

Raport z Projektu Indywidualnego

„Akwarium”

Klaudia Goczał, 195172

Informatyka, sem VI, WEEIA

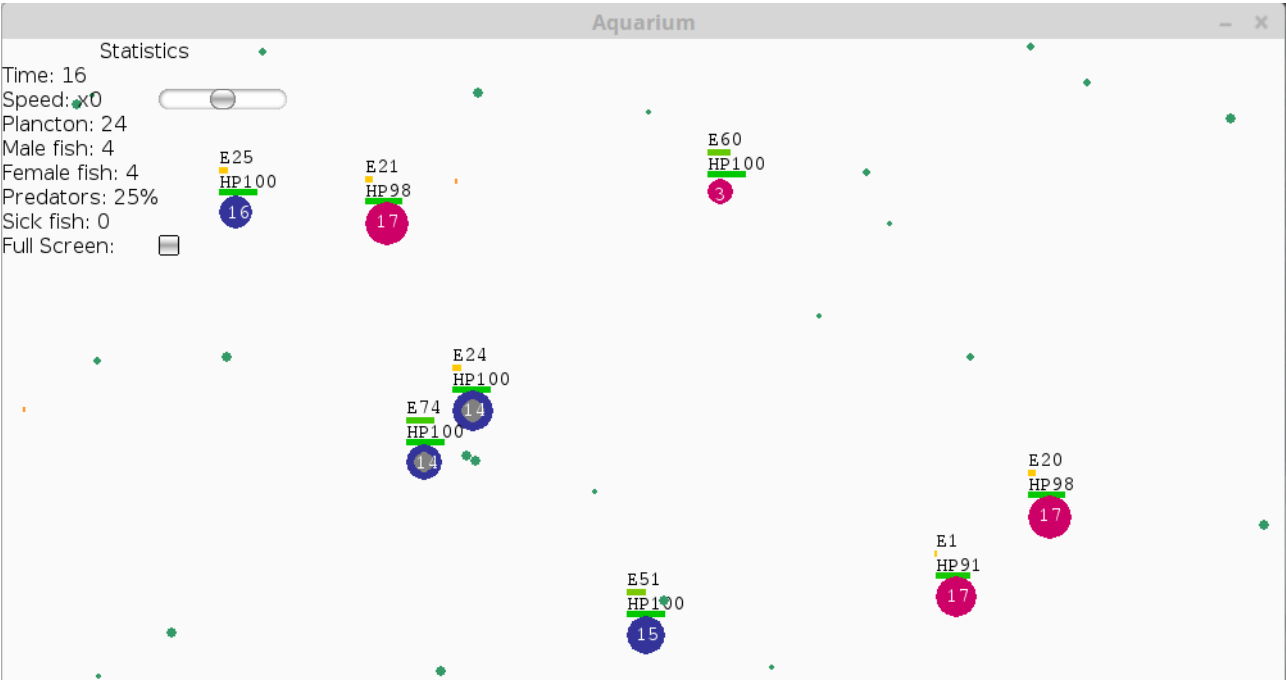
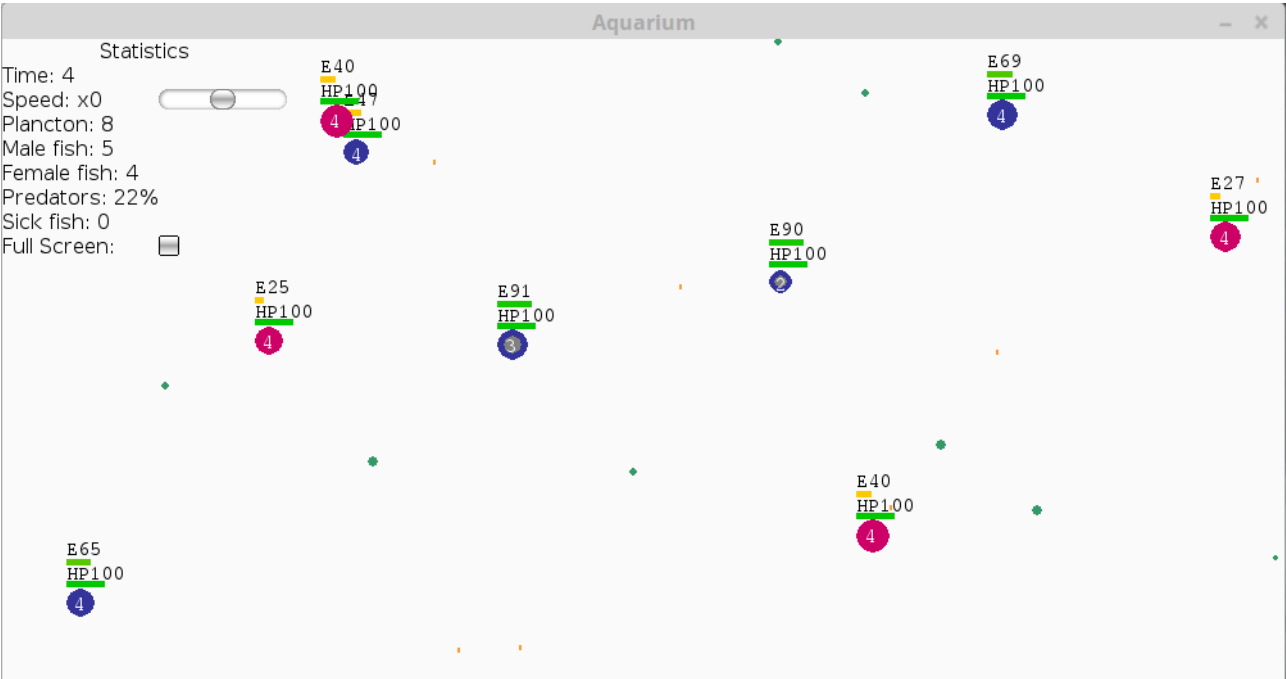
sem. letni 2018

