

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Факультет інтелектуальних інформаційних технологій та автоматизації

(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних наук

(повна назва кафедри (предметної, циклової комісії))

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Інформаційна технологія онтологічного моделювання бази

знань гри «Dota 2»»

Виконав: студент 2-го курсу, групи 1КН-23м
спеціальності 122 «Комп'ютерні науки»

(шифр і назва напрямку підготовки, спеціальності)

Струнь В. А.

(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН

Сілагін О. В.

(прізвище та ініціали)

« 05 » 12 2024 р.

Опонент: к.т.н., доцент каф. АПТ

Гармаш В. В.

(прізвище та ініціали)

« 05 » 12 2024 р.

Допущено до захисту

Завідувач кафедри КН

д.т.н., проф. Яровий А.А.

(прізвище та ініціали)

« 06 » 12 2024 р.

Вінниця ВНТУ - 2024 рік

Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра Комп'ютерних наук
Рівень вищої освіти другий (магістерський)
Галузь знань – 12 «Інформаційні технології»
Спеціальність – 122 «Комп'ютерні науки»
Освітньо-професійна програма – «системи штучного інтелекту»

ЗАТВЕРДЖУЮ

Завідувач кафедри КН
проф., д.т.н. Яровий А.А.

«11» 10 2024 року

**ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ**

Струню Владиславу Андрійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: «Інформаційна технологія онтологічного моделювання бази знань гри «Dota 2»».

Керівник роботи: Сілагін Олексій Віталійович, к.т.н., доцент.

затверджені наказом вищого навчального закладу від «17» 2024 року № 310

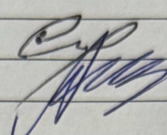
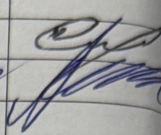
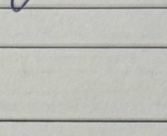
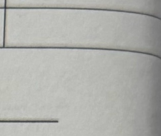
2. Термін подання студентом роботи 11. 11. 2024

3. Вихідні дані до роботи: об'єктно-орієнтована парадигма програмування, об'єктно-орієнтована база знань, можливість формування SPARQL-запитів, кількість основних класів – не менше 5 одиниць; кількість підкласів – не менше 10 одиниць; кількість індивідів – не менше 20 одиниць, ступінь ієрархії – не менше 2 гілки.

4. Зміст текстової частини: вступ, обґрунтування доцільності розробки інформаційної технології онтологічного моделювання бази знань гри «Dota 2», моделювання інформаційної інформаційної технології онтологічного моделювання бази знань гри «Dota 2», програмна реалізація інформаційної технології онтологічного моделювання бази знань гри «Dota 2», економічна частина, висновки, список використаних джерел, додатки.

5. Перелік ілюстративного матеріалу (з точним зазначенням обов'язкових креслень):
ER-модель предметної галузі гри «Dota 2»; UML-діаграма з предметної галузі гри «Dota 2», діаграма прецедентів роботи онтологічної системи, частина графу онтології гри «Dota 2»

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-3	Сілагін О.В., к.т.н., доц. каф. КН		
4	Лесько О. Й., к.т.н., проф. каф. ЕПВМ		

7. Дата видачі завдання 02.09.2024р.

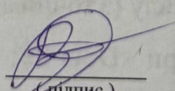
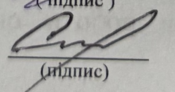
КАЛЕНДАРНИЙ ПЛАН

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітки
1	Аналіз предметної галузі. Огляд існуючих систем аналогів. Постановка задач дослідження	02.09.24 - 16.09.24	
2	Розробка покращеної інформаційної технології онтологічного моделювання бази знань	17.09.24 - 27.09.24	
3	Моделювання інформаційної технології онтологічного моделювання бази знань	28.09.24 - 05.10.24	
4	Підготовка економічної частини	06.10.24 - 24.10.24	
5	Апробація та/або впровадження результатів дослідження	25.10.24 - 08.11.24	
6	Оформлення пояснювальної записки, графічного матеріалу та презентації	09.11.24 - 11.11.24	

Студент

Керівник роботи


(підпис)

(підпис)

Струнь В. А.
(прізвище та ініціали)

Сілагін О.В.
(прізвище та ініціали)

АНОТАЦІЯ

УДК 004.8

Струнь В. А. Інформаційна технологія онтологічного моделювання бази знань гри «Dota 2». Магістерська кваліфікаційна робота зі спеціальності 122 – комп'ютерні науки, освітня програма - комп'ютерні науки. Вінниця: ВНТУ, 2024. 114 с.

На укр. мові. Бібліогр.: 35 назв; рис.: 29; табл. 12.

У магістерській кваліфікаційній роботі проведено аналіз предметної галузі гри «Dota 2». Досліджено основні підходи щодо структуризації знань. Виконано аналіз аналогів, розглянуто методи та технології, які використовуються для вирішення подібних задач та обґрунтована доцільність розробки. Розроблено структурну модель інформаційної технології, а також розроблено онтологічну модель бази знань гри «Dota 2», яка сприяє структуризації знань в цій галузі та спрощує роботу користувачів. Така онтологія дає можливість користувачам поліпшити свої знання про гру «Dota 2» на основі побудованих запитів. В роботі подано результати досліджень та аналіз роботи розробленої інформаційної технології онтологічного моделювання бази знань гри «Dota 2».

Графічна частина складається з 10 плакатів.

У економічному розділі розраховано суму витрат на виконання та впровадження результатів роботи складає 87470,7 грн, спрогнозовано орієнтовану величину витрат по кожній з статей витрат, розраховано чистий прибуток, термін окупності витрат для виробника 1,9 роки та економічний ефект для споживача при використанні даної розробки.

Ключові слова: онтологія, гра, «Dota 2», герої, класи героїв, предмети, база знань, Protégé, класи.

ABSTRACT

Strun V. A. Information technology for modeling the ontological base of the game “Dota 2”. Master's qualification work in specialty 122 - Computer Science, educational program - Artificial Intelligence Systems. Vinnytsia: VNTU, 2024, 114 p.

In Ukrainian language. Bibliographer: 35 titles; fig .: 29; table 12.

The master's thesis analyzes the subject area of the game “Dota 2”. The main approaches to structuring knowledge are investigated. An analysis of analogues is carried out, methods and technologies used to solve similar problems are considered, and the feasibility of development is substantiated. A structural model of information technology has been developed and an ontological model of the knowledge base of the game “Dota 2” has been developed, which helps to structure knowledge in this area and simplifies the work of users. Such an ontology allows users to improve their knowledge of the game “Dota 2” based on the built queries. The article presents the results of the study and analysis of the developed information technology for ontological modeling of the knowledge base of the game “Dota 2”.

The graphic part consists of 10 posters.

The economic section calculates the amount of costs for the implementation and realization of the results of the work, which is 87470,7 UAH, predicts the estimated cost for each of the cost items, calculates the net profit, the payback period for the manufacturer of 1.9 years and the economic effect for the consumer when using this development.

Keywords: ontology, game, "Dota 2", heroes, hero classes, items, knowledge base, Protégé, classes.

ЗМІСТ

ВСТУП.....	4
1 ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ БАЗИ ЗНАНЬ ГРИ «DOTA 2».....	7
1.1 Аналіз предметної галузі гра «Dota 2».....	7
1.2 Огляд існуючих систем аналогів.....	19
1.3 Постановка задач дослідження	23
1.4 Висновок до розділу 1	25
2 РОЗРОБКА ПОКРАЩЕНОЇ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ТА ОНТОЛОГІЧНЕ МОДЕЛЮВАННЯ БАЗИ ЗНАНЬ ГРИ «DOTA 2».....	26
2.1 Обґрунтування вибору методів реалізації інформаційної технології.....	26
2.2 Розробка покращеної інформаційної технології онтологічного моделювання гри «Dota 2»	30
2.3 Моделювання бази знань засобами технологій ER та UML	33
2.4 Розробка онтологічної моделі бази знань гри «Dota 2»	39
2.5 Реалізація онтологічної моделі бази знань гри «Dota 2».....	42
2.6 Висновок до розділу 2	46
3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ОНТОЛОГІЧНОГО МОДЕЛЮВАННЯ БАЗИ ЗНАНЬ ГРИ «DOTA 2».....	47
3.1 Програмні засоби для реалізації інформаційної технології онтологічного моделювання бази знань з гри «Dota 2».....	47
3.2 Реалізація онтологічної бази знань з гри «Dota 2»	51
3.3 Написання стандартних SPARQL запитів для пошуку інформації	62
3.4 Тестування онтологічної бази знань з гри «Dota 2»	65
3.5 Висновок до розділу 3	70
4 ЕКОНОМІЧНА ЧАСТИНА.....	71
4.1 Оцінювання комерційного потенціалу розробки	71
4.2 Прогнозування витрат на виконання науково-дослідної роботи.....	74
4.3 Розрахунок економічної ефективності науково-технічної розробки.....	79
4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності	81

4.5 Висновок до розділу 4.....	84
ВИСНОВКИ.....	85
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	87
Додаток А (обов'язковий) Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень	91
Додаток Б (обов'язковий) Лістинг програми	92
Додаток В (обов'язковий) Ілюстративна частина.....	107
Додаток Г (довідниковий) Інструкція користувача.....	113

ВСТУП

Актуальність теми. За останні кілька років відеоігри перетворилися з нішевого хобі на масову індустрію з річним доходом, що вимірюється сотнями мільярдів доларів.

Світовий ринок відеоігор постійно зростає, забираючи прибутки та час реклами, що раніше припадали на платне телебачення, відеостримінгові сервіси (Netflix, Hulu, HBO Max, Amazon Prime Video) та кінематографію. Грає у відеоігри понад третину земного населення, а кількість власників ігрових консолей у віці 35-44 років перевищує кількість молодих людей віком 16-24 років. Тому компанії витрачають сотні мільйонів доларів на розробку ігор, збираючи з ринку не менші доходи. Проте, схоже, це тільки початок.

Ринок відеоігор завойовує індустрію розваг і за останні роки обігнав традиційні ринки дозвілля людей: книжковий, музичний та кінематографічний. У 2024 році дохід ігрової індустрії може сягнути 187,7 млрд дол. У 2023 році цей показник становив 156 млрд дол. Щорічно він зростає в середньому на 20 млрд доларів.

Одним з прикладів є компанія Valve та її культова гра «Dota 2» яка набрала свою першу популярність після турніру The International у 2011 році, де перемогла українська кіберспортивна організація NAVI, з головним призом 1 мільйон доларів, що значно перевершувало призові в усіх інших кіберспортивних дисциплінах. Це дало значний поштовх для розвитку гри в яку сьогодні щоденно грає більше 600тис. гравців. Найбільшим показником гравців за день є майже 1 300 000, що доказує, що цю гру грає вагома частка населення світу.

Навіть через 13 років після впуску бети та 11 років після релізу гри в ній з'являється багато нових гравців, але через те що ця гра є МОВА поріг входження в неї є досить високим. Тому розробка онтологічної бази знань гри «Dota 2» є досить актуальною темою щоб знизити цей поріг входження та було зручно дізнатись інформацію по ній. Наприклад : базові механіки гри, цілі гри, герої, їхні здібності, предмети, які потрібно купляти протягом гри та інше.

Отже, розробка інформаційної технології онтологічного моделювання бази знань гри «Dota 2», знизить поріг входження та спростить доступ до інформації по цій грі, як новачку так і уже досвідченому гравцю, є доцільною та має практичне значення. Така розробка може застосовуватись професійними гравцями, тренерами, звичайними гравцями або просто людьми, які хочуть спробувати її пограти, але через багато інформації та її складність може відштовхувати людей від запуску своєї першої гри з реальними гравцями у «Dota 2».

Зв'язок роботи з науковими програмами, планами, темами. Магістерська кваліфікаційна робота виконана відповідно до напрямку наукових досліджень кафедри комп'ютерних наук Вінницького національного технічного університету 22 К1 «Розробка прикладних інтелектуальних інформаційних технологій та систем».

Мета і завдання дослідження. Метою магістерської кваліфікаційної роботи є розширення функціональних можливостей інформаційної технології онтологічного моделювання бази знань гри «Dota 2», що сприятиме ефективному пошуку потрібної інформації.

Для досягнення поставленої мети необхідно вирішити такі завдання:

- проаналізувати предметну галузь гри «Dota 2» та обґрунтувати доцільність розробки інформаційної технології онтологічного моделювання бази знань;
- провести порівняння існуючих технологій;
- розробити покращену інформаційну технологію онтологічного моделювання бази знань;
- здійснити моделювання предметної галузі гри «Dota 2» засобами проектування та побудови діаграм, закодувати онтологію в інструментальному середовищі;
- розробити онтологічну модель бази знань гри «Dota 2»;
- розробити загальну структурну схему системи;
- здійснити програмну реалізацію онтологічної бази знань;
- виконати тестування програми та провести аналіз отриманих результатів;
- обґрунтувати економічну доцільність розробки.

Об'єктом дослідження – процес структуризації знань галузі «комп'ютерна гра «Dota 2»»

Предметом дослідження є програмні засоби та технології онтологічного моделювання бази знань гри «Dota 2».

Методи дослідження. Для досягнення поставленої мети було застосовано методи класифікації інформації та систематизації інформації, теорія про моделі представлення знань, методи узагальнення, моделювання та абстрагування, методи моделювання UML, інтелектуальний аналізу даних, методи розробки та аналізу онтологічних систем.

Наукова новизна одержаних результатів полягає у вдосконаленні інформаційної технології онтологічного моделювання бази знань гри «Dota 2». А також пропонується нова онтологічна модель бази знань гри «Dota 2». Використання цієї моделі сприяє структуризації знань в цій галузі та спрощує роботу користувачів.

Практичне значення одержаних результатів полягає у наступному:

1. Розроблено структуру онтологічної бази знань гри «Dota 2».
2. Програмно реалізовано онтологічну базу знань гри «Dota 2», надано можливості для її модифікації та розширення.

Достовірність теоретичних положень магістерської кваліфікаційної роботи підтверджується чіткою постановкою задач, коректним використанням різних методів для доведення наукових положень, строгим виведенням аналітичних співвідношень, порівнянням результатів з відомими та збіжністю результатів математичного моделювання з результатами, отриманими під час впровадження розроблених програмних засобів.

Особистий внесок здобувача. Усі результати магістерської кваліфікаційної роботи були отримані здобувачем самостійно. У тезах, опублікованих у співавторстві, здобувачу належать такі результати: обґрунтування доцільності розробки інформаційної технології онтологічного моделювання бази знань гри «Dota 2»[1].

1 ОБГРУНТУВАННЯ ДОЦІЛЬНОСТІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ БАЗИ ЗНАНЬ ГРИ «DOTA 2»

1.1 Аналіз предметної області гра «Dota 2»

Індустрія комп'ютерних ігор є однією з найдинамічніших та найприбутковіших галузей у світі розваг. Вона охоплює розробку, виробництво, маркетинг та продаж комп'ютерних і відеоігор. Розглянемо кілька ключових аспектів цієї індустрії[2]:

1. Історія та розвиток : індустрія комп'ютерних ігор бере свій початок у 1950-х роках, коли з'явилися перші прості комп'ютерні ігри. З часом технології вдосконалювались, і вже в 1970-х роках з'явилися перші аркадні ігри, такі як "Pong" та "Space Invaders". У 1980-х роках почали з'являтися домашні ігрові консолі, а 1990-ті роки стали епохою 3D-графіки та персональних комп'ютерів.

2. Розробка ігор : процес розробки ігор складається з кількох етапів: концепція, дизайн, програмування, тестування та випуск. Команди розробників можуть включати дизайнерів, програмістів, художників, звукоінженерів та тестувальників. Великі ігрові студії, такі як Ubisoft, Electronic Arts та Blizzard, часто мають сотні співробітників.

3. Платформи : ігри розробляються для різних платформ, включаючи персональні комп'ютери, ігрові консолі (наприклад, PlayStation, Xbox, Nintendo Switch) та мобільні пристрої. Кожна платформа має свої особливості та вимоги до розробки.

4. Жанри ігор : існує безліч жанрів комп'ютерних ігор, серед яких: шутери, рольові ігри (RPG), стратегії, симулятори, спортивні ігри та багато інших. Кожен жанр має свої унікальні характеристики та цільову аудиторію.

5. Економічний вплив : індустрія комп'ютерних ігор приносить мільярди доларів щорічно. Вона включає продажі ігор, підписки, внутрішньоігрові покупки та кіберспорт. Кіберспорт став глобальним феноменом, приваблюючи мільйони глядачів та значні інвестиції.

6. Українські розробники : Україна також має своїх успішних розробників у цій індустрії. Наприклад, компанія GSC Game World відома своєю серією ігор "S.T.A.L.K.E.R.", а студія 4A Games створила популярну серію "Metro".

Індустрія комп'ютерних ігор продовжує розвиватися, впроваджуючи нові технології, такі як віртуальна реальність (VR) та штучний інтелект, що обіцяє ще більше цікавих можливостей у майбутньому.

MOBA (Multiplayer Online Battle Arena)[2] – це жанр комп'ютерних ігор, який поєднує елементи стратегії в реальному часі (RTS) та рольових ігор (RPG). У МОБА-іграх дві команди гравців змагаються на обмеженій арені, кожен гравець керує одним персонажем з унікальними здібностями. Метою зазвичай є знищення головної структури команди суперника, водночас захищаючи свою власну. Деякі з найвідоміших ігор цього жанру – "Dota 2" і "League of Legends".

Основні характеристики МОБА-ігор

– Командна гра: У більшості МОБА-ігор беруть участь дві команди, зазвичай по п'ять гравців у кожній. Кожен гравець керує одним персонажем, який називається "герой" або "чемпіон".

– Унікальні персонажі: Кожен герой має унікальні здібності та роль у команді. Наприклад, деякі герої спеціалізуються на завданні шкоди (дд), тоді як інші можуть лікувати союзників або захищати їх від атак (танки та саппорти).

– Карти та об'єкти: Арена зазвичай поділена на три основні шляхи (лінії), якими рухаються мінйони (керовані комп'ютером створіння). Гравці повинні знищувати ворожих мінйонів і структури, щоб просуватися по карті.

– Рівні та предмети: Під час гри герої набирають досвід і підвищують свої рівні, отримуючи нові здібності. Вони також можуть купувати предмети, які покращують їхні характеристики та допомагають у бою.

– Тактичний геймплей: МОБА-ігри вимагають від гравців високого рівня стратегії та командної роботи. Успіх залежить від координації дій, вибору правильних моментів для атак і захисту, а також від управління ресурсами.

Однією з представників цього жанру є Dota 2[3], Defense of the Ancients. Розроблена компанією Valve, "Dota 2" є однією з найпопулярніших МОБА-ігор.

У Dota 2 грають 2 команди по 5 осіб. Обидві команди мають свою окрему базу у протилежних кутах карти, та захищають її під час гри. Кожен із десяти гравців незалежно керує могутнім персонажем, якого називають героєм, кожен з яких має унікальні здібності та різний стиль гри. Виграє команда, яка першою знищить центральну будівлю на базі ворога, для цього герої протягом гри намагаються вбити ворожих героїв та допоміжних істот (кріпів) і зруйнувати захисні ворожі будівлі, здобуваючи таким чином досвід та золото для свого розвитку. Якщо герой помирає, то відроджується через деякий час на базі.

Це була одна з ігор, в яку найчастіше грали в Steam з моменту її виходу, з більш ніж мільйоном одночасних гравців на піку. Популярність гри призвела до створення офіційних товарів і медіа-адаптацій для неї, включно з коміксами та аніме-серіалами, а також до рекламних зв'язків з іншими іграми та ЗМІ. Гра також дозволяє спільноті створювати власні ігрові режими, карти та косметику, які завантажуються в Майстерню Steam. Дві додаткові гри, Artifact і Dota Underlords, також були випущені Valve. Dota 2 також використовувалася в експериментах з машинним навчанням, коли команда ботів, відома як OpenAI Five, показала здатність перемагати професійних гравців.

Dota 2[4] — командна онлайн гра у жанрі МОБА. Ігровий процес будується навколо боротьби двох команд по п'ять героїв, кожен з яких контролюється одним із гравців. Кожна команда має свій табір: у правому верхньому куті мапи розташовується Темна сторона (The Dire), у лівому нижньому — Світла сторона (The Radiant).

Гра проходить у вигляді окремих партій, жодна з попередніх партій не впливає на майбутні. Для участі в партії гравець має розпочати пошук доступної гри. Коли на певному сервері набереться 10 гравців приблизно однакового рівня майстерності, партія розпочинається. На початку партії кожен гравець має вибрати одного з доступних героїв. Після того, як усі гравці оберуть собі героя, вони з'являються на карті й починається боротьба. Перемогою вважається знищення головної споруди на ворожій базі — Древнього (англ. Ancient, на ігровому сленгу — трон).

Герої[5] — основа гри Dota 2, саме їхні дії визначають результат партії. Кожен із героїв має особливі навички та здібності та дерево талантів, які визначають їх роль і застосування в грі. Для досягнення мети герої протягом партії здобувають досвід і золото, знищуючи ворогів, зокрема — кріпів — ігрових істот, що контролюються штучним інтелектом і діють за встановленим алгоритмом. Здобутий досвід дає можливість героям підвищити рівень розвитку, що дає їм додаткові вміння або посилює ті, якими вони вже володіють.

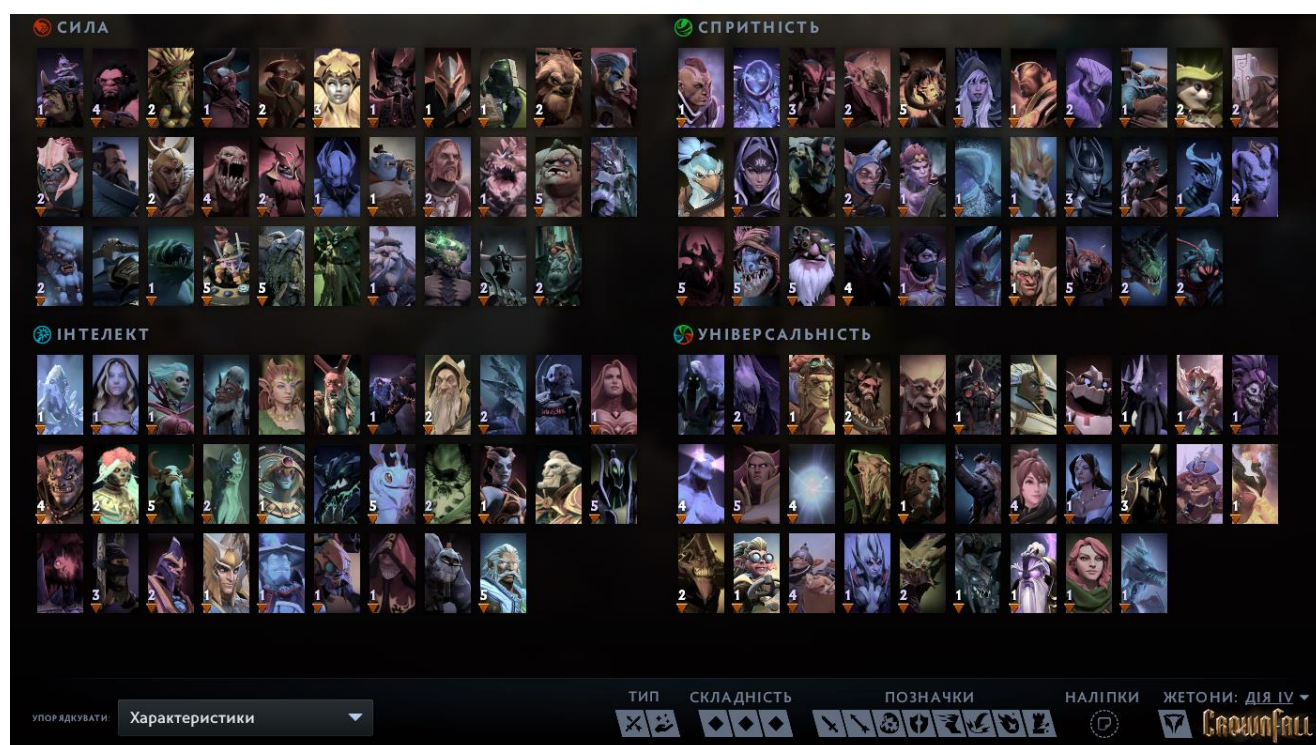


Рисунок 1.1 – Всі наявні герої у Dota 2

Здебільшого перебіг матчу залежить від дій персонажів, якими керують гравці. Кожен герой має набір характеристик, які проявляються тією чи іншою мірою. Один із атрибутів виступає для героя основним. Його підвищення додатково збільшує шкоду, яка завдається опонентам під час атак.

До базових атрибутів персонажа належать[6]:

- Сила – відповідає за кількість НР та швидкість регенерації.
- Спритність – регулює швидкість атаки, а також рівень броні.
- Інтелект – визначає число МР, швидкість відновлення та захист від магії.

Другорядні параметри персонажа:

- Швидкість пересування залежить від спритності.
- Швидкість атаки теж залежить від спритності.
- Дальність атаки – визначається зброєю героїв, яку вони використовують.
- Шкода від атаки – залежить від основного атрибуту.
- Броня – залежить від спритності.

Практично всі параметри можуть бути покращені протягом матчу шляхом прокачування рівнів, навичок та купівлі предметів. Так, купуючи предмет, який збільшує параметр спритності, герой отримує бонусну броню, збільшує швидкість переміщення та атаки. Також він завдає суперникам більшої шкоди під час атак, якщо основний параметр персонажа – спритність.

