

Projekt 01 [o nietypowych przygodach z ewolucją]

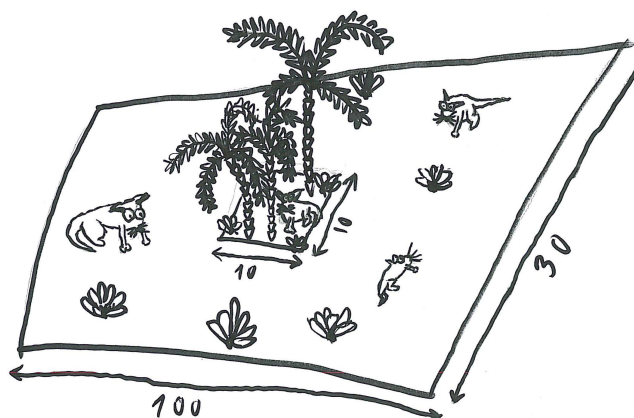
Niniejsza treść została zaadaptowana przez Aleksandra Smywińskiego-Pohla na podstawie opisu oraz ilustracji przygotowanych przez Wojciecha Kosiora. Inspiracją dla niego była z kolei książka "Land of Lisp" Conrada Barskiego, który zaś zainspirował się artykułem w "Scientific American". A na końcu modyfikacje wprowadził Radosław Łazarz, bazując częściowo na książce "Algorytmy genetyczne i ich zastosowania" Davida E. Goldberga. Dużo ludzi jak na jeden projekcik. :-)

Cel projektu

Stwórzmy grę! Nie będzie to jednak gra, w którą my gramy. Zamiast tego będzie to świat, który ewoluuje na naszych oczach! Stworzymy środowisko stepów i dżungli ze zwierzętami, które biegają, buszują w zaroślach, jedzą i rozmnażają się. A po kilku milionach lat zobaczymy, że wyewoluowały w różne gatunki!

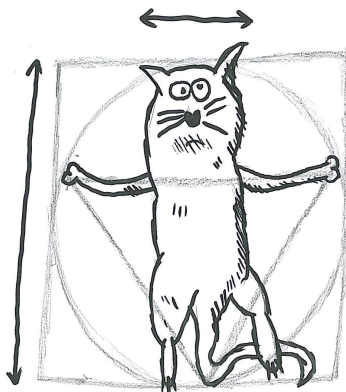


Świat naszej gry jest dość prosty. Składa się ze zwykłej, prostokątnej polaci podzielonej na kwadratowe pola. Większość świata pokrywają stepy, na których rośnie niewiele roślin stanowiących pożywienie zwierząt. Niektóre rejeony porasta jednak dżungla, gdzie rośliny rosną dużo szybciej. Rośliny będą wyrastały w losowych miejscach, ale ich koncentracja będzie większa w dżungli niż na stepie.



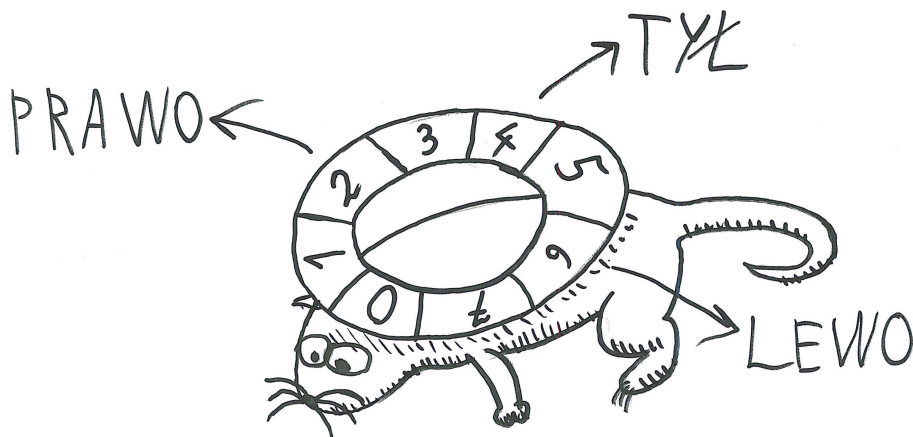
Nasze zwierzęta, które są roślinożercami, będą przemierzały ten świat w poszukiwaniu pożywienia. Każde zwierzę ma określoną energię, która zmniejsza się co dnia. Znalezienie i zjedzenie rośliny zwiększa poziom energii o pewną wartość.

