

Федеральное государственное автономное образовательное учреждение высшего образования  
"Национальный Исследовательский Университет ИТМО"  
Мегафакультет Компьютерных Технологий и Управления  
Факультет Программной Инженерии и Компьютерной Техники



**Модуль №1**  
по дисциплине  
**'Системы искусственного интеллекта'**

Выполнил Студент группы Р33102  
**Лапин Алексей Александрович**  
Преподаватель:  
**Королёва Юлия Александровна**

г. Санкт-Петербург  
2023г.

# Содержание

<b>1 Введение:</b>	<b>4</b>
1.1 Описание целей проекта и его значимости. . . . .	4
Цели проекта: . . . . .	4
Описание лабораторных работ: . . . . .	4
Значимость проекта: . . . . .	4
<b>2 Анализ требований:</b>	<b>5</b>
2.1 Определение основных требований к системе поддержки принятия решений. . . .	5
Критерии оценки включают в себя: . . . . .	5
2.2 Выявление требований к базе знаний и онтологии для представления знаний. . .	5
Требования к базе знаний: . . . . .	5
Требования к запросам к базе знаний: . . . . .	6
Требования к онтологии: . . . . .	6
<b>3 Изучение основных концепций и инструментов:</b>	<b>6</b>
3.1 Обзор основных концепций баз знаний и онтологий. . . . .	6
База знаний: . . . . .	6
Онтология: . . . . .	6
3.2 Изучение Prolog и его возможностей для разработки систем искусственного ин- теллекта. . . . .	7
Изучение Prolog . . . . .	7
Prolog . . . . .	7
Преимущества: . . . . .	7
3.3 Ознакомление с инструментами и библиотеками, подходящими для работы с ба- зами знаний и онтологиями на Prolog. . . . .	7
SWI-Prolog: . . . . .	7
SWI-Prolog Semantic Web Library 3.0: . . . . .	7
JPL (Java-Prolog Interface): . . . . .	8
Protege: . . . . .	8
<b>4 Реализация системы искусственного интеллекта на Prolog:</b>	<b>8</b>
4.1 Создание правил и логики вывода для принятия решений на основе базы знаний и онтологии. . . . .	8
База знаний: . . . . .	8
Онтология: . . . . .	13
Система поддержки принятия решения на основе базы знаний: . . . . .	17
<b>5 Оценка и интерпретация результатов:</b>	<b>21</b>
5.1 Примеры запросов для БЗ и онтологии, сравнение разницы реализации. . . . .	21
Запрос к базе знаний: . . . . .	21
Запрос к онтологии: . . . . .	21
Запрос к базе знаний: . . . . .	22
Запрос к онтологии: . . . . .	23
5.2 Оценка соответствия системы поставленным требованиям и достижению целей проекта. . . . .	24
5.3 Интерпретация результатов и описание дальнейших возможностей развития и улучшения системы. . . . .	24

<b>6</b>	<b>Заключение:</b>	<b>25</b>
6.1	Описание преимуществ и потенциальных применений разработанной системы искусственного интеллекта на базе Prolog, баз знаний и онтологий. . . . .	25
	Преимущества системы: . . . . .	25
	Потенциальные применения системы: . . . . .	25

# 1 Введение:

## 1.1 Описание целей проекта и его значимости.

**Цели проекта:** Целью данного проекта является разработка и применение системы, основанной на базе знаний и онтологии, для поддержки пользователей в принятии решений в контексте выбора видеоигры. Проект включает в себя следующие лабораторные работы:

### Описание лабораторных работ:

**Лабораторная 1. Создание базы знаний и выполнение запросов в Prolog** Эта лабораторная работа направлена на развитие навыков работы с фактами, предикатами и правилами в логическом программировании с использованием языка Prolog. Важными задачами являются создание базы знаний и выполнение разнообразных запросов к этой базе.

**Лабораторная 2. Создание онтологии в Protege** Цель этой лабораторной работы - преобразовать базу знаний, созданную в лабораторной работе 1, в онтологическую форму с использованием среды разработки онтологий Protege. Это позволит более формально и структурированно описать объекты и отношения между ними, а также определить классы и свойства.

**Лабораторная 3. Разработка системы поддержки принятия решения на основе базы знаний или онтологии** В этой лабораторной работе разрабатывается программа, которая использует базу знаний или онтологию для предоставления рекомендаций пользователю на основе введенных им данных о своих интересах и предпочтениях в контексте выбора видеоигры. Система должна уметь принимать запросы, проводить логические операции, и предоставлять рекомендации на основе полученных результатов.

### Значимость проекта:

- Лабораторная 1 позволит студентам развить навыки логического программирования и создания базы знаний, что может быть полезно в различных областях информационных технологий.
- Лабораторная 2 учит студентов работать с онтологиями, что является важной компетенцией в сфере искусственного интеллекта и семантического веба.
- Лабораторная 3 предоставляет студентам опыт в разработке систем поддержки принятия решений, что актуально в многих областях, включая рекомендательные системы и искусственный интеллект.

Проект ориентирован на развитие ключевых компетенций в области информационных технологий и имеет практическое применение в помощи пользователям при выборе видеоигр, что делает его значимым и актуальным.

## **2 Анализ требований:**

### **2.1 Определение основных требований к системе поддержки принятия решений.**

1. Создание программы, позволяющей пользователю вводить запрос через командную строку.
2. Использование введенных пользователем данных для формулирования логических запросов к базе знаний или онтологии.
3. На основе результатов выполнения запросов, система должна предоставлять рекомендации или советы, связанные с выбором из базы знаний или онтологии.
4. Обработка ввода пользователя и разбор введенных данных.
5. Документирование работы системы и описание ее функциональности.

#### **Критерии оценки включают в себя:**

1. Корректность и эффективность реализации системы поддержки принятия решений.
2. Способность программы адекватно использовать базу знаний или онтологию для выдачи рекомендаций.
3. Качество тестирования и обработка ввода пользователя.
4. Качество документации и описания работы системы.

### **2.2 Выявление требований к базе знаний и онтологии для представления знаний.**

#### **Требования к базе знаний:**

1. База знаний должна содержать не менее 20 фактов с одним аргументом.
2. База знаний должна содержать 10-15 фактов с двумя аргументами, которые дополняют и показывают связь с другими фактами.
3. В базе знаний должны присутствовать 5-7 правил.
4. Факты могут описывать объекты, их свойства и отношения между ними.
5. Предикаты должны описывать различные атрибуты объектов.
6. Правила должны представлять собой логические законы и выводы, которые можно сделать на основе фактов и предикатов.
7. В коде базы знаний должны быть комментарии, описывающие факты, предикаты и правила.

### **Требования к запросам к базе знаний:**

Запросы к базе знаний должны быть разной сложности и включать в себя:

1. Простые запросы для поиска фактов.
2. Запросы, использующие логические операторы (и, или, не) для формулирования сложных условий.
3. Запросы, использующие переменные для поиска объектов с определенными характеристиками.
4. Запросы, требующие выполнения правил для получения результата.

### **Требования к онтологии:**

1. Преобразование фактов и отношений из Prolog в концепты и свойства в онтологии.
2. Описание классов и свойств в онтологии, соответствующих объектам и отношениям из базы знаний.
3. Создание иерархии классов в Protege, если необходимо.
4. Тестирование онтологии и демонстрация ее функциональности.

## **3 Изучение основных концепций и инструментов:**

### **3.1 Обзор основных концепций баз знаний и онтологий.**

#### **База знаний:**

- **Факт** - это то, что известно.
- **Правило** - это способ порождения новых фактов на основе имеющихся.
- **Предикат** - это функция, которая возвращает бинарное значение (истина или ложь).
- **Процесс уннификации** - это процесс нахождения решения, который заключается в сопоставлении предиката цели с предикатами базы знаний.
- Интерпретатор Пролога автоматически выполняет поиск решения. Механизм поиска реализован с помощью отката после неудачи. Откат происходит на следующий экземпляр неоднозначного предиката. Выполнение программы на Прологе (резолюция цели) заключается в уннификации цели с базой знаний.