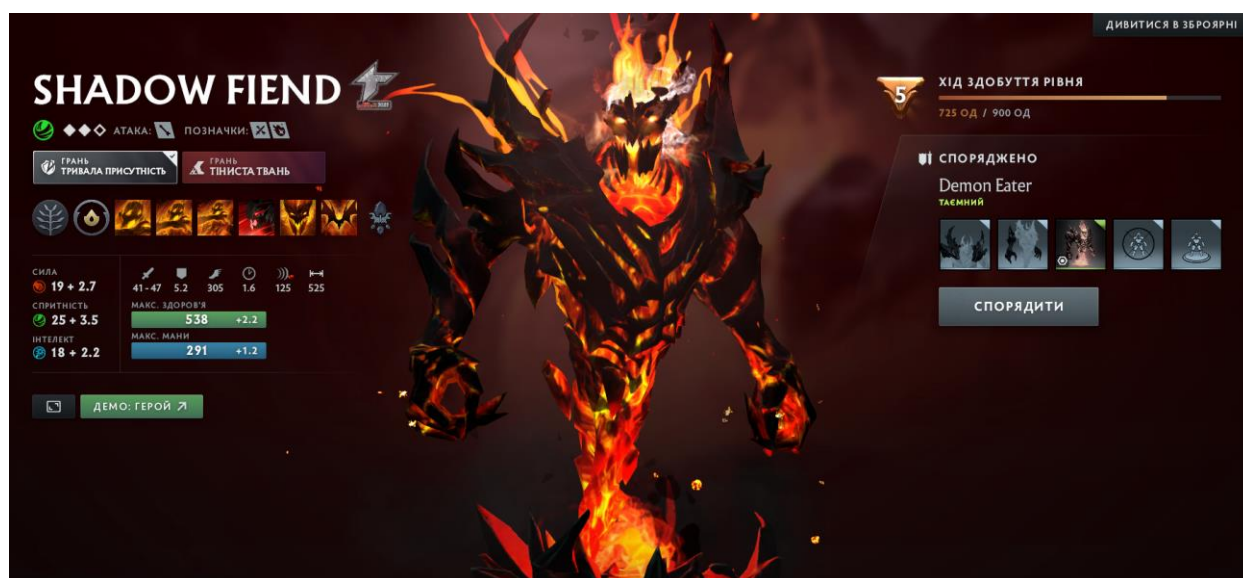


Рисунок 1.2 – Приклад вигляду у грі героя та опису його характеристик

Поєднання показників і здібностей героя робить його більш пристосованим до виконання певної ролі у Dota 2. До основних позицій у грі можна віднести такі:

- керрі;
- мідер;
- саппорт;
- хардлайнер;
- роумер.

Керрі (carry, core, late)[7] – головний персонаж для пізніх стадій матчу. На початку гри він відносно слабкий. Для нього важливі захист та підтримка саппортів. Весь потенціал гравця розкривається ближче до кінця матчу за умов отримання необхідних предметів. Завдяки активному фарму на початку і в середині гри герой на цій позиції отримує можливість вирішити долю протистояння. Граючи в Доту за керрі, геймер бере більшу частину відповідальності за результат партії.

Мідер (mider, mid) – персонаж, який отримує перевагу за досвідом на початку гри, щоб ефективно самотужки протистояти суперникам на середній лінії. Якщо у його команди виникають проблеми на верхній або нижній лініях, сила мідера може допомогти тиммейтам у непростій ситуації.

Саппорт (support) – герой, ефективність якого мінімально залежить від золота. Має яскраво виражені ефективні здібності на ранніх стадіях гри. Покликаний сприяти розкриттю потенціалу мідера та керрі. Він бере всі витрати на себе та віддає кор-героям весь можливий фарм. Грати за саппорта складніше, ніж може здатися на перший погляд.

Хардлайнер (hardliner, offlaner) – персонаж, здатний на початкових етапах протистояти 2–3 ворожим героям лінії. Характеризується великою кількістю НР або можливістю ескейпу (втечі від бою). Може отримувати ресурси поза лінією, якщо умови на ній занадто важкі. Граючи за хардлайнера, геймер має бути готовим до дефіциту ресурсів.

Роумер (roamer) – герой, який постійно перебуває в русі картою і не прив'язаний до жодної з ліній на старті. Його завдання – забезпечення підтримки там, де вона потрібна. Це може бути захист лінії від ворога, що атакує, або допомога у вбивстві противника.

Для кожної ролі характерна своя поведінка на різних етапах матчу. Незважаючи на нелінійність ігрового процесу, геймери роблять схожі дії у кожній партії[8]:

1) Керрі :

- Початок гри (early game). Купує предмети, які полегшують фарм та дозволяють вижити на перших хвилинах матчу. Разом із саппортом відправляється на

легку лінію. Усі зусилля спрямовує на отримання максимальної кількості золота. Намагається уникати пошкоджень і не вплутуватися в бійки.

- Середина гри (mid game). Купуючи перші серйозні предмети, може заробляти золото швидше. Використовує всі шанси для фарму, граючи в лісі та на лініях. Якщо має можливість, бере участь у бійках. Може знищити поодинокого ворожого героя підтримки.

- Пізня стадія гри (late game). Отримує необхідні предмети. Бере активну участь у масових битвах, роблячи великий внесок у їхній результат.

2) Мідер :

- Початок гри. Купує ті стартові предмети, які допоможуть протистояти ворожому мідеру. Концентрується на добиванні союзних та ворожих крипів, щоб здобути перевагу над суперником. Уважно стежить за картою, щоб заздалегідь помітити небезпеку в особі ворожого роумера або саппорта.

- Середина гри. Отримує вищий рівень на карті (іноді вдвічі вище середнього) та перші предмети. Починає активно допомагати у вбивствах ворогів. Нав'язує масові зіткнення, у яких реалізує свою перевагу, тим самим закріплюючи її.

- Пізня стадія гри. Стає другим за силою героєм після керрі. Разом із командою бере участь у фінальних битвах, завдаючи серйозної шкоди суперникам за допомогою своїх здібностей та предметів.

3) Хардлайнер :

- Початок гри. Базова закупівля складається з предметів та розхідників, які підвищують запас здоров'я, його регенерацію, а також покращують захищеність персонажа. Проводить старт гри на важкій лінії. З перших хвилин матчу йому протистоять ворожий керрі та саппорт (іноді 2). Завдання хардлайнера – вижити за цих умов і отримати максимальну вигоду. За сприятливих обставин, отримавши перевагу за досвідом та рівнем (2 до 1), хардлайнер заважає ворожому керрі-герою заробляти золото та досвід. Це змушує саппорта суперників постійно бути присутнім на лінії.

- Середина гри. Дочекавшись допомоги від мідера чи роумера, може спробувати вбити суперників по лінії. З цієї стадії перестає бути прив'язаним до своєї

позиції. Разом із роумером він веде боротьбу за контроль над картою, щоб створити простір для розвитку керрі.

- Пізня стадія гри. Безпосередньо бере участь у битвах на перших ролях. Приймає великий удар на себе. За наявності сильних здібностей контролю героїв противника (оглушення, уповільнення) ефективно ініціює битви на вигідних для своєї команди умовах.

4) Роумер :

- Початок гри. У списку початкових покупок у роумера знаходяться предмети, які дають змогу ефективно вбивати ворожих героїв на перших хвилинах гри та якомога рідше повертатися на базу. Відправляється на ту лінію, де вищий шанс успішного ганка (ganking), за проведення якого роумер отримує необхідний досвід та золото.

- Середина гри. Веде активну боротьбу за контроль над картою. Вистежує поодинокі цілі, з якими зможе впоратися наодинці. За можливості бере участь у всіх битвах на карті. Поєднуючи свої сили з союзними героями, ініціює вигідні для своєї команди битви або допомагає в атаках на ворожі вежі.

- Пізня стадія гри. На пізньому етапі приєднується до команди та бере участь у вирішальних битвах.

5) Саппорт :

- Початок гри. Отримує базові предмети і розхідники для підтримки кор-героїв. На початку матчу забезпечує безпечний фарм та простір для керрі, допомагає мідеру отримати перевагу над мідером ворожої команди. У той же час проводить розвідку та розставляє варди. Бере участь у всіх можливих зіткненнях, щоб уберегти союзників від загибелі чи допомогти вбити опонентів.

- Середина гри. Купує предмети, призначені для допомоги та захисту команди. Регулярно оновлює варди для своєї команди, а також виявляє та знищує ворожі. Бере участь у всіх битвах. Золото та досвід отримує під час битв, а також із ліній, які вільні від союзних керрі.

- Пізня стадія гри. Закінчує збір сильних предметів підтримки. У фінальних битвах допомагає основним силам команди своїми здібностями, намагаючись не підставлятися під удар.



Рисунок 1.3 – Вигляд мапи у Dota 2

За основу створюваної карти Dota 2 було взято карту Dota з гри cft III: Reign of Chaos.

Карта гри майже симетрична[8]. Кожна команда має власний табір у кутах карти — головною метою гри є захист власного табору і знищення ворожого. На базі кожної команди також розташований фонтан — споруда, що відновлює життєві та магічні сили героїв та атакує ворогів при їх наближенні, Древній трон (англ. Ancient) — головна споруда, яку необхідно знищити/захистити, 6 бараків — споруд, із яких час від часу з'являються «кріпи» (при знищенні чужого бараку команда посилює своїх

кріпів ближнього чи дальнього бою на кожній із трьох ліній), по 15 «колодязів», знищення яких приносить герою невелику кількість золота, 3 башти (T1, T2, T3) — захисні споруди, що атакують ворогів при їх наближенні. Бази поєднані трьома лініями: верхня — топ-лінія; нижня — бот-лінія; середня — мід-лінія. На кожній лінії розташовано по 2 башти кожної з команд. Що далі башта від бази, то вона слабша. Простір між лініями заповнено лісом або джунглями. У джунглях є табори нейтральних кріпів — по 12 у лісі кожної команди, Чотири з цих таборів зайнятий Древніми кріпами — сильними істотами, вбити яких на початкових рівнях дуже важко. Із північного заходу на південний схід перпендикулярно до мід-лінії по карті протікає річка, що поділяє карту на дві частини — світлу та темну.

Предмети[9] - внутрішньоігрове спорядження героїв, яке може давати героям додаткові характеристики та спеціальні здібності. Їх можна купити в кількох крамницях на ігровій карті. Деякі предмети низького розряду складаються в предмети вищого розряду, за допомогою рецепта. Герої мають шість осередків для предметів у їхньому інвентарі, три осередки в ранці, окремий осередок для нейтральних предметів і окремий осередок для Town Portal Scroll icon Town Portal Scroll, а також додаткових шість у їхній схованці. Кур'єр може піднімати і відсилати предмети їхньому власникові.

Нейтральні предмети - це предмети, які видобуваються з жетонів, що випадають з нейтральних кріпів. Їх не можна купити або продати і не можна знищити.

Кожен гравець може бути споряджений лише одним нейтральним предметом у комірці для нейтрального предмета. Невикористані нейтральні предмети можна помістити в схованку біля фонтану.

Майже кожен предмет у Dota 2 можна викинути, дозволяючи іншим героям взяти його, або передати його своїм союзникам. Незважаючи на це, більшістю предметів не можна ділитися з іншими героями: герой, який купив предмет, вважається його власником, і тільки він може використовувати, продавати або покращувати його. Якщо він опиниться в інвентарі когось іншого, то предмет буде заглушений.

Проте більшість розхідників повністю передаються, оскільки їх не можна поліпшити і вони дають тимчасові ефекти. Також ці предмети можуть використовувати герої, які не купували їх.

Ця особливість дозволяє героям підтримки купувати предмети і ділитися ними з командою. Деякі невитратні предмети відносяться до умовно переданих: інші герої можуть отримувати ефекти від отриманих предметів, але не можуть їх продати або поліпшити.

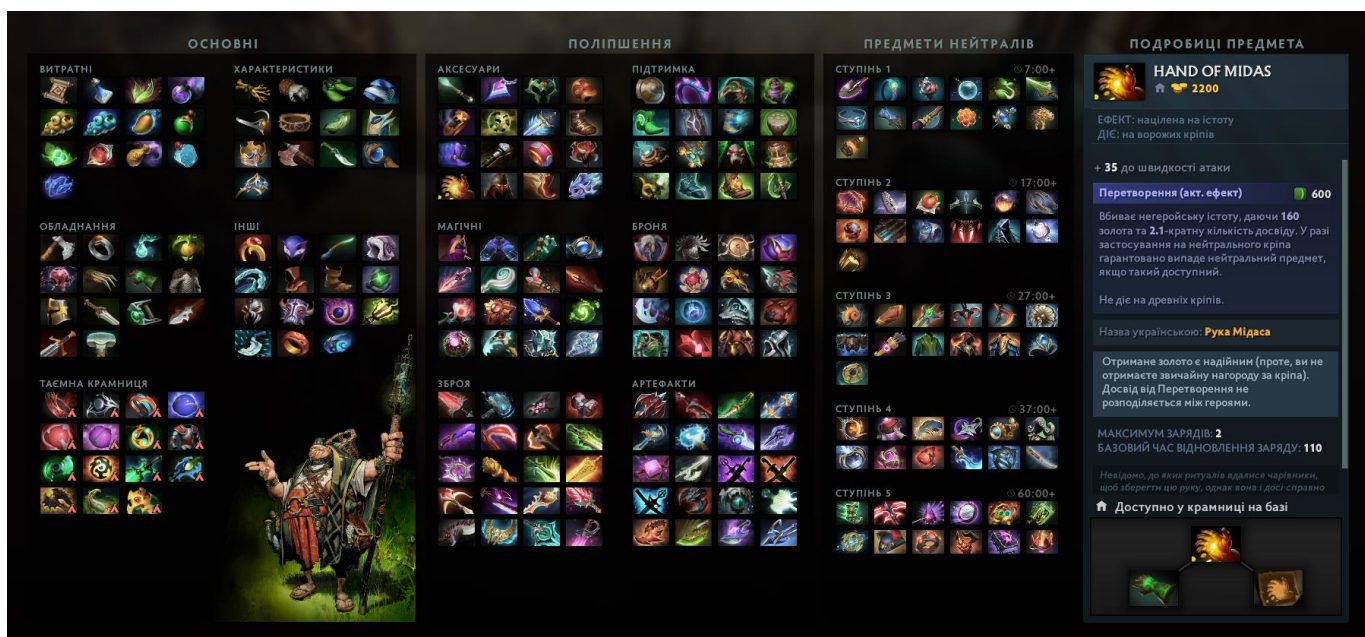


Рисунок 1.4 – Список усіх предметів у Dota 2

Dota 2[10], як і її попередниця DotA, є кіберспортивною дисципліною. Ще під час бета-тестування було проведено кілька великих турнірів, найбільшим з яких є The International, Starladder, також існує велика кількість регіональних турнірів та дрібних аматорських турнірів. Вона має величезну базу гравців і регулярно проводить великі турніри з великими призовими фондами, наприклад, The International.

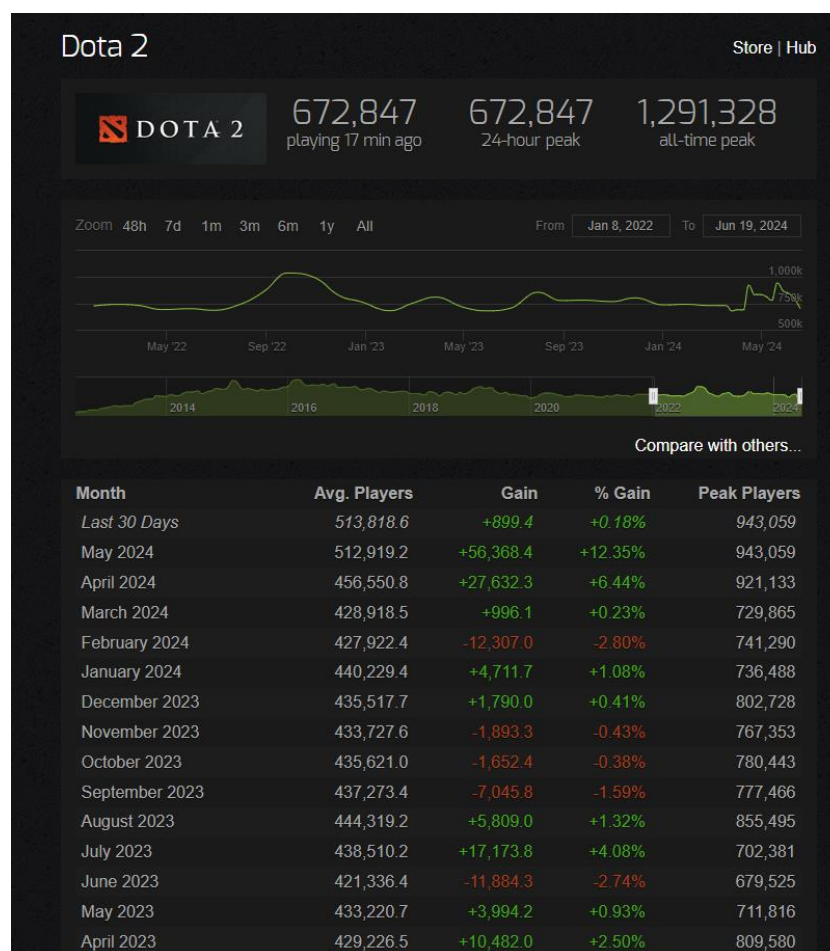


Рисунок 1.5 – Статистика онлайн гравців у Dota 2

МОВА-ігри займають важливе місце в кіберспорті. Великі турніри привертають мільйони глядачів і спонсорів. Крім того, ці ігри мають велику і активну спільноту гравців, що сприяє їхньому постійному розвитку та популярності. Перший The International було проведено 2011 року, у перші дні бета-тестування гри, на кіберспортивному чемпіонаті GamesCom. Організувала турнір розробник гри — Valve. На ньому змагалися 16 команд, а сума призових становила 1,6 млн доларів США. Переможцем турніру, а також власником одного мільйона доларів, стала українська команда Natus Vincere. Через рік у Сієтлі було проведено другий турнір The International 2. Переможцем турніру та власником мільйона доларів стала китайська команда iG, команда Natus Vincere посіла друге місце.

В наступні роки призовий фонд продовжував зростати, поки не досягнув свого максимуму у 2021 році — понад 40 млн доларів.

У 2020 році турнір було скасовано через пандемію коронавірусу. У 2022 році призовий фонд вперше зменшився і склав всього 18.9 млн доларів.

МОВА-ігри поєднують глибину стратегічного планування зі швидкими реакціями та індивідуальною майстерністю, що робить їх одними з найзахопливіших та найпопулярніших жанрів в індустрії комп'ютерних ігор.

1.2 Огляд існуючих систем аналогів

Простори інтернету дають нам величезну кількість інформації для вирішення питання створення бази знань. Безліч застосунків, додатків та сайтів, які переповнені інформацією, дають можливість дізнатися всі аспекти гри. Тому не варто залишати без уваги наступні веб-додатки, які є найпопулярнішими у використанні серед туристів-відпочивальників.

Перед тим як аналізувати кожен додаток потрібно звернути увагу на їх класифікацію. Адже вони мають свої функції, переваги, а також недоліки.

Розглянемо веб-ресурси, які містять інформацію про гру «Dota 2»:

- Dota 2 Fandom : Dota 2 Wiki;
- офіційний сайт Dota 2 – Dota2.com;
- Liquipedia;
- гра «Dota 2».

Dota 2 Fandom : Dota 2 Wiki[11] – це фанатська та спільнота-орієнтована платформа, присвячена популярній грі Dota 2. Вона містить велику кількість матеріалів, зокрема: інформація про персонажів: детальні сторінки про героїв, їх здібності, історії та стратегії гри, предмети та механіки: інформація про предмети, ігрові механізми, патчі, оновлення та зміни, режими гри: опис різних режимів гри, стратегії та поради, лор та всесвіт: історії, стосунки та фони персонажів і фракцій у всесвіті Dota 2, внески спільноти: спільний простір, де фанати можуть редагувати чи створювати нові сторінки про гру. Цей сайт часто оновлюється новими патчами гри, змінами балансу та матеріалами, створеними спільнотою. Тобто він містить в собі майже всю інформацію по грі, яка змінюється на актуальну при виході оновлення в

грі. Але недоліком є перенасищення сайту різною навігацією не пов'язаною з грою, що значно робить важчим пошук потрібної інформації.

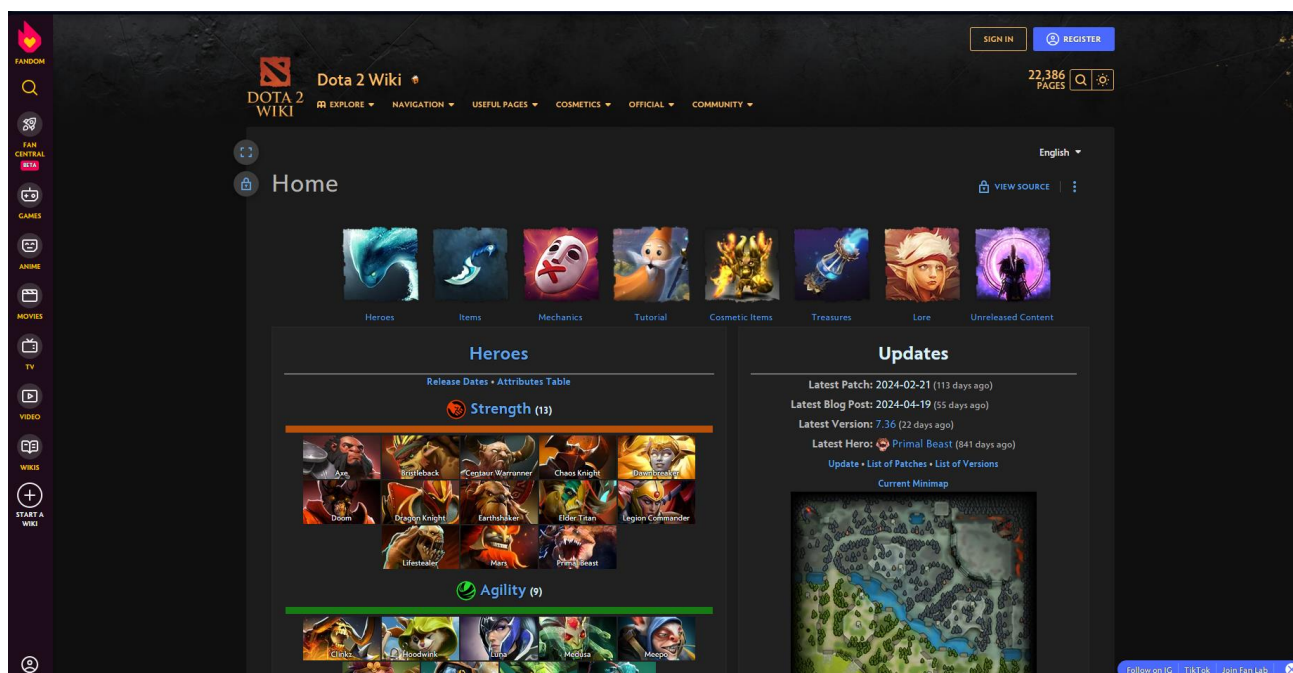


Рисунок 1.6 – Інтерфейс веб-сайту Dota 2 Fandom : Dota 2 Wiki

Dota2.com[12] – це офіційний веб-сайт гри Dota 2, розроблений компанією Valve. Цей сайт містить кілька основних розділів, включаючи Герої, Гра, Новини та Кіберспорт, які дають гравцям доступ до різноманітної інформації про гру. У розділі гра можна знайти інформацію про оновлення, але щоб знайти потрібну зміну може бути досить важко, адже потрібно знати номер оновлення. В розділі герої є лише актуальна інформація про базові характеристики та його здібності, що буває не достатнім. Тобто недоліком цього ресурсу є неповна інформація про гру, важкість пошуку певної інформації.

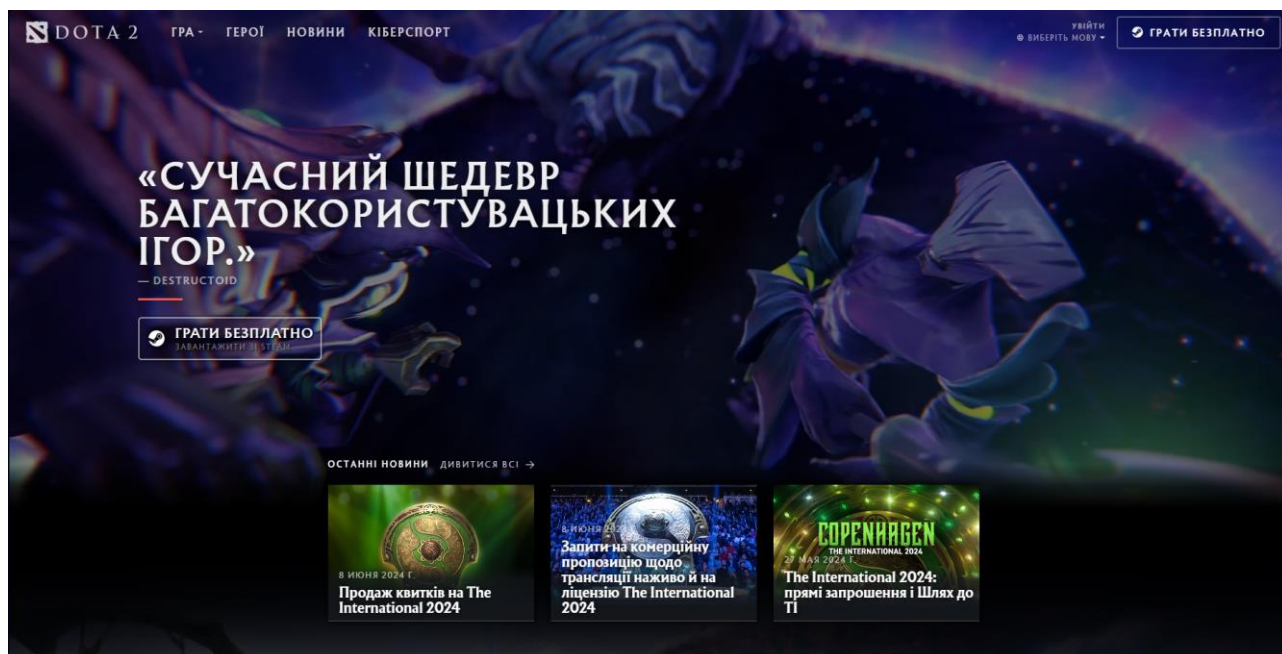


Рисунок 1.7 – Інтерфейс сервісу Dota2.com

Liquipedia[13] – ресурс, який зазвичай використовується для пошуку інформації про турніри ігор, але також є окремий розділ про гру «Dota 2», який зображено на рисунку 1.8. Недоліком є цього веб-сайту є те, що він спеціалізується на інформації про турніри, а не про гру, через що страждає якість цього розділу. Тобто недостовірність та неповнота інформації.