Anatomia zwierzęcia



Musimy śledzić kilka cech każdego zwierzęcia. Po pierwsze, zarówno w przypadku rośliny jak i tych, którzy je zjadają, musimy znać koordynaty x i y . Wskazują nam one, gdzie dane zwierzę lub roślina jest na mapie.

Musimy także wiedzieć, ile energii ma dane zwierzę. To darwinowska gra o przetrwanie, więc jeśli zwierzątko nie zdoła zdobyć odpowiedniej ilości pożywienia, będzie głodować i zdechnie... Energia mówi nam o tym, ile dni funkcjonowania zostało jeszcze danemu zwierzątku. Musi ono koniecznie znaleźć więcej jedzenia, zanim jej zapas się wyczerpie.



Musimy również pamiętać, w którą stronę zwierzę jest zwrócone. Jest to ważne, ponieważ każdego dnia będzie ono poruszać się na mapie w tym właśnie kierunku. Istnieje osiem różnych możliwych pozycji i tyle samo możliwych obrotów. Obrót **0** oznacza, że zwierzę nie zmienia swojej orientacji, obrót **1** oznacza, że zwierzę obraca się o 45° , **2**, o 90° , itd. Przykładowo: jeśli zwierzę było skierowane na północ i obrót wynosi **1**, to zwierzę skierowane jest teraz na północny wschód.

Na koniec musimy także przechowywać geny zwierzaka. Każde zwierzę ma N genów, z których każdy jest jedną liczbą z zakresu od **0** do **7**. Geny te opisują (w bardzo uproszczony sposób) schemat zachowania danej istoty. Egzystencja naszych zwierczaków ma cykliczną naturę. Każdy z nich przechowuje informację o tym, z którego fragmentu swojego genomu będzie korzystał najbliższego dnia. Podczas każdego ruchu zwierzak zmienia najpierw swoje ustawienie, obracając się zgodnie z aktualnie aktywnym genem, a potem porusza się o jedno pole w wyznaczonym kierunku. Następnie gen ulega dezaktywacji, a aktywuje się gen na prawo od niego (będzie sterował zwierzakiem kolejnego dnia). Gdy geny skończą się, to aktywacja wraca na początek ich listy. Przykładowo - genom:

oznacza, że żyjątko będzie kolejno: szło przed siebie, szło przed siebie, szło nieco w lewo, szło przed siebie, zawracało, szło przed siebie, ... - itd.

Konsumpcja i rozmnażanie

Jedzenie jest prostym procesem. Zakładamy, że zwierzę zjada roślinę, gdy stanie na jej polu, a jego energia wzrasta wtedy o z góry zdefiniowaną wartość.

Rozmnażanie jest zwykle najciekawszą częścią każdej symulacji ze zwierzętami. Zdrowe młode może mieć tylko zdrowa para rodziców, dlatego nasze zwierzęta będą się rozmnażać tylko jeśli mają odpowiednią ilość energii. Przy reprodukcji rodzice tracą na rzecz młodego pewną część swojej energii - ta energia będzie równocześnie stanowić startową energię ich potomka.

Urodzone zwierzę otrzymuje genotyp będący krzyżówką genotypów rodziców. Udział genów jest proporcjonalny do energii rodziców i wyznacza miejsce podziału genotypu. Przykładowo, jeśli jeden rodzic ma 50, a drugi 150 punktów energii, to dziecko otrzyma 25% genów pierwszego oraz 75% genów drugiego rodzica. Udział ten określa miejsce przecięcia genotypu, przyjmując, że geny są uporządkowane. W pierwszym kroku losowana jest strona genotypu, z której zostanie wzięta część osobnika silniejszego, np. *prawa*. W tym przypadku dziecko otrzymałoby odcinek obejmujący 25% *lewych* genów pierwszego rodzica oraz 75% *prawych* genów drugiego rodzica. Jeśli jednak wylosowana byłaby strona *lewa*, to dziecko otrzymałoby 75% *lewych* genów silniejszego osobnika oraz 25% *prawych* genów. Na koniec mają zaś miejsce mutacje: losowa liczba (wybranych również losowo) genów potomka zmienia swoje wartości na zupełnie nowe.

Symulacja

Symulacja każdego dnia składa się z poniższej sekwencji kroków:

- usunięcie martwych zwierząt z mapy,
- skręt i przemieszczenie każdego zwierzęcia,
- konsumpcja roślin na których pola weszły zwierzęta,
- rozmnażanie się najedzonych zwierząt znajdujących się na tym samym polu,
- wzrastanie nowych roślin na wybranych polach mapy.

