

Civilization

dokumentacja projektu zaliczeniowego
z przedmiotu “Modelowanie
i Symulacja Systemów”

Nikodem Tokarski

Albert Podraza

Przemysław Haptaś

Informatyka, III rok,

WEAiIB AGH w Krakowie, 2015

Wstęp

Projekt "Civilization" powstał w celu odpowiedzi na pytanie, jak duży wpływ na rozwój cywilizacji mają czynniki związane z rolnictwem. Innymi słowy, chcieliśmy dowiedzieć się, jak wyglądałoby rozmieszczenie ludności, gdyby najważniejszym powodem jej przemieszczania się była jakość gleby na danym terenie.

W tym celu postanowiliśmy najpierw sprawdzić dostępną literaturę, aby dowiedzieć się, czy istnieją publikacje poruszające ten problem oraz poznać, jak go rozwiązywano, a następnie zasymulować proces migracji ludności korzystając z technik programowania agentowego. Na końcu porównaliśmy wyniki symulacji z rzeczywistymi mapami gęstości zaludnienia oraz wysnuliśmy wnioski.

Jako obszar symulacji wybraliśmy basen Morza Śródziemnego.

Naszą pracę przedstawiamy poniżej.

Omówienie publikacji

Starożytny Egipt – model agentowy [1]

Artykuł opisuje modelowanie agentowe społeczeństwa starożytnego Egiptu. Omawiany model oparty jest na abstrakcyjnym krajobrazie Egiptu zawierającym wioski, farmy i rzeki. Agenci reprezentują gospodarstwa rolnicze zdolne do wymiany informacji i migracji motywowanej możliwością zdobycia większej ilości pożywienia (która traktowana jest jako wskaźnik jakości życia). Model w podstawowej formie ilustruje kluczowe idee modelowania agentowego.

Założeniem projektu nie jest próba pełnego odwzorowania społeczeństwa, mimo tego, że dysponowano wystarczającą ilością danych, by to zrobić – dane wyjściowe nie dostarczyłyby więcej informacji niż dane wejściowe. Model oparty o najbardziej podstawowe założenia jest w stanie symulować zachowanie starożytnych społeczeństw.

Prezentowany w artykule model został zaprojektowany by zbadać związek między przekazywaniem informacji i skupiskami ludzkimi w społeczeństwie opartym na rolnictwie (takie społeczeństwa powszechnie występowały na świecie przed nastaniem ery dominacji przemysłu). Model nie przewiduje klęsk pogodowych, które mają wpływ na ilość żywności. Czas odwzorowany w modelu to lata 3800 – 3500 p.n.e; miejsce to brzeg Nilu. Mapa reprezentowana jest jako zbiór kwadratów o boku długości 5 km. Przyjęte zostało, że wraz z oddalaniem się od rzeki żyzność gleby stopniowo maleje

W symulacji dokonywane są szacunki zagęszczenia ludności na danym terenie i średniej ilości mieszkańców w osadach. Migracja osady była zależna od: żyzności gleby na docelowym terytorium, odległości od niego oraz ilości ludzi przy danym polu/kwadracie.

Otrzymane w wyniku symulacji pokazały, że przeciętna jakość życia wyznaczana na podstawie ilości pożywienia wzrosła średnio o ponad 50%. Początkowe proporcje (4 dla odległości od nowego terenu i 1 dla „różnorodności” sąsiadów) zostały odwrócone.

Należy zauważyć, że po kilkudziesięciu iteracjach, niezależnie od początkowych proporcji dochodzi do uzyskania optymalnego wskaźnika jakości życia i oscylacji wokół niego. Prawdopodobnie wynika to z jednorodnego systemu probabilistycznego zastosowanego modelu. Dochodzi również do rozczarowującego odkrycia, pokazującego, że jakość życia spada kiedy agenci (gospodarstwa) posiadają zbyt dużo wiedzy (przy nierealnym wyniku zdobycia globalnej wiedzy agenci rozpraszają się). Zjawisko to jest znane w klasycznej ekonomii jako „paradox of partial knowledge”.

Modelowanie agentowe a starożytna cywilizacja minojska [2]

Artykuł opisuje użycie modelowania agentowego do zbadania wykorzystania ziemi, osiedlania się i schematów społecznego organizowania się ludności pewnego fragmentu Krety w Epoce Brązu. Badanie symuluje okres ponad 2000 lat, zakłada rolnictwo jako najbardziej krytyczną aktywność populacji i ewaluuje wpływ różnych modeli organizacji społeczeństwa oraz strategii agralnych na żywotność populacji i przestrzenne rozmieszczenie osad. Wyniki symulacji wyraźnie pokazują, że populacje, w których występują warstwy społeczne są większe niż te, w których społeczeństwo jest egalitarne, co sugeruje prawdziwość teorii mówiących, że w społeczeństwie Krety wczesnej Epoki Brązu istniały warstwy społeczne, które stały się załącznikiem złożonych struktur społecznych widocznych w późniejszych okresach. Co więcej, uzyskane rozproszenie populacji również zgadza się z dostępną wiedzą archeologiczną.

Użyty system ABM (ang. *agent-based model*) został stworzony na potrzeby symulacji, jednak twórcy twierdzą, że jest on ogólny, a jego celem nie jest udowodnienie, czy zaprzeczenie konkretnej teorii. Od innych systemów różni go to, że agenci są od siebie zupełnie niezależni oraz potrafią tworzyć złożone struktury społeczne.

Agenci oraz zasoby są umieszczone w dwuwymiarowej przestrzeni przedstawiającej teren 20 x 25 km podzielonej na kwadratowe komórki o boku długości 100m. Czas symulacji to ok. 2000 lat z krokiem równym 1 rok. Agenci symbolizują domostwa jako główne jednostki społeczne wytwarzające pewną wartość produkcyjną. Wiele agentów może dzielić daną komórkę, zaś przyległe, "zamieszkane" komórki tworzą osadę. Zasoby znajdują się w ustalonych miejscach, a ich dostępność może zmieniać się w czasie. Produktywność komórki jest zaś funkcją jej geo-morfologicznych parametrów (np. nachylenia powierzchni), żyzności gleby oraz czasu zajmowania jej przez populację - z czasem żyzność gleby zmniejsza się z powodu erozji, a wzrasta doświadczenie uprawiających glebę. Na wielkość populacji wpływa migracja oraz procesy naturalne - narodziny i śmierć agentów.