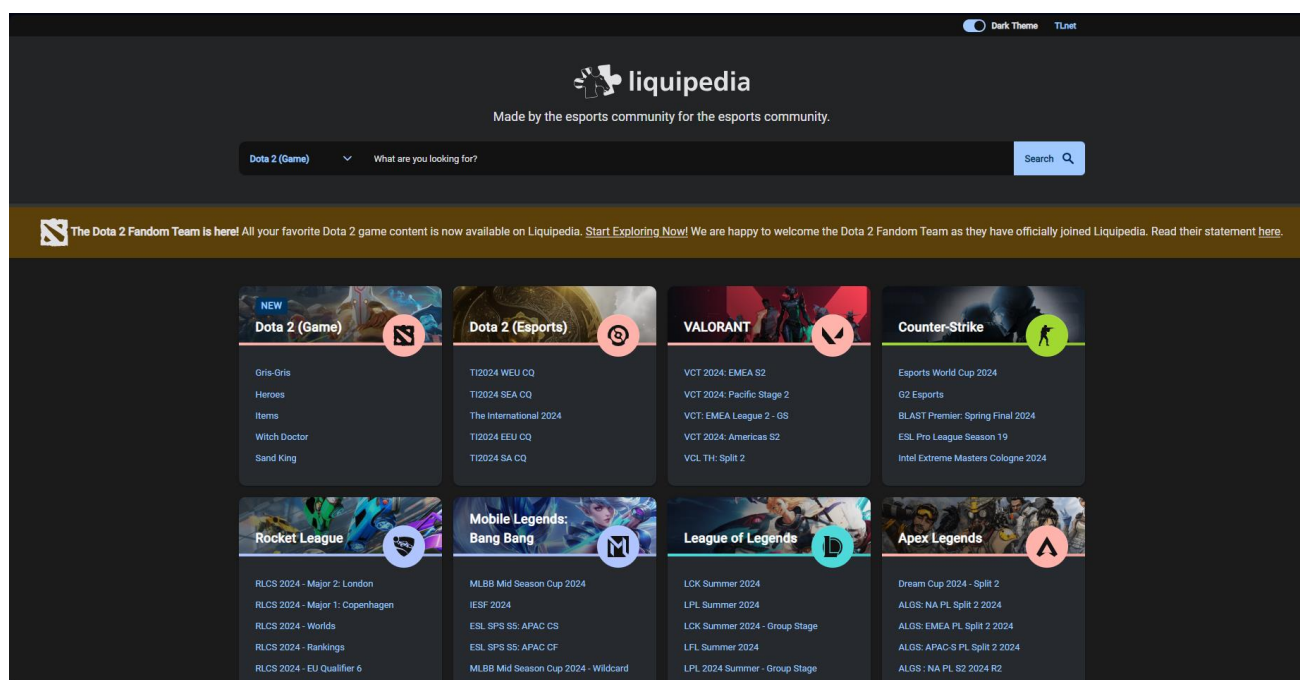


Рисунок 1.8 – Інтерфейс ресурсу Liquipedia.

Гра «Dota 2» - власне сама гра, в якій є вся інформація про гру, можна спробувати усі механіки на практиці та є навчальний посібник, який дасть базове розуміння гри. У грі доступна інформація про героїв, їх здібності та характеристики в меню вибору героя, але глибші стратегії чи зміни механік часто відсутні. Є навчальний режим, який допомагає новачкам освоїти основи гри, але він обмежений лише базовими знаннями. Під час гри можна отримувати підказки, але для більш детальної інформації про предмети чи стратегії потрібно виходити з гри або звертатися до зовнішніх джерел. Недолік у тому, що неможливо швидко знайти допомогу в реальному часі, що може спантеличити новачка, якому потрібна підказка на даному етапі.

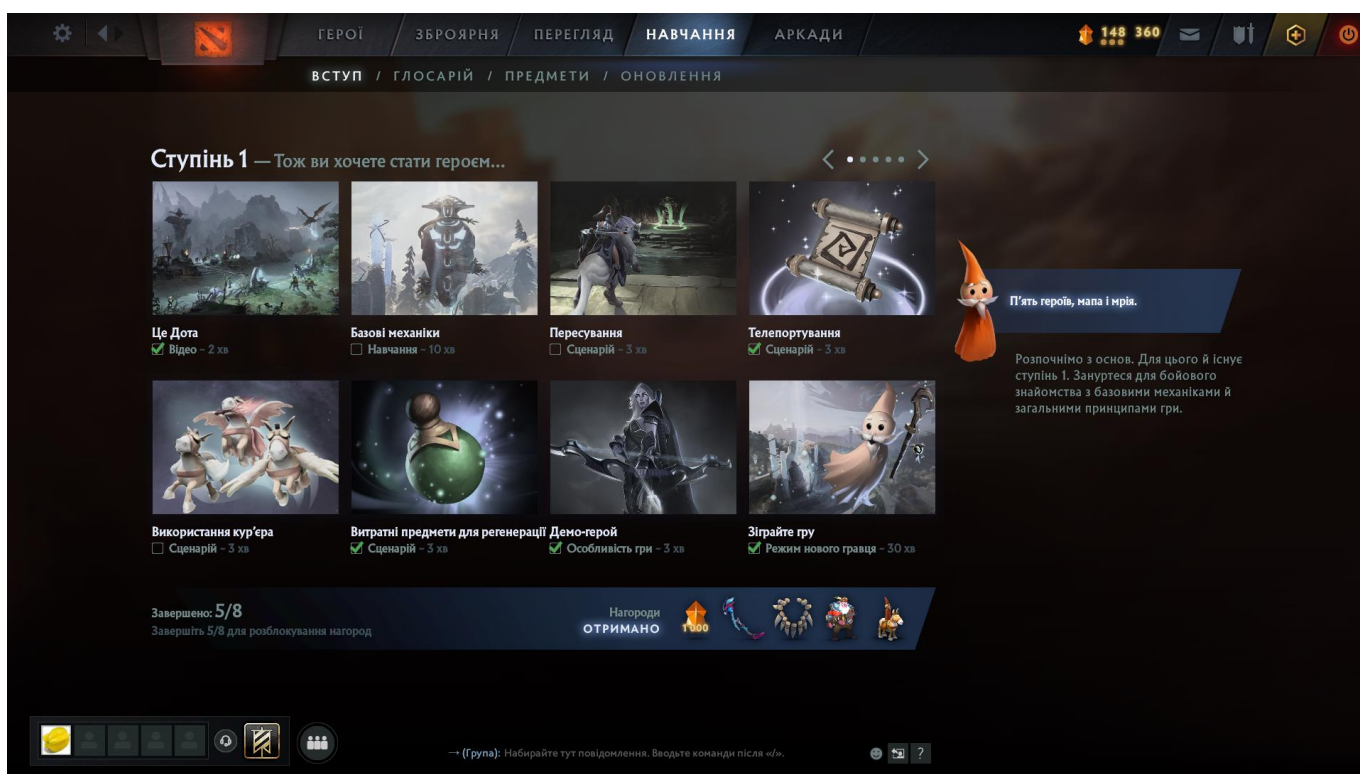


Рисунок 1.9 – Інтерфейс розділу навчання у грі «Dota 2».

Не варто лишати уваги і інші ресурси, які пов'язані із темою гри «Dota 2». Щоб створити новий продукт, важливо не оминати увагою інших, схожих, що створили раніше. Але потрібно зважувати всі переваги та недоліки уже розглянутих ресурсів, та брати їх до уваги. Розглянуті вище додатки мають ряд переваг, але також містять

недоліки. Недоліки у них усі схожі : недостовірність, неповнота інформації, перевантажений інтерфейс. Найкращим з них є сама гра, адже в ній можна спробувати це все на манекенах власними руками, але новачку запустити карту з манекеном може стати викликом. Тому розробка онтологічної бази знань гри «Dota 2» є досить актуальною, оскільки вона зможе виправити ті недоліки, які були зазначені у вищеперерахованих додатках.

1.3 Постановка задач дослідження

Сьогодні всесвіт Dota 2 набув значної популярності, привертаючи увагу як нових гравців, так і шанувальників жанру. Постійні оновлення гри, участь у змаганнях, а також розвиток різноманітних медіапродуктів (від коміксів до анімаційних серіалів) сприяють її глобалізації та поширенню серед широкої аудиторії. Dota 2 не обмежується лише комп'ютерною грою — це цілий культурний феномен, який притягує різних людей до вивчення особливостей гри та всесвіту, що стоїть за нею.

Dota 2 є багатогранною грою, яка пропонує величезну кількість героїв, стратегій та сценаріїв. Вона приваблює не лише гравців, зацікавлених у стратегічних іграх, а й тих, хто захоплюється епічними історіями, динамічними битвами та командною взаємодією. Регулярні оновлення, нові персонажі, механіки та події підтримують інтерес гравців і дозволяють постійно відкривати нові аспекти гри. Також важливим аспектом є доступність різноманітних ресурсів та платформ для взаємодії з фанатами, що допомагає підтримувати живий інтерес до гри.

Зважаючи на розмаїття персонажів, подій і всесвітніх зв'язків, виникає потреба в організації великої кількості даних, пов'язаних з Dota 2. Виникає необхідність створення ефективної системи для зберігання, структурування та візуалізації цих даних. Онтологічний підхід у цьому випадку може бути надзвичайно корисним, оскільки він дозволяє створювати логічні структури для відображення зв'язків між різними елементами гри.

Онтології є потужним інструментом у сфері інформаційних технологій. Вони дозволяють структурувати знання та покращити їх обробку, надаючи комп'ютерам можливість краще розуміти складні взаємозв'язки між об'єктами. У випадку Dota 2 це може бути корисним для створення бази даних, яка б об'єднувала відомості про героїв, їх здібності, предмети, локації та стратегії. Така система дозволить користувачам легше орієнтуватися в складному всесвіті гри, а також сприятиме стандартизації даних і покращенню взаємодії між різними ігровими платформами та системами.

Використання онтологічного підходу є доцільним для розробки технологій, які можуть забезпечити більш глибоке розуміння ігрового світу Dota 2, сприяючи розвитку інтелектуальних систем, що полегшать взаємодію гравців з великою кількістю даних і збереження зв'язків між ними.

Метою цієї роботи є організація, класифікація і керування даними великого обсягу, тобто інформацією про гру «Dota 2». А також розробка інформаційної технології, яка буде надавати інформацію за запитам. Ця онтологія, яка містить класи героїв, предметів та базових механік, їх відмінності та особливості. Дотримання правильної структури онтології зменшить час, необхідний для пошуку інформації та спростить проведення аналізу. Потрібну інформацію користувач буде отримувати на основі введених ним запитів, тому організація розробленої інформаційної технології має бути зрозумілою для користувача. Для досягнення поставленої задачі потрібно вирішити такі завдання:

1. Проаналізувати існуючу інформацію про гру «Dota 2».
2. Побудувати та систематизувати інформацію про механіки, героїв та предмети.
3. Спроекувати моделі інформаційної технології онтологічного моделювання бази знань гри «Dota 2».
4. Розробити структуру та відносини онтологічної бази знань.
5. Провести тестування онтології на основі запитів.

1.4 Висновок до розділу 1

У першому розділі було здійснено детальний огляд гри «Dota 2». Обґрунтовано доцільність розробки інформаційної технології онтологічного моделювання бази знань цієї гри. Використання такої технології може значно полегшити гравцям швидкий доступ до потрібної інформації про гру. Під час огляду предметної галузі було визначено проблеми в існуючих аналогах баз знань та поставлено за мету виправити їх у своїй роботі.

Аналіз інформаційних ресурсів показав наявність кількох відомих баз знань з Dota 2, реалізованих у вигляді онлайн-ресурсів: вікі-хостинг Fandom (Dota 2 Wiki), офіційний сайт Dota 2 – Dota2.com, Liquipedia, а також сама гра Dota 2. Описано їх переваги та недоліки. Виявлено, що хоча ці ресурси містять великий обсяг інформації, вони не завжди надають структуровані дані, що ускладнює їхнє використання, особливо для нових користувачів. Найкращим аналогом серед них визначено вікі-хостинг Fandom, оскільки він пропонує зручну навігацію і містить велику кількість необхідної інформації.

Таким чином, було обґрунтовано необхідність розробки системи, яка дозволяє гравцям швидко знаходити відповідні дані, структуруючи інформацію за допомогою онтологічного підходу. Це дозволить створити зручну і ефективну базу знань, що буде відповідати потребам користувачів і значно покращить досвід гри.

Задачі дослідження були сформульовані таким чином, щоб вирішити наявні проблеми в аналізі і систематизації інформації про Dota 2, а також обґрунтовано використання онтологічного підходу для моделювання бази знань цієї гри.

2 РОЗРОБКА ПОКРАЩЕНОЇ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ТА ОНТОЛОГІЧНЕ МОДЕЛЮВАННЯ БАЗИ ЗНАНЬ ГРИ «DOTA 2»

2.1 Обґрунтування вибору методів реалізації інформаційної технології

Представлення знань – це множина синтаксичних і семантичних угод, що роблять можливим формальне вираження знань про предметну галузь у комп'ютерно інтерпретованій формі [14]. Модель подання знань це абстрактна структура або методологія, яка описує як інформацію так і знання, які можна структурувати, організувати та представити, щоб зрозуміти та використовувати її ефективно. Модель подання знань відображає, спосіб представлення знань та взаємодію компонентів у даній моделі. Найрозповсюдженішими є такі моделі представлення знань: логічні моделі, продукційні моделі, фреймові моделі, семантичні мережі [15].

Логічні моделі представлення знань це способи організації та виразу знань за допомогою логічних структур і правил. Вони використовують математичну логіку для формалізації знань і дають можливість комп'ютерам розуміти, робити висновки та використовувати ці знання для вирішення завдань. Основні логічні моделі представлення знань включають предикатну логіку: ця модель використовує логічне висловлювання для опису знань. Вона має предикати, які вказують на властивості об'єктів і відносини між ними [16].

Продукційна модель або модель, що заснована на правилах, дозволяє представити знання у вигляді речень-продукцій типу: «Якщо (умова), тоді (дія)»[17]. Під умовою розуміється деяке речення-зразок, за яким здійснюється пошук у базі знань, а під дією – дії, які виконуються при успішному результаті пошуку. При використанні продукційної моделі база знань складається з набору правил. Дані у такій моделі це вихідні факти, на підставі яких запускається машина виведення [18].

Фреймові моделі представлення знань – це способи організації та структурування знань, що базуються на концепції «фреймів». Фрейми представляють собою структуру даних, які містять атрибути (слоти) та їх значення для опису об'єкта або концепції. Ця модель дозволяє зручно відображати знання про світ, описуючи

характеристики та відносини між об'єктами. Фрейм – це основний об'єкт фреймової моделі. Він містить атрибути (слоти) та значення, які описують конкретний об'єкт або концепцію [19].

Семантичні мережі – це модель представлення знань, яка використовує графову структуру для опису взаємозв'язків між концепціями або об'єктами. У семантичних мережах вузли графа представляють собою концепції або об'єкти, а ребра (зв'язки) вказують на відносини між цими вузлами. Ця модель дозволяє структурувати та виразно відображати знання, а також використовувати їх для вирішення завдань розуміння та виведення інформації [20].

Онтологічна модель подання знань – це спосіб організації та представлення знань за допомогою онтології. Онтологія – це формально визначена система концепцій або термінів, що використовується для опису інформації та встановлення взаємозв'язків між ними в конкретній галузі знань або домені [21]. Онтології допомагають структурувати знання, надаючи їм семантичний контекст та поняттєву рамку. Загалом, онтології допомагають забезпечити однозначне розуміння та інтерпретацію інформації у конкретній галузі знань, що сприяє кращому управлінню та використанню цієї інформації.

Використання онтологій для побудови бази знань має багато переваг і може значно полегшити організацію, розуміння та роботу з інформацією. Ось деякі з основних переваг:

- структурована інформація;
- семантичне розуміння;
- інтелектуальний пошук і фільтрація;
- інтеграція даних;
- підтримка інтелектуальних систем;
- легка модифікація і розширення;
- спільне розуміння;
- мультидисциплінарність.

Онтології надають можливість структурувати інформацію в базі даних, що

допомагає організувати дані в логічний ієрархічний порядок. Це полегшує пошук, розуміння та навігацію по інформації. Також надають семантичний контекст інформації. Вони визначають семантику та взаємозв'язки між різними термінами і концептами, що полегшує розуміння значення даних. Онтології дозволяють створювати інтелектуальні алгоритми пошуку та фільтрації. Ви можете шукати не тільки за ключовими словами, але і за семантичним значенням, що робить пошук більш точним. За допомогою онтологій можна об'єднувати дані з різних джерел та джерел даних, навіть якщо вони мають різну структуру. Це спрощує роботу з різнорідними джерелами інформації.

Онтології створюють базу для розвитку інтелектуальних систем, таких як експертні системи, семантичні пошукові системи та системи рекомендацій. Вони можуть допомогти автоматизувати прийняття рішень та аналіз даних. Їх можна легко модифікувати і розширювати з метою врахування нової інформації та змін у знаннях.

Онтології дозволяють створювати спільне розуміння між різними системами та людьми, це сприяє зменшенню непорозумінь та покращенню співпраці. Їх можна застосовувати в різних галузях і для різних завдань, що робить їх універсальним інструментом для побудови баз знань. Загалом, використання онтологій допомагає створювати більш ефективні та інтелектуальні системи обробки та розуміння інформації, що робить їх корисними для багатьох галузей, включаючи науку, бізнес, освіту та інформаційні системи пов'язанні з описом художніх сетингів.

Protégé є одним з найпопулярніших і широко використовуваних безкоштовних програмних засобів для розробки, редагування та управління онтологіями. Розроблений на базі відкритого коду за підтримки Стенфордського університету, цей інструмент пропонує широкий спектр функцій для ефективного створення та редагування онтологій за допомогою добре структурованого графічного інтерфейсу.

Особливості Protégé включають:

1. Графічний користувацький інтерфейс: Protégé надає зручний графічний інтерфейс з наочною візуалізацією класів, властивостей та відносин, що дозволяє легко створювати і редагувати онтології (рис 2.2).

2. Підтримка стандартів онтології: Protégé підтримує всі основні стандарти, такі як RDF, OWL, SKOS, SPARQL, SHACL та інші, що забезпечує гнучкість та сумісність з різними джерелами даних та системами.

3. Інтеграція з іншими джерелами даних: Protégé дозволяє імпортувати онтології та дані з різних джерел, таких як файлові системи, бази даних, веб-сервіси або інші формати файлів, забезпечуючи можливість працювати зі складними сценаріями інтеграції.

4. Розширення та плагіни: Protégé підтримує великий набір плагінів та розширень, які додають додаткову функціональність та можливості. Це включає різні редактори властивостей, SPARQL-запитів, візуалізацій графів та ін.

5. Підтримка спільноти: Protégé має активну спільноту користувачів та розробників, які надають підтримку, обмін ідеями та обговорення нових функцій, проблем, методик та підходів у розробці онтологій.

Оскільки магістерська робота стосується "Онтологічного моделювання бази знань з організації бібліотеки", Протеже може виявитися дуже корисним інструментом для проекту. Він забезпечує ефективний спосіб створення, редагування та управління онтологіями, дозволяє інтегрувати дані з різних джерел, підтримує сумісність зі стандартами RDF та OWL та надає підтримку у вирішенні проблем та питань через активну спільноту.

Незважаючи на численні переваги Protégé, існують також деякі мінуси та обмеження, які варто врахувати при використанні цього програмного засобу для магістерської роботи:

1. Крива навчання: Протеже може бути дещо складним для початківців через його багаті можливості та налаштування. Нові користувачі можуть відчувати, що їм потрібно додатковий час для засвоєння функціональності та розуміння різних вкладок, інструментів та параметрів.

2. Продуктивність і ресурси: Протеже може вимагати помітних обчислювальних ресурсів, особливо при роботі з великими онтологіями або складними операціями. Це може призвести до його зниженої продуктивності та відповідності на менш потужних комп'ютерах.

3. Сумісність з форматами файлів: Протеже підтримує стандартні формати онтологій, такі як RDF, OWL і так далі, але може мати обмежену підтримку для інших менш поширених або специфічних форматів файлів.

4. Відсутність комерційної підтримки: Хоча Protégé має відкритий код і активну спільноту, гарантії щодо підтримки на комерційному рівні відсіюються. Це може стати проблемою для деяких організацій або проектів, які вимагають високого рівня підтримки та оновлень.

5. Безпека та обмеження користувацького інтерфейсу: Протеже може мати обмежені функції безпеки та налаштування доступу для захисту даних та контролю дозволів користувачів у порівнянні з комерційними рішеннями. Дивлячись на вищезазначені мінуси, Protégé все ж залишається корисним та надійним варіантом для онтологічного моделювання, особливо враховуючи його безкоштовність, розширюваність та широку функціональність. Попри обмеження, Protégé приносить значні переваги в контексті створення та обробки онтологій та доставляє значущий внесок у розвиток досліджень з онтологічного моделювання. Вибір програмного забезпечення в цілому залежить від специфіки предметної області, доступних ресурсів та технічних вимог [24].

Загалом, використання онтологій допомагає створювати більш ефективні та інтелектуальні системи обробки та розуміння інформації, що робить їх корисними для багатьох галузей, включаючи науку, бізнес, освіту та інформаційні системи пов'язанні з описом художніх сетингів.

2.2 Розробка покращеної інформаційної технології онтологічного моделювання гри «Dota 2»

Розробка інформаційної технології для онтологічного моделювання бази знань гри «Dota 2» полягає у створенні онтологічної моделі, яка дозволяє організувати знання про основні компоненти та процеси, що пов'язані з цією грою. Вибір методології є важливим етапом, оскільки він визначає ефективність і масштабованість майбутньої системи. Для досягнення цієї мети було обрано використання простого

підходу інженерії знань, який складається з семи основних кроків. Цю методологію було вдосконалено, додавши створення ER-моделі та діаграми класів для предметної області. Такий підхід дає змогу побудувати базову структуру моделі, що наочно демонструє сутності (класи) і допомагає виявити взаємозв'язки між ключовими елементами, спрощуючи тим самим подальші етапи розробки онтології. Ця методологія дозволяє ітеративно та систематично створювати онтології, що забезпечує їх гнучкість і адаптивність до конкретної предметної області. Покращена версія методології зображена на рисунку 2.1.

Метою цього підрозділу є детальний опис процесу створення онтології гри «Dota 2», з урахуванням усіх етапів методології. Виконання всіх кроків дозволить розробити робочу онтологію, яка ефективно моделюватиме взаємозв'язки між основними елементами гри та забезпечить їх узгоджене й повне представлення.



Рисунок 2.1 – Покращена інформаційна технологія створення онтологічної моделі

Отже, методологія включає в себе наступні етапи:

1. Визначення мети та предметної області

На першому етапі важливо зрозуміти, яку проблему вирішує онтологія, для якої предметної області буде створена модель. Для цього потрібно визначити цілі використання онтології (наприклад, автоматизація пошуку, підтримка прийняття рішень, інтеграція даних). Також визначити, які знання необхідно представити в онтології та сформулювати питання, на які онтологія має дати відповіді. Це дозволить зосередитися на ключових аспектах, забезпечити структуроване представлення інформації та уникнути зайвих деталей, що не стосуються предметної області.

2. Визначення масштабу онтології

Визначити, які частини предметної області будуть включені в онтологію. Цей етап полягає в визначенні меж онтології, що дозволяє зосередити увагу на найважливіших аспектах і не перенавантажувати модель зайвими елементами.

3. Побудова ER-моделі та діаграми класів

На цьому етапі створюються початкові моделі структури даних, що визначають основні сутності (класи) і взаємозв'язки між ними. ER-модель (модель сутність-зв'язок) допомагає візуалізувати взаємозв'язки між основними об'єктами предметної області. Діаграма класів дозволяє точніше визначити структуру класів та їх атрибутів.

4. Визначення основних концепцій та класів

На цьому етапі потрібно формалізувати основні елементи гри «Dota 2» через класи та концепти. Це можуть бути такі класи, як «Герой», «Об'єкт», «Навичка» або «Подія». Кожен з цих класів можна розділити на підкласи. Наприклад, клас «Герой» можна поділити на підкласи «Маги», «Воїни» тощо. Також важливо створити ієрархію класів, яка показує, як загальні та спеціалізовані концепти взаємодіють в межах гри.

5. Визначення атрибутів і властивостей

Для кожного класу необхідно визначити атрибути та властивості. Наприклад, для класу «Герой» атрибути можуть бути «Здоров'я», «Мана», «Шкода» тощо. Для класу «Навичка» атрибути можуть включати «Тип шкоди», «Рівень навички» або «Час охолодження». Визначаються також типи атрибутів — чи це числові значення (наприклад, рівень здоров'я героя), текстові значення (наприклад, опис навички) чи булеві значення (наприклад, чи має герой здатність до відновлення манни).