Daną symulację opisuje szereg parametrów:

- wysokość i szerokość mapy,
- wariant mapy (wyjaśnione w sekcji poniżej),
- startowa liczba roślin,
- energia zapewniana przez zjedzenie jednej rośliny,
- liczba roślin wyrastająca każdego dnia,
- wariant wzrostu roślin (wyjaśnione w sekcji poniżej),
- startowa liczba zwierzaków,
- startowa energia zwierzaków,
- energia konieczna, by uznać zwierzaka za najedzonego (i gotowego do rozmnażania),
- energia rodziców zużywana by stworzyć potomka,
- minimalna i maksymalna liczba mutacji u potomków (może być równa 0),
- wariant mutacji (wyjaśnione w sekcji poniżej),

- długość genomu zwierzaków,
- wariant zachowania zwierzaków (wyjaśnione w sekcji poniżej).

Warianty

Pewne aspekty symulacji są konfigurowalne i mogą silnie zmieniać jej przebieg. Część to zwykłe parametry liczbowe (np. początkowe rozmiary populacji). Część z nich jednak dość znacząco modyfikuje jej zasady. Dotyczy to w szczególności: działania mapy, działania wzrostu roślin, działania mutacji, zachowania zwierzaków.

W przypadku mapy kluczowe jest to, jak obsługujemy jej krawędzie. Zrealizujemy następujące warianty:

- **kula ziemska** - lewa i prawa krawędź mapy zapętla się (jeżeli zwierzak wyjdzie za lewą krawędź, to pojawi się po prawej stronie - a jeżeli za prawą, to po lewej); górna i dolna krawędź mapy to bieguny - nie można tam wejść (jeżeli zwierzak próbuje wyjść poza te krawędzie mapy, to pozostaje na polu na którym był, a jego kierunek zmienia się na odwrotny);
- **piekielny portal** - jeżeli zwierzak wyjdzie poza krawędź mapy, to trafia do magicznego portalu; jego energia zmniejsza się o pewną wartość (taką samą jak w przypadku generacji potomka), a następnie jest teleportowany w nowe, losowe miejsce na mapie.

W przypadku wzrostu roślin pewne pola są silnie preferowane, zgodnie z zasadą Pareto. Istnieje 80% szansy, że nowa roślina wyrośnie na preferowanym polu, a tylko 20% szans, że wyrośnie na polu drugiej kategorii. Preferowanych jest około 20% wszystkich miejsc na mapie, 80% miejsc jest uznawane za nieatrakcyjne. Implementujemy następujące warianty::

- **zalesione równiki** - preferowany przez rośliny jest poziomy pas pól w centralnej części mapy (udający równik i okolice);
- **toksyczne trupy** - rośliny preferują te pola, na których zwierzęta umierają najrzadziej - rosną na tych polach, na których najmniej zwierząt skończyło swój żywot w trakcie symulacji.

W przypadku mutacji mamy do czynienia z dwoma prostymi opcjami:

- **pełna losowość** - mutacja zmienia gen na dowolny inny gen;
- **lekka korekta** - mutacja zmienia gen o 1 w górę lub w dół (np. gen 3 może zostać zamieniony na 2 lub 4, a gen 0 na 1 lub 7).

Podobnie proste są warianty zachowania:

- **pełna predestynacja** - zwierzak zawsze wykonuje kolejno geny, jeden po drugim;
- **niewielkie szaleństwo** - w 80% przypadków zwierzak po wykonaniu genu aktywuje gen następujący zaraz po nim, w 20% przypadków przeskakuje jednak do innego, losowego genu.

Wymagania dla aplikacji

1. Aplikacja ma być realizowana z użyciem graficznego interfejsu użytkownika z wykorzystaniem biblioteki *JavaFX*.
2. Jej głównym zadaniem jest umożliwienie uruchamiania symulacji o wybranych konfiguracjach.
 1. Powinna umożliwić wybranie jednej z uprzednio przygotowanych gotowych konfiguracji.