W każdym kroku agenci:

- zbierają plony w komórkach, które do nich należą
- sprawdzają, czy suma plonów i zmagazynowanych we wcześniejszych krokach zasobów wystarcza na zaspokojenie ich potrzeb
- jeśli w.w. suma nie wystarcza, w zależności od przyjętego modelu organizacji społeczeństwa, mogą prosić o pomoc innych agentów
- jeśli wciąż brakuje im zasobów, osiągają zmniejszoną zdolność reprodukcyjną oraz rozważają migrowanie do innych komórek.

Istnieją trzy podstawowe modele określające dystrybucję zasobów wewnątrz społeczeństwa:

1. niezależne (ang. *independent*) - agenci nie dzielą się zasobami
2. dzielące się (ang. *sharing*) - agenci posiadają wspólne zasoby
3. samo-organizujące się (ang. *self-sharing*) - agenci przyjmują model niezależny lub dzielący się zależnie od potrzeby.

Na podstawie wiedzy historycznej i archeologicznej, autorzy zdecydowali się stworzyć algorytm implementujący model samo-organizującego się społeczeństwa na podstawie

publikacji R. Kota, N. Gibbins, N. R. Jennings - *Self-Organising Agent Organisations*. Zakłada on istnienie relacji między agentami, gdzie agent może być:

1. znajomym (ang. *acquaintance*) - brak interakcji z innym agentem
2. rówieśnikiem (ang. *peer*) - jego zasoby są porównywalne z zasobami innego agenta
3. zwierzchnikiem/ podwładnym (ang. *authority*) - występuje znaczna różnica w wielkości zasobów między agentami.

Model zakłada również stosowanie zarówno rolnictwa ekstensywnego (ang. *extensive farming*), jak i rolnictwa intensywnego (ang. *intensive farming*).

Wyniki symulacji pokazały, niezależnie od przyjętego rodzaju rolnictwa (intensywne/ ekstensywne), największy wzrost populacji samo-organizujących się, a następnie populacji niezależnych, będących z kolei w przewadze nad populacjami egalitarnymi, w których dzielono wszystkie zasoby. Niezależnie od przyjętych strategii, w odpowiedniej liczbie kroków, symulowane rozmieszczenie osad wiernie odwzorowywało historyczne dane.

Socjalno-ekologiczny system Majów [3]

Artykuł traktuje o symulacji MayaSim. Jest to połączenie modeli:

- ABM(ang. *agent-based model*),
- automatów komórkowych
- modeli sieciowych.

Całość przedstawia socjalno-ekologiczny system starożytnych Majów. Agenci, komórki i sieci reprezentują jego poszczególne elementy. Przykładowo, agentami są tutaj grupy ludzi, komórkami terytoria posiadające różne właściwości, a modelem sieciowym jest tu handel. Ponadto uwzględnione są tu inne zmienne takie jak: demografia, agrokultura, degradacja gleb, wahania klimatyczne. Wpływając na te atrybuty możemy w pewnym przybliżeniu obserwować rozwój cywilizacji, tutaj - Majów. Dla zebrania dokładniejszych rezultatów potrzebowalibyśmy jednak więcej danych archeologicznych. Model zakłada obszar o wielkości 516,484 km² podzielony na tereny o powierzchni 20km². Symulacja obejmuje 650 iteracji, gdzie jeden skok to 2 lata, a więc cały okres trwania cywilizacji majów (tj. okres preklasyczny, klasyczny, schyłkowy). Dzięki setkom uruchomień, zebrano wiele informacji, pozwalających sprawdzić szereg teorii dot. dynamiki systemu.

Jednym z ciekawszych wniosków było to, że Majowie osiągnęli największy rozwój, kiedy degradacja ziemi i handel ograniczone były pewnymi wartościami; kiedy ziemia była zbyt zdegradowana, handel zanikał, co miało swoje odwzorowanie historyczne. Z drugiej strony, nie uzyskano niestety efektu kryzysu, który dotknął Majów pod koniec okresu klasycznego, czego powodem może być nieuwzględnienie kilku czynników takich jak konkurencja, klęski żywiołowe, etc. Celem stworzenia modelu była chęć lepszego zrozumienia złożoności dynamiki systemu socjalno-ekologicznego oraz przetestowanie czynników na niego wpływających, co w pewnym stopniu osiągnięto.

War, Space, and the evolution of Old World complex societies [4]

Artykuł przedstawia zaawansowany model agentowy mający na celu symulację ewoluowania społeczeństw uwzględniając różnorodne czynniki społeczne takie jak wojny, czy podboje. Modelowanym terenem jest Afroeurazja, czyli połączenie dwóch kontynentów - Eurazji rozróżnianej na Europę i Azję oraz Afryki. Krajobraz reprezentowany jest poprzez siatkę kwadratów o wymiarach 100 x 100 km. Każda tak powstała komórka posiada swój zestaw cech: rolnictwo, rodzaj powierzchni (np. pustynia), przewyższenia. Na początku symulacji każda komórka jest zasiedlona przez niezależne społeczności. Komórki przyległe do stepów posiadają "ziarno" cechy militarnej, które stopniowo rozprzestrzenia się na sąsiadujące tereny.