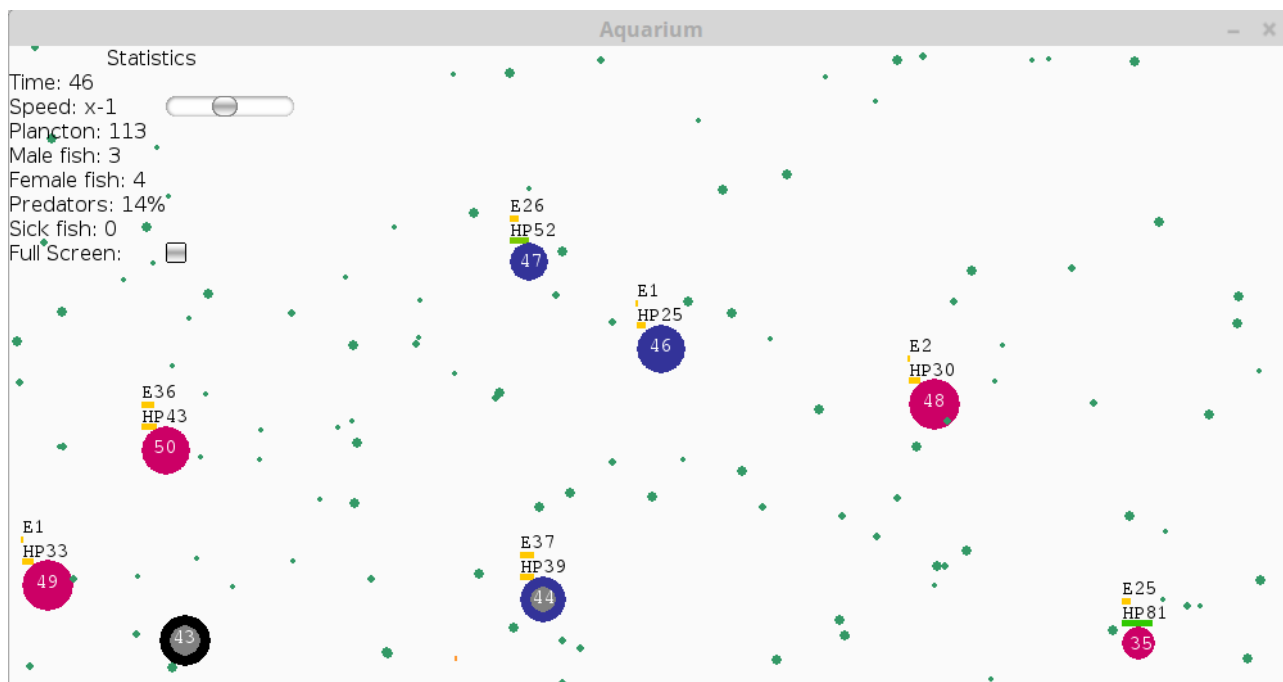
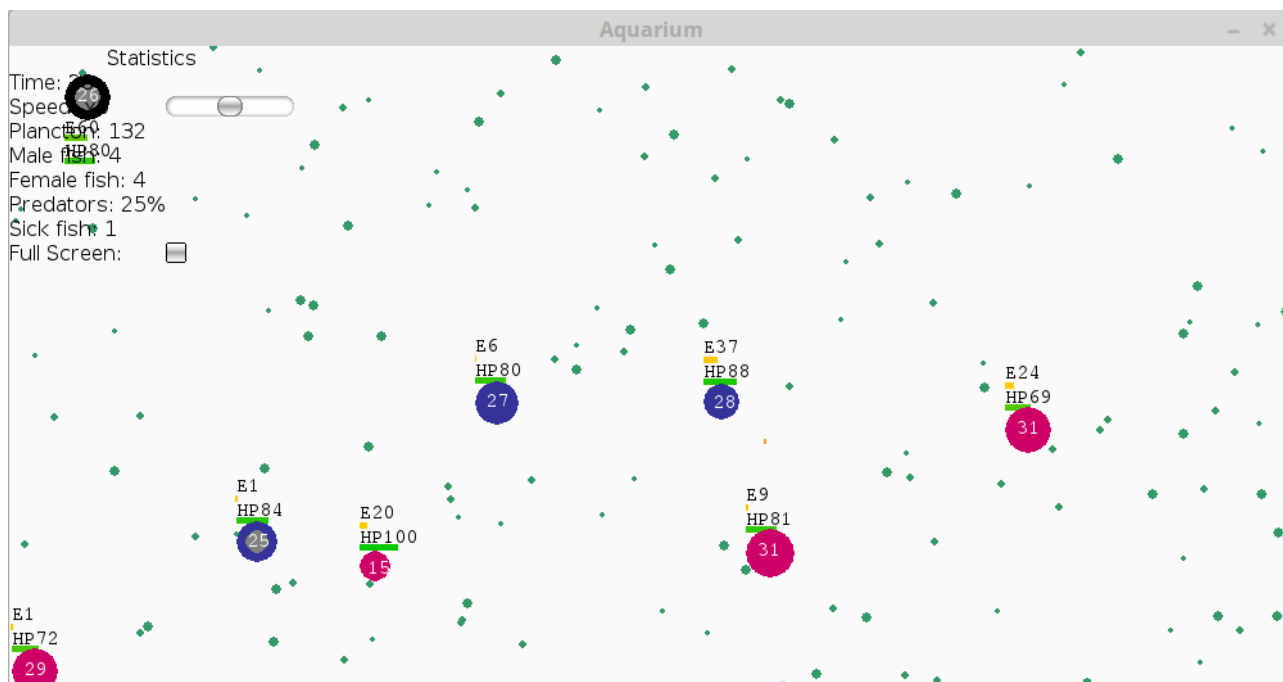
Cel projektu