#### **Онтология:**

- Онтологии содержат концепты или классы, которые представляют сущности или идеи в области. Концепты могут иметь атрибуты и отношения с другими концептами.
- Онтологии определяют отношения между концептами для отражения взаимосвязей и зависимостей между ними. К распространенным типам отношений относятся "is-a", "part-of", "has-property" и "has-value".

- Свойства описывают характеристики или атрибуты концептов. Они могут использоваться для определения дополнительной информации о концептах, например, их размера, цвета или местоположения.
- Экземпляры - это конкретные индивидуумы или примеры понятий в области. Например, "Apple" может быть экземпляром понятия "Fruit" в онтологии фруктов.
- Онтологии часто используют иерархическую структуру, называемую таксономией, для организации понятий на основе их отношений обобщения и специализации. Такая структура помогает классифицировать и организовывать знания.
- Онтологии позволяют проводить рассуждения и делать выводы, применяя логические правила для получения новых знаний из существующих. Умозаключения помогают отвечать на сложные запросы или делать логические выводы.

### 3.2 Изучение Prolog и его возможностей для разработки систем искусственного интеллекта.

**Изучение Prolog** является важным шагом для разработчиков, занимающихся созданием систем искусственного интеллекта. Prolog (Programming in Logic) - это язык программирования, основанный на логике предикатов, который широко используется для решения задач, связанных с логическим выводом.

**Prolog** - это язык логического программирования, который использует базу знаний для решения задач. Программа на Прологе состоит из фактов и правил, которые описывают отношения между объектами. Пролог использует механизм унификации для поиска решения. Пролог используется для разработки систем искусственного интеллекта, таких как экспертные системы, системы поддержки принятия решений, системы обработки естественного языка и другие.

#### Преимущества:

- Лаконичный язык. Это означает, что знания могут быть представлены в очень компактной форме, что важно для эффективного вывода. Другими словами, вы можете выражать сложные идеи очень компактным способом, что делает Prolog хорошо подходящим для приложений искусственного интеллекта.
- Декларативный язык. Это означает, что вы можете описать, чего хотите достичь, не беспокоясь о том, как этого достичь.

### 3.3 Ознакомление с инструментами и библиотеками, подходящими для работы с базами знаний и онтологиями на Prolog.

**SWI-Prolog:** SWI-Prolog - это мощная и распространенная реализация языка Prolog. Она предоставляет широкий набор инструментов и библиотек для работы с базами знаний и онтологиями.

**SWI-Prolog Semantic Web Library 3.0:** Semantic Web Library - это библиотека для SWI-Prolog, которая предоставляет инструменты для работы с семантическими технологиями, такими как RDF и OWL. Она позволяет загружать, создавать и манипулировать онтологиями, а также выполнять запросы и рассуждения на основе семантических данных.

**JPL (Java-Prolog Interface):** Если вы предпочитаете использовать Java для разработки, то JPL может быть полезным инструментом. Он предоставляет интерфейс между Java и Prolog, позволяя вам использовать функциональность Prolog в своих Java-приложениях.

**Protege:** Protege - это мощное и широко используемое средство для разработки и управления онтологиями. Он предоставляет графический интерфейс для создания, редактирования и визуализации онтологий в формате OWL. Protege также поддерживает рассуждения и выполнение запросов к онтологиям.

## 4 Реализация системы искусственного интеллекта на Prolog:

### 4.1 Создание правил и логики вывода для принятия решений на основе базы знаний и онтологии.

База знаний:

```
1 % Facts: Information about the release year of Super Mario games
2 release_year('Super Mario Bros.', 1985).
3 release_year('Super Mario Bros.: The Lost Levels', 1986).
4 release_year('Super Mario Bros. 2', 1988).
5 release_year('Super Mario Bros. 3', 1988).
6 release_year('Super Mario Land', 1989).
7 release_year('Super Mario World', 1990).
8 release_year('Super Mario Land 2: 6 Golden Coins.', 1992).
9 release_year('Super Mario 64', 1996).
10 release_year('Super Mario Sunshine', 2002).
11 release_year('New Super Mario Bros.', 2006).
12 release_year('Super Mario Galaxy', 2007).
13 release_year('New Super Mario Bros. Wii', 2009).
14 release_year('Super Mario Galaxy 2', 2010).
15 release_year('Super Mario 3D Land', 2011).
16 release_year('New Super Mario Bros. 2', 2012).
17 release_year('New Super Mario Bros. U', 2012).
18 release_year('Super Mario 3D World', 2013).
19 release_year('Super Mario Run', 2016).
20 release_year('Super Mario Odyssey', 2017).
21 release_year('Super Mario Bros. Wonder', 2023).
22
23 % Facts: Characters that appear in each Super Mario game
24 appears_in_game('Mario', 'Super Mario Bros.').
25 appears_in_game('Mario', 'Super Mario Bros.: The Lost Levels').
26 appears_in_game('Mario', 'Super Mario Bros. 2').
27 appears_in_game('Mario', 'Super Mario Bros. 3').
28 appears_in_game('Mario', 'Super Mario Land').
29 appears_in_game('Mario', 'Super Mario World').
30 appears_in_game('Mario', 'Super Mario Land 2: 6 Golden Coins.').
31 appears_in_game('Mario', 'Super Mario 64').
32 appears_in_game('Mario', 'Super Mario Sunshine').
33 appears_in_game('Mario', 'New Super Mario Bros.').
```



```

34 appears_in_game('Mario', 'Super Mario Galaxy').
35 appears_in_game('Mario', 'New Super Mario Bros. Wii').
36 appears_in_game('Mario', 'Super Mario Galaxy 2').
37 appears_in_game('Mario', 'Super Mario 3D Land').
38 appears_in_game('Mario', 'New Super Mario Bros. 2').
39 appears_in_game('Mario', 'New Super Mario Bros. U').
40 appears_in_game('Mario', 'Super Mario 3D World').
41 appears_in_game('Mario', 'Super Mario Run').
42 appears_in_game('Mario', 'Super Mario Odyssey').
43 appears_in_game('Mario', 'Super Mario Bros. Wonder').
44
45 appears_in_game('Princess Peach', 'Super Mario Bros.').
46 appears_in_game('Princess Peach', 'Super Mario Bros.: The Lost Levels').
47 appears_in_game('Princess Peach', 'Super Mario Bros. 2').
48 appears_in_game('Princess Peach', 'Super Mario Bros. 3').
49 appears_in_game('Princess Peach', 'Super Mario World').
50 appears_in_game('Princess Peach', 'Super Mario 64').
51 appears_in_game('Princess Peach', 'Super Mario Sunshine').
52 appears_in_game('Princess Peach', 'New Super Mario Bros.').
53 appears_in_game('Princess Peach', 'Super Mario Galaxy').
54 appears_in_game('Princess Peach', 'New Super Mario Bros. Wii').
55 appears_in_game('Princess Peach', 'Super Mario Galaxy 2').
56 appears_in_game('Princess Peach', 'Super Mario 3D Land').
57 appears_in_game('Princess Peach', 'New Super Mario Bros. 2').
58 appears_in_game('Princess Peach', 'New Super Mario Bros. U').
59 appears_in_game('Princess Peach', 'Super Mario 3D World').
60 appears_in_game('Princess Peach', 'Super Mario Run').
61 appears_in_game('Princess Peach', 'Super Mario Odyssey').
62 appears_in_game('Princess Peach', 'Super Mario Bros. Wonder').
63
64 appears_in_game('Luigi', 'Super Mario Bros.').
65 appears_in_game('Luigi', 'Super Mario Bros. 2').
66 appears_in_game('Luigi', 'Super Mario Bros. 3').
67 appears_in_game('Luigi', 'Super Mario World').
68 appears_in_game('Luigi', 'Super Mario 64').
69 appears_in_game('Luigi', 'New Super Mario Bros. Wii').
70 appears_in_game('Luigi', 'Super Mario 3D Land').
71 appears_in_game('Luigi', 'New Super Mario Bros. 2').
72 appears_in_game('Luigi', 'New Super Mario Bros. U').
73 appears_in_game('Luigi', 'Super Mario 3D World').
74 appears_in_game('Luigi', 'Super Mario Bros. Wonder').
75
76 appears_in_game('Princess Daisy', 'Super Mario Land').
77 appears_in_game('Princess Daisy', 'Super Mario Run').
78
79 appears_in_game('Bowser', 'Super Mario Bros.').
80 appears_in_game('Bowser', 'Super Mario Bros.: The Lost Levels').
81 appears_in_game('Bowser', 'Super Mario Bros. 3').
82 appears_in_game('Bowser', 'Super Mario World').
83 appears_in_game('Bowser', 'Super Mario 64').
84 appears_in_game('Bowser', 'Super Mario Sunshine').