Kluczowa dla projektu jest dynamika cech ultraspołecznych (wg. autorów to słowo podkreśla wybitne przystosowanie człowieka do zachowań społecznych). Każda komórka jest zamieszkała przez społeczność posiadającą własny "cultural genome" - wektor o wartości binarnej, przybierający 1 gdy społeczność posiada cechy ultraspołeczne lub 0 w przeciwnym wypadku. Wymieniona cecha jest nabywalna jeśli komórkę zamieszkuje społeczność, jednak w symulacji owa cecha jest częściej tracona niż nabywana. Czynnikiem, dzięki któremu cecha jest nabywana jest wojna. Komórki są w stanie podbijać inne komórki i w ten sposób budować wielokomórkowe organizmy - społeczeństwa. Prawdopodobieństwo wygranej konfliktu zależy od relatywnej siły atakującego i od umiejętności obrony. Siła ustalana jest na podstawie rozmiaru społeczeństwa (ilości komórek) i średniej ilości cech społecznych. Komórka przegranej społeczności jest zajmowana i może przyjąć genom kulturowy zwycięzcy. Ta zmiana odzwierciedla fakt narzucania przegranemu społeczeństwu kultury swoich najeźdźców (np. religia, język, zamiana/eksterminacja piśmiennych elit społecznych). Jednak jak zwracają uwagę autorzy, nie jest to istotny czynnik w przenoszeniu cech kulturowych. Prawdopodobieństwo takiej wymiany wzrasta wraz ze zwiększeniem cech militarnych; bardziej efektywna forma ofensywy wojennej zwiększa szansę na wygraną. Symulacja uwzględnia również wpływ terenu na obronność komórki (np. góry są trudniejsze do atakowania i łatwiejsze w obronie).

Testowany czas to lata 1500 p.n.e - 1500 n.e. Symulacja przeprowadzona na tym okresie dała w wyniku dane zauważalnie zbliżone do realnych. Pierwsze imperia pojawiły się w Mezopotamii, Egipcie i na północy Chin ze względu na sprzyjające warunki oraz fakt rozprzestrzeniania siły militarnej. Co ciekawe, usunięcie z branych pod uwagę czynników przewyższenia skutkowało dużym zmniejszeniem dokładności modelu i powodowało prowadzenie działań wojennych równomiernie na całym terenie.

Wartym odnotowania jest również fakt, że autorzy używają dalece zaawansowanego określania wartości siły militarnej. Szansę na wygraną zmienia nawet fakt pojawienia się kawalerii. W publikacji zamieszczono obszerny opis matematyczny dotyczący wojny i zjawisk społecznych. Podstawowy wzór, na podstawie którego określano szansę sukcesu w działaniach wojennych biorąc pod uwagę współczynnik obrony i ataku:

$$P_{succ} = \frac{P_{att} - P_{deff}}{P_{att} + P_{deff}}$$

gdzie:

- P_{att} - współczynnik ataku
- P_{succ} - szansa sukcesu
- P_{deff} - współczynnik obrony

Podsumowanie publikacji

Porównując powyższe artykuły, nasuwają się między nimi następujące podobieństwa:

1. W każdym artykule użyte jest modelowanie ABM
2. W artykułach [1] i [2] agentami są stosunkowo małe grupy społeczne (gospodarstwa domowe)
3. W każdym artykule brana jest pod uwagę atrakcyjność terenu oraz populacja
4. Wszędzie symulacje zaczynają się stosunkowo wcześnie, kiedy cywilizacje zaczynały się tworzyć zaraz po okresie koczowniczym, najwcześniej w artykułach: [1] i [2], gdzie te okresy obejmowały kilka tysięcy lat p.n.e .
5. Ogólna idea przebiegu symulacji w każdym artykule była podobna, ale w każdym z nich autorzy koncentrowali się na innych skutkach, i tak:
 - a. w [1] na pozyskaniu informacji na temat wzrostu jakości życia i tego skutkach,
 - b. w [2] na określeniu, który rodzaj społecznej współpracy daje największy wzrost populacji,
 - c. w [3] na zasymulowaniu tworzenia się systemów społeczno-ekologicznych i ich wpływu na migrację,
 - d. w [4] na skutkach wojen i zebraniu danych dot. rozprzestrzeniania się kultur.
6. W każdym z artykułów są używane automaty komórkowe,
7. Długość kroku to odpowiednio:
 - a. w [1] nie został podany, ale można domniemać, że to przedział 5-10 lat,
 - b. w [2] jeden rok,
 - c. w [3] dwa lata,
 - d. w [4] jeden rok.

Konfrontując ze sobą powyższe artykuły można również zauważyć kilka różnic, najczęściej przy doborze zmiennych kalibrujących symulacje oraz zauważyć pewne uniwersalne zasady dotyczące symulacji rozwoju cywilizacji, jak np. w artykule [3], w którym położono nacisk na zależności handlowo-ekologiczne, podczas gdy w innych artykułach zostało to zwyczajnie pominięte. Z drugiej strony mamy wiele podobieństw takich jak: korzystanie z modelowania ABM, aparatów komórkowych, uwzględnienie atrakcyjności terenu jako jedna z głównych zmiennych, podobne długości skoków, etc., które są swego rodzaju drogowskazami dla naszej symulacji.

Krótki opis pojęć ABM i automatów komórkowych

ABM (ang. *agent-based model*) - to jedno z podejść stosowanych przy modelowaniu systemów. To szereg elementów zaczerpniętych z teorii gier, socjologii obliczeniowej, algorytmów ewolucyjnych pozwalających odwzorować losowość. Agentami mogą być podprogramy lub algorytmy, mające wpływ na cały system oraz potrafiące przewidywać wpływ wyarty przez ich decyzje na system. Co do reprezentacji wewnętrznej, mogą to być jednostki decyzyjne, społeczności, etc. W naszym modelu będą to grupy osób decydujące o dalszym rozwoju. Cały proces polega na interakcji między poszczególnymi agentami, którzy są od siebie niezależni, w celu odtworzenia i przewidzenia złożonych zjawisk. Ponadto, proces przewiduje skalowalność, maksymalne uproszczenia (zastosowanie zasady K.I.S.S.) przy jednoczesnym zachowaniu pewnej złożoności. Takie podejście pozwala na uwzględnienie wielu zmiennych zaczerpniętych z nauk społecznych, takich jak status społeczny, ekonomiczne wskaźniki, co daje nam narzędzia i możliwość do bardzo dokładnej analizy systemów socjologicznych.

Automaty komórkowe (ang. *cellular automaton*) – system składający się z pojedynczych komórek rozmieszczonych obok siebie. Ich układ może przypominać planszę do gry. Komórki muszą przyjąć jeden ze ściśle określonych stanów (dowolnie duża liczba skończona). Stan komórki zmienia się synchronicznie e wedle reguł określających sposób

w jaki nowy stan komórki zależy od jej obecnego stanu i stanów wszystkich jej sąsiadów. Automat składa się z n -wymiarowej regularnej siatki komórek, z których każda jest taka sama. Cała przestrzeń powinna być zajmowana w całości przez komórki ułożone obok siebie. Ewolucja komórek przebiega według tych samych, określonych reguł. Za twórcę automatów komórkowych uważa się Janosa von Neumanna.