Celem projektu jest stworzenie symulacji środowiska półnaturalnego organizmów żywych. W dokładności będzie to symulacja akwarium. W akwarium znajdują się rybki, które między sobą interakują, mogą płodzić potomstwo, zjadać, gonić się czy przed sobą uciekać. Niektóre cechy akwarium nie będą dotrzymane całkowicie. Ze względów technicznych, na przykład narybek nie jest “odławiany” po urodzeniu, a także nie martwimy się ilością tlenu czy azotu w wodzie. Z drugiej strony zasadniczo niegłodne ryby nie goniłyby się, więc prawdopodobnie dobry właściciel nie chciałby doprowadzić swoich pupili do kanibalizmu i dosypywałby im odpowiednią ilość pokarmu. W naszej symulacji, aby pokazać różne zachowanie ryb, zmuszamy je czasem do czynów moralnie niepoprawnych.

Projekt pisany jest w Pythonie oraz przy pomocy biblioteki pygame.

Screenshots z przykładowej symulacji





Dokładny opis działania

PRĘDKOŚĆ SYMULACJI

Prędkość symulacji można sterować za pomocą suwaka, dostępne prędkości to od 10 razy wolniej do 10 razy szybciej, przy czym przy prędkościach "-1, 0, 1" prędkość symulacji jest taka sama.

PLANKTON

Początkową wartość planktonu określa `PLANCTON_START_NUM`. Planktonu przybywa raz na `PLANCTON_TIMER` o wartość wyliczoną z funkcji sinusa.

Ilość dodawanego planktonu to % z `PLANCTON_MAX_TO_ADD`. % ten to wartość funkcji sinus, wyliczoną z argumentu, który jest losową wartością. Wartości spośród, których argument jest losowany są zwiększane co `PLANCTON_TIMER` o `RADIANS_CHANGE`.

Plankton to zielone kulki, których wielkość waha się od `MIN_PL_RADIUS` do `MAX_PL_RADIUS`.

Im większy plankton tym więcej energii może dać rybce zjedzenie go.

Pojawia się w losowych miejscach na ekranie.

RYBKA

Rybka przedstawiona jest jako koło o danym kolorze. (niebieski/różowy)

Podczas tworzenia rybki:

- losowana jest jej płeć, na tej podstawie wybierany jest także kolor

- losowana jest pozycja początkowa

- przydzielane są punkty energii oraz zdrowia (health points - HP), odpowiednio `MAX_ENERGY` oraz `MAX_HP`

- losowany jest wektor ruchu

- liczby związane z rybką: E (energia), HP (punkty zdrowia), na środku rybki jest jej wiek w `FISH_YEAR`

- losowana jest wielkość początkowa między `INIT_MIN_FISH_SIZE`, a `INIT_MAX_FISH_SIZE` oraz szybkość rośnięcia rybki

- jeśli jest predatorem ma także szarą kropkę na środku

JEDZENIE/ZDOBYWANIE ENERGII

Każdorazowo jeśli rybka wpłynie na plankton otrzymuje ilość energii odpowiadająca rozmiarowi planktonu przemnożonego przez `MULTIPLIER_FOR_FOOD`.

Oczywiście rybka nie może mieć energii większej niż `MAX_ENERGY`.

Predatorzy mogą zjadać inne, mniejsze od siebie, rybki. Jeśli rybka zje rybkę, która była chora, sama zachorowuje.

ROŚNIĘCIE RYBKI

Rybki rosną w nieskończoność (tak jak w życiu) jednak coraz wolniej. Podczas tworzenia rybki losowana jest szybkość jej rośnięcia.

Co jeden FISH_YEAR obliczana jest wartość ile rybka urosła, obliczamy ją na podstawie funkcji: y/x , gdzie x to wiek rybki, a y to szybkość rośnięcia rybki (wartość między GROWING_UP_SPEED_MIN a GROWING_UP_SPEED_MAX).

PRĘDKOŚĆ RYBKİ

Prędkość poruszania się rybki zależy od jej energii. Powyżej/bądź równo z wartością ENERGY_CHANGE_VELOCITY rybka porusza się z prędkością MAX_SPEED, poniżej porusza się z prędkością MIN_SPEED.