```

```

85 appears_in_game('Bowser', 'New Super Mario Bros.').
86 appears_in_game('Bowser', 'Super Mario Galaxy').
87 appears_in_game('Bowser', 'New Super Mario Bros. Wii').
88 appears_in_game('Bowser', 'Super Mario Galaxy 2').
89 appears_in_game('Bowser', 'Super Mario 3D Land').
90 appears_in_game('Bowser', 'New Super Mario Bros. 2').
91 appears_in_game('Bowser', 'New Super Mario Bros. U').
92 appears_in_game('Bowser', 'Super Mario 3D World').
93 appears_in_game('Bowser', 'Super Mario Run').
94 appears_in_game('Bowser', 'Super Mario Odyssey').
95 appears_in_game('Bowser', 'Super Mario Bros. Wonder').
96
97 appears_in_game('Kings', 'Super Mario Bros. 3').
98 appears_in_game('Hoot', 'Super Mario 64').
99
100 appears_in_game('Dorrie', 'Super Mario 64').
101 appears_in_game('Dorrie', 'New Super Mario Bros.').
102 appears_in_game('Dorrie', 'Super Mario Odyssey').
103
104 appears_in_game('Toadsworth', 'Super Mario Sunshine').
105 appears_in_game('Baby Luma', 'Super Mario Galaxy').
106 appears_in_game('Baby Luma', 'Super Mario Galaxy 2').
107 appears_in_game('Lubba', 'Super Mario Galaxy 2').
108 appears_in_game('FLUDD', 'Super Mario Sunshine').
109 appears_in_game('Ray', 'Super Mario Galaxy').
110 appears_in_game('Fluzzard', 'Super Mario Galaxy 2').
111 appears_in_game('Cosmic Spirit', 'Super Mario Galaxy 2').
112 appears_in_game('Cappy', 'Super Mario Odyssey').
113
114 % Facts: Platforms on which Super Mario games were released
115 game_on_platform('Super Mario Bros.', 'NES').
116 game_on_platform('Super Mario Bros.: The Lost Levels', 'NES').
117 game_on_platform('Super Mario Bros. 2', 'NES').
118 game_on_platform('Super Mario Bros. 3', 'NES').
119 game_on_platform('Super Mario Land', 'Game Boy').
120 game_on_platform('Super Mario World', 'SNES').
121 game_on_platform('Super Mario Land 2: 6 Golden Coins.', 'Game Boy').
122 game_on_platform('Super Mario 64', 'Nintendo 64').
123 game_on_platform('Super Mario Sunshine', 'GameCube').
124 game_on_platform('New Super Mario Bros.', 'Nintendo DS').
125 game_on_platform('Super Mario Galaxy', 'Wii').
126 game_on_platform('New Super Mario Bros. Wii', 'Wii').
127 game_on_platform('Super Mario Galaxy 2', 'Wii').
128 game_on_platform('Super Mario 3D Land', 'Nintendo 3DS').
129 game_on_platform('New Super Mario Bros. 2', 'Nintendo 3DS').
130 game_on_platform('New Super Mario Bros. U', 'Wii U').
131 game_on_platform('Super Mario 3D World', 'Wii U').
132 game_on_platform('Super Mario Run', 'Mobile').
133 game_on_platform('Super Mario Odyssey', 'Nintendo Switch').
134 game_on_platform('Super Mario Bros. Wonder', 'Nintendo Switch').
135

```

```

136 % Facts: Graphics style of Super Mario games
137 game_graphics('Super Mario Bros.', '2D').
138 game_graphics('Super Mario Bros.: The Lost Levels', '2D').
139 game_graphics('Super Mario Bros. 2', '2D').
140 game_graphics('Super Mario Bros. 3', '2D').
141 game_graphics('Super Mario Land', '2D').
142 game_graphics('Super Mario World', '2D').
143 game_graphics('Super Mario Land 2: 6 Golden Coins.', '2D').
144 game_graphics('Super Mario 64', '3D').
145 game_graphics('Super Mario Sunshine', '3D').
146 game_graphics('New Super Mario Bros.', '2D').
147 game_graphics('Super Mario Galaxy', '3D').
148 game_graphics('New Super Mario Bros. Wii', '2D').
149 game_graphics('Super Mario Galaxy 2', '3D').
150 game_graphics('Super Mario 3D Land', '3D').
151 game_graphics('New Super Mario Bros. 2', '2D').
152 game_graphics('New Super Mario Bros. U', '2D').
153 game_graphics('Super Mario 3D World', '3D').
154 game_graphics('Super Mario Run', '2D').
155 game_graphics('Super Mario Odyssey', '3D').
156 game_graphics('Super Mario Bros. Wonder', '2D').
157
158 % Facts: Main characters in the Super Mario series
159 main_character('Mario').
160 main_character('Luigi').
161 main_character('Princess Peach').
162 main_character('Princess Daisy').
163 main_character('Bowser').
164
165 % Facts: Protagonist characters in the Super Mario series
166 protagonist('Mario').
167 protagonist('Luigi').
168 protagonist('Princess Peach').
169 protagonist('Princess Daisy').
170
171 % Facts: Antagonist character in the Super Mario series
172 antagonist('Bowser').
173
174 % Facts: Supporting characters in the Super Mario series
175 supporting_character('Kings').
176 supporting_character('Hoot').
177 supporting_character('Dorrie').
178 supporting_character('Toadsworth').
179 supporting_character('Baby Luma').
180 supporting_character('Lubba').
181 supporting_character('FLUDD').
182 supporting_character('Ray').
183 supporting_character('Fluzzard').
184 supporting_character('Cosmic Spirit').
185 supporting_character('Cappy').
186

```

```

187
188 % Rules:
189
190 % Rule: Character appears in a 3D Super Mario game
191 appears_in_3d_game(X, Y) :- appears_in_game(X, Y), game_graphics(Y, '3D').
192 % Rule: Character appears in a 2D Super Mario game
193 appears_in_2d_game(X, Y) :- appears_in_game(X, Y), game_graphics(Y, '2D').
194 % Rule: Character appears in a Super Mario game released on a specific
    platform
195 appears_in_game_on_platform(X, Y) :- appears_in_game(X, Z),
    game_on_platform(Z, Y).
196 % Rule: Two characters appear in the same Super Mario game
197 characters_in_one_game(X, Y) :- appears_in_game(X, Z), appears_in_game(Y,
    Z), X \= Y.
198 % Rule: Main character appears in a Super Mario game
199 main_character_in_game(X, Y) :- appears_in_game(X, Y), main_character(X).
200 % Rule: Games that were released in the same year
201 games_that_realise_in_same_year(X, Y) :- release_year(X, Z),
    release_year(Y, Z), X \= Y.

```

### Тестирование и отладка системы, обеспечение ее функциональности и эффективности.

```

1 % Queries: Example queries to retrieve information
2 ?-appears_in_game('Luigi', X).
3 ?-protagonist('Bowser').
4 ?-appears_in_game_on_platform('Mario', 'Nintendo Switch').
5 ?-release_year(X, Y), Y > 2020.
6 ?-appears_in_3d_game('Mario', X).
7 ?-(appears_in_game('Hoot', X); appears_in_game('Dorrie', X)),
    (game_on_platform(X, 'Nintendo 64')), game_graphics(X, '3D').

```

```

?- appears_in_game('Luigi', X).
X = 'Super Mario Bros.' ;
X = 'Super Mario Bros. 2' ;
X = 'Super Mario Bros. 3' ;
X = 'Super Mario World' ;
X = 'Super Mario 64' ;
X = 'New Super Mario Bros. Wii' ;
X = 'Super Mario 3D Land' ;
X = 'New Super Mario Bros. 2' ;
X = 'New Super Mario Bros. U' ;
X = 'Super Mario 3D World' ;
X = 'Super Mario Bros. Wonder'.

