|  |  |
| --- | --- |
| Informatyka, studia dzienne, | inż I st. semestr VI |
| **Sztuczna inteligencja oraz systemy ekspertowe**  Prowadzący: mgr inż. Jagoda Lazarek | 2012/2013  Środa,8.30 |
|  |  |
| Data oddania:\_\_\_\_\_\_\_ | Ocena:\_\_\_\_\_\_\_ |

Karol Górecki 165405

Tomasz Trębski 165535

Robert Glonek 165404

Maciej Bąk 157802

Łukasz Rowiński 165508

Tomasz Szopka 165530

Zadanie 2: Symulator świata

# Opis świata

Jako nasz pomysł na świat wybraliśmy kawałek środowiska naturalnego, w tym przypadku jezioro z otaczającym je lasem. Głównymi agentami w tym świecie są: ryby drapieżne, niedrapieżne, wędkarz, kłusownik, leśniczy oraz ptaki łowiące ryby.

Ryby niedrapieżne mają poziom głodu, który będzie zmuszał je do jedzenia najbliższego dobrego dla nich pokarmu – może to być naturalny pokarm z jeziora lub przynęta na wędce, jeżeli ryba będzie dostatecznie blisko. Ryby te widzą tylko pokarm, który znajduję się jedno pole przezd nimi. Ilość : 4+, w tym 2 różne. Dla tego typu ryby przewidujemy następujące czynniki efektywności: ilość zjedzonego pokarmu, ilość pomyślnych ucieczek z wędki lub przed rybą drapieżną..

Ryby drapieżne patrolują jezioro w poszukiwaniu pokarmu, czyli ryb niedrapieżnych. Wyczuwają ryby znajdujące się w określonej odległości pola na około nich. Ryby drapieżne mają następujące czynniki efektywności: ilość upolowanych ryb oraz współczynnik przeżywalności, czyli ilość iteracji, których udało się jej przetrwać.

Wędkarz jest obiektem nieruchomym, który wabi głodne ryby na przynętę. Zajmuje jedno pole lądu oraz jedno pole jeziora. Jego czynniki efektywność to ilość złapanych ryb oraz suma wag złapanych ryb.

Kłusownik jest odmiana wędkarza, który ma dwa zadania – złowienie ryb oraz unikanie leśniczego, który może go zatrzymać. Zajmuje te same pola, co wędkarz, jednak jest agentem ruchomym. Czynniki efektywności dla kłusownika ilość iteracji bez złapania, suma danych łapówek.

Leśniczy patroluje brzeg jeziora i sprawdza karty wędkarskie łowiących. Ma możliwość aresztowania kogoś bez pozwolenia lub może dać się przekupić łapówką. Jest agentem ruchomym, który patroluje pojedyncze pole, na którym się znajduj. Efektywności leśniczego jest mierzona przy pomocy ilość wziętych łapówek (czynnik skorumpowania), ilości złapanych kłusowników.

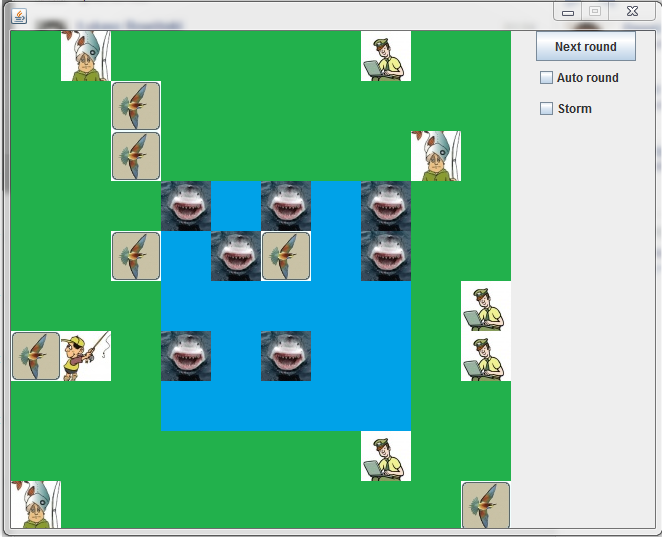
Ptak jest agentem ruchomym, który przelatuje nad jednym polem jeziora i jeżeli jest głodny to próbuje złowić znajdujące się pod nim ryby. Jego czynniki efektywności to ilości złapanych ryb.

Każdy agent ma określoną maksymalną ilość pól, które może się przesunąć w danej iteracji. Co iteracja ilości pól, o które się agent przesuwa, jest losowana.

Czynnikiem niedeterministycznym w naszym świecie jest pogoda, która zmienia się, co losową ilość iteracji. Czynnik ten może przyjąć następujące stany: słonecznie, deszcz, burza.

# Opis implementacji

Projekt jest to połączenie aplikacji desktopowej napisanej w Javie oraz zestawu skryptów programu Clisp. Program w Javie odpowiada za inicjalizacje świata w Clispie, odbieranie informacji po ich przetworzeniu oraz z wyświetlanie wizualizacji świata w postaci graficznego interfejsu użytkownika.



Rysunek 1Przykładowy screenshot z działania programu.

Cześć projektu, która została napisana w Clispie odpowiada za logikę. W tym fragmencie zadania są funkcje (rules) odpowiedzialne m.in. za:

* Poruszanie się aktorów
* „zabijanie”
* Unikatowe funkcje zależne od implementacji

**Wymagania implementacyjne aktorów**

1. Aktorzy muszą implementować logikę odpowiedzialną za odczyt/zapis informacji ze struktur wejściowych/wyjściowym. Jest to ważne z tego powodu, że następnie informacja o tych danych są serializowane do Stringa (java.lang.String) i przekazywane do instancji Clisp’a danego aktora
2. Aktorzy muszą implementować logikę pozwalająca na losowe przeprowadzenie następnej tury, tj. informacje zależne losowo (kierunek przemieszczenie, czy losowo wybrać kierunek, czy przyjąć łapówkę) muszą zostać ustalone na podstawie danych ze struktur wejściowych na poziomie logiki danego aktora.
3. Implementacja aktorów konkretnych osób należy umieścić w osobnych paczkach
4. W paczce, które zawiera implementacja aktorów powinien również znaleźć się plik \*.properties, który zawiera dane wejściowe do pierwszej tury gry. Format pliku musi odpowiadać strukturze danych wejściowych, aby program mógł go odczytać   
   i przeprowadzić inicjalizację instancji aktora w prawidłowy sposób.

Powyższe wymagania można zrealizować implementując dla swojego aktora interfejs **org.kornicameister.sise.lake.types.ClispType**, który wymusza implementacją odpowiednich metod.

**Jak działa tura**

Aplikacja trzyma załadowane deskryptory aktorów. Deskryptor jest w tym miejscu takim obiektem, który wiąże ze sobą środowiska Clisp oraz instancję aktora po stronie Javy. Tura jest kontrolowana przez świat, który uruchamia aktorów.