6. Визначення зв'язків

На цьому етапі потрібно створити зв'язки між різними класами. Це можуть бути відносини «використовує» (наприклад, зв'язок між класами «Герой» і «Навичка»), «вбиває» (між класами «Герой» і «Об'єкт») або «належить до» (між класами «Герой» і «Команда»). Визначаються також напрямки зв'язку (односторонній чи двосторонній) і кардинальність (наприклад, «один до багатьох», «багато до багатьох»).

7. Створення екземплярів

Після визначення класів і зв'язків необхідно створити конкретні екземпляри для кожного класу. Наприклад, для класу «Герой» можна створити екземпляри, такі як «Invoker», «Pudge», «Lina» з конкретними атрибутами (захист, рівень здоров'я, швидкість руху). Кожен екземпляр повинен мати значення для відповідних атрибутів, наприклад, «Invoker» може мати атрибути «Мана» 500, «Здоров'я» 650 тощо.

8. Створення можливості використання SPARQL-запитів

Після того як вже є створена онтологічна база знань потрібно під'єднати модуль SPARQL-запитів для сортування та вибірки даних користувачем.

2.3 Моделювання бази знань засобами технологій ER та UML

При моделюванні онтологічної бази знань необхідне застосування сучасних засобів моделювання, оскільки є важливим коректно впорядкувати дані та зв'язки між ними. Розглянемо процес моделювання бази знань гри «Dota 2» з використанням методології ER-моделювання.

ER-моделювання (Entity-Relationship modeling) – це методологія проектування і моделювання баз даних, яка базується на використанні концепцій «сутностей» (entities) та «відносин» (relationships) для опису структури і взаємозв'язків між даними в інформаційній системі [22]. Основною метою ER-моделювання є визначення сутностей, їх атрибутів та взаємозв'язків між ними, щоб створити абстрактну модель даних, яка може бути використана для подальшого проектування і створення реальної бази даних. Сутності представляють об'єкти або концепції, про які збирається інформація. Відносини вказують на зв'язки між сутностями. Атрибути – це властивості сутностей, які описують їхні характеристики. Наприклад, атрибутами сутності «Товар» можуть бути «Назва», «Ціна», «Кількість на складі» тощо.

ER-модель може бути візуалізована у вигляді схеми або діаграми, відомої як Entity-Relationship Diagram (ERD), яка допомагає зрозуміти структуру даних та їхні взаємозв'язки перед розробкою. ER-модель є важливою частиною процесу проектування і розробки баз даних. ER-діаграма (Entity-Relationship Diagram,

або ERD) – це графічне зображення ER-моделі, яке використовується для візуалізації структури даних в інформаційній системі або базі даних. ER-діаграми допомагають легше розуміти структуру та зв'язки між різними сутностями та об'єктами даних [22].

ER-модель (Entity-Relationship model) є потужним інструментом для моделювання структури даних в інформаційних системах і проектування баз даних. Вона має свої переваги і недоліки. На рисунку 2.2 зображено фрагмент ER-моделі з предметної галузі гри «Dota 2».

У таблиці 2.1 подано повне розшифрування зв'язків між предметною галуззю гри «Dota 2» та сутностями фрагменту предметної галузі.

Таблиця 2.1 – Опис зв'язків фрагменту предметної галузі гри «Dota 2»

Ім'я сутності 1	Ім'я сутності 2	Тип зв'язку	Ім'я зв'язку
Герой	Предмети	1:Б	Купує
Предмети	Характеристики	1:Б	Містить
Герой	Тип	1:Б	Належить
Герой	Характеристики	1:Б	Містить
Герой	Здібності	1:Б	Використовує
Здібності	Характеристики	1:Б	Містить

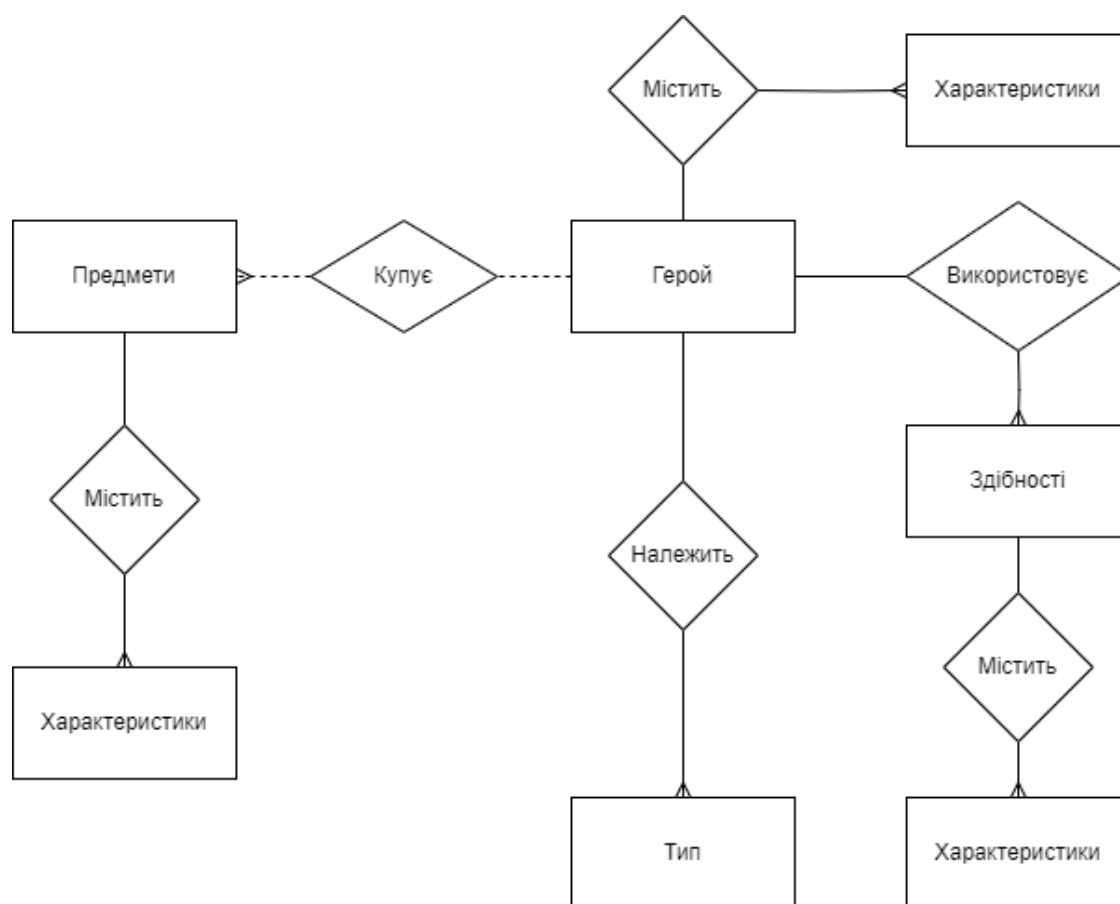


Рисунок 2.2 – ER-модель предметної галузі гри «Dota 2»

До переваг можна віднести простоту і зрозумілість, ER-модель надає зручний і зрозумілий спосіб візуалізації структури даних та взаємозв'язків між ними. Вона легко розуміється розробниками, аналітиками і замовниками проекту. ER-модель допомагає визначити сутності, відносини та атрибути, це спрощує процес проектування бази даних. Вона дозволяє планувати структуру даних до розробки фактичної бази даних. Чіткість взаємозв'язків, модель дозволяє чітко визначити, як сутності взаємодіють між собою і це допомагає уникнути непорозумінь у проекті. ER-модель є стандартизованою інструментальною методологією, яка використовується в багатьох галузях і є загальноприйнятою для моделювання баз даних. Завдяки ER-моделі можна аналізувати структуру даних і виявляти можливість оптимізації запитів та операцій з даними.

До недоліків можна віднести: спрощеність ідеалізації, ER-модель надто спрощує дійсну складність даних і взаємозв'язків в реальних системах. У деяких випадках це може призвести до втрати деталей. Обмеженість виразності, ER-модель

не завжди може відобразити всі аспекти дійсної системи. В деяких випадках можуть бути потрібні додаткові моделі або розширення ER-моделі. Деякі взаємозв'язки в реальних системах складніше відобразити за допомогою звичайних ER-діаграм. В ER-моделі важко моделювати великі і складні системи, оскільки діаграми можуть стати надмірно великими та нечитабельними.

У практиці, ER-модель є важливим інструментом для початкового аналізу та проектування баз даних, але вона може бути комбінована з іншими методами і моделями, такими як UML (Unified Modeling Language), для більш повного опису і розробки інформаційних систем і баз даних.

UML, або Unified Modeling Language, є стандартною мовою для моделювання та проектування програмного забезпечення. UML надає набір графічних і текстових нотацій, що дозволяють інженерам програмного забезпечення, аналітикам та розробникам створювати моделі програмних систем, їх структури, функцій та взаємодій. Основні концепції UML включають такі елементи: діаграми, компоненти, відносини, атрибути [23].

UML включає різні види діаграм, які представляють різні аспекти програмної системи. Основні типи діаграм UML включають:

Діаграми класів (Class Diagrams) показують структуру класів, атрибутів і методів, а також взаємозв'язки між класами.

Діаграми послідовності (Sequence Diagrams) відображають послідовність обміну повідомленнями між об'єктами в часі.

Діаграми взаємодії (Interaction Diagrams) включають у себе діаграми послідовності та діаграми співробітництва, які моделюють взаємодію об'єктів.

Діаграми діяльності (Activity Diagrams) використовуються для моделювання процесів та послідовностей дій в системі.

Діаграми станів (State Diagrams) показують різні стани об'єкта та переходи між ними.

UML дозволяє описувати компоненти програмної системи, такі як класи, об'єкти, інтерфейси, пакети, модулі тощо. Надає можливість визначити взаємозв'язки між об'єктами і класами, такі як агрегація, композиція, залежності, спадкування тощо.

Крім того, атрибути об'єктів і класів можна описати за допомогою UML. UML передбачає можливість створювати документацію, що допомагає пояснити структуру і поведінку програмної системи.

Переваги UML включають стандартизацію, що дозволяє розробникам спілкуватися і співпрацювати, а також можливість детального аналізу та проектування програмного забезпечення перед його реалізацією. Однак UML також може бути складним і часом вимагати значного обсягу роботи, особливо для великих та складних проектів. Також важливо враховувати, що успішна модель UML не завжди гарантує успішну реалізацію програмного продукту.

Діаграма прецедентів (Use Case Diagram) – це один з видів діаграм в рамках UML і використовується для візуалізації функціональності або можливостей системи з точки зору зовнішніх користувачів або акторів. Діаграма прецедентів допомагає ідентифікувати та відобразити ролі користувачів системи та способи їх взаємодії з системою через різні сценарії (прецеденти) [24].

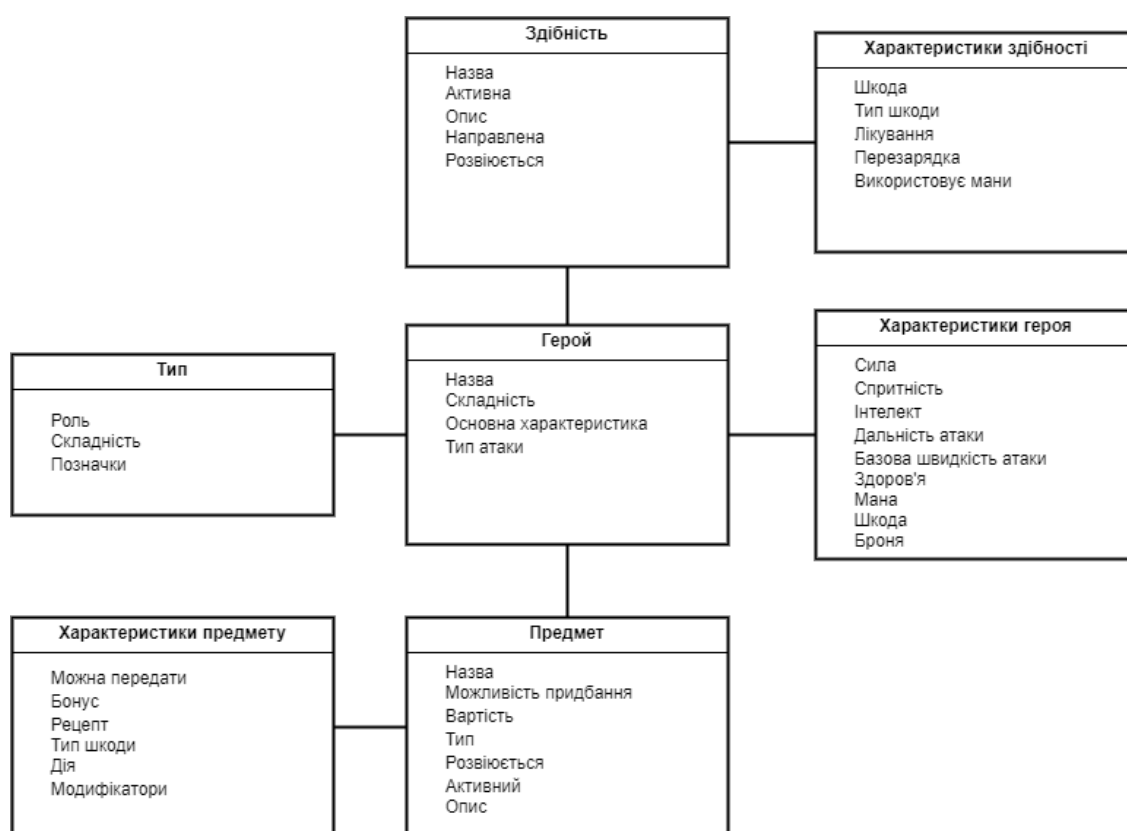


Рисунок 2.3 – UML-діаграма з предметної галузі гри «Dota 2»

Діаграма прецедентів має свої переваги, які роблять її корисною при моделюванні та розробці програмних систем:

- легкість спілкування;
- визначення функціональності;
- зосередження на користувачах;
- виявлення потенційних проблем;
- основа для інших діаграм;
- структурованість і стандартизація

Діаграми прецедентів легко читати та розуміти навіть людям, які не є технічними спеціалістами. Це дозволяє використовувати їх для комунікації з замовниками та стейкхолдерами, які можуть вказати на важливість різних можливостей системи. Діаграми прецедентів допомагають ідентифікувати і описувати функціональні можливості системи, які важливі для користувачів. Це допомагає розробникам зрозуміти, що система має робити. Аналіз діаграм прецедентів може допомогти виявити можливі невідповідності вимог та суперечності між різними прецедентами. Це дозволяє уникнути проблем в подальшому проектуванні і розробці.

Наведемо діаграму прецедентів роботи в онтологічній системі гри «Dota 2» на рисунку 2.4.

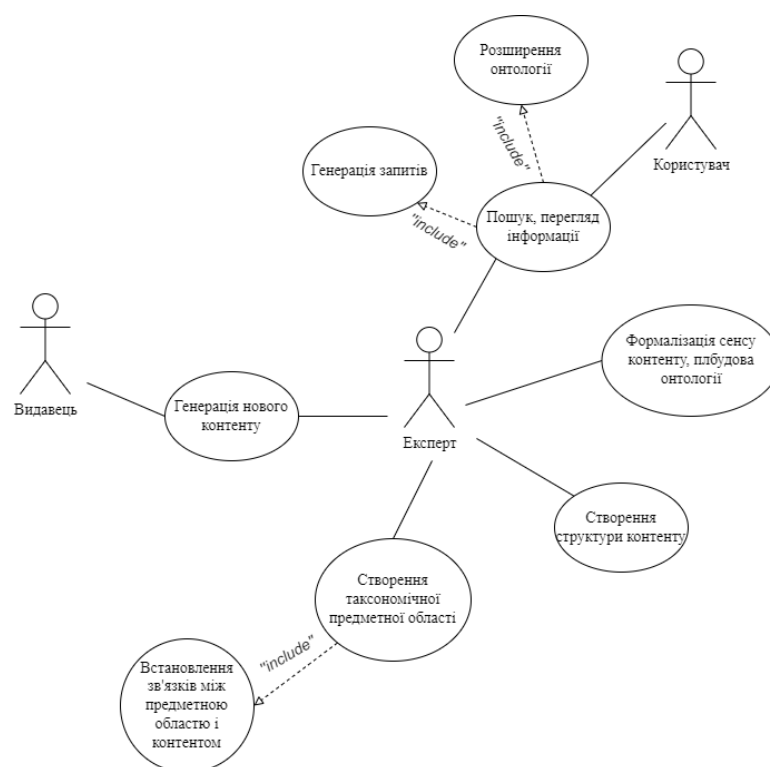


Рисунок 2.4 – Діаграма прецедентів роботи онтологічної системи гри «Dota 2»

Загалом, діаграми прецедентів є потужним інструментом для аналізу та документування вимог до програмних систем і забезпечують підставу для подальшої розробки та тестування інформаційної технології онтологічного моделювання бази знань гри «Dota 2» з точки зору функціональності.

2.4 Розробка онтологічної моделі бази знань гри «Dota 2»

Створення моделі бази знань — це процес формування впорядкованого сховища інформації, де дані організовані і зручні для доступу. Така модель дозволяє зберігати, упорядковувати та поширювати знання, що мають значення для певної сфери, компанії або проекту. Основними етапами розробки моделі бази знань є: визначення цілей, визначення цільової аудиторії, збір інформації, вибір відповідних технологій, розробка структури бази знань і наповнення її змістом.

Розробка моделі бази знань вимагає декілька кроків і враховується контекстом і метою, для яких ця база знань буде використовуватися. При розробці звернемо увагу на такі пункти:

- визначення мети та галузі застосування;
- збір та аналіз знань;
- вибір інструментів і технологій;
- створення структури бази знань;
- заповнення бази знань.

Аудиторією для такої бази знань є гравці, які хочуть покращити свої навички і зрозуміти, як використовувати певних героїв або предмети в різних стратегіях, тренери та аналітики, які досліджують ефективність героїв, предметів та тактик у грі, дослідники та розробники, які аналізують механіку гри, її баланс та вплив змін на геймплей.

Отже, ціль та аудиторія визначатимуть, які саме знання та інформацію буде містити база знань і коли буде організовувати ці дані. Важливо знати потреби користувачів та визначити, яким чином розроблювальний продукт буде корисною для них.

При розробці бази знань потрібно звернути увагу на такі аспекти, як герої та їх характеристики, предмети, NPC(non-playable character), останні зміни у грі “Dota 2”, здібності героїв. На основі цих аспектів створюємо головні класи нашої бази знань.

На цьому етапі розумно створити ієрархію предметної галузі та відобразити її графічно. Розробимо кілька схем та проведемо аналіз, щоб визначити, яка саме структура відповідає нашим потребам.

Однією з особливостей першого варіанту представлення є акцент на «герої». База знань вибудована на кількості класів, яка відповідає кількості описаних класів. Однак недоліком такого підходу є те, що відносини встановлюються тільки між одним аспектом області, в той час як інші сегменти представлені у формі даних. Цей підхід обмежує можливість додавання нових знань, які не стосуються елементу «герої».

Графічне зображення фрагменту семантичної мережі для побудови онтології гри «Dota 2» можна побачити на рисунку 2.5.

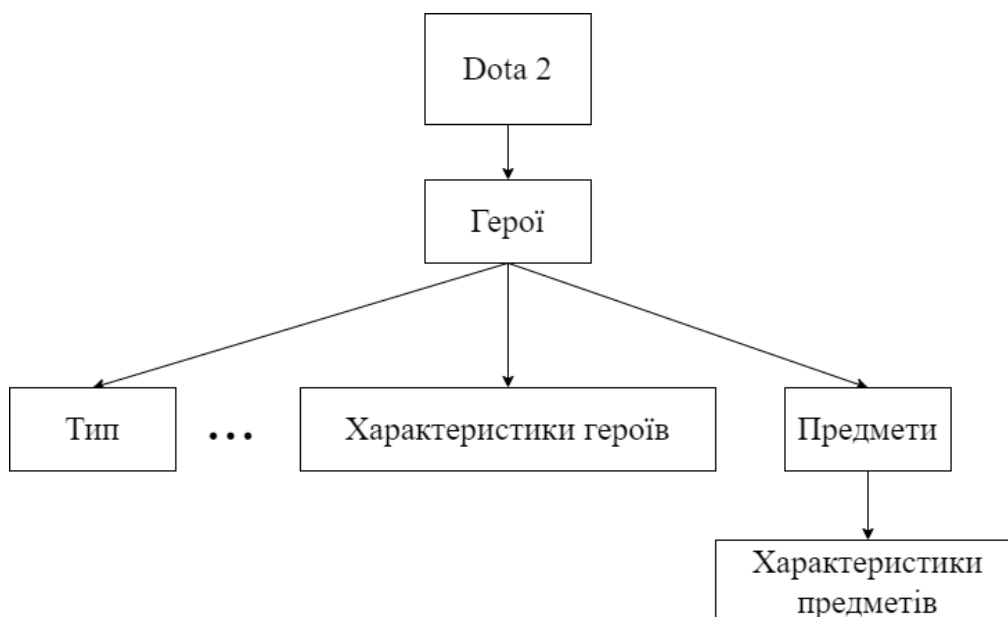


Рисунок 2.5 – Перший варіант вигляду онтології «Dota 2»

Важливо відзначити, що було проведено аналіз кількох основних варіантів представлення структури онтології. Другий варіант схеми включає в себе головні аспекти предметної галузі «Dota 2» (рис. 2.6)

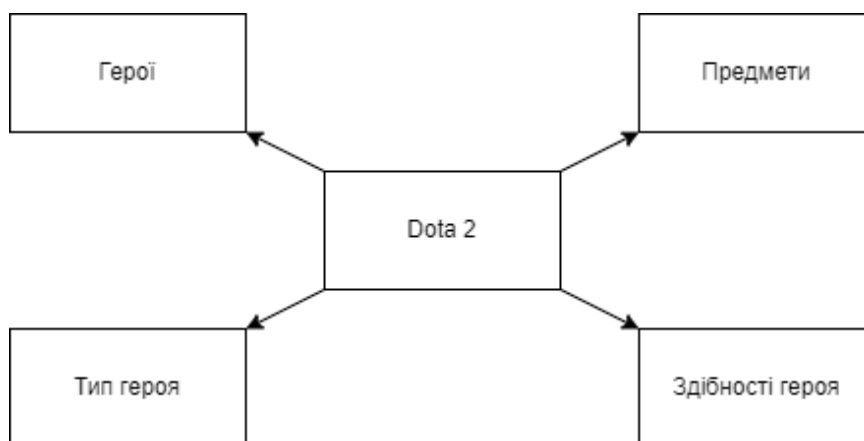


Рисунок 2.6 – Другий варіант вигляду онтології «Dota 2»

Для користувача була розроблена схема функціонування онтологічної бази знань гри «Dota 2» (рис. 2.7). Далі таку схему можна модифікувати та розширити.



Рисунок 2.7 – Схема функціонування онтологічної бази знань гри «Dota 2»

Для початку роботи користувач має запустити середовище та завантажити наявну онтологію. Після створення запиту, система здійснить пошук по гілкам онтографу та надасть результат.

2.5 Реалізація онтологічної моделі бази знань гри «Dota 2»

Для створення онтологічної моделі бази знань гри «Dota 2» у середовищі Protege потрібно створи відношення та класи. Однією з переваг такого середовища є простота у додаванні, редагуванні та видаленні будь-якого сегменту.

У Protégé, класи тлумачаться як сукупності, а елементи цих класів розглядаються як конкретні випадки. Побудова класів ґрунтується на використанні формальних конструкцій, які визначають вимоги до екземплярів даного класу. Protégé дозволяє створювати ієрархію класів, що відображається у формі «Суперклас – Підклас». Для того щоб створити класи в Protégé, потрібно перейти на вкладку

«Classes». За замовчуванням порожня онтологія містить клас Thing, який виступає як суперклас, і всі створені об'єкти входять в цей клас.

В Protégé існують обмеження на назви класів. Наприклад, не можна починати назву класу з цифри, якщо ви введете «1class», Protégé відобразить його як «class». Рекомендується використовувати більш зручні варіанти, такі як «class1» або «class_1». Також не можна використовувати апостроф у назві класу. Наприклад, якщо ви введете слово «м'яч», Protégé збереже лише частину слова після апострофа, тобто «яч». Об'єкти цих класів утворюють екземпляри.

Також можна створювати підкласи на різних рівнях від основного класу, який є класом першого рівня. Для цього потрібно скористатися функцією створення ієрархії класів, яка доступна через інструмент «Create Class hierarchy». У вікні, що відкрилося, потрібно вибрати основний клас. В першому полі вказуємо рівні для класів, а в полі «Prefix» задаємо назву для класів (рис. 2.8).

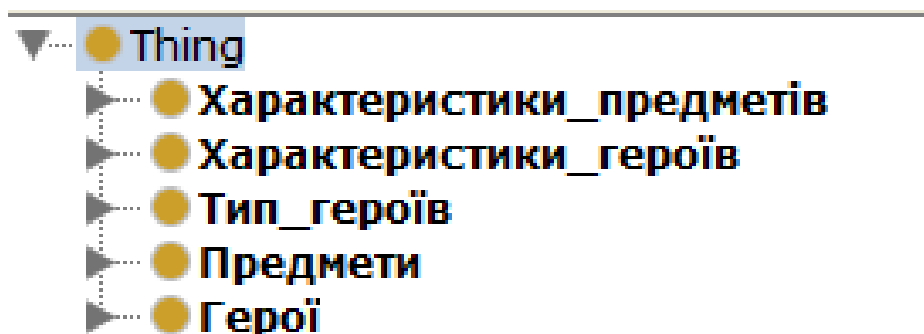


Рисунок 2.8 – Приклад створення класів в Protégé

Після того, як завершено створення класів предметної галузі ми маємо можливість зобразити їх у вигляді графу. Для цього потрібно перейти на вкладку «OntoGraf» та обрати зручний спосіб розміщення класів. У результаті отримаємо граф, що відображає створені класи (рис. 2.9). Одними з головних класів онтології гри «Dota 2» є «Герої», «Предмети», «Тип героїв», «Характеристики героїв», «Характеристики предметів».