```

```

?- protagonist('Bowser').
false.

```

```

?- appears_in_game_on_platform('Mario', 'Nintendo Switch').
true

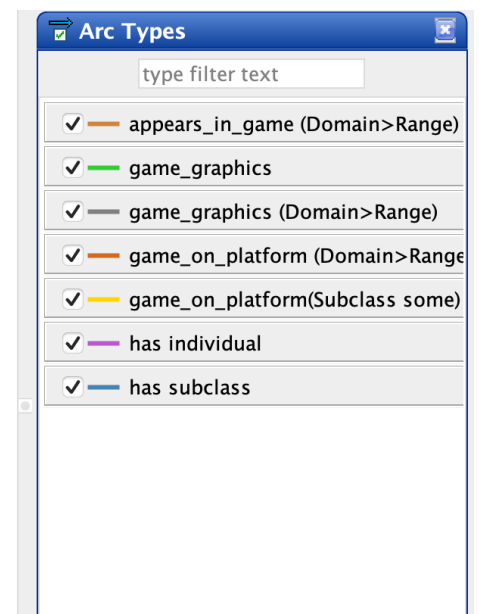
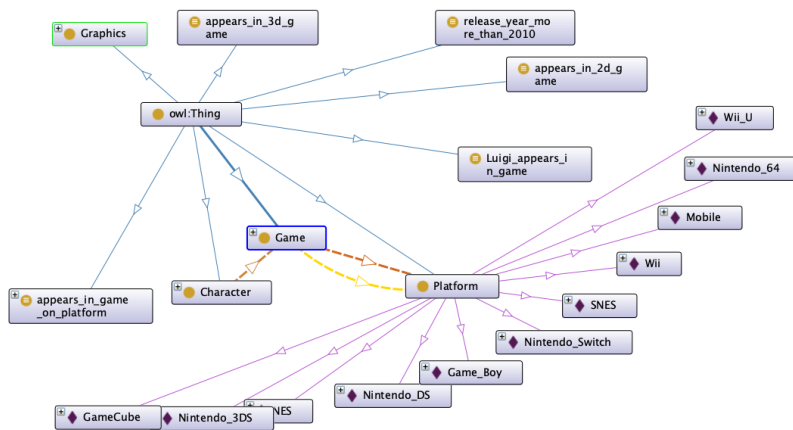
```

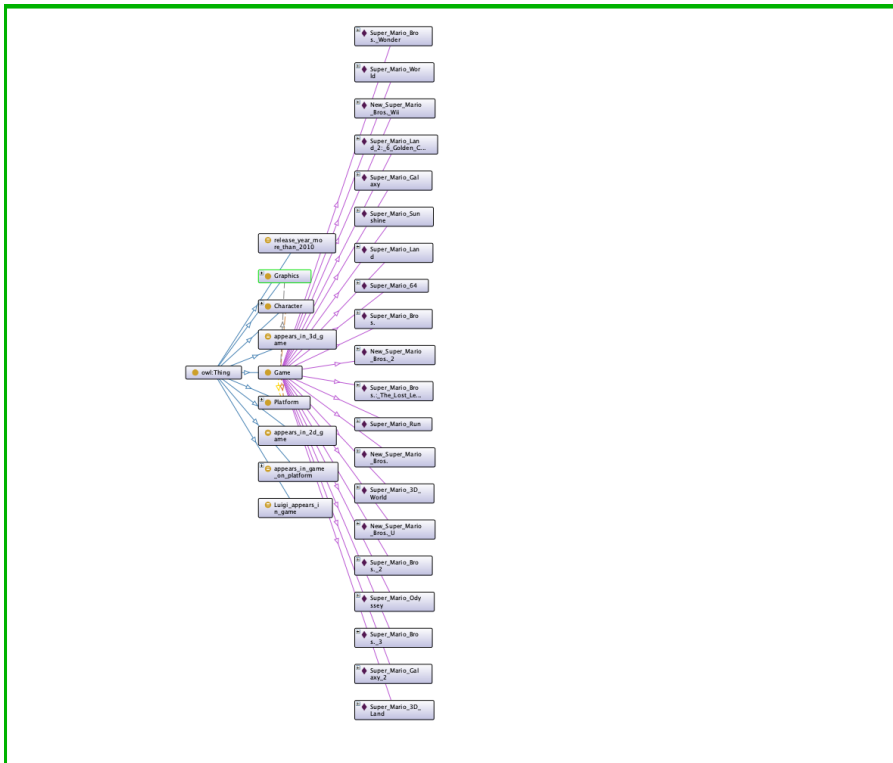
```
?- release_year(X, Y), Y > 2020.
X = 'Super Mario Bros. Wonder',
Y = 2023.
```

```
?- appears_in_3d_game('Mario', X).
X = 'Super Mario 64' ;
X = 'Super Mario Sunshine' ;
X = 'Super Mario Galaxy' ;
X = 'Super Mario Galaxy 2' ;
X = 'Super Mario 3D Land' ;
X = 'Super Mario 3D World' ;
X = 'Super Mario Odyssey' ;
false.
```

```
?- (appears_in_game('Hoot', X); appears_in_game('Dorrie', X)), (game_on_platform(X, 'Nintendo 64')), game_graphics(X, '3D').
X = 'Super Mario 64' ■
```