Definicja matematyczna [5]:

Automat komórkowy jest to:

- sieć komórek $\{i\}$ przestrzeni D -wymiarowej,
- zbiór $\{s_i\}$ stanów pojedynczej komórki, zwykle ten sam dla wszystkich komórek, zawierający k elementów,
- reguła F określająca stan komórki w chwili $t+1$ w zależności od stanu w chwili t tej komórki i komórek ją otaczających; $s_i(t+1) = F(\{s_j(t)\})$, gdzie: j należy do $O(i)$, gdzie: $O(i)$ jest otoczeniem i -tej komórki

Z automatów można korzystać na rozmaite sposoby:

- jako narzędzie do symulacji fizycznych, w których oddziałuje ze sobą wiele obiektów
- jako obiekty matematyczne w teorii procesów dynamicznych
- do tworzenia fraktali

Najsłynniejszym automatem komórkowym jest bez wątpienia gra „Life” autorstwa Johna H. Conwaya. W grze można dostrzegać model żyjącego organizmu/środowiska. Jest ona automatem uniwersalnym, czyli zdolnym do wszystkich operacji logicznych.

Źródła:

- [1] D. R. Sarah Symons - *Agent-based models of ancient egypt*
- [2] A. Chliaoutakis, G. Chalkiadakis - *Utilizing Agent-Based Modeling to Gain New Insights into the Ancient Minoan Civilization*
- [3] Scott Heckbert - *MayaSim: An Agent-Based Model of the Ancient Maya Social-Ecological System*
- [4] E. A. L. T. S. G. Peter Turchin, Thomas E. Currie - *War, space and the evolution of old*
- [5] Krzysztof Kułakowski - *Automaty Komórkowe*
- [6] <http://webarchive.iiasa.ac.at/Research/LUC/External-World-soil-database/>

Proponowany model

Proponowany przez nas model składa się z trzech głównych elementów: mapy automatów komórkowych, zamieszkujących je agentów oraz algorytmu przedstawiającego zależność między automatami a agentami.

a) automat komórkowy *Cell* - jest jednostką modelującą fragment symulowanego terytorium.

Symulowany rejon to obszar między południkami: -10° i 40° oraz między równoleżnikami: 50° i 25° , co daje 180 900 komórek o średnich rozmiarach 9,2 km x 7 km i powierzchni 65 km².

Głównym atrybutem komórki jest żyzność gleby *fertility*, która determinuje liczbę agentów, którzy mogą go zamieszkać.

Fertility określone jest na podstawie danych o parametrach gleby pochodzących z projektu Harmonized World Soil Database

(<http://webarchive.iiasa.ac.at/Research/LUC/External-World-soil-database/>).

Dane te składały się z jednego pliku dla każdego z następujących parametrów:

1. Nutrient availability
2. Nutrient retention capacity
3. Rooting conditions
4. Oxygen availability to roots
5. Excess salts
6. Toxicity
7. Workability (constraining field management).

Każdy plik składał się z:

1. Nagłówek
 - o liczba kolumn macierzy
 - o liczba wierszy macierzy
 - o długość geograficzna lewego dolnego rogu macierzy
 - o szerokość geograficzna lewego dolnego rogu macierzy
 - o wielkość komórki (w stopniach)

czyli np. wiersz 1080 będzie przedstawiał wartości parametru żyzności dla równika
2. Macierzy wartości parametru żyzności dla danej komórki (0-7), gdzie:
 - o 0 - brak danych (w praktyce morza i oceany)
 - o 1 - brak lub lekkie ograniczenia
 - o 2 - średnie ograniczenia
 - o 3 - duże ograniczenia
 - o 4 - bardzo duże ograniczenia
 - o 5 - głównie nie-gleba
 - o 6 - wieczna zmarzlizna
 - o 7 - akweny wodne (rzeki, jeziora)

[illegible]

Na wstępie przekształciliśmy te wartości tak, aby większa oznaczała lepszą glebę:

```
double val = map[i][j];
if (val == 0 || val == 7) // seas or waterbodies
    val = 0;
else if (val == 6 || val == 5) // permafrost or non-soil
    val = 1;
else if (val == 4) // very severe constraints
    val = 2;
else if (val == 3) // severe constraints
    val = 4;
else if (val == 2) // moderate constraints
    val = 6;
else // non or slight constraints
    val = 7;
map[i][j] = val;
```

I przypisaliśmy *fertility* średnią arytmetyczną z wartości komórki macierzy z każdego z 7 plików.

Oprócz tego, *Cell* zawiera:

- *agentsSizeLimit* - oznacza maksymalną liczbę agentów zamieszkujących daną komórkę, wprost proporcjonalny do *fertility*
- indeksy położenia komórki na mapie *row*, *col*
- zamieszkujących ją agentów *agents*.
- *color* - obrazujący poziom zaludnienia oraz dominującą cywilizację poprzez odpowiednio: jasność (im ciemniejszy, tym większe zaludnienie) oraz barwę.
- *citizens* - informacja o liczbie agentów poszczególnych cywilizacji.

b) agent *Agent* - jest interpretowany, przy standardowych parametrach symulacji (poniżej), jako 100 rodzin czteroosobowych.

Zawiera atrybuty:

1. Liczbę tur, które przeżył *lifeTime*
2. Kolor cywilizacji, do której przynależy.

c) algorytm *Algorithm*

Algorytm korzysta z niektórych z opisanych niżej parametrów symulacji.