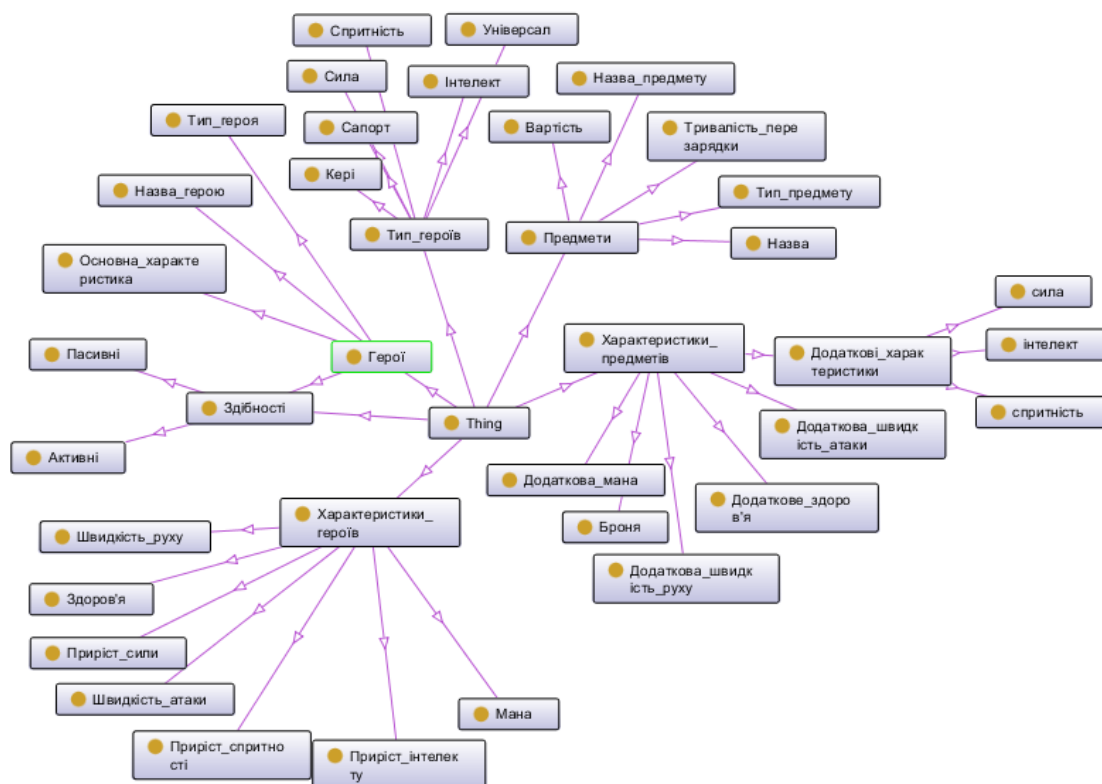


Рисунок 2.9 – Базовий граф онтологічної бази знань з гри «Dota 2»

Розглянемо основні функції роботи з класами та типи зав'язків і відносин між можливими класами. Опишемо такі функції:

Add Subclass – функція, яка додає новий підклас.

Add Sibling class – функція додання класу однієї ієрархії.

Delete classes – видалення обраного класу. Subclass of – встановлення підкласу у клас.

Equivalent to – встановлення еквівалентності певних класів.

Disjoint With – встановлення заборони використання взаємно суперечливих понять або термінів при їх входженні як підкласів до одного й того ж класу.

General class axioms – функція відображає аксіоми класів, що не можуть бути пов'язані напряду з іменованим класом.

Disjoint Union Of – робить клас головним у аксіомі.

Instances – визначення індивіда як тип в аксіомі твердження класу.

Target For Key – визначає змішаний список властивостей об'єктів і даних, які діють як ключ для екземплярів поточного вибраного класу.

Редактор Protégé має два типи властивостей(properties): Data Properties – властивості типів даних; Object Properties – властивості об'єктів.

Властивості об'єктів – це відношення між двома екземплярами (рис. 2.10)[25]. У даному редакторі є два основних типи властивостей: властивості об'єктів та властивості типів даних (Object Properties та Data Properties відповідно). Властивості об'єктів верхнього рівня автоматично створюються за замовчуванням, так само як і вищезгаданий клас Thing. Для того, щоб створити властивість для конкретного об'єкта, необхідно перейти на вкладку Object Properties.

Функціональність властивості виражається тим, що для кожного екземпляра цієї властивості існує не більше одного іншого екземпляра, який має зв'язок з першим екземпляром через ту саму властивість. Цю властивість також можна називати однозначною.

Зворотна функціональність властивості виявляється у тому, що для конкретного екземпляра може існувати багато інших екземплярів, які мають зв'язок з першим через цю саму властивість.

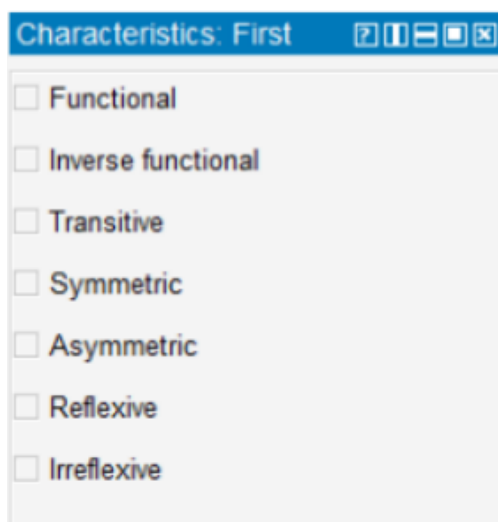


Рисунок 2.10 – Зображення видів властивостей об'єкту

Транзитивна властивість встановлює зв'язок між екземпляром x та екземпляром y , а також між y та z , з чого випливає наявність зв'язку між екземпляром x та екземпляром z . Важливо зауважити, що транзитивна властивість не може одночасно бути функціональною.

Якщо властивість x є симетричною і пов'язує екземпляри a та b , то це означає, що екземпляр b також має зв'язок з екземпляром a через цю властивість.

Якщо властивість x є асиметричною і з'єднує два екземпляри a та b , то важливо відзначити, що екземпляр b не може мати зв'язок з екземпляром a за допомогою властивості x .

Рефлексивність властивості виявляється в тому, що екземпляр може мати зв'язок з собою через цю властивість.

Іррефлексивність властивості виявляється у тому, що екземпляр не можна пов'язати сам з собою через цю властивість.

2.6 Висновок до розділу 2

У другому розділі магістерської кваліфікаційної роботи проаналізовано методи подання знань, розроблено моделі з предметної галузі гри «Dota 2» у різних формах подання, в тому числі засобами ER-моделювання та UML-діаграм.

На основі аналізу основних сутностей та зв'язків між ними побудовано ER-модель предметної галузі. Обґрунтовано важливість UML-діаграм та побудовано діаграму класів і діаграму прецедентів бази знань. Здійснено моделювання структури предметної галузі.

Проведено кодування онтології шляхом використання інструментарію Protégé. Проведено аналіз логічної, продукційної, фреймової моделей, семантичної мережі та онтологічної моделі подання знань.

Наведено переваги онтологічного підходу представлення знань та обґрунтовано доцільність його використання. Наведена концептуальна схема онтології гри «Dota 2», яка охоплює потрібні галузі, як «Герої», «Предмети», «Тип героїв», «Характеристики героїв», «Характеристики предметів».

Проведено кодування онтології засобами Protégé, змодельовано ієрархію основних класів. Здійснено огляд використовуваних функцій та зображено граф онтології.

3 ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОЇ ТЕХНОЛОГІЇ ОНТОЛОГІЧНОГО МОДЕЛЮВАННЯ БАЗИ ЗНАНЬ ГРИ «DOTA 2»

3.1 Програмні засоби для реалізації інформаційної технології онтологічного моделювання бази знань з гри «Dota 2»

Онтологічні програмні засоби (Ontology Tools) – це програми і середовища, призначені для створення, управління та використання онтології. Існує багато різних варіацій і типів, у залежності від їх функціональності та призначення. Ось деякі відомі варіації онтологічних програмних засобів:

- OWL API.
- PoolParty.
- Ontolingua.
- Protégé.

OWL API – це Java API для роботи з мовою OWL (Web Ontology Language) і онтологіями. Він розроблений для створення, редагування та опрацювання онтологічних знань. Перевагами OWL API є підтримка OWL. OWL API підтримує специфікацію OWL, яка є стандартом для представлення семантичних мереж і онтології. Вона дозволяє створювати і редагувати онтологію у відповідності з цим стандартом [26].

OWL API є потужним інструментом для роботи з онтологіями, але він також має свої недоліки та обмеження, ось деякі з них. OWL API вимагає глибокого розуміння онтологічних концепцій і структур даних, які можуть бути складними для початківців. Робота з онтологіями може бути складним завданням і навіть досвідчені розробники можуть стикатися з труднощами в процесі створення та обслуговування складних онтологій. Обсяги пам'яті, необхідні для роботи з великими або складними онтологіями, можуть бути значними. Це може створити проблеми, якщо не вистачає ресурсів для ефективної обробки даних. Управління версіями онтології може бути складним завданням, а OWL API не надає вбудованих інструментів для легкої роботи з версіями онтології та їх об'єднання. OWL API – це API на основі Java, і він не має

графічного інтерфейсу для створення або редагування онтології. Це може зробити процес створення онтології менш інтуїтивним для користувачів, які не мають досвіду програмування. Відсутність вбудованої роботи з реляційними базами даних.

PoolParty – це комерційний продукт, який надає інтегровані інструменти для роботи з онтологіями та семантичними технологіями. Він використовується для створення, управління та застосування семантичних моделей знань у різних аспектах, включаючи розуміння текстів, пошуку інформації та багато іншого. PoolParty надає інтегровані пакети, включаючи інструменти для моделювання онтології, автоматичного видалення семантики з текстів, розмічення даних та інструментів візуалізації знань. Це робить його потужним інструментом для створення та управління семантичними даними. PoolParty підтримує ключові стандарти семантичного вебу, такі як RDF (Resource Description Framework), OWL (Web Ontology Language) і SPARQL (SPARQL Protocol and RDF Query Language) [26]. Це робить його сумісним із семантичними інструментами та інфраструктурою. Платформа підтримує візуалізацію семантичних моделей знань. Це допомагає користувачам краще розуміти структуру та взаємозв'язки між концепціями. Він може легко інтегруватися з існуючими системами та джерелами даних, що дозволяє використовувати семантичні дані в різних контекстах.

Серйозним недоліком такого інструменту є висока вартість ліцензії для малих компаній або проектів з обмеженим бюджетом. Ontolingua – це була одна з ранніх онтологічних мов і система для роботи з онтологіями та семантичними даними. Вона була розроблена в Мітському інституті технологій (MIT) і використовувалася для створення та управління онтологіями з надання опису знань у різних сферах, таких як біологія, медицина, інформаційні системи та інші. Підтримувала формальну мову, проте має значні недоліки. Один із найбільших недоліків Ontolingua – це те, що розробка та підтримка системи були припинені на деякому етапі. Відсутність активного розвитку означає, що система може бути несумісною із сучасними стандартами та потребами. Ontolingua була складною системою для новачків, і для її використання потрібно було мати глибоке розуміння семантичних технологій та формальних методів. Ontolingua в основному базувалася на текстовому інтерфейсі, що

зробило її менш доступною для користувачів, які звикли до графічних інтерфейсів. Ontolingua була спрямована в основному на англomовних користувачів та мала обмежену підтримку інших мов.

Protégé – це відома і безкоштовна платформа для створення, редагування та управління онтологіями. Вона підтримує мови OWL і RDF і має багатий набір редагувальних інструментів для моделювання онтології. Protégé дозволяє імпортувати, експортувати та обробляти онтологію для різних виробництв [26]. Наведемо основні переваги [27]:

- Графічний інтерфейс: Protégé надає графічний інтерфейс, який спрощує створення та редагування онтології. Ви можете перевірити класи, властивості, індивідуали та зв'язки між ними.

- Підтримка OWL.

- Плагіни і розширення: Protégé має розширену систему плагінів, яка дозволяє розширити його.

- Підтримка імпорту та експорту.

- Підтримка спеціального розширення: Protégé підтримує спеціалізоване розширення для роботи з конкретними доменами, такими як медицина, біологія та географія.

- Валідація і перевірка правил: ви можете застосувати правила та обмеження для вашої онтології та перевірити їх відповідність.

- Підтримка спільної роботи: Protégé дозволяє кільком користувачам спільно працювати над однією онтологією та вести версії.

До недоліків можна віднести: складність використання для новачків, оскільки для досягнення повного розуміння функціональності і можливостей інструменту може знадобитися час та навчання. Велика кількість опцій і налаштувань, Protégé має велику кількість функцій, що може спричинити завантаженість для користувачів, які шукають простий інструмент.

Наведемо переваги та недоліки цих інструментів та проаналізуємо їх у таблиці 3.1.

Таблиця 3.1 – Порівняльна характеристика онтологічних програмних засобів

Назва	Переваги	Недоліки
OWL API	Підтримка OWL	Відсутній графічний інтерфейс. Необхідний великий обсяг пам'яті. Відсутність вбудованої роботи з реляційними базами даних.
PoolParty	Підтримка OWL. Графічний інтерфейс. Підтримка імпорту та експорту. Може легко інтегруватися з існуючими системами та джерелами даних.	Висока вартість ліцензії.
Ontolingua	Функції імпорту та моделювання. Підтримка формальної мови.	Відсутній графічний інтерфейс. Система може бути несумісною із сучасними стандартами та потребами. Обмежена підтримка інших мов.
Protégé	Підтримка OWL. Графічний інтерфейс. Плагіни і розширення. Підтримка імпорту та експорту. Валідація і перевірка правил. Підтримка спільної роботи. Безкоштовність використання.	Навантаженість інструментів.

Проаналізувавши характеристики, можна зробити висновок, що найкращим інструментом є Protégé, оскільки підтримує формальну мову, має графічний інтерфейс, має інструментарій для роботи з запитамі, сама програма є безкоштовною та знаходиться у вільному доступі.

Для побудови запитів було вирішено використовувати SPARQL. SPARQL – це мова запитів, яка дозволяє правильно і точно визначити, яку інформацію потрібно отримати з графів RDF. Запити SPARQL подібні до SQL-запитів для реляційних баз даних, але вони працюють із семантичними даними. SPARQL надає багатий

синтаксис для визначення умов пошуку, об'єднання та фільтрації даних. Ви можете виразити складні запити, які охоплюють багато типів даних і відносини між ними. SPARQL дозволяє об'єднувати дані з різних джерел і розподілених систем [28].

SPARQL використовується в багатьох семантичних веб-додатках, семантичних репозиторіях та інших проектах, де важливо робити запити до семантичних даних і отримувати значущу інформацію.

3.2 Реалізація онтологічної бази знань з гри «Dota 2»

Першим кроком реалізації онтологічної бази знань є побудова структури класів та підкласів. За допомогою функцій Add Subclass, Add Sibling class, Delete classes створимо необхідні класи у середовищі Protégé (рис. 3.1).

Ontograp є стандартним інструментом для візуалізації онтологій і є типом плагіну SlotWidget, вбудованим у Protege за замовчуванням[29]. Такий інструмент є досить зручним і саме його буде використано для відображення онтологічної бази знань з гри «Dota 2». Проте цей плагін має деякі обмеження :

- необхідність розкривати усі гілки вручну;
- необхідність відновлення правила і відношень при встановленні неправильних властивостей між класами;
- не відображає у структурі окремих індивідів.

OntoGraf підтримує інтерактивну навігацію зв'язками розроблюваних онтологій OWL. Підтримуються різні макети для автоматичної організації структури онтології. Підтримуються різні зв'язки: підклас, індивід, властивості об'єкта домену/діапазону та еквівалентність. Зв'язки та типи вузлів можна відфільтрувати, щоб допомогти створити потрібне подання. Графічне представлення онтології гри «Dota 2» включатиме орієнтований граф, який буде відображений у Protégé за допомогою інструменту Ontograp. Повна версія цього графу подана на рисунку 3.2.

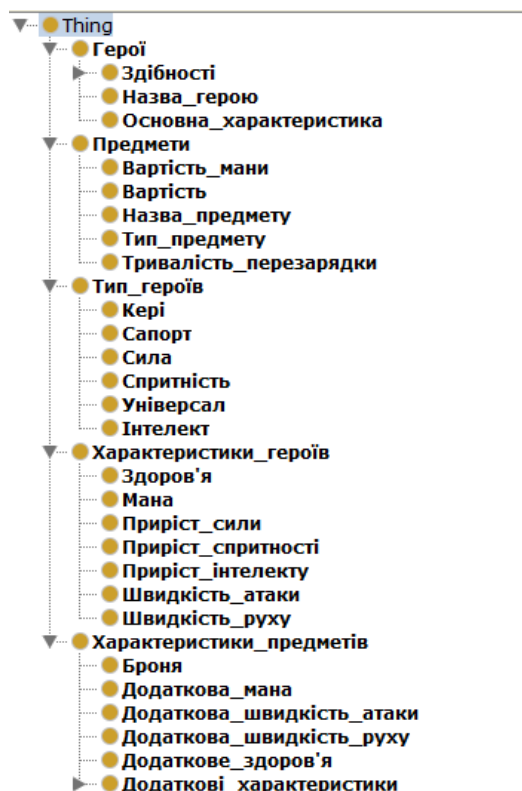


Рисунок 3.1 – Розробка класів онтології

Побудова онтографу (онтологічного графу) зазвичай починається з визначення основних концептів (класів) та їхніх зв'язків. Ось кілька ключових етапів, які часто застосовуються:

1. Визначення основних класів: Це основні концепти, які повинні бути представлені в онтографі. Наприклад, у домені гри Dota 2 такими класами можуть бути Герої, Предмети, Механіки тощо.

2. Визначення властивостей: Це характеристики або атрибути класів. Наприклад, для класу Герої це можуть бути властивості, як здоров'я, мана, рівень.

3. Встановлення відносин: Визначення зв'язків між класами, наприклад, "належить до", "має властивість" або "використовує". Наприклад, герої можуть мати зв'язки з класами Предмети (герой може мати предмет), або Механіки (герой може мати здібності).

4. Інстанціювання (створення індивідів): Після визначення класів і властивостей можна створювати індивідів, що належать до цих класів. Наприклад,


```

</owl:Class>

<!-- http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Активні -->

<owl:Class
rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Активні">

  <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Здібності"/>

</owl:Class>

```

Надалі потрібно створити властивості об'єктів між ними. Властивості в OWL можуть утворювати ієрархії, називаючись підвластивостями, так само, як і класи, оскільки класи, описані вище, характеризуються своїми супервластивостями. Властивості в OWL служать для опису бінарних відносин, в той час як властивості типів даних визначають зв'язки між екземплярами та конкретними значеннями даних. У мові OWL є певні обмеження:

- обмеження кардинальності; це дозволяє обирати кількість відношень в якому може бути застосований бажаний клас;
- обмеження кванторів;
- обмеження змінних.

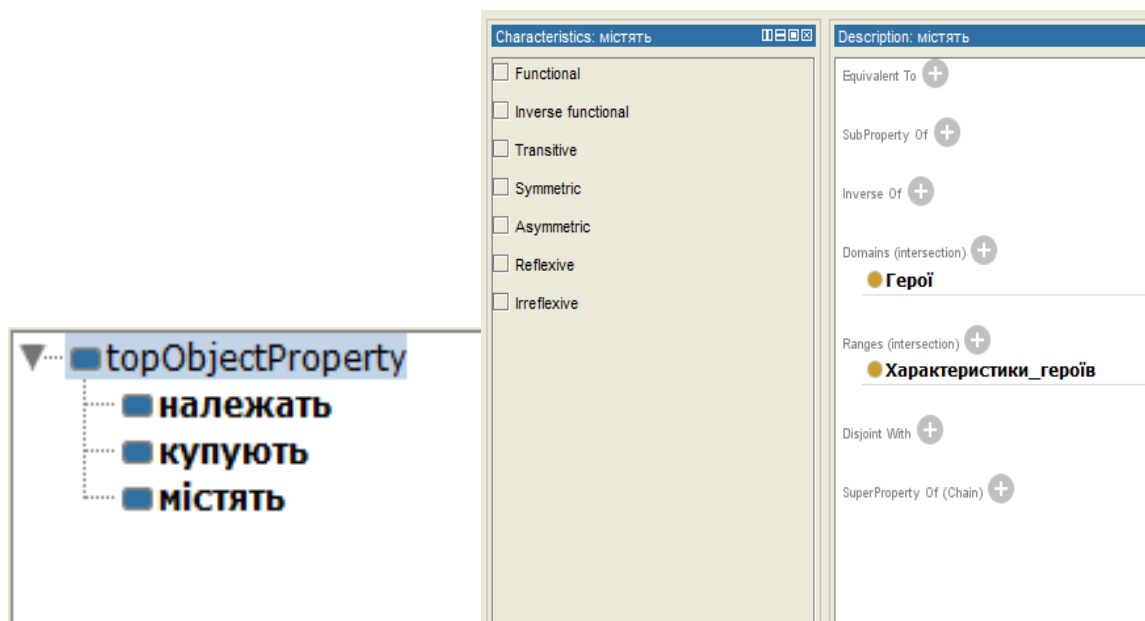


Рисунок 3.3 – Створення властивостей між об'єктами

На рисунку 3.4 показані розроблені властивості для типів даних. Для кожної властивості типів даних є свій домен та тип підтримувальних даних.

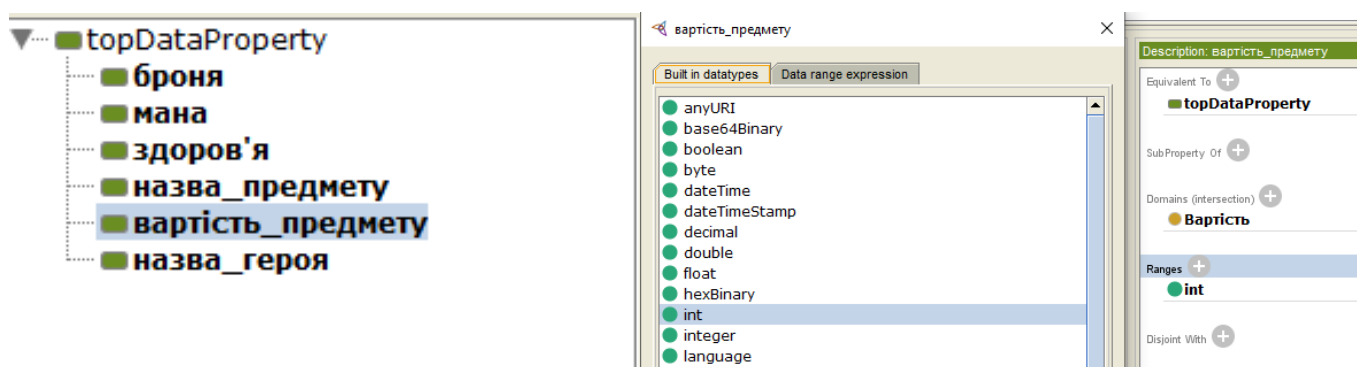


Рисунок 3.4 – Розробка властивостей типів даних

Мова OWL надає можливість створювати екземпляри та визначати їх характеристики. Екземпляри класів можуть використовуватись у властивостях класів, зокрема в обмеженнях типу «має значення» (hasValue).

Створення індивідів, або екземплярів класів, є процесом, який відображає онтологію, виявляючи специфічні об'єкти у запропонованому контексті. За допомогою індивідів онтологія ілюструє дійсні об'єкти, які належать до класів, підтверджуючи структуру онтопроцесу поряд із відповідними даними.

Використання такого підходу в створенні індивідів дозволяє отримувати масштабовані та повні онтологічні структури, що відповідають реальному світові під час забезпечення стабільної форми даних. Вирішуючи задачі і цілі з формування і підтримки актуальності масштабів даних, системи, що використовують індивіди, можуть забезпечити кращу якість класифікації, категоризації та відображення різноманітної інформації в межах домену.

Для того щоб створити екземпляр класу, необхідно перейти на вкладку «Individuals by class», обрати клас та натиснути на іконку створення екземпляру (рис. 3.5)

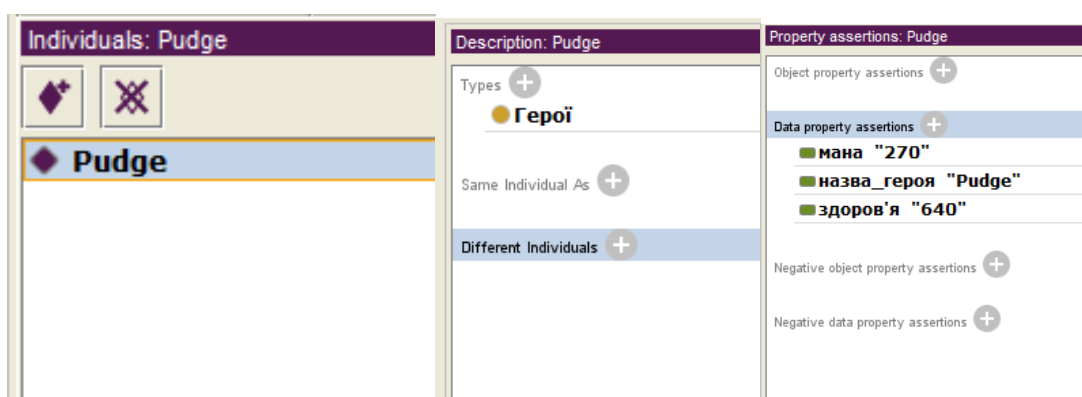


Рисунок 3.5 – Створення індивідів

На рисунку 3.6 зображена частина графу онтології, що описує зв'язки між підкласом «Основні характеристики» з його дочірніми класами. Також на рисунку 3.7 зображені індивіди класів, що належать до цих дочірніх класів.

