Министерство образования и науки РФ

Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

**факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА №1**

по дисциплине

“Системы искусственного интеллекта”

**Студент:**

Чернова Анна Ивановна

Группа P33301

**Преподаватель:**

Кугаевских Александр Владимирович



Санкт-Петербург, 2023

**Введение:**

Цель проекта: изучение и использование языка программирования Пролог для решения задач логического вывода и искусственного интеллекта. Пролог позволяет описывать отношения между объектами, что позволяет решать задачи, которые традиционно сложно решать с помощью других языков программирования.

В качестве темы выбрана игра “Mobile Legens”.

**Анализ требований:**

Программа для взаимодействия с пользователем должна:

* Обрабатывать ввод пользователя и сообщать об ошибке в нём, если она обнаружится
* Поддерживать несколько типов запросов
* Быть масштабируемой

База знаний должна:

* Основываться на информации, актуальной и корректной для игры Mobile Legens
* Поддерживать иерархию игровых персонажей
* Поддерживать проверку выбора напарника по игре
* Поддерживать проверку возможности вступления в бой двух персонажей

**Изучение основных концепций и инструментов:**

База знаний - хранилище структурированных данных, фактов, понятий и знаний, созданная и устроенная для доступа к ним и их упорядоченного использования. База знаний создаётся для некоторой предметной области и позволяет как человеку, так и программе извлекать и использовать данные о ней.

Знание – факты/информация о предметной области в базе знаний

Факты – данные/утверждения. Описывающие сущности и их характеристики в базе знаний

Правила – логические конструкции, основывающиеся на фактах и других правилах, выводящее связи и характеристики сущностей

Запросы – обращения к базе знаний с целью получения информации о сущностях, характеристиках и их связях по заданным условиям

Онтология – формализация предметной области с помощью концептуальной схемы, определяющая понятия, их свойства и отношения между ними

Классы – типы/категории сущностей в онтологии

Свойства – атрибуты/характеристики сущностей

Экземпляры – конкретные объекты, принадлежащие к классам и обладающие свойствами

Отношения – определяют связи и взаимодействие между сущностями и классами

Prolog – декларативный язык программирования, использующийся для решения задач искусственного интеллекта и логического программирования.

Prolog позволяет создать базу данных и исполнять запросы к ней.

В данном проекте для взаимодействия с Prolog использовались:

* SWI Prolog
* Библиотека jpl для языка Java

**Реализация системы искусственного интелекта на Prolog:**

name\1 – имя персонажа

rank\2 – имя персонажа и ранг, соответствующий ему

playerPosition\2 – имя персонажа и позиция, на которой он играет

position\2 - пара персонажей с любыми позициями

samePosition\2 – пара персонажей с одинаковой позицией

sameRank\2 – пара персонажей с одинаковыми рангами

opponentPair\2 – проверка возможности игры персонажей против друг друга (условие возможности: одинаковые позиции + ранг)

command\2 – проверка возможности игра персонажей в паре против другой команде (условие возможности: одинаковый ранг + разные позиции)

manyPosition\1 - персонажи, которые могут играть на двух позициях (например, поддержка + танк)

name('Natan').  
name('Melissa').  
name('Hanabi').  
name('Tigreal').  
name('Akai').  
name('Franco').  
name('Atlas').  
name('Balmond').  
name('Freya').  
name('Saber').  
name('Karina').  
name('Fanny').  
name('Hayabusa').  
name('Nana').  
name('Eudora').  
name('Angela').  
name('Floryn').  
name('Johnson').  
name('Lesley').  
  
  
rank('Natan', '3' ).  
rank('Melissa', '1').  
rank('Hanabi', '2').  
rank('Tigreal', '2').  
rank('Akai','2').  
rank('Franco','1').  
rank('Atlas', '3').  
rank('Balmond', '1').  
rank('Freya', '2').  
rank('Saber', '2').  
rank('Karina','1').  
rank('Fanny', '3').  
rank('Hayabusa', '3').  
rank('Balmond', '3').  
rank('Nana', '1').  
rank('Eudora', '3').  
rank('Angela','3').  
rank('Nana', '2').  
rank('Floryn', '2').  
rank('Johnson', '3').  
rank('lesley', '3').  
  