Na początku każdej tury w każdej komórce:

a. *deathsAndBirths()*

i. dla każdego agenta:

1. Jeśli *lifeTime* == *DEATH_TIME*, zabijamy agenta
2. Inkrementujemy licznik *lifeTime*
3. Jeśli w komórce jest wolne miejsce, próbujemy powiększyć rodzinę, czyli utworzyć nowego agenta tej samej cywilizacji
4. Wielkość szansy (w procentach) to:

$$success = 100 - lifeTime * 20$$

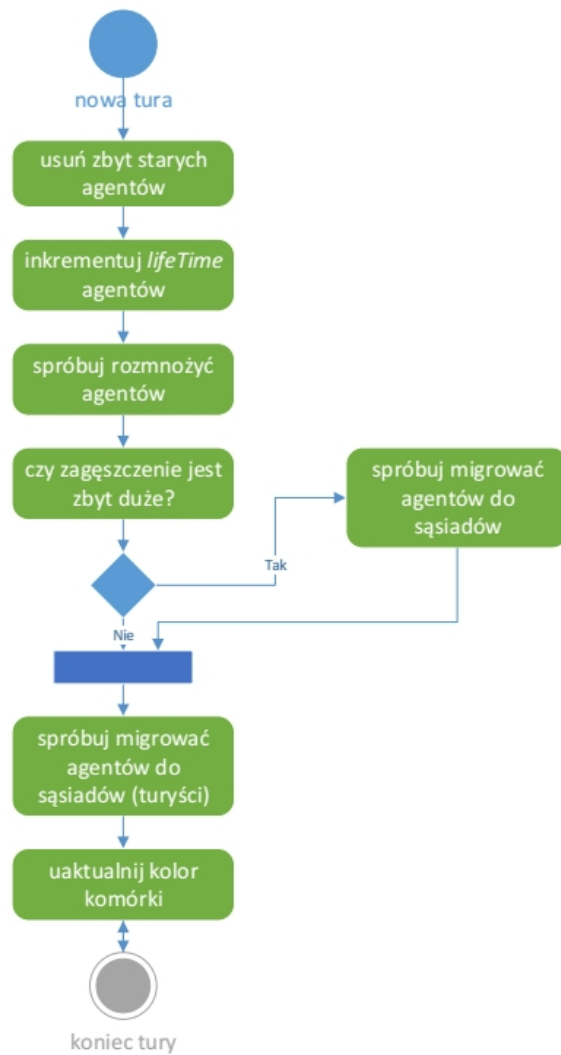
Im starszy agent, tym mniejsze prawdopodobieństwo na stworzenie nowego agenta

b. *migrations()*

- i. jeśli wolne miejsce w komórce jest mniejsze niż *MIGRATION_CAUSE*, to losujemy *MIGRATION_PERCENT* agentów, którzy chcą migrować
- ii. dla każdego wylosowanego agenta:
 1. losujemy sąsiednią komórkę i sprawdzamy, czy jest w niej wolne miejsce
 2. jeśli jest, przenosimy do niej agenta
 3. jeśli nie, agent zostaje w swojej komórce
- iii. losujemy *TRAVEL_PERCENT* agentów, którzy chcą migrować
- iv. powtarzamy dla nich punkt ii.

c. *elections()*

- i. wybieramy kolor cywilizacji, której agentów jest najwięcej
- ii. zwiększamy lub zmniejszamy jasność tego koloru w zależności od liczby wszystkich zamieszkujących ją agentów
- iii. jasność koloru przedstawia zaludnienie, a jego barwa cywilizację rządzącą na terenie komórki



Symulacja zjawiska

Parametry symulacji

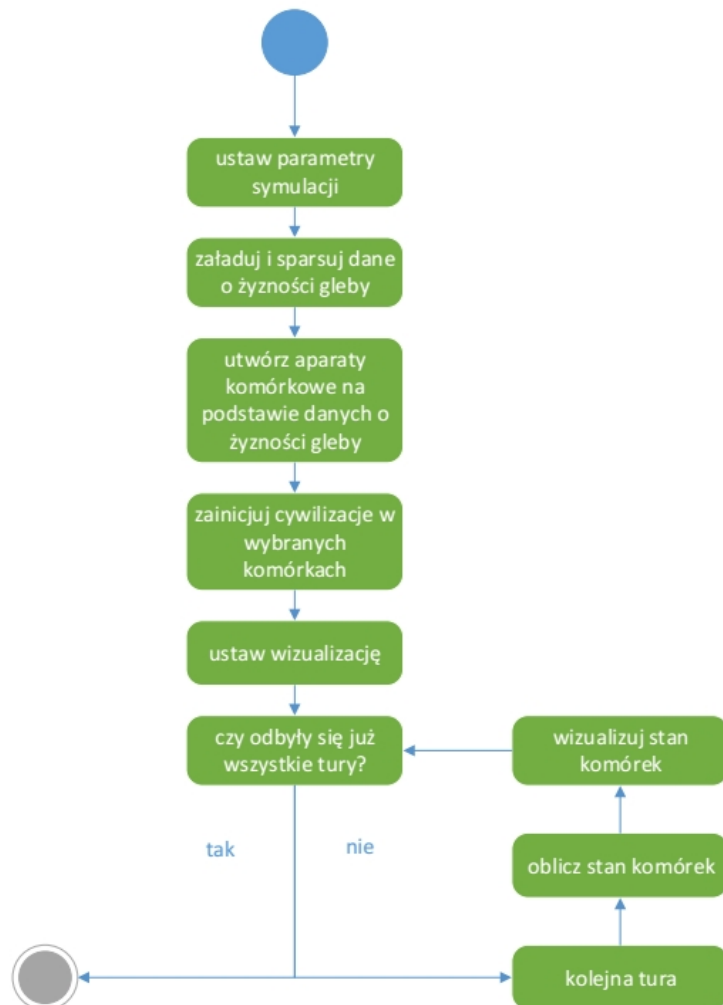
lp.	klasa	nazwa	wartość domyślna	opis
1.	main:Global	TURN_TIME	10	liczby lat, które symbolizuje jedna tura symulacji
2.	main:Global	URNS	500	liczba tur symulacji
3.	main:Global	CIVILIZATIONS_NR	10	liczba symulowanych cywilizacji
4.	main:Global	MAX_INIT_CIVIL_SIZE	10	maksymalna początkowa liczba agentów w każdej cywilizacji
5.	main:Global	MAX_AGENTS_CELL_LIMIT	70	maksymalna liczba agentów, jaką może pomieścić komórka o maksymalnym <i>fertility</i>
6.	abm:Cell	POPULATION_SIZE_RANGES	{ 0, 0.50, 0.80, 0.90, 0.95 }	określa, ilu musi być agentów, by zaznaczyć kolor komórki jako ciemniejszy
7.	abm:Algorithm	DEATH_TIME	5	liczba tur, po których zabijany jest agent
8.	abm:Algorithm	MIGRATION_CAUSE	5	wielkość wolnego miejsca w komórce, przy której agenci decydują się migrować do mniej zaludnionych, sąsiednich komórek
9.	abm:Algorithm	MIGRATION_PERCENT	20	procent agentów, którzy próbują migrować jeśli wolne miejsce w komórce jest mniejsze niż MIGRATION_CAUSE
10.	abm:Algorithm	TRAVEL_PERCENT	10	procent agentów, którzy niezależnie od wolnego miejsca decydują się migrować