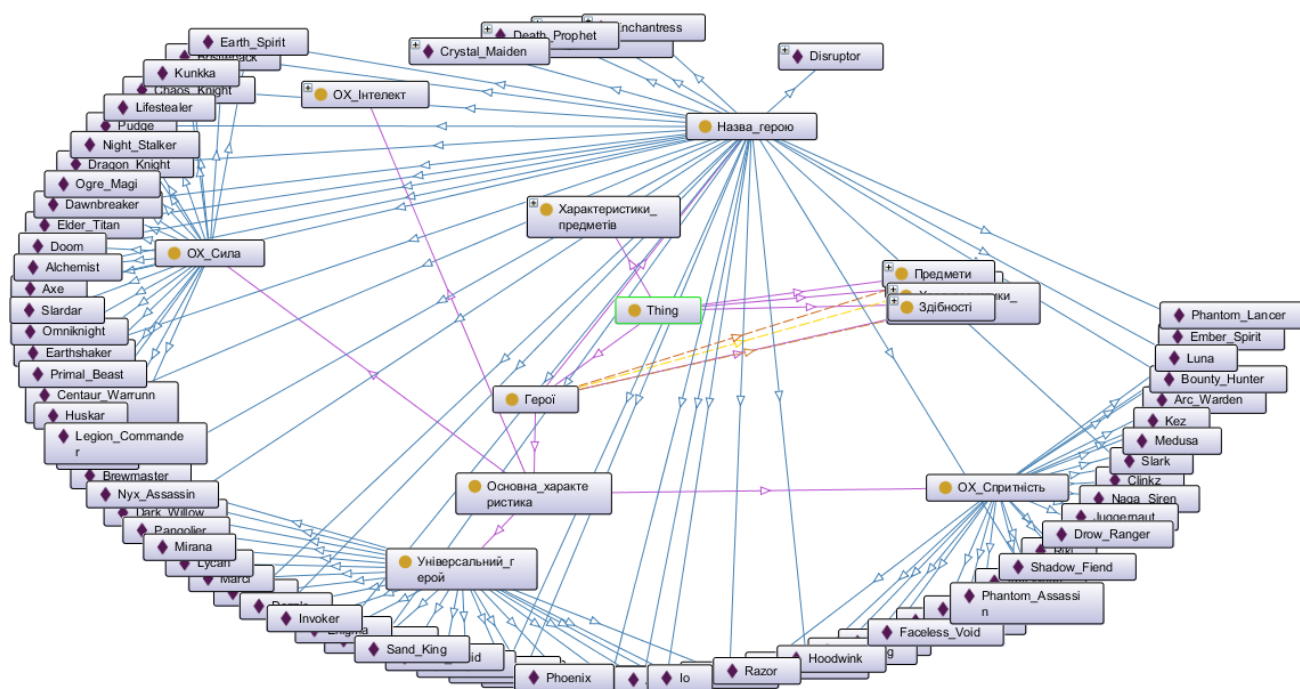


Рисунок 3.6 – Частина графу онтології гри «Dota 2»

Ці приклади демонструють, як індивіди створюються в онтології, а також як правила використовуються для автоматичного виведення нових властивостей або оновлення існуючих. Цей підхід дозволяє ефективно керувати структурованими даними та автоматизувати процеси управління бібліотечним каталогом.

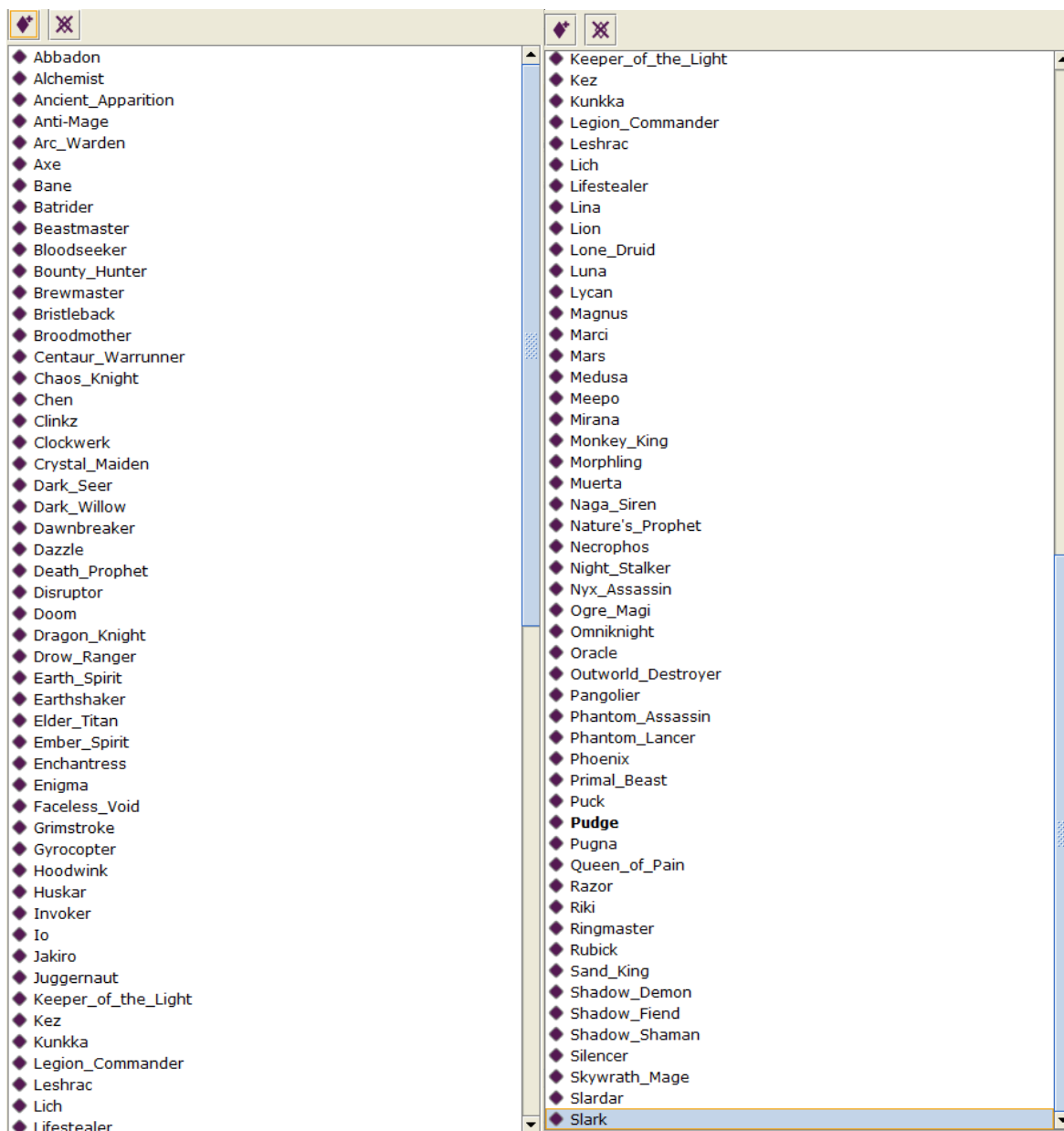


Рисунок 3.7 – Індивіди класу «Герої»

У кодї цієї онтологічної системи індивіди будуть відображатися таким чином:

```
<owl:NamedIndividual
```

```
  rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Alchemist">
```

```
    <rdf:type
```

```
      rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Герої"/>
```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Назва_герою"/>

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#ОХ_Сила"/>

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Основна_характеристика"/>
</owl:NamedIndividual>

```

```

    <!-- http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Ancient_Apparition -->

```

```

    <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Ancient_Apparition">

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Герої"/>

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Назва_герою"/>

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#ОХ_Інтелект"/>

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Основна_характеристика"/>
</owl:NamedIndividual>

```

```
<!-- http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Anti-Mage -->
```

```
<owl:NamedIndividual
rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Anti-Mage">
  <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Герої"/>
  <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Назва_герою"/>
  <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#ОХ_Спритність"/>
  <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Основна_характеристика"/>
</owl:NamedIndividual>
```

```
<!-- http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Arc_Warden -->
```

```
<owl:NamedIndividual
rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Arc_Warden">
  <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Герої"/>
```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Назва_герою"/>

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#ОХ_Спритність"/>

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Основна_характеристика"/>

```

```

</owl:NamedIndividual>

```

```

    <!-- http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Ахе -->

```

```

    <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Ахе">

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Герої"/>

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Назва_герою"/>

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#ОХ_Сила"/>

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Основна_характеристика"/>
    </owl:NamedIndividual>

```

```

    <!-- http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Bane -->

```

```

    <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Bane">

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Герої"/>

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Назва_герою"/>

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Основна_характеристика"/>

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Універсальний_герой"/>
    </owl:NamedIndividual>

```

3.3 Написання стандартних SPARQL запитів для пошуку інформації

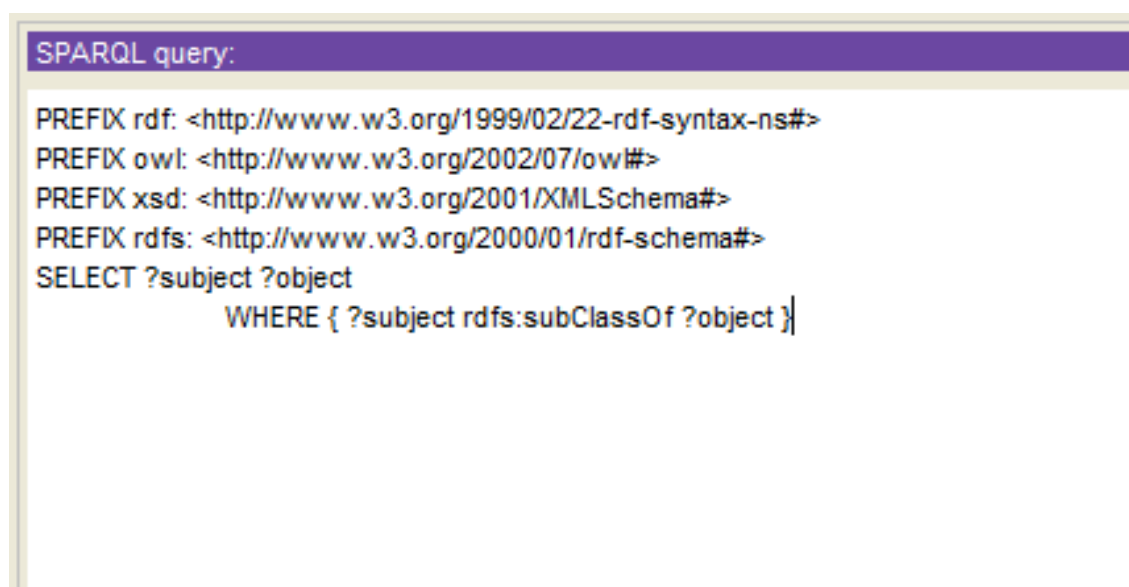
SPARQL – це мова запитів для взаємодії з даними та є однією з ключових технологій семантичної павутини. Вона призначена не лише для отримання даних шляхом виконання запитів, але й для зміни даних, таких як видалення існуючої

інформації або додавання нових даних до бази знань. Надання SPARQL-точок доступу є рекомендованою практикою при публікації даних у всесвітньому павутинні [30].

Для того щоб виконати запит SPARQL, потрібно перейти до вкладки «SPARQL Query» (рис. 3.8). Якщо вкладка не відображається, в меню Window переходимо по вкладці Tabs і виберемо «SPARQL Query».

Для створення запитів на мові SPARQL спочатку потрібно в текстовому полі вкладки «SPARQL Query» задати простори імен, які будуть використовуватися. В цій розробці використовуються простори імен. При чому простори rdf, owl, rdfs та xsd є стандартними і прописуються автоматично.

За замовчуванням на цій вкладці буде відображатись:



```
SPARQL query:
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?subject ?object
WHERE { ?subject rdfs:subClassOf ?object }
```

Рисунок 3.8 – Вигляд SPARQL запитів

Для виконання цього запиту потрібно натиснути «Execute» і під кодом відобразиться результат у вигляді таблиці «subject» та «object».

На рисунку 3.9 можна побачити результат цього запиту

subject	object
спритність	Додаткові_характеристики
Активні	Здібності
Приріст_спритності	Характеристики_героїв
Швидкість_атаки	Характеристики_героїв
Мана	Характеристики_героїв
Універсальний_герой	Основна_характеристика
Поліпшений	Тип_предмету
Додаткове_здоров'я	Характеристики_предметів
Основний	Тип_предмету
Броня	Характеристики_предметів
Здоров'я	Характеристики_героїв
Тривалість_перезарядки_активної_здібності	Активні
Додаткові_характеристики	Характеристики_предметів
Рівність_сил	Характеристики_героїв

Рисунок 3.9 – Результат роботи запиту

До мови SPARQL належать чотири варіанта запитів: Запит типу SELECT використовується для витягування необроблених значень із точки доступу SPARQL, результати представляються у вигляді таблиці. Запит типу CONSTRUCT застосовується для витягування інформації з точки доступу SPARQL у форматі RDF та перетворення результатів до визначеної форми. Запит типу ASK використовують для створення запитів типу істина/хибність. Запит DESCRIBE використовують для отримання опису RDF-ресурсу. Реалізація запиту DESCRIBE визначається розробником SPARQL-точки доступу. Кожна форма запиту має включати в себе блок WHERE, щоб обмежити запит. Нижче наведена частина часто використовуваних ключових слів в SPARQL запитів. Повний список є в офіційній документації:

DISTINCT – показує у відповіді на запит, що він буде унікальним буде унікальним; LIMIT – використовується для задання максимальних кількостей виведених результатів; OFFSET – дозволяє не показувати в результаті перші n рішень; ORDER BY – використовується для сортування результатів за збільшенням чи зменшенням; PREFIX – скорочує URI; OPTIONAL – визначає необов'язковий шаблон; GRAPH – використовується для формування запитів, що будуть застосовані до шаблону іменованих графів. SPARQL-точка доступу, також відома як SPARQL-endpoint, представляє собою службу, сумісну із мовою запитів SPARQL. Ця точка доступу дозволяє користувачам ставити запити до бази знань та отримувати результати у різних форматах. Отже, SPARQL-точки доступу призначені для надання зручного інтерфейсу до бази знань у вигляді сервісу. Існують два основних типи точок доступу: загального та локального призначення. Точки доступу загального призначення можуть обробляти запити до будь-яких зазначених RDF-документів, що

знаходяться в Мережі. З іншого боку, локальні точки доступу можуть отримувати дані лише від одного визначеного ресурсу [31].

3.4 Тестування онтологічної бази знань з гри «Dota 2»

Тестування онтологічної моделі полягає в перевірці правильності та відповідності властивостей, класів та відносин між об'єктами в онтології. Добре побудована онтологія повинна точно відображати відносини між об'єктами та адекватно реагувати на запити або взаємодії. Тестування проходить через ряд кроків, таких як перевірка синтаксису, внутрішньої узгодженості, коректності роботи правил та відповідності результатів запитів.

Одним з методів тестування онтології є створення та виконання SPARQL-запитів, що відображають типові сценарії використання моделі. Застосування SPARQL-запитів дозволяє перевірити, наскільки ефективно і точно онтологія може відповісти на різноманітні запити чи вимоги користувачів, а також допомагає виявити потенційні недоліки або проблеми в онтологічній моделі [32].

Для отримання максимальної користі від етапу тестування, рекомендується створити цілеспрямований набір запитів, що протестують різні аспекти видавничої справи, з використанням різних ресурсів та відносин між об'єктами онтології.

Ітеративний процес тестування може сприяти удосконаленню онтологічної моделі, також може допомогти зрозуміти зміни та модифікації, які повинні бути внесені для більш точного відображення реального світу, поліпшення корисності та працездатності системи. Тестування включає перевірку роботи моделі за допомогою запитів SPARQL. Застосування SPARQL-запитів допомагає перевірити, як добре онтологічна модель може відповісти на найпоширеніші сценарії використання.

Оцінка якості онтології включає різні аспекти, серед яких можна виділити структурні, функціональні та орієнтовані на користувача елементи. Структурні аспекти стосуються правильності побудови таксономії сутностей в онтології, тобто, як грамотно організовано її структуру. Функціональні аспекти оцінюють, наскільки онтологія підходить для вирішення конкретних задач, які на неї покладаються.

Користувацькі аспекти враховують, як зручно кінцевим користувачам працювати з онтологією, чи є у них необхідні інструменти для ефективного використання.

Деякі аспекти якості онтології можна перевіряти автоматично, зокрема, перевірку структурних елементів. Однак для оцінки функціональних аспектів часто потрібна експертна оцінка, яку можуть надати автори онтології або потенційні користувачі системи. Оцінку якості доцільно розділяти на два основні етапи: верифікацію та валідацію. Верифікація включає перевірку формальної та структурної цілісності онтології, тоді як валідація зосереджується на перевірці функціональних аспектів, щоб переконатися в її ефективності для вирішення конкретних задач.

Щоб оцінити коректність роботи пошукової системи на основі онтології, планується залучити 10 користувачів, які оцінять роботу запитів за десятибальною системою. Результати оцінювання будуть занесені в таблицю для подальшого аналізу. Кожен користувач оцінюватиме кілька критеріїв, серед яких: «Результат» — наскільки задоволений користувач отриманим результатом запиту, «Відсутність помилок» — чи не було помилок у відповіді (якщо помилок немає, оцінка максимальна), «Час відгуку» — наскільки швидко система надала результат (мінімальний час отримує максимальну оцінку), та «Користувацький досвід» — загальне враження користувача від тестування системи та її можливостей.

Таблиця 3.2 – Результати оцінювання розробленої онтології

Користувачі	Результат	Відсутність помилок	Час відгуку	Користувацький досвід	Середнє значення
1	8	9	10	9	90%
2	9	8	9	8	85%
3	9	9	10	10	95%
4	8	9	8	9	85%
5	10	8	9	8	88%
6	9	10	9	10	95%
7	10	10	10	10	100%
8	8	9	9	9	88%
9	9	10	9	8	90%
10	10	9	10	9	95%

Для обробки результатів було обрано метрику Single Usability Metrics, результат обчислено за формулою:

$$SUM = \frac{SM1+SM2+...+SMn}{n}, \quad (3.1)$$

де $SM1, SM2, \dots, SMn$ – середнє значення оцінювання одного користувача;

n – кількість середніх значень оцінки користувачів.

Середній показник SUM для всіх користувачів дорівнює 91%, що є досить гарним показником для онтологічних баз знань.

Проведемо подібну перевірку для сервісів-аналогів Dota 2 Fandom : Dota 2 Wiki, офіційний сайт Dota 2 – Dota2.com, Liquipedia, власне самої гри «Dota 2», де так само тестування проведуть десять користувачів та виставлять свої бали від 1 до 10.

У таблицях 3.3, 3.4, 3.5, 3.6 наведено усереднені результати, середня оцінка Dota 2 Fandom : Dota 2 Wiki становить 84%, офіційний сайт Dota 2 – Dota2.com – 83,7%, Liquipedia - 85%, а власне можливості самої гри «Dota 2» становлять 83%.

Таблиця 3.3 – Результати оцінювання «Dota 2 Fandom : Dota 2 Wiki»

Користувачі	Середнє значення
1	90
2	95
3	92
4	88
5	77
6	65
7	84
8	90
9	96
10	66

Таблиця 3.4 – Результати оцінювання «офіційний сайт Dota 2 – Dota2.com»

Користувачі	Середнє значення
1	90
2	89
3	76
4	59
5	78
6	94
7	74
8	94
9	91
10	92

Таблиця 3.5 – Результати оцінювання «Liquipedia»

Користувачі	Середнє значення
1	95
2	82
3	86
4	90
5	92
6	67
7	78
8	90
9	81
10	89

Таблиця 3.6 – Результати оцінювання «гра «Dota 2»»

Користувачі	Середнє значення
1	92
2	82
3	80
4	74
5	80
6	68
7	94
8	90
9	88
10	82

На рисунку 3.10 подано порівняльний графік сервісів представлення бази знань з гри «Dota 2».

На графіку (рис. 3.10) чітко видно, що незважаючи на деякі відмінності через окремі експерименти експертів, розроблена онтологічна модель випереджає аналоги.

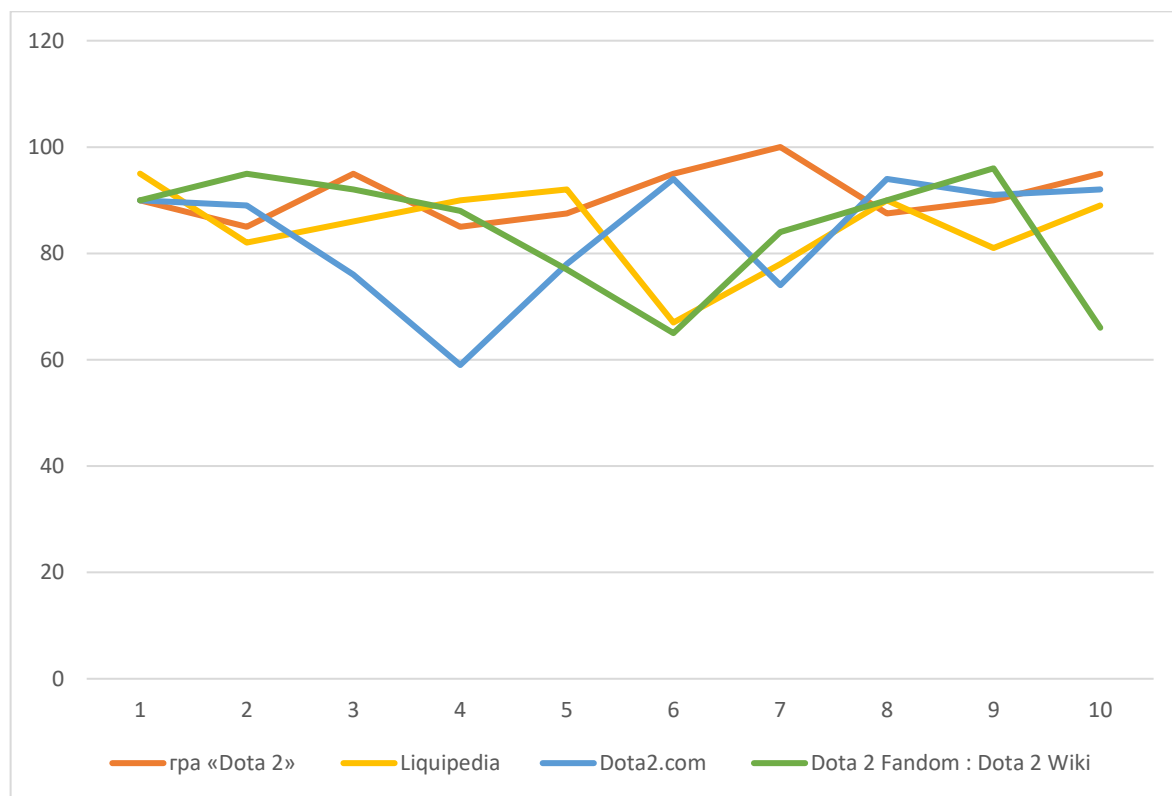


Рисунок 3.10 – Порівняльний графік сервісів представлення бази знань гри «Dota 2»

Також було використано та протестовано Reasoner, який допомагає робити логічні висновки на основі введених даних, що покращує розуміння та пошук інформації у онтологічній базі знань.

Основною відмінністю від аналогів є наявність вибірки та сортування даних за допомогою SPARQL-запитів, яка допомагає підібрати, наприклад, найкращого героя відштовхуючись від вподобань та потреб користувача та дасть лише потрібну інформацію, яку користувач потребує. Наприклад, користувач може зробити вибірку лише сапорти, універсали, швидка шкода та відсортувати по здоров'ю.

3.5 Висновок до розділу 3

У цьому розділі на основі аналізу програмних засобів розробки онтологічних баз знань визначено поточний стан розвитку онтологічних програмних засобів, виділено OWL API, PoolParty, Ontolingua, Protégé. Обрано Protégé як найбільш ефективний засіб для реалізації онтології гри «Dota 2» за рахунок доступності функціоналу та зручності використання.

Побудовано повну структуру класів та підкласів, розроблено відносини між класами та розширено програму екземплярами класів. Для візуалізації був обраний інструмент Ontograf на основі його зручності використання та зрозумілості результату візуалізації.

Під час тестування ефективності онтологічної бази знань гри «Dota 2» було створено декілька SPARQL запитів для пошуку інформації в онтологічній базі знань. Досліджено можливості мови SPARQL та її інтеграцію з Protégé, а також протестовано роботу запитів на практиці.

Розробка запитів для тестування роботи семантичного пошуку підтвердила коректність розробленої онтології. Проведено оцінювання онтологічної бази знань гри «Dota 2» у вигляді колективної експертизи та порівняно з основними аналогами. Порівняльне тестування ефективності онтологічної бази знань та її аналогів у вигляді онлайн-ресурсу Dota 2 Fandom : Dota 2 Wiki, офіційний сайт Dota 2 – Dota2.com, Liquipedia, власне самої гри «Dota 2» показало, що ефективність розроблювальної онтологічної моделі є вищою, а загальна оцінка становить 91% зі 100%.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання комерційного потенціалу розробки

Метою проведення комерційного та технологічного аудиту є оцінювання комерційного потенціалу інформаційної веб-системи аналізу динаміки географічної структури зовнішньої торгівлі товарами України.

Для проведення технологічного аудиту було залучено 3-х незалежних експертів Вінницького національного технічного університету кафедри комп'ютерних наук: к.т.н., доц. Озеранський В. С., к.т.н., доц. Сілагін О. В., к.т.н., доц. Крилик Л. В. Для проведення технологічного аудиту було використано таблицю 4.1 в якій за п'ятибальною шкалою використовуючи 12 критеріїв здійснено оцінку комерційного потенціалу.

