym sens tej idei jest wspomniany już problem komiwojażera, w przypadku którego w zadanym grafie ważonym należy odnaleźć najmniejszy cykl obejmujący wszystkie jego wierzchołki. Graf pozostaje zawsze taki sam – w trakcie rozwiązywania nie zmienia się ani liczba wierzchołków, ani połączenia między nimi, ani wagi krawędzi. Jakąkolwiek z wymienionych zmian należałoby zinterpretować jako wygenerowanie nowego grafu, a w konsekwencji – nowego problemu do rozwiązania. Analogicznie przykładem problemu nie spełniającego tego założenia mogłoby być rozpoznawanie pisma – zadanie wymagające analizy wielu próbek, z których każda stanowi właściwie osobny problem do rozwikłania. W tym przypadku zastosowanie znajdują sieci neuronowe, które na podstawie dotychczasowych problemów „uczą się” rozwiązywać kolejne. Co prawda algorytmy genetyczne również są stosowane do tworzenia systemów uczących się, jednak jest to „nauka” metodą prób i błędów przez wielokrotne podchodzenie do tego samego zadania. Możliwość zastosowania takiej metody jest istotną cechą łączącą omawiane tu problemy.

Kolejnym elementem, na który warto zwrócić uwagę, jest zakres dopuszczalnych rozwiązań. Stosowanie algorytmów genetycznych jest zasadne w przypadkach, w których możliwy kompromis pomiędzy jakością wyniku a czasem uzyskania go. Akceptowalność rozwiązań reprezentowanych przez osobniki jest określana poprzez porównanie przyporządkowanej im wartości funkcji celu z ustalonym wcześniej progiem. Wspomniany kompromis powoduje, że wartość tego progu jest istotnie niższa od maksymalnej możliwej dla danego problemu. To z kolei oznacza, że osobnik będący końcowym rezultatem pracy algorytmu genetycznego jedno z rozwiązań „wystarczająco dobrych”, nie koniecznie najlepsze. W przypadkach, w których taki kompromis nie jest możliwy, o wiele bardziej zasadne jest zastosowanie wyspecjalizowanego algorytmu analitycznego.

2.1.2.2 [O zastosowaniu w grach – problem interaktywności]

Podstawowym czynnikiem stojącym na przeszkodzie zastosowaniu mechanizmu algorytmów genetycznych w grach komputerowych zdaje się być ich interaktywność. Działania gracza powodują zmianę stanu gry, z punktu widzenia algorytmu tworząc nowy problem do rozwiązania.

Odbija się to przede wszystkim na ocenie populacji, gdyż wartości zwracane przez dotychczasową funkcję celu mogą okazać się błędne w nowych okolicznościach. Utrata jednoznaczności przyporządkowania wartości do danego osobnika wydaje się być nieuchronna. Nie jest przy tym pewne, czy w takich warunkach wartości genów mają bezpośrednie przełożenie na jakość reprezentowanego rozwiązania. Być może należy zidentyfikować zewnętrzne czynniki wpływające na „sprawność” osobników i oprzeć nową metodę ewaluacji właśnie na nich.

Niestety takie podejście wiąże się z ryzykiem wydłużenia czasu potrzebnego na wykonanie jednego cyklu pracy algorytmu genetycznego. Jeśli osobniki miałyby być powiązane z obiektami w grze, a ocena miała wynikać z jakości ich interakcji z innymi elementami gry (w tym z graczem), to selekcja potrwałaby zdecydowanie dłużej niż gdyby została przeprowadzona przy użyciu funkcji matematycznej przyjmującej geny jako argumenty.

Opisanych wyżej problemów można uniknąć, stosując algorytmy genetyczne do zadań niezależnych od działań gracza i innych źródeł zmienności, a nawet nie związanych samą rozgrywką, lecz z tworzeniem gry lub jej elementów. Zagadnienia te nie wchodzą jednak w zakres niniejszej pracy, a samo rozwiązanie jest jedynie unikiem.

Istnieje natomiast prawdopodobieństwo, że ukierunkowana losowość zapewniana przez algorytmy genetyczne wystarczy, by dać zadowalające wyniki nawet w tak niesprzyjających warunkach. Z punktu widzenia gracza dużo ważniejsza może okazać się obecność i ciągłość działania procesu dostosowywania elementów rozgrywki niż uzyskanie przez ten proces jakiegoś konkretnego rozwiązania.