Ustalając wartość domyślną *MAX_AGENTS_CELL_LIMIT* kierowaliśmy się następującym rozumowaniem:

- średnie rozmiary komórki: 9,2 km x 7 km, przyjmijmy powierzchnię komórki na 64 km²
- miasto świata o największej gęstości zaludnienia: Manila, ok. 43 000 ludzi/km²
- z powodu nieporównanie gorszej technologii przyjmijmy maksymalną gęstość ludności na 430 osób/km²
- przyjmijmy maksymalną liczbę ludności w komórce na 28 000 osób
- przyjmijmy, że agenci będą związani z 4-osobowymi rodzinami
- przyjmijmy agenta jako 100 4-osobowych rodzin
- przyjmijmy maksymalną liczbę agentów w komórce na 70

Przebieg symulacji

Symulacja składa się z następujących etapów:



Użyte technologie

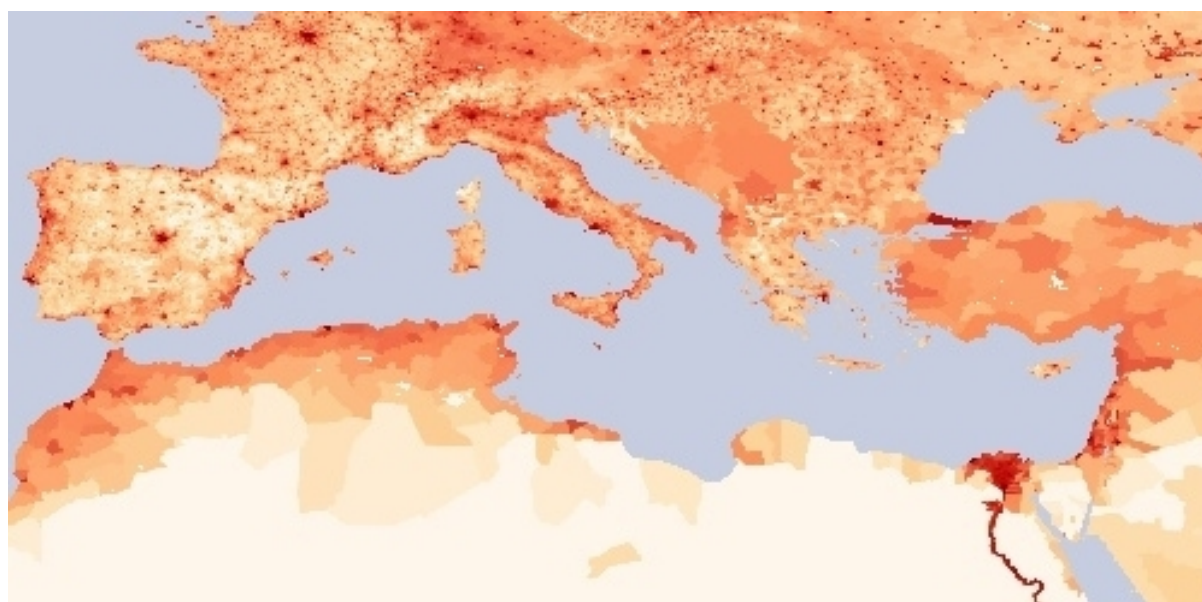
Projekt napisany jest w całości w języku Java.

Oprócz standardowych bibliotek zastosowaliśmy:

- AWT
- Swingbibliotekę do parsowania plików ESRI ASCII Raster napisaną i udostępnioną przez Dave Murray-Rust (<https://code.google.com/archive/p/java-esri-ascii/>), którą dostosowaliśmy do naszych potrzeb