PORUSZANIE SIĘ RYBKİ

Każdy ruch rybki kosztuje ją ENERGY_POINT przemnożony przez jej prędkość i rozmiar. Oznacza to, że rybka zużywa więcej energii jeśli porusza się szybciej i im większa (więc jednocześnie i starsza) jest.

[Tutaj rozwiązujemy problem, że rybka starsza może krócej uciekać/gonić -> im starsza tym większa więc straci więcej energii więc krócej będzie mogła uciekać.]

Energia rybki nie może spaść poniżej MIN_ENERGY.

Jeśli rybka niczego nie goni, ani nie ucieka, porusza się próbując wpłynąć na punkt, który jest dla niej losowany.

Drapieżne rybki zjadają mniejsze rybki otrzymując analogiczną ilość energii.

GONIENIE INNYCH RYBEK

Rybka jest w stanie gonić inne rybki jeśli jest predatorem. Prawdopodobieństwo bycia predatorem to 1/3.

Predator oznaczany jest poprzez szarą kropkę na środku swojego ciała.

Ponieważ rybki nie zjadają swoich pobratymców jeśli nie muszą, gonienie innej rybki następuje tylko gdy w akwarium

jest za mało pożywienia (mniej niż HUNGER_PLANKTON_LIMIT).

Gonić mogą tylko drapieżne rybki gdyż tylko one mogą zjeść inną rybkę.

Rybka zacznie gonić najbliższą rybkę, którą dostrzeże i która jest mniejsza od niej. Zasięg wzroku rybki zależy od wielkości rybki, rybki większe łatwiej jest zauważyć niż mniejsze.

Aby uzyskać wartość zasięgu wzroku mnożymy rozmiar potencjalnego celu przez VISION_MULTIPLIER. Zasięg wzroku nie może być większy niż MAX_FISH_VISION.

Jeśli najbliższa rybka została wzięta na celownik zostaje goniona, wtedy prędkość naszej rybki zwiększa się do CHASING_SPEED.

Za większą prędkość poruszania się zabieramy więcej punktów energii.

Nawet jeśli rybki normalnie nie byłoby stać na szybkie poruszanie się (ze względu na ograniczenie ENERGY_CHANGE_VELOCITY), jeśli spełnione są warunki do gonienia, zacznie ona gonić rybkę!

Toż to przecież chodzi o przetrwanie! Jedynie przy energii równej MIN_ENERGY nie robi tego, będzie poruszać się z MIN_SPEED.

UCIEKANIE PRZED INNYMI RYBKAMI

Rybka ucieka przed każdym większym od siebie drapieżnikiem (niezależnie od ilości planktonu - przecież ona nie wie kiedy drapieżnik zaatakuje więc profilaktycznie ratuje łuski).

Uciekanie jest identyczne do gonienia jeśli chodzi o mechanikę, punkt wybrany jest odbiciem predatora jednocześnie w osi OX i OY.

STARZENIE SIĘ RYBK

Rybka naturalnie "starzeje się" - podwyższany jest jej wiek (AGE) wraz z przemijającym czasem. W momencie gdy mienie FISH_YEAR jednostek czasu, rybka traci punkt życia.

Starzenie się można przyspieszyć bądź zwolnić. Jeśli pod rząd przez SLOWER_FASTER_AGING_COUNTER jednostek czasu rybka utrzyma poziom HP poniżej/powyżej HP_FASTER_AGING/HP_SLOWER_AGING rybka może zacząć starzeć się wolniej/szybciej.

Od licznika "dni" w jednym FISH_YEAR odpowiednio jest dodawana bądź odejmowana wartość ADDITIONAL_FISH_YEAR. Co oznacza, że rybka się szybciej bądź wolniej starzeje.

Na przykładzie człowieka: założmy, że człowiek A ma 35 HUMAN_YEARS i 60 "dni". Ponieważ odżywia się zdrowo i jego HP przez jakiś czas (na przykład 30 dni) jest na wysokim poziomie to niejako "odmładza się", więc odejmujemy od jego 'licznika wieku' na przykład 20 dni. W ten sposób traktujemy jakoby człowiek A miał 35 HUMAN_YEARS i 40 dni. Odmłodziliśmy go. W drogą stronę działa to analogicznie.