## Онтология:

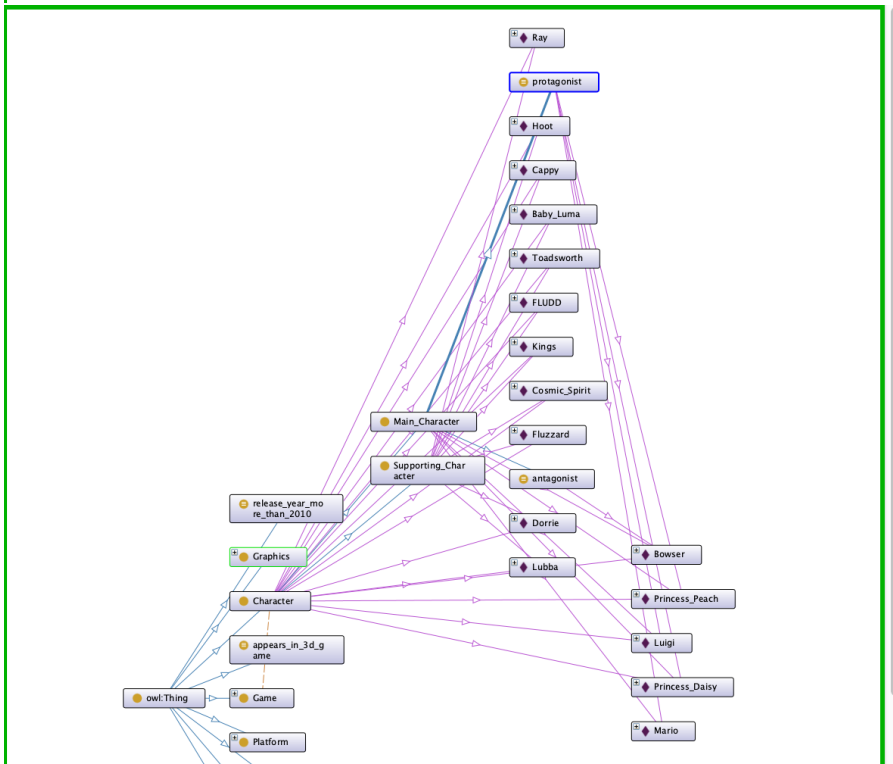




### Arc Types

type filter text

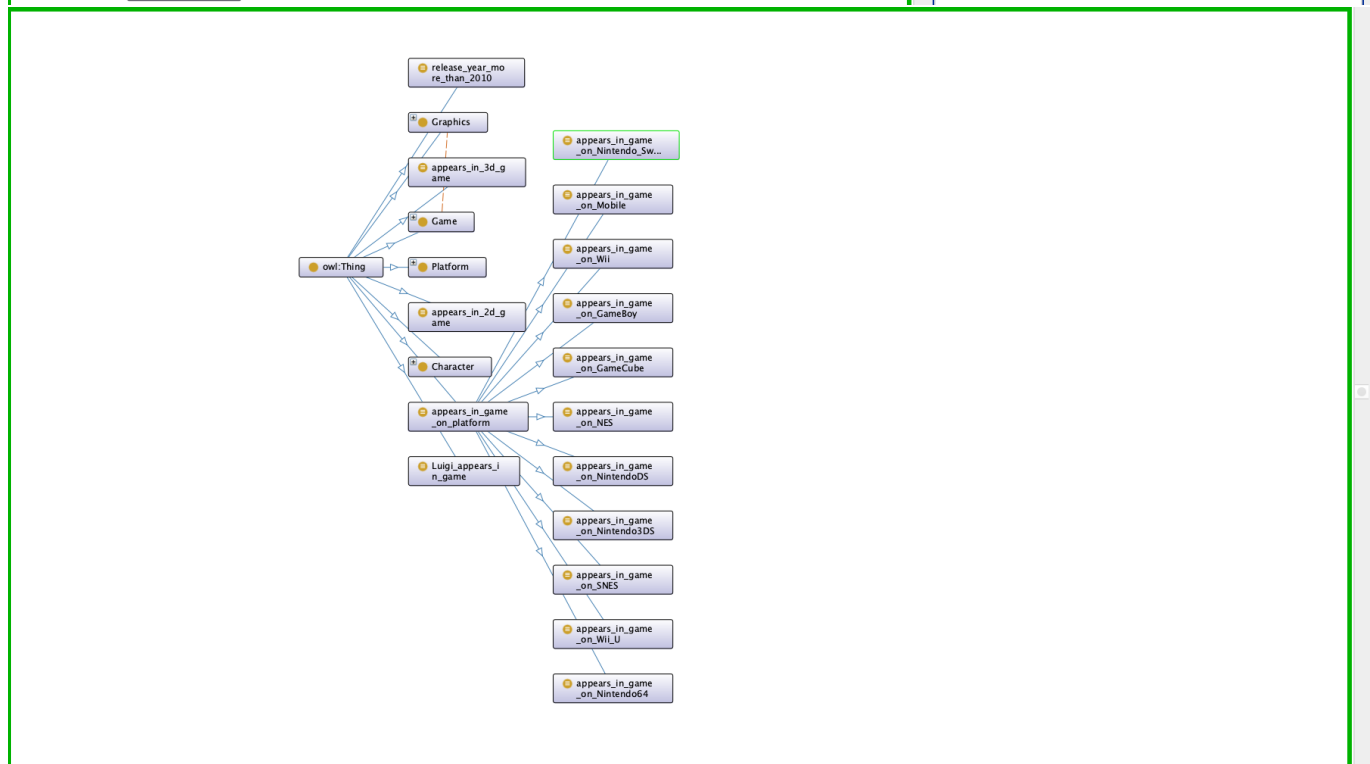
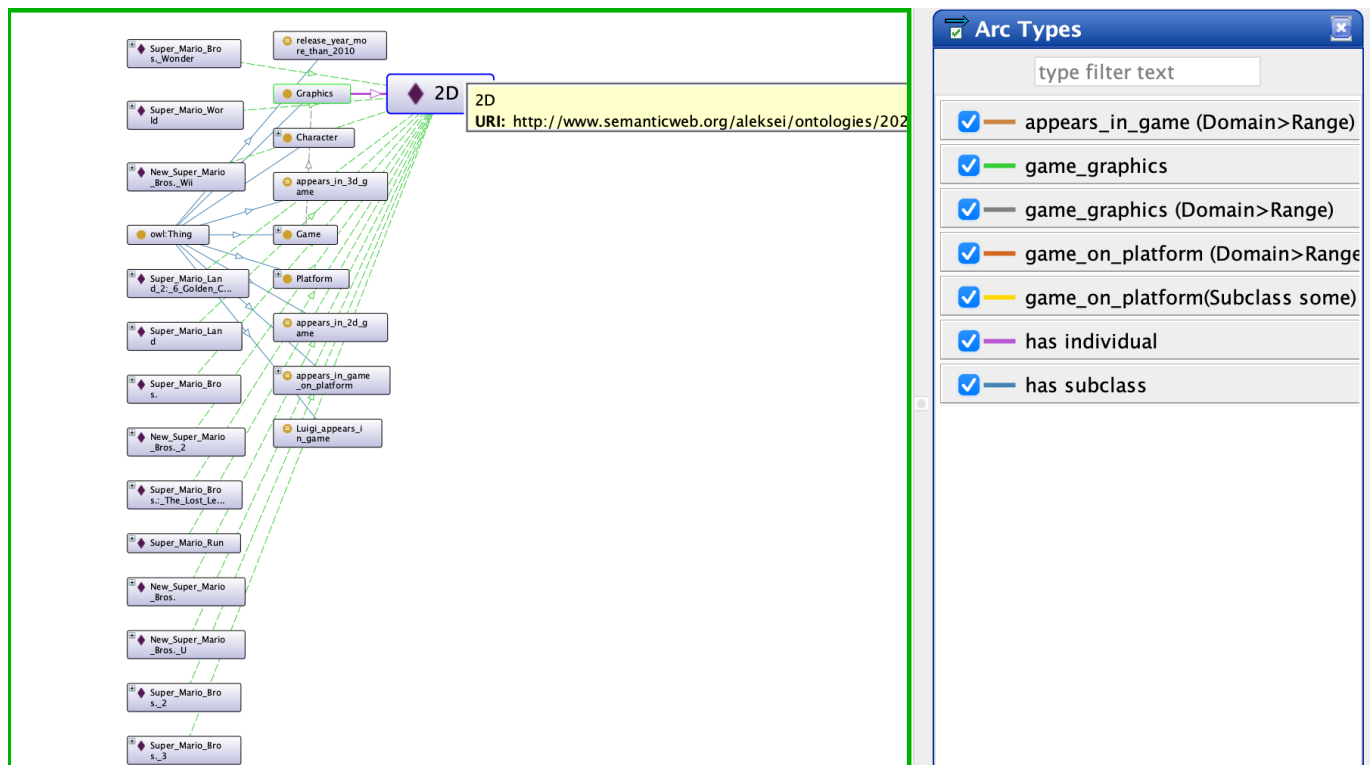
- ☒ appears\_in\_game (Domain>Range)
- ☒ game\_graphics
- ☒ game\_graphics (Domain>Range)
- ☒ game\_on\_platform (Domain>Range)
- ☒ game\_on\_platform(Subclass some)
- ☒ has individual
- ☒ has subclass



### Arc Types

type filter text

- ☒ appears\_in\_game (Domain>Range)
- ☒ game\_graphics
- ☒ game\_graphics (Domain>Range)
- ☒ game\_on\_platform (Domain>Range)
- ☒ game\_on\_platform(Subclass some)
- ☒ has individual
- ☒ has subclass