2. Dostępne konfiguracje mają formę znajdujących się w odpowiednim folderze na dysku plików konfiguracyjnych (plik może zawierać po prostu pary klucz + wartość, po jednej na linię).
3. Istnieje możliwość wczytania alternatywnej, sporządzonej przez użytkownika konfiguracji.
3. Uruchomienie symulacji powinno skutkować pojawieniem się nowego okna obsługującego daną symulację.
 1. Jednocześnie uruchomionych może być wiele symulacji, każda w swoim oknie, każda na osobnej mapie.
4. Sekcja symulacji ma wyświetlać animację pokazującą pozycje zwierząt, ich energię w dowolnej formie (np. koloru lub paska zdrowia) oraz pozycje roślin - i ich zmiany.
5. Program musi umożliwiać zatrzymywanie oraz wznowianie animacji w dowolnym momencie (niezależnie dla każdej mapy - patrz niżej).
6. Program ma pozwalać na śledzenie następujących statystyk dla aktualnej sytuacji w symulacji:
 - liczby wszystkich zwierząt,
 - liczby wszystkich roślin,
 - liczby wolnych pól,
 - najpopularniejszych genotypów,
 - średniego poziomu energii dla żyjących zwierząt,
 - średniej długości życia zwierząt dla martwych zwierząt (wartość uwzględnia wszystkie nieżyjące zwierzęta - od początku symulacji),
 - średniej liczby dzieci dla żyjących zwierząt (wartość uwzględnia wszystkie powstałe zwierzęta, a nie tylko zwierzęta powstałe w danej epoce).
7. Jeżeli zdecydowano się na to w momencie uruchamiania symulacji, to jej statystyki powinny być zapisywane (każdego dnia) do pliku CSV. Plik ten powinien być "otwieralny" przez dowolny rozumiejący ten format program (np. MS Excel).
8. Po zatrzymaniu programu można oznaczyć jedno zwierzę jako wybrane do śledzenia. Od tego momentu (do zatrzymania śledzenia) UI powinien przekazywać nam informacje o jego statusie i historii:
 - jaki ma genom,
 - która jego część jest aktywowana,
 - ile ma energii,
 - ile zjadł roślin,
 - ile posiada dzieci,
 - ile dni już żyje (jeżeli żyje),
 - którego dnia zmarło (jeżeli żywot już skończyło).
9. Po zatrzymaniu programu powinno być też możliwe pokazanie, które ze zwierząt mają dominujący (najpopularniejszy) genotyp (np. poprzez wyróżnienie ich wizualnie).
10. Aplikacja powinna być możliwa do zbudowania i uruchomienia z wykorzystaniem Gradle'a.
11. Wymogi traktujemy "zdroworozsądkowo", a nie "literalnie" - wczujmy się w badacza ewoluujących zwierzątek, który chciałby wygodnie sprawdzać i obserwować jak różne parametry świata wpływają na przyszłe losy ich populacji. To nasz hipotetyczny klient, który na koniec powinien być zadowolony. ;]

FAQ

(A właściwie odpowiedzi na nie)

- Nowo narodzony (lub wygenerowany) zwierzak jest ustawiony w losowym kierunku. Ma też aktywowany losowy gen (niekoniecznie pierwszy).
- Narodzone dziecko pojawia się na tym samym polu co jego rodzice.
- Nie wszystkie wartości parametrów muszą być dozwolone. Lepiej ograniczyć dopuszczalne zakresy (w szczególności do takich, które nie spowodują natychmiastowego zawieszenia aplikacji) - i poinformować użytkownika, że jego konfiguracja nie jest prawidłowa, jeżeli poza nie wykroczy.
- Energię traktujemy całkowitoliczbowo. Pilnujemy jednak, by jej jedynym źródłem były rośliny (po rozmnażaniu się suma energii organizmów na danym polu powinna być taka sama jak przed rozmnażaniem).
- Jeżeli na jednym polu kilka zwierzków rywalizuje o roślinę (albo o możliwość rozmnażania), to konflikt ten jest rozwiązywany w następujący sposób:
 - pierwszeństwo mają organizmy o największej energii,
 - jeżeli to nie pozwala rozstrzygnąć, to pierwszeństwo mają organizmy najstarsze,
 - jeżeli to nie pozwala rozstrzygnąć, to pierwszeństwo mają organizmy o największej liczbie dzieci,
 - jeżeli to nie pozwala rozstrzygnąć, to wśród remisujących organizmów wybieramy w dowolny sposób.
- Rośliny mogą rosnąć tam, gdzie stoją zwierzęta. Zjadanie ma miejsce w momencie wchodzenia na pole. Potem zwierze nie przeszkadza już w istnieniu rośliny.
- Nowe rośliny nie pojawiają się, jeżeli nie ma już dla nich miejsca na mapie.
- Statystyki nie muszą być prezentowane w formie wykresu (choć na pewno byłoby to ciekawe usprawnienie).
- Powyższa specyfikacja może różnić się trochę (lub bardzo) od analogicznych dokumentów znanych ubiegłym rocznikom. Zaleca się czujność i unikanie dróg na skróty. :-)

Przykładowe implementacje-inspiracje

Uwaga: przedstawione implementacje niekoniecznie spełniają tegoroczne wymagania.

- <https://www.youtube.com/watch?v=4FangGEpwe4>