Таблиця 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

Продовження таблиці 4.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витратити значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років

Продовження таблиці 4.1

Критерії оцінювання та бали (за 5-ти бальною шкалою)					
Кри-терій	0	1	2	3	4
				3-х до 5-ти років	
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Таблиця 4.2 – Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ, розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0-10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

В таблиці 4.3 наведено результати оцінювання експертами комерційного потенціалу розробки.

Таблиця 4.3 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада експерта		
	Сілагін О. В.	Озеранський В. С	Крилик Л. В.
	Бали, виставлені експертами:		
1	4	1	2
2	3	1	1
3	3	5	4
4	4	4	2
5	2	4	2
6	1	2	3
7	2	2	3
8	3	2	4

Продовження таблиці 4.3

Критерії	Прізвище, ініціали, посада експерта		
	Сілагін О. В.	Озеранський В. С	Крилик Л. В.
	Бали, виставлені експертами:		
9	4	4	2
10	3	3	2
11	5	3	4
12	2	3	2
Сума балів	СБ ₁ =35	СБ ₂ =33	СБ ₃ =34
Середньоарифметична сума балів $\overline{СБ}$	$\overline{СБ} = \frac{\sum_1^3 СБ_i}{3} = \frac{31 + 34 + 34}{3} = 34$		

Середньоарифметична сума балів, розрахована на основі висновків експертів склала 34 бали, що згідно таблиці 4.2 вважається, що рівень комерційного потенціалу проведених досліджень є вище середнього.

Інформаційної веб-система аналізу динаміки географічної структури зовнішньої торгівлі товарами України, а саме програма, яка допомагає проводити моніторинг, аналізувати та порівнювати дані. Дана програма буде цікава різним організаціям та компаніям в яких цільовий ринок направлений закордон.

4.2 Прогнозування витрат на виконання науково-дослідної роботи

Витрати, пов'язані з проведенням науково-дослідної роботи групуються за такими статтями: витрати на оплату праці, витрати на соціальні заходи, матеріали, паливо та енергія для науково-виробничих цілей, витрати на службові відрядження, програмне забезпечення для наукових робіт, інші витрати, накладні витрати.

1. Основна заробітна плата кожного із дослідників $З_0$, якщо вони працюють в наукових установах бюджетної сфери визначається за формулою:

$$Z_0 = \frac{M}{T_p} * t \text{ (грн)}, \quad (4.1)$$

де M – місячний посадовий оклад конкретного розробника (інженера, дослідника, науковця тощо), грн.;

T_p – число робочих днів в місяці; приблизно $T_p \approx 21...23$ дні;

t – число робочих днів роботи дослідника.

Для розробки програмні засоби необхідно залучити програміста з посадовим окладом 17000 грн. Кількість робочих днів у місяці складає 35, а кількість робочих днів програміста складає 22. Зведемо сумарні розрахунки до таблиця 4.4.

$$Z_0 = \frac{18000}{22} * 10 = 8182 \text{ (грн)} - \text{основна заробітна плата керівника.}$$

$$Z_0 = \frac{15500}{22} * 35 = 24589 \text{ (грн)} - \text{основна заробітна плата програмного інженера.}$$

Таблиця 4.4 – Заробітна плата дослідника в науковій установі бюджетної сфери

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату грн.
Керівник	18000	818,2	10	8182
Програмний інженер	15500	702,5	35	24589
Всього				32771

2. Розрахунок додаткової заробітної плати робітників

Додаткова заробітна плата Z_d всіх розробників та робітників, які приймали участь в розробці нового технічного рішення розраховується як 10 - 12 % від основної заробітної плати робітників.

На даному підприємстві додаткова заробітна плата начисляється в розмірі 10% від основної заробітної плати.

$$З_д = (З_о + З_р) * \frac{Н_{\text{дод}}}{100\%} \quad (4.2)$$

$$З_д = 0,1 * 32771 = 3277 \text{ (грн).}$$

3. Нарахування на заробітну плату $Н_{зп}$ дослідників та робітників, які брали участь у виконанні даного етапу роботи, розраховуються за формулою:

$$Н_{зп} = (З_о + З_д) * \frac{\beta}{100}, \quad (4.3)$$

де $З_о$ – основна заробітна плата розробників, грн.;

$З_д$ – додаткова заробітна плата всіх розробників та робітників, грн.;

β – ставка єдиного внеску на загальнообов'язкове державне соціальне страхування, % .

Дана діяльність відноситься до бюджетної сфери, тому ставка єдиного внеску на загальнообов'язкове державне соціальне страхування буде складати 22%, тоді:

$$Н_{зп} = (32771 + 3277) * \frac{22}{100} = 8524,34 \text{ (грн).}$$

4. Витрати на комплектуючі вироби, які використовують при виготовленні одиниці продукції, розраховуються, згідно їх номенклатури, за формулою:

$$К = \sum_{i=1}^n Н_i * Ц_i * K_i, \quad (4.5)$$

де $Н_i$ – кількість комплектуючих i -го виду, шт.;

$Ц_i$ – покупна ціна комплектуючих i -го найменування, грн.;

K_i – коефіцієнт транспортних витрат (1,1...1,15).

Таблиця 4.5 – Комплектуючі, що використані на розробку

Найменування матеріалу	Ціна за одиницю, грн.	Витрачено	Вартість витраченого матеріалу, грн.
Папір	150	1	150
Ручка	20	2	40
CD-диск	30	1	30
Флешка	250	1	250
Всього			470
З врахуванням коефіцієнта транспортування			517

5. Програмне забезпечення для наукової роботи включає витрати на розробку та придбання спеціальних програмних засобів і програмного забезпечення необхідного для проведення дослідження.

Для написання магістерської роботи використовувалися редактор Protégé.

6. Амортизація обладнання, комп'ютерів та приміщень, які використовувались під час виконання даного етапу роботи

Дані відрахування розраховують по кожному виду обладнання, приміщенням тощо.

$$A = \frac{Ц \cdot T}{T_{кор} \cdot 12}, \quad (4.6)$$

де Ц – балансова вартість даного виду обладнання (приміщень), грн.;

$T_{кор}$ – час користування;

T – термін використання обладнання (приміщень), цілі місяці.

Згідно пункта 137.3.3 Податкового кодекса амортизація нараховується на основні засоби вартістю понад 2500 грн. В нашому випадку для написання магістерської роботи використовувався персональний комп'ютер вартістю 30000 грн.

$$A = \frac{30000 \cdot 1}{2 \cdot 12} = 1150 \text{ (грн.)}$$

7. До статті «Паливо та енергія для науково-виробничих цілей» відносяться витрати на всі види палива й енергії, що безпосередньо використовуються з технологічною метою на проведення досліджень.

$$B_e = \sum_{i=1}^n \frac{W_{yt} \cdot t_i \cdot C_e \cdot K_{впi}}{\eta_i}, \quad (4.7)$$

де W_{yt} – встановлена потужність обладнання на певному етапі розробки, кВт;

t_i – тривалість роботи обладнання на етапі дослідження, год;

C_e – вартість 1 кВт-години електроенергії, грн;

$K_{впi}$ – коефіцієнт, що враховує використання потужності, $K_{впi} < 1$;

η_i – коефіцієнт корисної дії обладнання, $\eta_i < 1$.

Для написання магістерської роботи використовується персональний комп'ютер для якого розрахуємо витрати на електроенергію.

$$B_e = \frac{0,283 \cdot 350 \cdot 4,32 \cdot 0,75}{0,88} = 307,2 \text{ (грн)}.$$

Витрати на службові відрядження, витрати на роботи, які виконують сторонні підприємства, установи, організації та інші витрати в нашому дослідженні не враховуються оскільки їх не було.

Накладні (загальновиробничі) витрати $B_{нзв}$ охоплюють: витрати на управління організацією, оплата службових відряджень, витрати на утримання, ремонт та експлуатацію основних засобів, витрати на опалення, освітлення, водопостачання, охорону праці тощо. Накладні (загальновиробничі) витрати $B_{нзв}$ можна прийняти як (100...150)% від суми основної заробітної плати розробників та робітників, які виконували дану МКНР, тобто:

$$B_{нзв} = (3_o + 3_p) \cdot \frac{H_{нзв}}{100\%}, \quad (4.8)$$

де $H_{\text{нзв}}$ – норма нарахування за статтею «Інші витрати».

$$B_{\text{нзв}} = 32771 \cdot \frac{100}{100\%} = 32771(\text{грн}).$$

Сума всіх попередніх статей витрат дає витрати, які безпосередньо стосуються даного розділу МКНР

$$B = 32771 + 3277 + 7930,5 + 517 + 1150 + 307,2 + 32771 = 78723,7(\text{грн}).$$

Прогнозування загальних втрат ЗВ на виконання та впровадження результатів виконаної МКНР здійснюється за формулою:

$$ЗВ = \frac{B}{\eta}, \quad (4.9)$$

де η – коефіцієнт, який характеризує стадію виконання даної НДР.

Оскільки, робота знаходиться на стадії науково-дослідних робіт, то коефіцієнт $\beta = 0,9$.

Звідси:

$$ЗВ = \frac{78723,7}{0,9} = 87470,7(\text{грн}).$$

4.3 Розрахунок економічної ефективності науково-технічної розробки

У даному підрозділі кількісно спрогнозуємо, яку вигоду, зиск можна отримати у майбутньому від впровадження результатів виконаної наукової роботи. Розрахуємо

збільшення чистого прибутку підприємства $\Delta\Pi_i$, для кожного із років, протягом яких очікується отримання позитивних результатів від впровадження розробки, за формулою

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_0 * N * \Pi_0 * \Delta N)_i * \lambda * \rho * \left(1 - \frac{v}{100}\right), \quad (4.10)$$

де $\Delta\Pi_0$ – покращення основного оціночного показника від впровадження результатів розробки у даному році.

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки:

Π_0 – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки:

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0,8333$.

ρ – коефіцієнт, який враховує рентабельність продукту. $\rho = 0,25$;

v – ставка податку на прибуток. У 2024 році – 18%.

Припустимо, що ціна за програмний продукт зростає на 600 грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року на 45 шт., протягом другого року – на 35 шт., протягом третього року на 25 шт. Реалізація продукції до впровадження розробки складала 1 шт., а її ціна до складає 10000 грн. Розрахуємо прибуток, яке отримає підприємство протягом трьох років.

$$\Delta\Pi_1 = [600 \cdot 1 + (10000 + 600) \cdot 45] \cdot 0,833 \cdot 0,25 \cdot \left(1 + \frac{18}{100}\right) = 117815.59 \text{ (грн)}.$$

$$\begin{aligned}\Delta\Pi_2 &= [600 \cdot 1 + (10000 + 600) \cdot (45 + 35)] \cdot 0,833 \cdot 0,25 \cdot \left(1 + \frac{18}{100}\right) \\ &= 208983.28 \text{ (грн)}.\end{aligned}$$

$$\begin{aligned}\Delta\Pi_3 &= [600 \cdot 1 + (10000 + 600) \cdot (45 + 35 + 25)] \cdot 0,833 \cdot 0,25 \cdot \left(1 + \frac{18}{100}\right) \\ &= 274103.05 \text{ (грн)}.\end{aligned}$$

4.4 Розрахунок ефективності вкладених інвестицій та періоду їх окупності

Розрахуємо основні показники, які визначають доцільність фінансування наукової розробки певним інвестором, є абсолютна і відносна ефективність вкладених інвестицій та термін їх окупності.

Розрахуємо величину початкових інвестицій PV , які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки.

$$PV = k_{\text{інв}} \cdot 3B, \quad (4.11)$$

де $k_{\text{інв}}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо ($k_{\text{інв}} = 2 \dots 5$).

$$PV = 2 \cdot 87470,7 = 174941,5 \text{ (грн)}.$$

Розрахуємо абсолютну ефективність вкладених інвестицій $E_{\text{абс}}$ згідно наступної формули:

$$E_{\text{абс}} = (\text{ПП} - PV), \quad (4.12)$$

де $\Pi\Pi$ – приведена вартість всіх чистих прибутків, що їх отримає підприємство від реалізації результатів наукової розробки, грн.;

$$\Pi\Pi = \sum_1^T \frac{\Delta\Pi_i}{(1+\tau)^t}, \quad (4.13)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн.;

T – період часу, протягом якого виявляються результати впровадженої НДДКР, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,2;

t – період часу (в роках).

$$\Pi\Pi = \frac{117815.59}{(1 + 0,2)^1} + \frac{208983.28}{(1 + 0,2)^2} + \frac{274103.05}{(1 + 0,2)^3} = 401931.37 \text{ (грн)}.$$

$$E_{abc} = (401931.37 - 174941,5) = 226\,989,8 \text{ (грн)}.$$

Оскільки $E_{abc} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР може бути доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_v . Для цього користуються формулою:

$$E_v = \sqrt[T_{ж}]{1 + \frac{E_{abc}}{PV}} - 1, \quad (4.14)$$

де $T_{ж}$ – життєвий цикл наукової розробки, роки.

$$E_b = \sqrt[3]{1 + \frac{226\,989,8}{174941,5}} - 1 = 0,52 = 52\%.$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \quad (4.15)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2024 році в Україні $d = (0,14 \dots 0,2)$;

f – показник, що характеризує ризикованість вкладень; зазвичай, величина $f = (0,05 \dots 0,1)$.

$$\tau_{min} = 0,16 + 0,05 = 0,21.$$

Так як $E_b > \tau_{min}$ то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_b}. \quad (4.16)$$

$$T_{ок} = \frac{1}{0,52} = 1,9 \text{ (роки)}.$$

Так як $T_{ок} \leq 3 \dots 5$ -ти років, то фінансування даної наукової розробки в принципі є доцільним.

4.5 Висновок до розділу 4

Проведено оцінку комерційного потенціалу інформаційної технології онтологічного моделювання бази знань гри «Dota 2», а саме онологічної бази знань, яка допомагає швидко та зручно взаємодіяти з інформацією про гру «Dota 2».

Прогнозування витрат на виконання науково-дослідної роботи по кожній з статей витрат складе 78723,7 грн. Загальна ж величина витрат на виконання та впровадження результатів даної НДР буде складати 87470,7 грн.

Вкладені інвестиції в даний проект окупляться через 1,9 роки, приведена вартість всіх чистих прибутків, що їх отримає підприємство від реалізації результатів наукової розробки склала 401931,37 грн.

ВИСНОВКИ

При виконанні магістерської кваліфікаційної роботи розв'язано задачу розширення функціональних можливостей інформаційної технології онтологічного моделювання бази знань гри «Dota 2», що сприятиме ефективному пошуку потрібної інформації. Всі завдання, поставлені для реалізації інформаційної технології онтологічного моделювання бази знань гри «Dota 2» виконані в повному обсязі, а саме: проаналізовано предметну галузь гри «Dota 2» та обґрунтовано доцільність розробки інформаційної технології онтологічного моделювання бази знань; проведено порівняння існуючих технологій; розроблено покращену інформаційну технологію онтологічного моделювання бази знань; здійснено моделювання предметної галузі гри «Dota 2» засобами проектування та побудови діаграм, закодовано онтологію в інструментальному середовищі; розроблено онтологічну модель бази знань гри «Dota 2» та загальну структурну схему системи; здійснено програмну реалізацію онтологічної бази знань; виконано тестування програми та проведено аналіз отриманих результатів; обґрунтовано економічну доцільність розробки.

У першому розділі було здійснено детальний огляд гри «Dota 2». Обґрунтовано доцільність розробки інформаційної технології онтологічного моделювання бази знань цієї гри. Використання такої технології може значно полегшити гравцям швидкий доступ до потрібної інформації про гру. Під час огляду предметної галузі було визначено проблеми в існуючих аналогах баз знань та поставлено за мету виправити їх у своїй роботі.

У другому розділі магістерської кваліфікаційної роботи проаналізовано методи подання знань, розроблено моделі з предметної галузі гри «Dota 2» у різних формах подання, в тому числі засобами ER-моделювання та UML-діаграм. Було розроблено покращену інформаційну технологію онтологічного моделювання бази знань гри «Dota 2». Основною відмінністю від аналогів є наявність вибірки та сортування даних за допомогою SPARQL-запитів. На основі аналізу основних сутностей та зв'язків між ними побудовано ER-модель предметної галузі. Обґрунтовано важливість UML-

діаграм та побудовано діаграму класів і діаграму прецедентів бази знань. Здійснено моделювання структури предметної галузі. Проведено кодування онтології шляхом використання інструментарію Protégé. Наведено переваги онтологічного підходу представлення знань та обґрунтовано доцільність його використання. Наведена концептуальна схема онтології гри «Dota 2».

У третьому розділі визначено поточний стан розвитку онтологічних програмних засобів та обрано Protégé як найбільш ефективний засіб для реалізації онтології гри «Dota 2» за рахунок доступності функціоналу та зручності використання. Для візуалізації був обраний інструмент Ontograf на основі його зручності використання та зрозумілості результату візуалізації. Досліджено можливості мови SPARQL та її інтеграцію з Protégé, а також протестовано роботу запитів на практиці. Проведено оцінювання онтологічної бази знань гри «Dota 2» у вигляді колективної експертизи та порівняно з основними аналогами. Порівняльне тестування ефективності онтологічної бази знань та її аналогів у вигляді онлайн-ресурсу Dota 2 Fandom : Dota 2 Wiki, офіційний сайт Dota 2 – Dota2.com, Liquipedia, власне самої гри «Dota 2» показало, що ефективність розроблювальної онтологічної моделі є вищою, а загальна оцінка становить 91% зі 100%.

Мета дослідження розширення функціональних можливостей інформаційної технології онтологічного моделювання баз знань досягається за рахунок появи у покращеній технології нових функцій вибірки та сортування даних за допомогою SPARQL-запитів.

У четвертому розділі було виконано розрахунок витрат на розробку та виготовлення нового технічного рішення, сума яких складає 87470,7 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який потенційно може отримати виробник від реалізації розробленого технічного рішення, знайдено термін окупності витрат виробника та економічний ефект для споживача при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розробка у виробництві та використання дешевша за аналог і є висококонкурентоспроможною. Період окупності складе близько 1,9 роки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Конференції ВНТУ електронні наукові видання, Молодь в науці: дослідження, проблеми, перспективи (МН-2025) [Електронний ресурс]. URL: <https://conferences.vntu.edu.ua/index.php/mn/mn2025/paper/view/22802> (дата звернення: 04.12.24).
2. Wikipedia Індустрія відеоігор [Електронний ресурс]. URL: https://uk.wikipedia.org/wiki/Індустрія_відеоігор (дата звернення: 10.11.24).
3. Індустрія комп'ютерних ігор перемогла кіно [Електронний ресурс]. URL: <https://glavcom.ua/publications/industriya-kompyuternih-igor-peremogla-kino-doslidzhennya-ukrajinskogo-rinku-737655.html> (дата звернення: 10.11.24).
4. Gamer Info — Dota 2 [Електронний ресурс]. URL: <https://gamerinfo.net/uk/dota-2/> (дата звернення: 10.11.24).
5. Dota 2 — огляди та новини на AIN.UA [Електронний ресурс]. URL: <https://ain.ua/tag/dota-2/> (дата звернення: 10.11.24). 92
6. Як новачкам грати в Dota 2 [Електронний ресурс]. URL: <https://cyberscore.live/ua/articles/how-to-play-dota-2-for-beginners/> (дата звернення: 10.11.24).
7. Dota 2 для тих, хто не в темі [Електронний ресурс]. URL: https://ua.tribuna.com/uk/blogs/exclamation_mark/2922146/ (дата звернення: 10.11.24).
8. Dota 2 Heroes [Електронний ресурс]. URL: <https://www.dota2.com/heroes> (дата звернення: 10.11.24).
9. Items - Dota 2 Wiki - Fandom [Електронний ресурс]. URL: <https://dota2.fandom.com/wiki/Items> (дата звернення: 10.11.24).
10. Dota 2 Wikipedia [Електронний ресурс]. URL: https://uk.wikipedia.org/wiki/Dota_2 (дата звернення: 10.11.24).
11. Dota 2 Wiki - Fandom [Електронний ресурс]. URL: https://dota2.fandom.com/wiki/Dota_2_Wiki (дата звернення: 10.11.24).
12. Dota 2 Official Website [Електронний ресурс]. URL: <https://www.dota2.com> (дата звернення: 10.11.24).

13. Liquipedia [Електронний ресурс]. URL: https://liquipedia.net/dota2/Main_Page (дата звернення: 10.11.24).
14. Представлення знань [Електронний ресурс]. URL: <https://uk.wikipedia.org/wiki> (дата звернення: 10.11.24).
15. Методи представлення знань [Електронний ресурс]. URL: <https://uk.wikipedia.org/wiki> (дата звернення: 10.11.24).
16. Логічна модель представлення знань [Електронний ресурс]. URL: <https://www.wikidata.uk-ua.nina.az> (дата звернення: 10.11.24).
17. ПРОДУКЦІЙНА МОДЕЛЬ [Електронний ресурс]. URL: https://stud.com.ua/158210/informatika/produksionnaya_model#google_vignette (дата звернення: 10.11.24). 93
18. Переваги та недоліки продукційних моделей [Електронний ресурс]. URL: <https://posibniki.com.ua/post-perevagi-ta-nedoliki-produkciynih-modeley> (дата звернення: 10.11.24).
19. Модель представлення знань в адаптивній системі дистанційного навчання та контролю знань «EduPRO» [Електронний ресурс]. URL: <http://dspace.nbu.gov.ua/bitstream/handle/123456789/60073/49-Fedoruk.pdf?sequence=1> (дата звернення: 10.11.24).
20. Семантична мережа [Електронний ресурс]. URL: https://uk.wikipedia.org/wiki/%D0%A1%D0%B5%D0%BC%D0%B0%D0%BD%D1%82%D0%B8%D1%87%D0%BD%D0%B0_%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D0%B0 (дата звернення: 10.11.24).
21. Онтології і подання знань [Електронний ресурс]. URL: <http://www.znannya.org/?view=ontology-give-knowledge> (дата звернення: 10.11.24).
22. ER-моделювання. Призначення і особливості в рівнянні з UMLдіаграмами. Нотації ER-діаграм [Електронний ресурс]. URL: <http://um.co.ua/8/8-2/8-209440.html> (дата звернення: 10.11.24).
23. Unified Modeling Language [Електронний ресурс]. URL: https://uk.wikipedia.org/wiki/Unified_Modeling_Language (дата звернення: 10.11.24).

24. Діаграма прецедентів [Електронний ресурс]. URL: <https://uk.wikipedia.org/wiki> (дата звернення: 10.11.24).
25. Властивості бінарних відносин [Електронний ресурс]. URL: <https://studfile.net/preview/8709160/> (дата звернення: 10.11.24).
26. OWLAPI [Електронний ресурс]. URL: <https://github.com/owlcs/owlapi> (дата звернення: 10.11.24).
27. Ontology Management [Електронний ресурс]. URL: <https://www.poolparty.biz/ontology-management> (дата звернення: 10.11.24).
28. Protégé [Електронний ресурс]. URL: <https://ru.wikipedia.org/wiki/Prot%C3%A9g%C3%A9> (дата звернення: 10.11.24). 94
29. Protege [Електронний ресурс]. URL: <https://protege.stanford.edu> (дата звернення: 10.11.24).
30. SPARQL [Електронний ресурс]. URL: <https://uk.wikipedia.org/wiki/SPARQL> (дата звернення: 10.11.24).
31. SPARQL Query Language for RDF [Електронний ресурс]. URL: <https://webcitation.org/6879LE0Gr?url=http://www.w3.org/TR/rdf-sparql-query/> (дата звернення: 10.11.24).
33. Менеджмент складних онтологій з використанням онтологічних моделей [Електронний ресурс]. URL: <http://dspace.nbuv.gov.ua/bitstream/handle/123456789/85226/20-Burov.pdf?sequence=1> (дата звернення: 10.11.24).
33. Методичні вказівки до виконання економічної частини магістерських кваліфікаційних робіт / Уклад.: В. О. Козловський, О. Й. Лесько, В. В. Кавецький. Вінниця : ВНТУ, 2021. 42 с.
34. Оцінка ефективності проектів [Електронний ресурс]. URL: <https://buklib.net/books/22886/> (дата звернення: 29.11.24).
35. Методичні вказівки до виконання магістерських кваліфікаційних робіт для студентів спеціальності 122 «Комп'ютерні науки» [Електронний ресурс] / уклад.: А. А. Яровий, О. К. Колесницький. – Вінниця : ВНТУ, 2023. – (PDF, 58 с.)

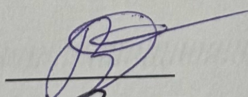
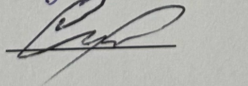
Додатки

Додаток А (обов'язковий)**Протокол перевірки кваліфікаційної роботи на наявність текстових запозичень****ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ**Назва роботи: Інформаційна технологія онтологічного моделювання бази знань гри «Dota 2»Тип роботи: магістерська кваліфікаційна робота
(БДР, МКР)Підрозділ кафедра комп'ютерних наук, ФІТА
(кафедра, факультет)**Показники звіту подібності Turnitin**Оригінальність 83,00% Схожість 17,00%**Аналіз звіту подібності (відмітити потрібне):**

- ☒ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.
- ☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її виконання автором. Роботу направити на розгляд експертної комісії кафедри.
- ☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Ознайомлені з повним звітом подібності, який був згенерований системою Turnitin щодо роботи.

Автор роботи

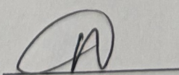
Струнь В.А.

Сілагін О.В.

Керівник роботи

Опис прийнятого рішенняМагістерську кваліфікаційну роботу допущено до захисту

Особа, відповідальна за перевірку



Озеранський В.С.