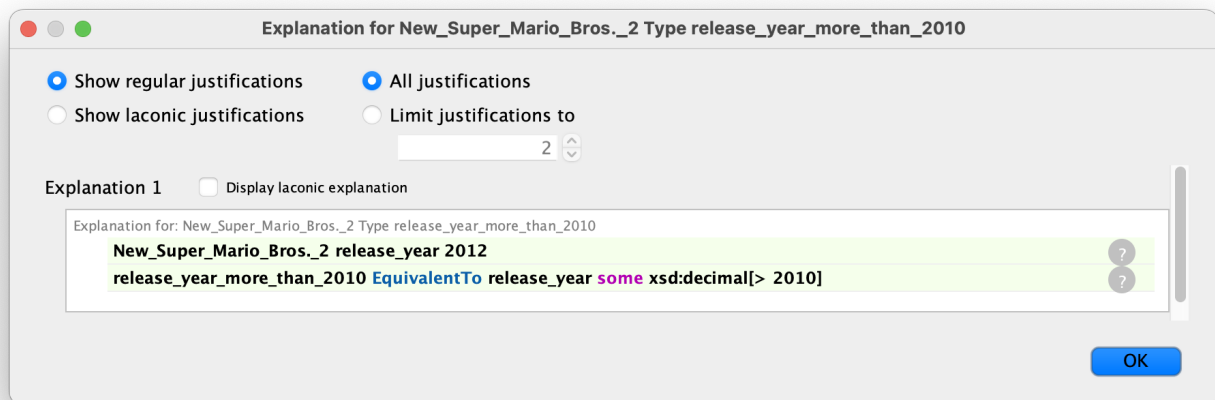
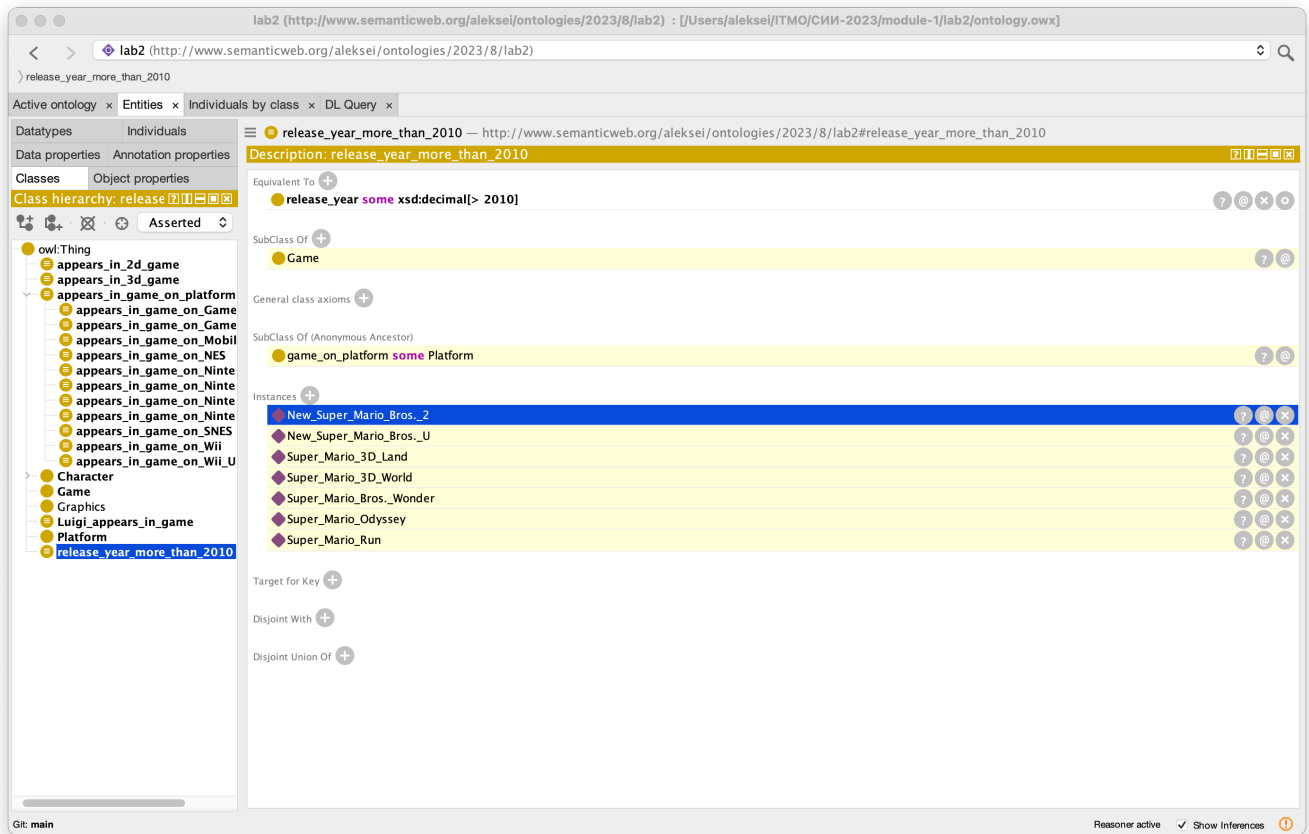


Тестирование и отладка системы, обеспечение ее функциональности и эффективности.









Система поддержки принятия решения на основе базы знаний:

```

1 package org.lapin;
2 import org.jpl7.*;
3 import org.lapin.parser.Parser;
4
5 import java.util.Scanner;
6
7 public class Main {
8     public static void main(String[] args) {

```

```

9
10 System.out.println("Hello and welcome!");
11 Scanner scanner = new Scanner(System.in);
12 System.out.println("Enter your query: ");
13 String query = scanner.nextLine();
14 Query q1 =
15     new Query(
16         "consult",
17         new Term[] {new
18             Atom("/Users/aleksei/ITMO/СММ-2023/module-1/new-lab-1/m
19
20     );
21 if (q1.hasSolution()) {
22
23     String res = Parser.run(query);
24     Query q3 = new Query(res);
25     var solutions = q3.allSolutions();
26     System.out.println("Вам подойдут игры: ");
27     for (int i = 0; i < solutions.length; i++) {
28         System.out.println(solutions[i].get("X"));
29     }
30 } else {
31     System.out.println("Connect to Prolog Failed");
32 }

```

```

1 package org.lapin.parser;
2
3 public class Parser {
4     public static String run(String str){
5         String queryResult = "";
6         // Splitting the input string on minus sign. Assumes the string
7         // contains multiple queries
8         var queries = str.split("\\s*~\\s*");
9         for(int queryIndex = 0; queryIndex < queries.length; queryIndex++){
10             if(queries[queryIndex].contains("персонаж")){
11                 queries[queryIndex] = queries[queryIndex]
12                     .substring(queries[queryIndex]
13                         .indexOf(":")+1).trim();
14                 // split the query into smaller components based on 'и'
15                 var components = queries[queryIndex].split("\\s*и\\s*");
16                 queryResult += "(";
17                 for (int componentIndex = 0; componentIndex <
18                     components.length; componentIndex++){
19                     queryResult += "(";
20                     // split the components based on ','
21                     var items = components[componentIndex].split(",\\s*");
22                     for (int itemIndex = 0; itemIndex < items.length;
23                         itemIndex++) {
24                         String item = "," + items[itemIndex].trim() + ",";
25                         item = item.replaceAll("\\s*\\.\\s*", "");

```

```

23         queryResult += "appears_in_game(" + item + ", X)";
24         if(itemIndex != items.length-1){
25             // add semicolon for multiple items in the
26             // same component
27             queryResult += "; ";
28         }
29         queryResult += ") ";
30         if(componentIndex != components.length-1){
31             // add comma for multiple components
32             queryResult += ", ";
33         }
34     }
35     queryResult += ") ";
36 }
37 if(queries[queryIndex].contains("приставка")){
38     queries[queryIndex] = queries[queryIndex]
39         .substring(queries[queryIndex]
40             .indexOf(":")+1).trim();
41     var components = queries[queryIndex].split("\s*и\s*");
42     queryResult += "(";
43     for (int componentIndex = 0; componentIndex <
44         components.length; componentIndex++){
45         queryResult += "(";
46         var items = components[componentIndex].split(",\s*");
47         for (int itemIndex = 0; itemIndex < items.length;
48             itemIndex++) {
49             String item = "'" + items[itemIndex].trim() + "'";
50             item = item.replaceAll("\\s*\\.\\s*", "");
51             queryResult += "game_on_platform( X," + item + ")";
52             if(itemIndex != items.length-1){
53                 queryResult += "; ";
54             }
55         }
56         queryResult += ") ";
57         if(componentIndex != components.length-1){
58             queryResult += ", ";
59         }
60     }
61     queryResult += ") ";
62 }
63 if(queries[queryIndex].contains("графикой")){
64     queries[queryIndex] = queries[queryIndex]
65         .substring(queries[queryIndex]
66             .indexOf(":")+1).trim();
67     var components = queries[queryIndex].split("\s*и\s*");
68     queryResult += "(";
69     for (int componentIndex = 0; componentIndex <
70         components.length; componentIndex++){
71         queryResult += "(";
72         var items = components[componentIndex].split(",\s*");

```

```

70         for (int itemIndex = 0; itemIndex < items.length;
71             itemIndex++) {
72             String item = "'" + items[itemIndex].trim() + "'";
73             item = item.replaceAll("\\s*\\.\\s*", "");
74             queryResult += "game_graphics( X, " + item + ")";
75             if(itemIndex != items.length-1){
76                 queryResult += "; ";
77             }
78             queryResult += ") ";
79             if(componentIndex != components.length-1){
80                 queryResult += ", ";
81             }
82         }
83         queryResult += ") ";
84     }
85     if(queryIndex != queries.length-1){
86         queryResult += ", ";
87     }
88 }
89 //      System.out.println(queryResult);
90 return queryResult;
91 }
92
93 }

```

**Тестирование и отладка системы, обеспечение ее функциональности и эффективности.**

Hello and welcome!