Wyniki symulacji

Parametry domyślne

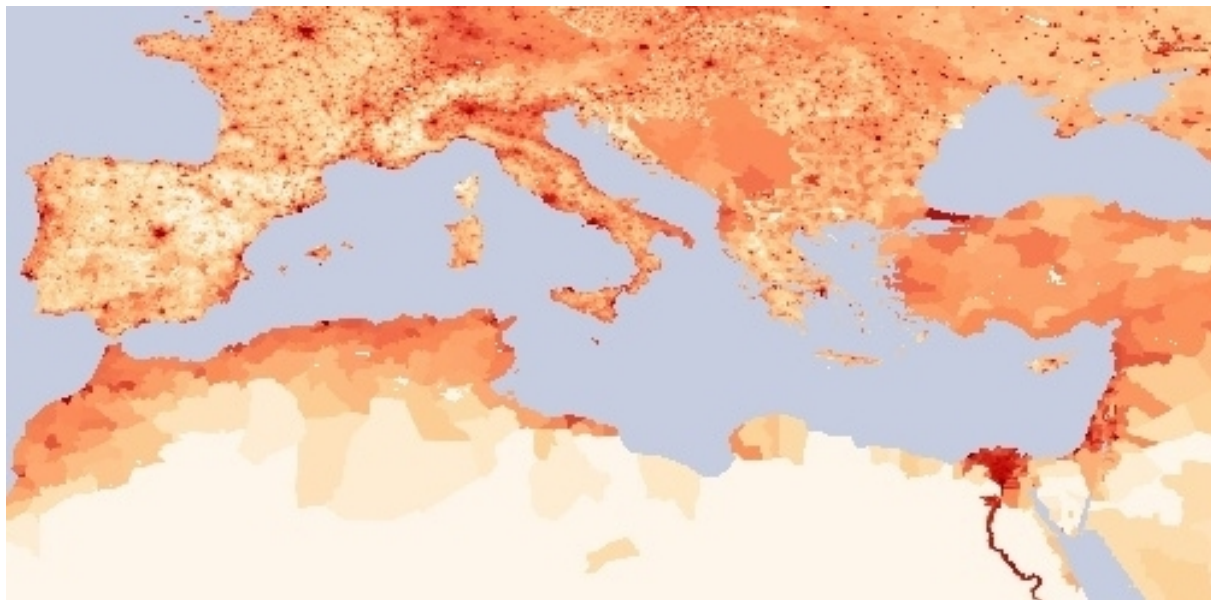


Symulowani agenci osiedlili się na północ od M. Czarnego, gdzie znaleźć można bardzo żyzne czarnoziemy, w rzeczywistości jednak teren ten nie jest tak gęsto zamieszkany. Ciekawe jest również to, że za dobre miejsce do zamieszkania uznali zachodnią część Sahary, teren bezludny. Innym przykładem niezgodności wyników symulacji z rzeczywistością jest delta Nilu - teren od stuleci gęsto zaludniony, symulowani agenci nie osiedlili się tam jednak w stopniu większym niż na innych terenach.

Zmienione parametry

`CIVILIZATIONS_NR = 5`

`MAX_INIT_CIVIL_SIZE = 1`



Zmniejszenie liczby cywilizacji o połowę oraz zmniejszenie liczby początkowych agentów zakładających te cywilizacje dziesięciokrotnie, czyli zmniejszenie początkowej liczby ludności 20-krotnie, poskutkowało znacznie mniejszym pokryciem terenu.

Teren, do którego jednak agenci dotarli, został przez nich tak samo gęsto pokryty, jak ten w symulacji z domyślnymi parametrami.

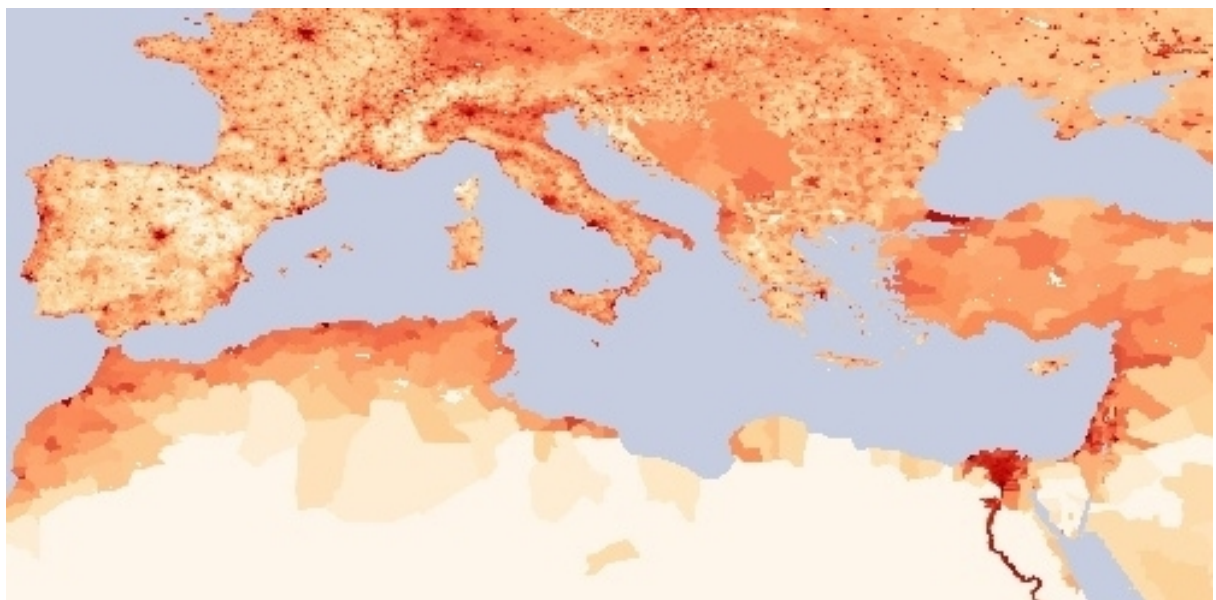
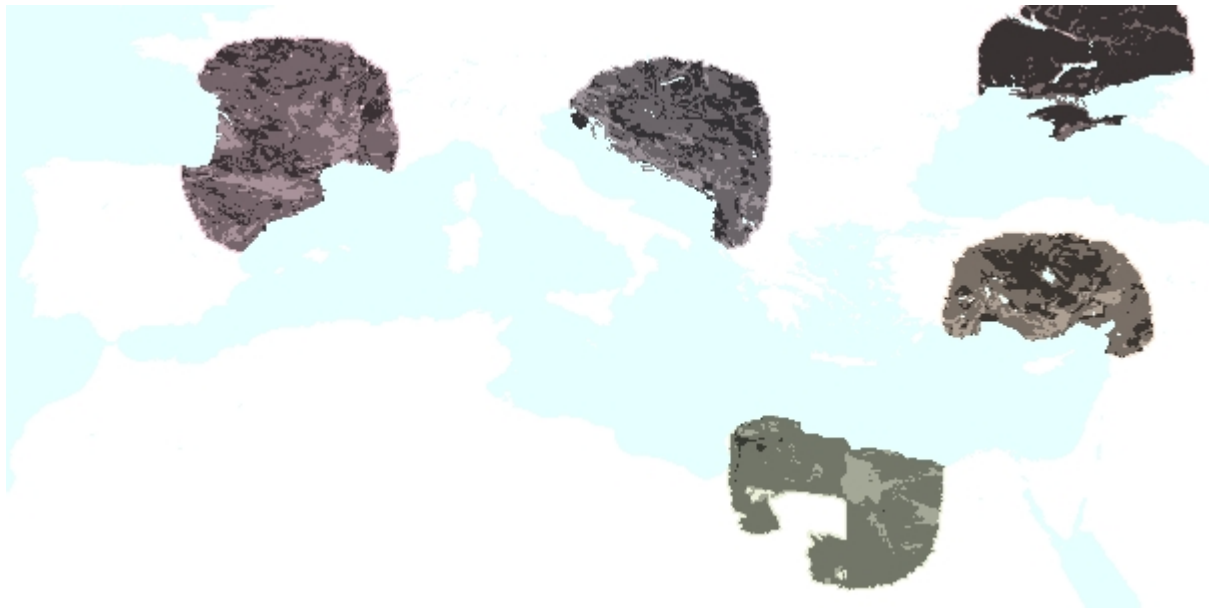
Zmienione parametry