2.2 Wykorzystane technologie

Najważniejszym elementem aplikacji wchodzącej w skład niniejszej pracy jest zaimplementowany algorytm genetyczny, natomiast gra stanowi jedynie platformę konieczną, by implementacja mogła mieć miejsce. Tak określona hierarchia sprawia, że strona wizualna programu ma relatywnie niewielkie znaczenie, dlatego też wybór języka Java z wykorzystaniem biblioteki AWT okazał się wystarczający do realizacji wszystkich istotnych założeń dotyczących funkcjonalności programu.

3. OPIS APLIKACJI

3.1 Okno aplikacji

Okno aplikacji ma wymiary 800x800 pikseli i jest podzielone na dwa obszary: właściwy obszar gry o szerokości 480 pikseli oraz dodatkowy obszar, w którym wyświetlane są dane związane z pracą algorytmu genetycznego, o szerokości 320 pikseli. Wymiary te można zmienić jedynie poprzez modyfikację wartości odpowiadających im zmiennych w kodzie źródłowym aplikacji.

3.2 Działanie aplikacji [front-end?]

3.2.1 Zasady gry

Aplikacja jest inspirowana grą „Space Inviders”, co jest widoczne w wielu elementach rozgrywki, w tym w warstwie wizualnej, mimo iż implementacja mechanizmu algorytmów genetycznych wymusiła wprowadzenie szeregu istotnych zmian.

Wrogowie są reprezentowani przez sprite’y wzorowane na kosmitach ze wspomnianej gry. Są oni tworzeni grupami (falami) i wprowadzani na górnym krańcu obszaru gry pojedynczo, w krótkich odstępach czasowych. Mogą poruszać się w lewo, w prawo lub w dół oraz strzelać. Wprowadzanie nowej fali rozpoczyna się dopiero w momencie wyczerpania poprzedniej, tj. kiedy wszyscy dotychczasowi wrogowie zostaną wyeliminowani lub opuszczą obszar gry.

Gracz steruje obiektem reprezentowanym przez biały kwadrat u dołu okna. Może on poruszać się w lewo lub w prawo albo strzelać.

Pociski wystrzelone przez gracza i wrogów są reprezentowane odpowiednio przez jasnoniebieskie i czerwone kwadraty. Różnią się ona także kierunkiem (właściwie: zwrotem) ruchu – te pierwsze poruszają się pionowo w górę, natomiast te drugie poruszają się pionowo w dół.

Pociski gracza pojawiają się w grze po naciśnięciu przez niego przycisku odpowiadającego komendzie strzału (domyślnie jest to „w” lub „strzałka w górę”), jednak nie częściej niż pozwala na to odgórnie ustalony limit czasowy.

Wrogie pociski są wystrzeliwane przez losowego wroga w odgórnie ustalonych odstępach czasowych. Długość tego odstępu podlega modyfikacjom zależnym od liczebności wrogów – wraz z ich ubywaniem w ramach danej fali czas pomiędzy kolejnymi wystrzałami staje się krótszy.

Zadaniem wrogów jest dotarcie do dolnego krańca obszaru gry. Zadaniem gracza jest uzyskanie jak najwyższego wyniku punktowego poprzez niszczenie wrogów, jednocześnie jak najdłużej unikając kolizji z nimi i ich pociskami.

3.2.2 Kolizje

W trakcie gry może dojść do następujących kolizji pomiędzy obiektami:

* pomiędzy pociskiem gracza a wrogiem – pocisk zostaje usunięty z gry a dany wróg „zniszczony”, a jego wartość zostaje dodana do aktualnej puli punktów.
* pomiędzy pociskiem wroga a obiektem gracza – pocisk zostaje usunięty z gry a wartość licznika „życia” gracza zostaje zmniejszona o 1.

Jeśli wartość tego licznika spadnie to zera, gra zostaje zakończona.

* pomiędzy obiektem gracza a wrogiem – wróg zostaje „zniszczony” a jego wartość dodana do wyniku punktowego. Licznik „życia” ulega zmniejszeniu o 1.

Jeśli dany wróg dotrze do dolnego krańca obszaru gry, zostanie on „zniszczony”, ale jego wartość będzie odjęta od wyniku punktowego.