Enter your query:

Мой любимый персонаж: Mario, Luigi - у меня есть приставка: NES, Nintendo DS  
- я люблю играть в игры с графикой: 2D, 3D.

Вам подойдут игры:

'Super Mario Bros.'  
'Super Mario Bros.: The Lost Levels'  
'Super Mario Bros. 2'  
'Super Mario Bros. 3'  
'New Super Mario Bros.'  
'Super Mario Bros.'  
'Super Mario Bros. 2'  
'Super Mario Bros. 3'

Hello and welcome!

Enter your query:

Я люблю играть в игры с графикой: 3D - Мой любимый персонаж: Mario, Luigi  
и Princess Peach

Вам подойдут игры:

'Super Mario 64'  
'Super Mario 64'

'Super Mario Sunshine'  
'Super Mario Galaxy'  
'Super Mario Galaxy 2'  
'Super Mario 3D Land'  
'Super Mario 3D Land'  
'Super Mario 3D World'  
'Super Mario 3D World'  
'Super Mario Odyssey'

## 5 Оценка и интерпретация результатов:

### 5.1 Примеры запросов для БЗ и онтологии, сравнение разницы реализации.

Запрос к базе знаний:

```
?- release_year(X, Y), Y > 2020.  
X = 'Super Mario Bros. Wonder',  
Y = 2023.
```

Запрос к онтологии:

lab2 (http://www.semanticweb.org/aleksei/ontologies/2023/8/lab2) : [Users/aleksei/ITMO/СИИ-2023/module-1/lab2/ontology.owlx]

lab2 (http://www.semanticweb.org/aleksei/ontologies/2023/8/lab2)

release\_year\_more\_than\_2010

Active ontology x Entities x Individuals by class x DL Query x

Data types Individuals

Data properties Annotation properties

Classes Object properties

Class hierarchy: release\_year\_more\_than\_2010

owl:Thing

- appears\_in\_2d\_game
- appears\_in\_3d\_game
- appears\_in\_game\_on\_platform
- appears\_in\_game\_on\_Game
- appears\_in\_game\_on\_Mobil
- appears\_in\_game\_on\_NES
- appears\_in\_game\_on\_Ninte
- appears\_in\_game\_on\_Ninte
- appears\_in\_game\_on\_SNES
- appears\_in\_game\_on\_Wii\_U
- Character
- Game
- Graphics
- Luigi\_appears\_in\_game
- Platform
- release\_year\_more\_than\_2010

release\_year\_more\_than\_2010 — http://www.semanticweb.org/aleksei/ontologies/2023/8/lab2#release\_year\_more\_than\_2010

Description: release\_year\_more\_than\_2010

Equivalent To

- release\_year xsd:decimal(> 2010)

SubClass Of

- Game

General class axioms

SubClass Of (Anonymous Ancestor)

- game\_on\_platform some Platform

Instances

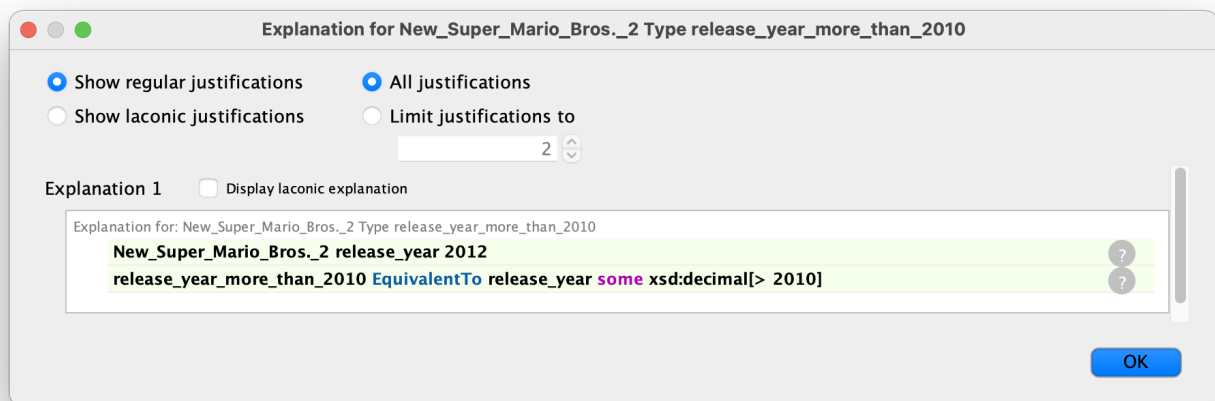
- New\_Super\_Mario\_Bros.\_2
- New\_Super\_Mario\_Bros.\_U
- Super\_Mario\_3D\_Land
- Super\_Mario\_3D\_World
- Super\_Mario\_Bros.\_Wonder
- Super\_Mario\_Odyssey
- Super\_Mario\_Run