Ponieważ nie można "odmładzać się" oraz "postarzać" w nieskończoność, nawet jeśli poziom HP będzie utrzymywał się na odpowiednim poziomie przez dłuższy czas, zmienić w ten sposób swój wiek można tylko 3 (hardcoded) razy pod rząd. Jeśli w międzyczasie HP rybki na chwilę spadnie, a później wzrośnie - licznik się resetuje.

REGENERACJA RYBK

Rybka może "zdecydować" o próbie regeneracji swojego życia kosztem posiadanej energii. Jeśli ilość punktów życia jest mniejsza bądź równa HP_REGENERATION_POSSIBLE oraz energia jest większa bądź równa ENERGY_REGENERATION_POSSIBLE rybka może zacząć się regenerować. Podczas regeneracji rybka nie porusza się, a trwa to tyle jednostek czasu ile określa REGENERATION_CYCLE. Następnie rybka dostaje jeden dodatkowy punkt życia (HP) - oczywiście nie powyżej wartości maksymalnej. W zamian rybka traci ENERGY_REGENERATION_COST energii.

ROZMNAŻANIE RYBEK

Rybki podzielone są na dwie płci (losowanie odbywa się podczas tworzenia rybki, kolor rybki także zależy od płci).

Z każdym ruchem dla rybki żeńskiej następuje decyzja czy rybka zostawi gdzieś ikrę. Rybka może to zrobić nie częściej niż REPRODUCTION_POSSIBLE, a decyzja jest losowana, prawdopodobieństwo na to wynosi 1/LAYING_EGG_PROBABILITY. W momencie gdy rybka zostawi ikrę, jest ona możliwa do zapłodnienia przez EGG_FRESHNESS jednostek czasu akwarium (odświeżanie ekranu).

W momencie gdy na ikrę wpłynie samiec, podejmowana jest decyzja czy chce ją zapłodnić (prawdopodobieństwo określa: $1/(LAYING_EGG_PROBABILITY+1)$).

Jeśli ikra zostanie zapłodniona: pojawi się NEW_FISH_NUM nowych rybek w akwarium.

Decyzja o rozmnażaniu następuje przed decyzją o regeneracji -> oznacza to, że reprodukcja jest "ważniejsza" dla rybki. Reprodukacja zabiera rybce ENERGY_REPRODUCTION_COST energii.

UTRATA ZDROWIA

Rybka traci HP gdy przez MAX_HUNGRY_TIME czasu jej energia wynosi poniżej MIN_ENERGY_HP_LOSE oraz gdy jest chora. Jeśli jest i głodna i chora, traci życie podwójnie. W momencie gdy HP = 0, rybka umiera.

CHOROBA

Co jakiś czas (DISEASE_DEADLINE) może pojawić się choroba z prawdopodobieństwem $1/(DISEASE_PROBABILITY+1)$. Jedna z rybek jest losowana i zarażana chorobą. Jeśli rybka zje rybkę, która była chora, sama zachorowuje.

Plany na przyszłość

W zamyśle projekt będzie dalej rozwijany, prawdopodobnie do pracy inżynierskiej. W dalszych planach jest dodanie następnych możliwości parametryzacji zachowania rybek jak, na przykład, dodanie im roślin do zjadania oraz kryjówek, w których mogłyby się chować przed drapieżnikami. Duża ilość zależności sprawia, że należałoby dostosować wszystkie parametry tak, aby symulacja była samowystarczalna i cykl życia byłby zachowany – rybki obumierają, a pojawiają się nowe I nigdy nie ma sytuacji kiedy akwarium wymarze całkowicie.

Dodatkowo w zamyśle jest wprowadzenie algorytmów samouczących się, aby rybkami mogły sterować algorytmy, które umiałyby dostosować się do warunków, które zastały.