3.2.3 Zakończenie gry

Gra kończy się w momencie, gdy wartość licznika „życia” osiągnie 0 lub gracz sam zakończy rozgrywkę.

3.3 Struktura aplikacji

3.3.1 Menu główne

Menu główne zostaje wyświetlone zaraz po uruchomieniu aplikacji. Składa się ono z czterech przycisków:

* Play – kliknięcie spowoduje rozpoczęcie nowej rozgrywki;
* Load – kliknięcie spowoduje odczyt danych dotyczących stanu poprzednio zapisanej rozgrywki i wznowienie jej, o ile istnieją pliki z tymi danymi;
* Options – kliknięcie spowoduje przejście do ekranu opcji;
* Exit – kliknięcie spowoduje zakończenie pracy aplikacji.

3.3.2 Menu pauzy

Jeśli w trakcie rozgrywki zostanie naciśnięty przycisk odpowiadający komendzie pauzy (domyślnie: klawisz „Escape”), to rozgrywka ta zostanie wstrzymana a w miejscu obszaru gry będzie wyświetlane menu pauzy. Obszar z danymi dotyczącymi pracy algorytmu genetycznego pozostanie nadal widoczny przez cały czas trwania pauzy.

Menu składa się z następujących przycisków:

* Resume – kliknięcie spowoduje zakończenie pauzy, powrót do aktualnej rozgrywki;
* Save – kliknięcie spowoduje zapis kluczowych danych dotyczących stanu aktualnej rozgrywki do odpowiednich plików;
* Quit – kliknięcie spowoduje zakończenie aktualnej rozgrywki i powrót do menu głównego.

Szczegółowe informacje o zapisie i odczycie gry znajdują się w sekcji 3.4.2.

3.3.3 Ekran opcji

Ekran opcji pozwala na modyfikację parametrów istotnych dla pracy algorytmów genetycznych zaimplementowanych w aplikacji, czyli: rozmiar populacji (pod nazwą „Liczba wrogów w fali”), rozmiar genomu, prawdopodobieństwo krzyżowania i mutacji oraz rozmiar populacji dla implementacji drugorzędnej (pod nazwą „Liczba fal do krzyżowania”). Szczegóły dotyczące znaczenia tych parametrów znajdują się w sekcji 3.4.1.

W skład ekranu opcji chodzą też dwa przyciski (poza przyciskami służącymi do modyfikacji wartości parametrów):

* Save – kliknięcie spowoduje przekazanie aktualnej wartości parametrów ze zmiennych tymczasowych do zmiennych globalnych, dzięki czemu będą one mogły zostać uwzględnione w najbliższej nowej rozgrywce i ewentualnie zapisane do pliku;
* Back – kliknięcie spowoduje powrót do menu głównego.

3.3.4 Ekran końca gry

Jeśli gracz straci w trakcie rozgrywki wszystkie dostępne „życia”, to zostanie ona zakończona oraz zostanie wyświetlony komunikat pożegnalny wraz z informacją o liczbie punktów uzyskanych w trakcie tej rozgrywki. Kliknięcie przycisku „Exit” spowoduje powrót do menu głównego.

3.4 [Back end]

3.4.1 Implementacja algorytmów genetycznych

Najważniejszym elementem opisywanej tu aplikacji są dwie niezależne implementacje algorytmów genetycznych. Pierwsza z nich obsługuje zachowanie wrogów i ma znaczenie priorytetowe – to jej działanie jest przedmiotem obserwacji będącej podstawowym celem tej aplikacji. Druga implementacja obsługuje proces ustalania początkowych pozycji na których mają pojawić się wrogowie wprowadzani do gry. Jej znaczenie jest jednak marginalne, co okazało się na stosunkowo późnym etapie tworzenia aplikacji, co z kolei sprawiło, że jej wymiana na lepsze rozwiązanie stała się nieopłacalna.

3.4.1.1 Kluczowe pojęcia

[Terminologia związana z grami i związek z pojęciami dotyczącymi algorytmów genetycznych]

Z punktu widzenia algorytmu obsługującego zachowania wrogów stanowią oni osobniki reprezentujące rozpatrywane rozwiązania.

[populacja, pokolenia = fale]

[genom = sekwencja ruchów]

3.4.1.2 [Opis]

3.4.2 Zapis i odczyt gry