Target for Key

Disjoint With

Disjoint Union Of

Git: main Reasoner active Show Inferences



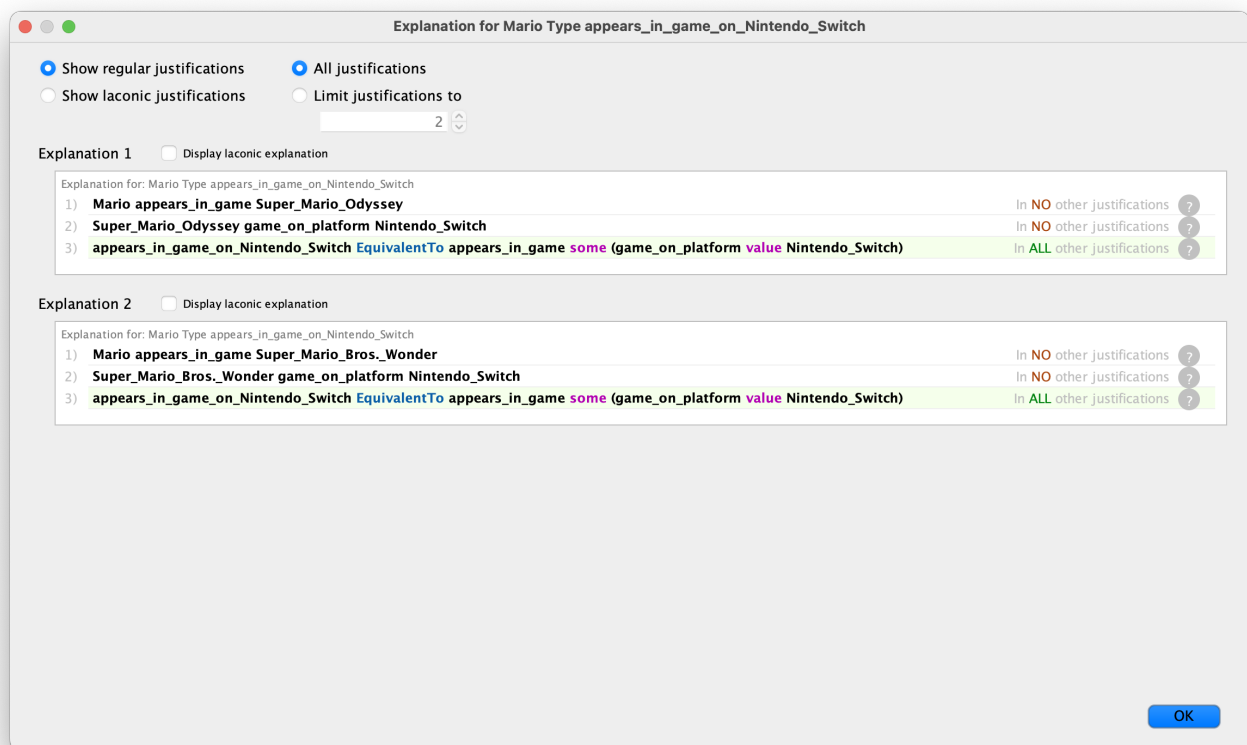
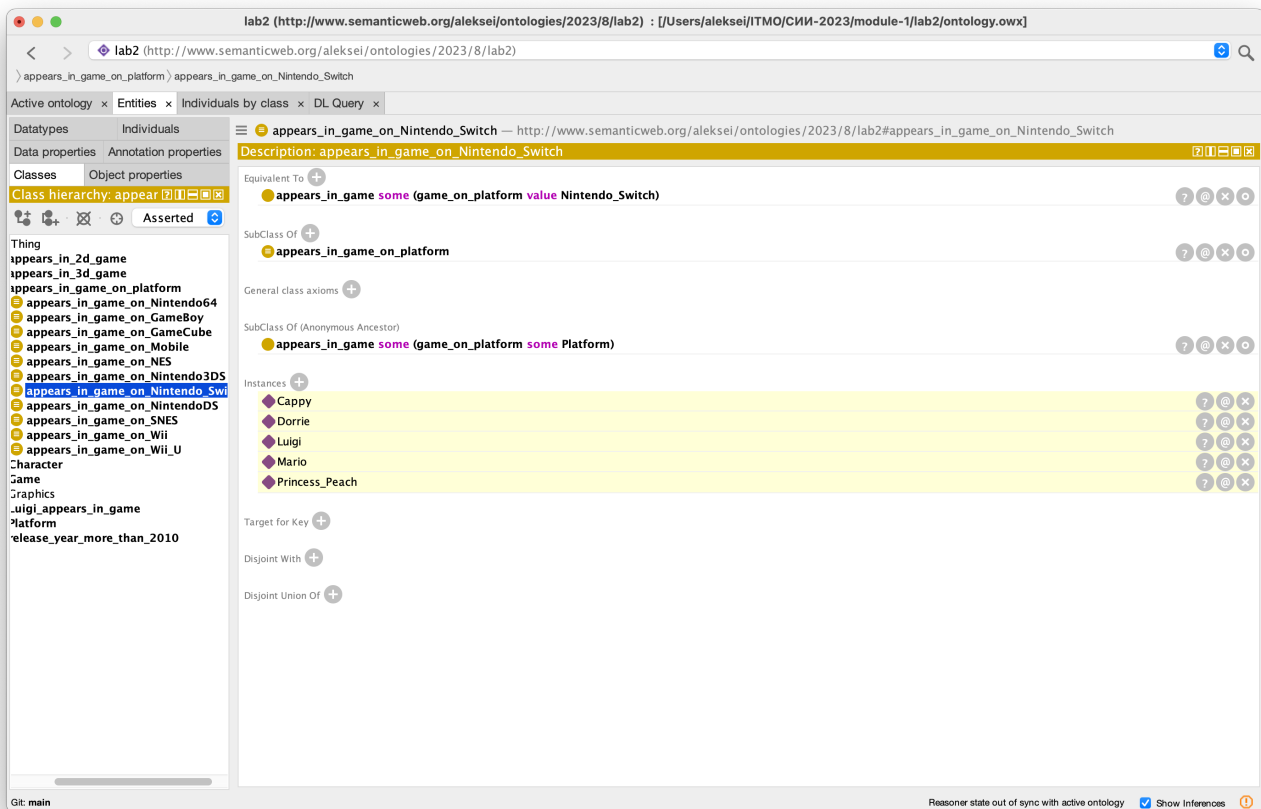
Реализация запроса через факт в Prolog и свойства в Protege похожи.

### Запрос к базе знаний:

```
% Rule: Character appears in a Super Mario game released on a specific platform
appears_in_game_on_platform(X, Y) :- appears_in_game(X, Z), game_on_platform(Z, Y).
```

```
?- appears_in_game_on_platform('Mario', 'Nintendo Switch').
true
```

## Запрос к онтологии:



Реализация запроса через правило в Prolog, отличается в удобстве использования от реализации

через отдельные классы эквивалентные определенным правилам для каждой платформы.

## **5.2 Оценка соответствия системы поставленным требованиям и достижению целей проекта.**

Система способна обрабатывать запросы пользователей, связанные с играми серии Super Mario, и выдавать релевантные рекомендации. Была реализована база знаний на языке Prolog и онтология на языке OWL с помощью Protege, которые позволяют хранить и структурировать информацию о персонажах, играх и платформах серии Super Mario.

Также была реализована система поддержки принятия решения на основе базы знаний, которая позволяет находить игры, соответствующие запросам пользователей. Были проведены тесты системы, которые показали ее эффективность и корректность.

Таким образом, система полностью соответствует поставленным требованиям и целям проекта, и может быть использована для обработки запросов пользователей, связанных с играми серии Super Mario.

Система поддержки принятия решений на основе базы знаний соответствует основным требованиям, таким как:

- Принимает запросы пользователя через командную строку и обрабатывает их с помощью парсера.
- Формулирует логические запросы к базе знаний Prolog, используя введенные пользователем данные.
- Предоставляет рекомендации, советы, связанные с выбором видеоигры, на основе результатов выполнения запросов.
- Есть документация работы системы и описание ее функциональности.

Система достигает целей проекта, таких как:

- Развитие навыков логического программирования и создания базы знаний на Prolog
- Развитие навыков работы с онтологиями и семантическими технологиями
- Развитие навыков разработки систем поддержки принятия решений, использующих базу знаний или онтологию для предоставления рекомендаций
- Практическое применение системы в помощи пользователям при выборе видеоигр

## **5.3 Интерпретация результатов и описание дальнейших возможностей развития и улучшения системы.**

В результате разработки системы поддержки принятия решений на основе базы знаний и онтологии для выбора видеоигр серии Super Mario были достигнуты поставленные цели и требования проекта. Система успешно обрабатывает запросы пользователей и предоставляет рекомендации, соответствующие их запросам.

Однако, существуют возможности для дальнейшего улучшения и развития системы. Например, можно добавить больше информации о персонажах, играх и платформах серии Super Mario, чтобы система могла предоставлять более точные и полезные рекомендации. Также



можно улучшить парсер запросов, чтобы система могла обрабатывать более сложные запросы и предоставлять более точные результаты.

Кроме того, можно рассмотреть возможность интеграции системы с другими сервисами и платформами, такими как онлайн-магазины игр, чтобы пользователи могли сразу же приобрести рекомендованные игры. Также можно рассмотреть возможность добавления функционала для сравнения игр и выбора наиболее подходящей для пользователя.

В целом, система поддержки принятия решений на основе базы знаний и онтологии для выбора видеоигр серии Super Mario является успешным проектом, который может быть дальше улучшен и развит для удовлетворения потребностей пользователей.

## **6 Заключение:**

### **6.1 Описание преимуществ и потенциальных применений разработанной системы искусственного интеллекта на базе Prolog, баз знаний и онтологий.**

#### **Преимущества системы:**

- Система позволяет пользователю получать рекомендации по выбору видеоигр на основе своих интересов и предпочтений, используя логический язык Prolog и базу знаний, содержащую информацию о различных играх серии Super Mario.
- Система использует декларативный подход к представлению знаний, что упрощает их описание и обновление. Система также способна проводить логический вывод и унификацию для поиска решения.
- Система демонстрирует возможности Prolog и семантических технологий для разработки систем искусственного интеллекта, таких как экспертные системы, рекомендательные системы, системы обработки естественного языка и другие
- Система имеет практическое значение для пользователей, которые хотят найти подходящую видеоигру из серии Super Mario

#### **Потенциальные применения системы:**

- Система может быть использована для помощи пользователям в выборе видеоигр не только из серии Super Mario, но и из других жанров и франшиз, расширяя базу знаний или онтологию соответствующими данными.
- Система может быть интегрирована с другими приложениями и сервисами, такими как веб-сайты, мобильные приложения, голосовые ассистенты и социальные сети.