  
playerPosition('Natan', 'Marksman').  
playerPosition('Melissa', 'Marksman').  
playerPosition('Hanabi', 'Marksman').  
  
playerPosition('Lesley', 'Marksman').  
playerPosition('Lesley', 'Assassin').  
  
playerPosition('Tigreal', 'Tank').  
playerPosition('Tigreal', 'Support').  
  
playerPosition('Akai','Tank').  
  
playerPosition('Franco','Tank').  
playerPosition('Franco','Support').  
  
playerPosition('Atlas','Tank').  
playerPosition('Freya','Fighter').  
playerPosition('Balmond', 'Fighter').  
playerPosition('Hayabusa','Assassin').  
playerPosition('Karina','Assassin').  
playerPosition('Fanny','Assassin').  
playerPosition('Saber','Assassin').  
playerPosition('Eudora','Mage' ).  
  
playerPosition('Nana','Mage').  
playerPosition('Nana','Support').  
  
playerPosition('Floryn', 'Support').  
playerPosition('Angela', 'Support').  
  
playerPosition('Johnson', 'Tank').  
playerPosition('Johnson','Support').  
  
%пара персонажей с любыми позициями  
position(X,Y):-playerPosition(X, \_), playerPosition(Y, \_ ), X\=Y.  
  
%персонажи с одинаковой позицией  
samePosition(X, Y):-playerPosition(X, Z), playerPosition(Y, Z), X\=Y.  
%персонажи с одинаковыми рангами  
sameRank(X, Y):- rank(X,Z), rank(Y,Z), X\=Y.  
  
% могут ли игроки играть против друг друга? (условие возможности: одинаковые позиции + ранг)  
opponentPair(X, Y):- samePosition(X, Y), sameRank(X, Y).  
  
% могут ли игроки играть в паре против другой команды?(условие возможности: одинаковый ранг + разные позиции)  
command(X,Y):- sameRank(X, Y), (position(X,Y))\=(samePosition(X,Y)) .  
  
% персонажи, которые могут играть на двух позициях (например, поддержка + танк)  
manyPosition(X):- (playerPosition(X, Y), playerPosition(X, Z), Y@<Z).

Примеры запросов prolog:

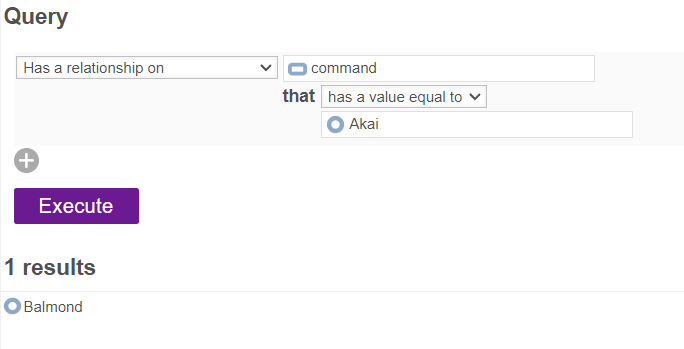
samePosition('Fanny', 'Johnson').

opponentPair('Hayabusa', 'Johnson').

command('Freya', 'Hanabi').

manyPosition('Johnson').

Пример запроса к онтологии:



Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

**Вывод**: разработанная система позволит проверить какого персонажа лучше выбрать в зависимости от уровня игры и предпочтений, предложит советующих персонажей, напарников и противников. Игрокам не придётся искать эту информацию на сайтах в интернете (что занимает время) или пытаться пройти игру, не зная, что делать дальше (что займёт ещё больше времени).