CIVILIZATIONS_NR = 5

MAX_INIT_CIVIL_SIZE = 1

MIGRATION_PERCENT = 4

TRAVEL_PERCENT = 2



W tej symulacji zniechęciliśmy agentów do przemieszczania się zmniejszając 5-krotnie parametry *MIGRATION_PERCENT* oraz *TRAVEL_PERCENT*. W rezultacie pokryty przez nich teren jest znacznie mniejszy i mniej zaludniony niż w poprzedniej symulacji (patrz: wschodnia Libia i zachodni Egipt).

Wnioski

Do najważniejszych wniosków, jakie możemy wyciągnąć z przeprowadzonej przez nas symulacji, możemy zaliczyć:

1. Czynniki, które wzięliśmy pod uwagę sprowadzały się do określenia jakości gleby. Okazuje się, że jakość gleby jest ważnym czynnikiem, który ma istotny wpływ na wybór miejsca do osiedlenia się, jednak nie wzięcie pod uwagę innych istotnych czynników takich jak ukształtowanie terenu, dostęp do morza, technologia, konflikty między cywilizacjami etc., powoduje, że wynik symulacji w wielu miejscach nie zgadza się z dostępnymi danymi o gęstości zaludnienia.
2. Brak zaimplementowanych mechanizmów interakcji między cywilizacjami powoduje, że po pewnej stałej liczbie tur, symulacja stabilizuje się. Następuje to w momencie zajęcia każdej wolnej, nadającej się do osiedlenia, komórki. Początkowo na granicach cywilizacji można zaobserwować przejścia terenu od obcych cywilizacji, ale po kilkunastu turach to zjawisko również przestaje występować. Agenci nie migrują, ponieważ nie mają miejsca, a w zajętych komórkach przyrost naturalny równoważy śmiertelność agentów.

Projekt stara się podejść do problemu w sposób ogólny, nie skupialiśmy się na poszczególnych społecznościach rozwijających się w niepowtarzalnych środowiskach tak jak w artykułach [1-5] .

Podczas tworzenia modelu przeprowadziliśmy wiele dyskusji dotyczących interpretacji agenta - czy jeden agent ma reprezentować jednego człowieka, rodzinę, a może 100 rodzin? Program wykonuje wiele obliczeń i okazało się, że interpretowanie agenta jako jednej 4-osobowej rodziny uniemożliwi obserwowanie symulacji. Dlatego zdecydowaliśmy się interpretować agenta jako 100 takich rodzin.

Jednym z największych wyzwań było zdobycie, przeanalizowanie i przetworzenie danych o żyzności gleby. Innym problemem było zaprojektowanie algorytmu, który byłby prosty do zrozumienia, odwzorowujący pewne rzeczywiste zachowania ludzi, a jednocześnie niezłożony obliczeniowo, a więc możliwy to zastosowania na dużej liczbie agentów.

Najważniejszym kierunkiem dalszego rozwoju projektu wydaje nam się poprawienie zastosowanych algorytmów poprzez dodanie większej ilości parametrów kształtujących symulację np. handel, czy symulacja konfliktów (w przypadku kiedy agenci nie mieliby już wolnych komórek do zdobycia, wojna o zajęte, atrakcyjne ziemie mogła by dać bardzo ciekawe rezultaty).

Drugim ciekawym pomysłem, który notabene realizujemy obecnie w ramach przedmiotu Programowanie Współbieżne i Rozproszone, byłaby próba zrównoleglenia części obliczeń. Do tego celu używamy języków Erlang oraz Go.

Uruchomienie symulacji

1. Przejdź do katalogu *Civilization*
2. Ustaw parametry symulacji edytując wartości zawarte w pliku *app.conf*.
Domyślne parametry symulacji zawarte są w pliku *default.conf*
3. Uruchom symulację za pomocą komendy *java -Xms512m -Xmx1024m -jar civilization.jar*

Przykład dla systemu Ubuntu 14.04

```
przemek@przemek-PC:~$ cd Civilization/
przemek@przemek-PC:~/Civilization$ ls -l
total 528
-rw-rw-r-- 1 przemek przemek 198 sty 18 16:33 app.conf
-rw-rw-r-- 1 przemek przemek 515437 sty 18 16:12 civilization.jar
-r--r--r-- 1 przemek przemek 198 sty 18 16:33 default.conf
drwxrwxr-x 2 przemek przemek 4096 sty 4 11:26 lib
-rw-rw-r-- 1 przemek przemek 32 sty 4 11:26 README.md
drwxrwxr-x 4 przemek przemek 4096 sty 4 21:47 resources
drwxrwxr-x 7 przemek przemek 4096 sty 11 22:45 src
przemek@przemek-PC:~/Civilization$ |
```

```
app.conf x
TURNS=200;
TURN_TIME=10;
CIVILIZATIONS_NR=5;
MAX_INIT_CIVIL_SIZE=10;
MAX_AGENTS_CELL_LIMIT=70;
MAX_FERTILITY=7;
DEATH_TIME=50;
MIGRATION_CAUSE=5;
MIGRATION_PERCENT=20;
TRAVEL_PERCENT=10;
```

```
przemek@przemek-PC:~/Civilization$ java -Xms512m -Xmx1024m -jar civilization.jar
Program started!
Setting up simulation parameters...

Map loading...
Map loaded!

Setting up the visualization...
Visualization set!

Setting civilizations' positions...
Societies set!

Starting simulation...
```

Załączniki

1. Kod źródłowy projektu
2. Aplikacja (runnable jar)
3. Gif przedstawiający przykładową symulację