Додаток Б (обов'язковий)

Лістинг програми

```
<?xml version="1.0"?>
```

```
<!DOCTYPE rdf:RDF [
```

```
  <!ENTITY owl "http://www.w3.org/2002/07/owl#" >
```

```
  <!ENTITY xsd "http://www.w3.org/2001/XMLSchema#" >
```

```
  <!ENTITY rdfs "http://www.w3.org/2000/01/rdf-schema#" >
```

```
  <!ENTITY rdf "http://www.w3.org/1999/02/22-rdf-syntax-ns#" >
```



```
<rdf:RDF xmlns="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-14#"
  xml:base="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-14"
```

```
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
```

```
<owl:Ontology
```

```
  rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-14"/>
```

```
<!--
```

```
////////////////////////////////////
```

```
//
```

```
// Object Properties
```

```
//
```

```
////////////////////////////////////
```

```
-->
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#купують -->
```

```
<rdf:Description
rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#купують"/>
```

```
<!-- http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#містять
-->
```

```
<rdf:Description
rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#містять"/>
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#належать
-->
```

```
<rdf:Description
rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#належать"/>
```

```
<!--
```

```
////////////////////////////////////
```

```
//
```

```
// Data properties
```

```
//
```

```
////////////////////////////////////
```

```
-->
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#вартість_предмету -->
```

```
<owl:DatatypeProperty
rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#вартість_предмету">
```

```
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
```

```
</owl:DatatypeProperty>
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#здоров'я -->
```

```
<rdf:Description
rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#здоров'я"/>
```

```
<!-- http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#мана -
->
```

```
<rdf:Description
rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#мана"/>
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#назва_героя -->
```

```
<rdf:Description
rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#назва_героя"/>
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#назва_предмету -->
```

```
<owl:DatatypeProperty
rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#назва_предмету">
```

```
<rdf:type rdf:resource="&owl;FunctionalProperty"/>
```

```
</owl:DatatypeProperty>
```

```
<!--
```

```
////////////////////////////////////
```

```
//
```

```
// Classes
```

```
//
```

```
////////////////////////////////////
```

```
-->
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Інтелект -->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Інтелект">
```

```
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Тип_героїв"/>
```

```
</owl:Class>
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Активні -->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Активні">
```

```
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Здібності"/>
```

```
</owl:Class>
```

```
<!-- http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Броня ->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Броня">
```

```
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Характеристики_предметів"/>
```

```
</owl:Class>
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Вартість -->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Вартість">
```

```
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Предмети"/>
```

```
</owl:Class>
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Вартість_мани -->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Вартість_мани">
```

```
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Предмети"/>
```

```
</owl:Class>
```

```
<!-- http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Грої ->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Грої"/>
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Додаткова_мана -->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Додаткова_мана">
```

```
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Характеристики_предметів"/>
```

```
</owl:Class>
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Додаткова_швидкість_атаки -->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Додаткова_швидкість_атаки">
```

```
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Характеристики_предметів"/>
```

```
</owl:Class>
```



```

<!--      http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Додаткова_швидкість_руху -->

<owl:Class   rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-
ontology-8#Додаткова_швидкість_руху">

    <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Характеристики_предметів"/>

</owl:Class>

<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Додаткове_здоров&apos;я -->

<owl:Class   rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-
ontology-8#Додаткове_здоров&apos;я">

    <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Характеристики_предметів"/>

</owl:Class>

<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Додаткові_характеристики -->

<owl:Class   rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-
ontology-8#Додаткові_характеристики">

    <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Характеристики_предметів"/>

</owl:Class>

<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Здоров&apos;я -->

<owl:Class   rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-
ontology-8#Здоров&apos;я">

    <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Характеристики_героїв"/>

</owl:Class>

```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Здібності -->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Здібності">
```

```
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Герої"/>
```

```
</owl:Class>
```

```
<!-- http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Кепі -->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Кепі">
```

```
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Тип_героїв"/>
```

```
</owl:Class>
```

```
<!-- http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Мана ->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Мана">
```

```
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Характеристики_героїв"/>
```

```
</owl:Class>
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Назва_активної_здібності -->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Назва_активної_здібності">
```

```
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Активні"/>
```

```
</owl:Class>
```

```
<!-- http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Назва_герою -->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Назва_герою">
```

```
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Герої"/>
```

```
</owl:Class>
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Назва_пасивної_здібності -->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Назва_пасивної_здібності">
```

```
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Пасивні"/>
```

```
</owl:Class>
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Назва_предмету -->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Назва_предмету">
```

```
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Предмети"/>
```

```
</owl:Class>
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Нейтральний -->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Нейтральний">
```

```
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Тип_предмету"/>
```

```
</owl:Class>
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#OX_Інтелект -->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#OX_Інтелект">
```

```
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Основна_характеристика"/>
```

```
</owl:Class>
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#OX_Сила -->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#OX_Сила">
```

```
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Основна_характеристика"/>
```

```
</owl:Class>
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#OX_Спритність -->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#OX_Спритність">
```

```
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Основна_характеристика"/>
```

```
</owl:Class>
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Основна_характеристика -->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Основна_характеристика">
```

```
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Герої"/>
```

```
</owl:Class>
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Основний -->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Основний">
```

```
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Тип_предмету"/>
```

```
</owl:Class>
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Пасивні -->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Пасивні">
```

```
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Здібності"/>
```

```
</owl:Class>
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Поліпшений -->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Поліпшений">
```

```
<rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Тип_предмету"/>
```

```
</owl:Class>
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Предмети -->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Предмети">
```

```
<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Приріст_сили -->
```

```
<owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Приріст_сили">
```

```

    <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Характеристики_героїв"/>

    </owl:Class>

    <!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Приріст_спритності -->

    <owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-
ontology-8#Приріст_спритності">

        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Характеристики_героїв"/>

        </owl:Class>

        <!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Приріст_інтелекту -->

        <owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-
ontology-8#Приріст_інтелекту">

            <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Характеристики_героїв"/>

            </owl:Class>

            <!-- http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Сапорт
-->

            <owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-
ontology-8#Сапорт">

                <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Тип_героїв"/>

                </owl:Class>

                <!-- http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Сила -
->

                <owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-
ontology-8#Сила">

```

```

    <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Тип_героїв"/>

    </owl:Class>

    <!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Спритність -->

    <owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-
ontology-8#Спритність">

        <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Тип_героїв"/>

        </owl:Class>

        <!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Тип_героїв -->

        <owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-
ontology-8#Тип_героїв"/>

        <!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Тип_предмету -->

        <owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-
ontology-8#Тип_предмету">

            <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Предмети"/>

            </owl:Class>

            <!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Тривалість_активної_здібності -->

            <owl:Class rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-
ontology-8#Тривалість_активної_здібності">

                <rdfs:subClassOf
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Активні"/>

                </owl:Class>

```

```

<!--
////////////////////////////////////
//
// Individuals
//
////////////////////////////////////
-->

<!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Abbadon -->

  <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Abbadon">

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Герой"/>

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Назва_герою"/>

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Основна_характеристика"/>

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Універсальний_герой"/>

  </owl:NamedIndividual>

  <!--      http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Alchemist -->

    <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Alchemist">

      <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Герой"/>

```



```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Назва_герою"/>

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#ОХ_Сила"/>

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Основна_характеристика"/>

    </owl:NamedIndividual>                                <!--
http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Ancient_Apparition -->

    <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Ancient_Apparition">

        <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Герої"/>

        <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Назва_герою"/>

        <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#ОХ_Інтелект"/>

        <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Основна_характеристика"/>

    </owl:NamedIndividual>

    <!--    http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Anti-
Mage -->

    <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Anti-Mage">

        <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Герої"/>

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Назва_герою"/>

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#ОХ_Спритність"/>

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Основна_характеристика"/>

```

```

    </owl:NamedIndividual>

```

```

    <!--http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Arc_Warden -->

```

```

    <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Arc_Warden">

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Герої"/>

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Назва_герою"/>

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#ОХ_Спритність"/>

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Основна_характеристика"/>

```

```

    </owl:NamedIndividual>

```

```

    <!-- http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-8#Axe -->

```

```

    <owl:NamedIndividual
rdf:about="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Axe">

```

```

    <rdf:type
rdf:resource="http://www.semanticweb.org/hallo/ontologies/2024/10/untitled-ontology-
8#Герої"/>

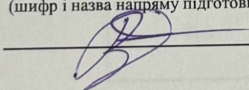
```

Додаток В (обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ ОНТОЛОГІЧНОГО
МОДЕЛЮВАННЯ БАЗИ ЗНАНЬ ГРИ «DOTA 2»

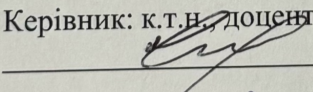
Виконав: студент 2-го курсу,
групи 1КН-23м
спеціальності 122 «Комп'ютерні науки»
(шифр і назва напрямку підготовки, спеціальності)



Струнь В. А.

(прізвище та ініціали)

Керівник: к.т.н., доцент каф. КН



Сілагін О. В.

(прізвище та ініціали)

« 05 » 12 2024 р.

Вінниця ВНТУ - 2024 рік



Рисунок В.1 – Покращена інформаційна технологія створення онтологічної моделі

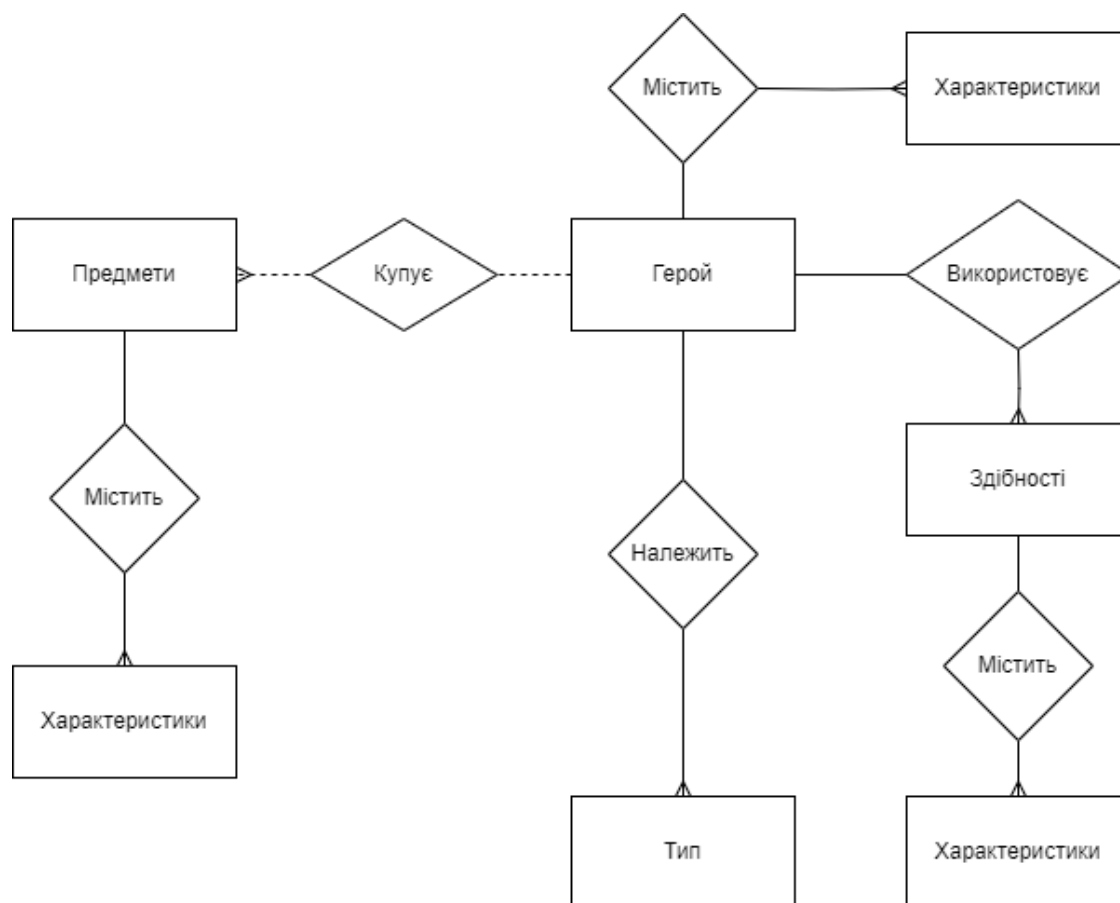


Рисунок В.2 – ER-модель предметної галузі гри «Dota 2»

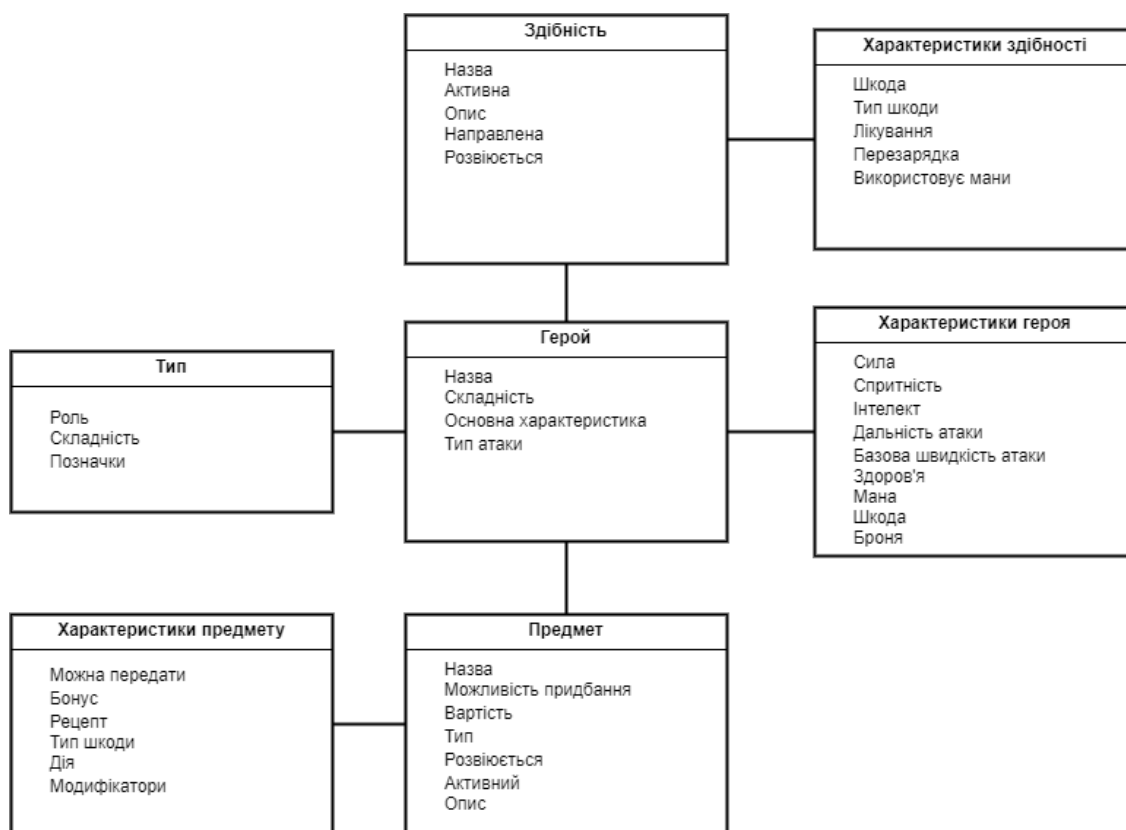


Рисунок В.3 – UML-діаграма з предметної галузі гри «Dota 2»

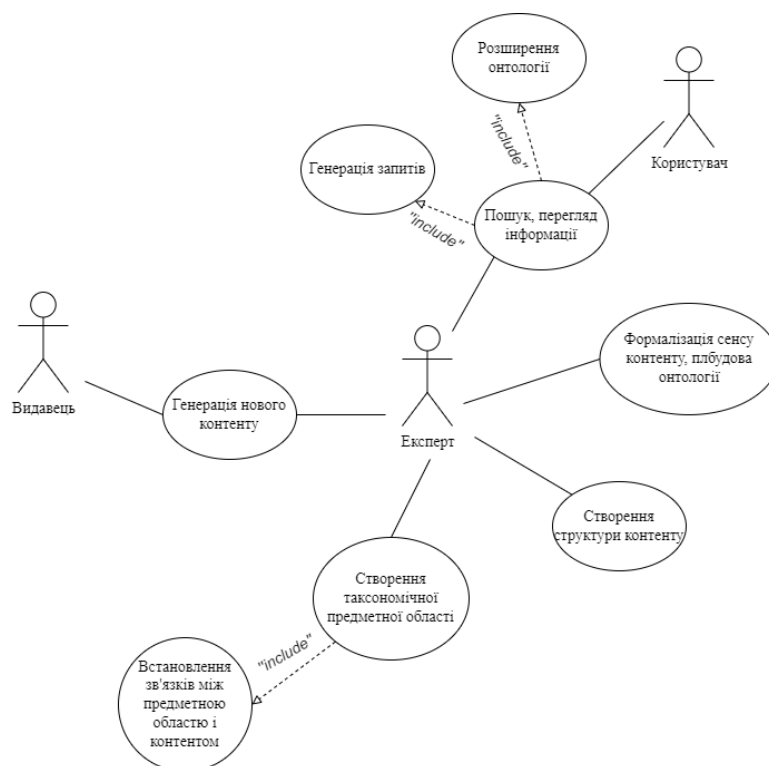


Рисунок В.4 – Діаграма прецедентів роботи онтологічної системи гри «Dota 2»

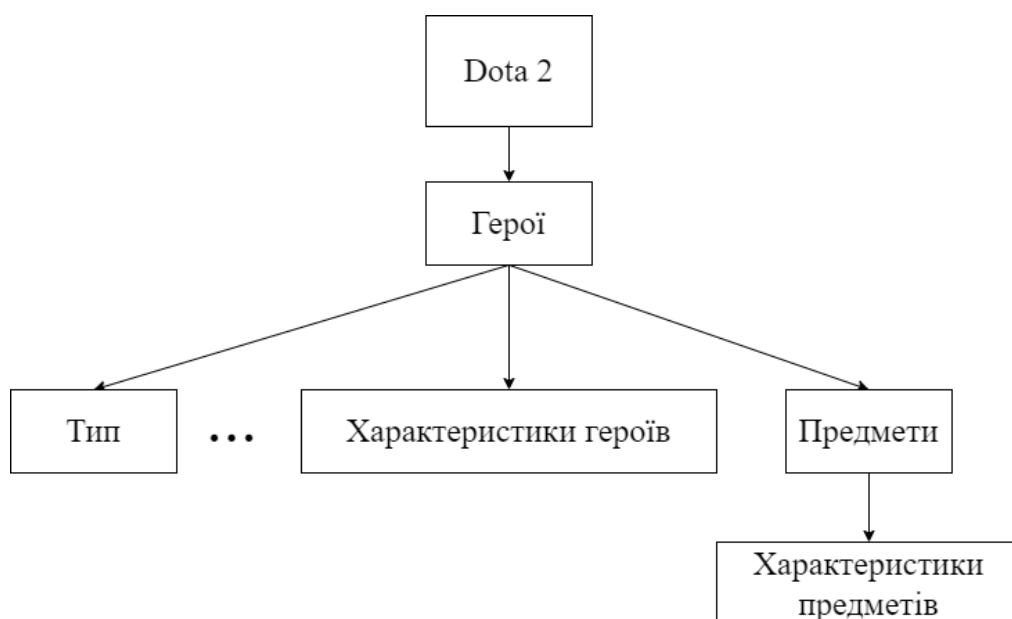


Рисунок В.5 – Перший варіант вигляду онтології «Dota 2»

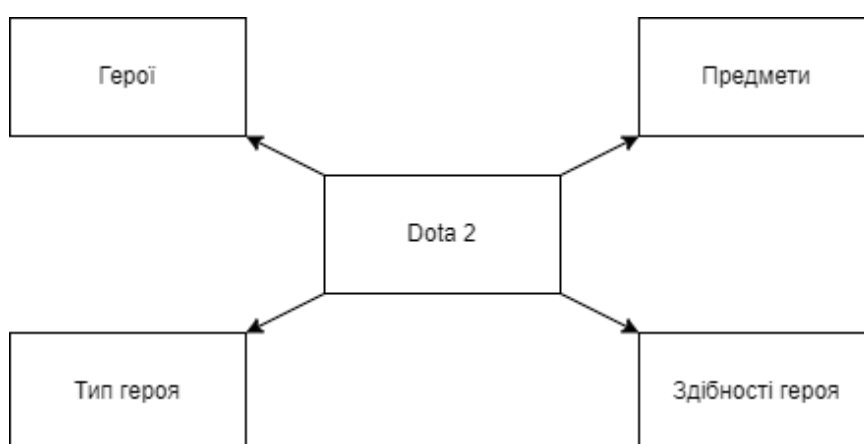


Рисунок В.6 – Другий варіант вигляду онтології «Dota 2»



Рисунок В.7 – Схема функціонування онтологічної бази знань гри «Dota 2»

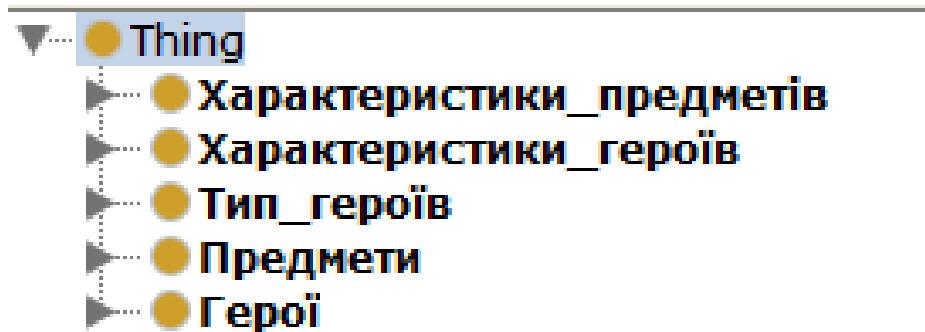


Рисунок В.8 – Приклад створення класів в Protégé

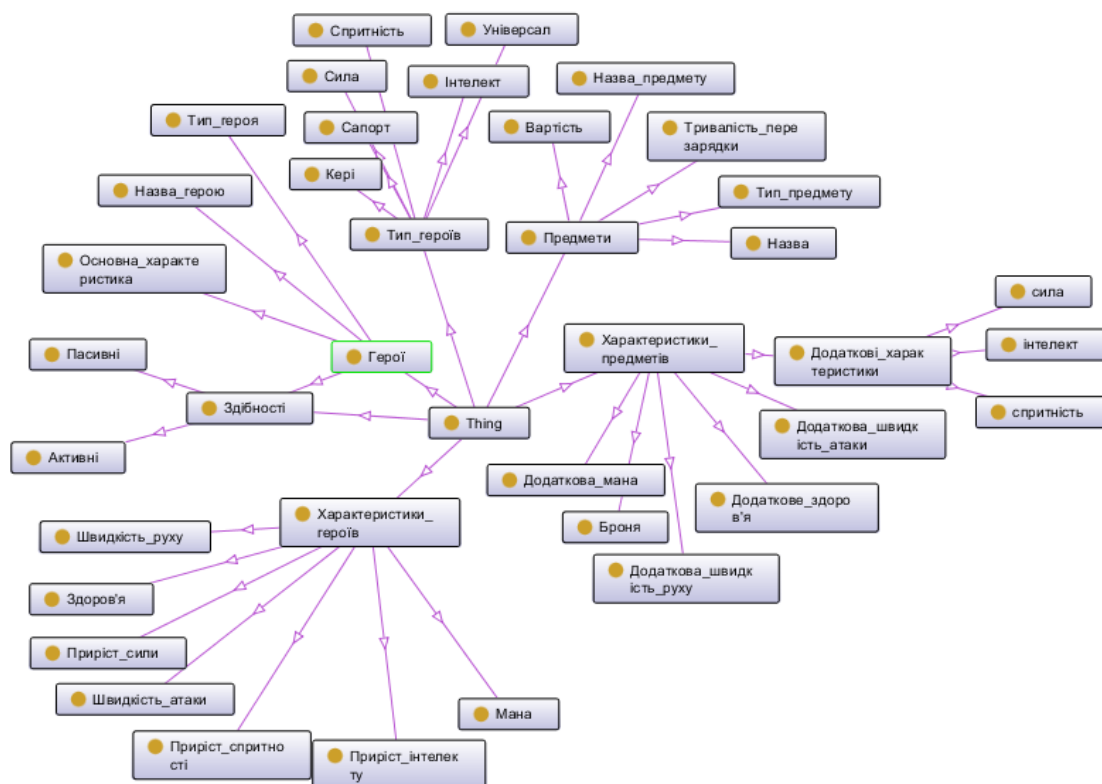


Рисунок В.9 – Базовий граф онтологічної бази знань з гри «Dota 2»

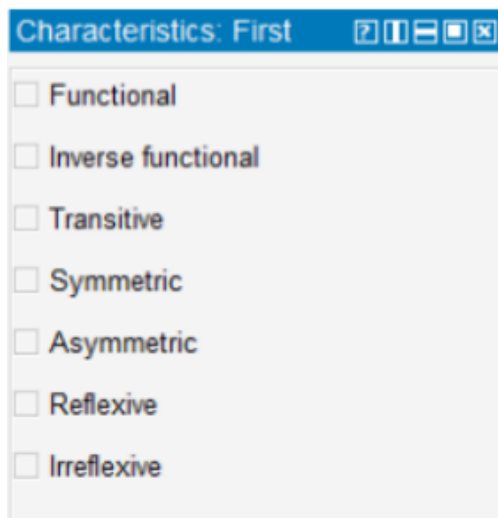


Рисунок В.10 – Зображення видів властивостей об'єкту

Додаток Г (довідниковий)

Інструкція користувача

Запускаємо середовище Protégé та завантажуюмо файл з онтологією. Обираємо вкладку Ontograf для пошуку інформації за гілками графа. Також можна розглядати ієрархію онтології, вивчати деталізацію предметної галузі (рис. Г.1).

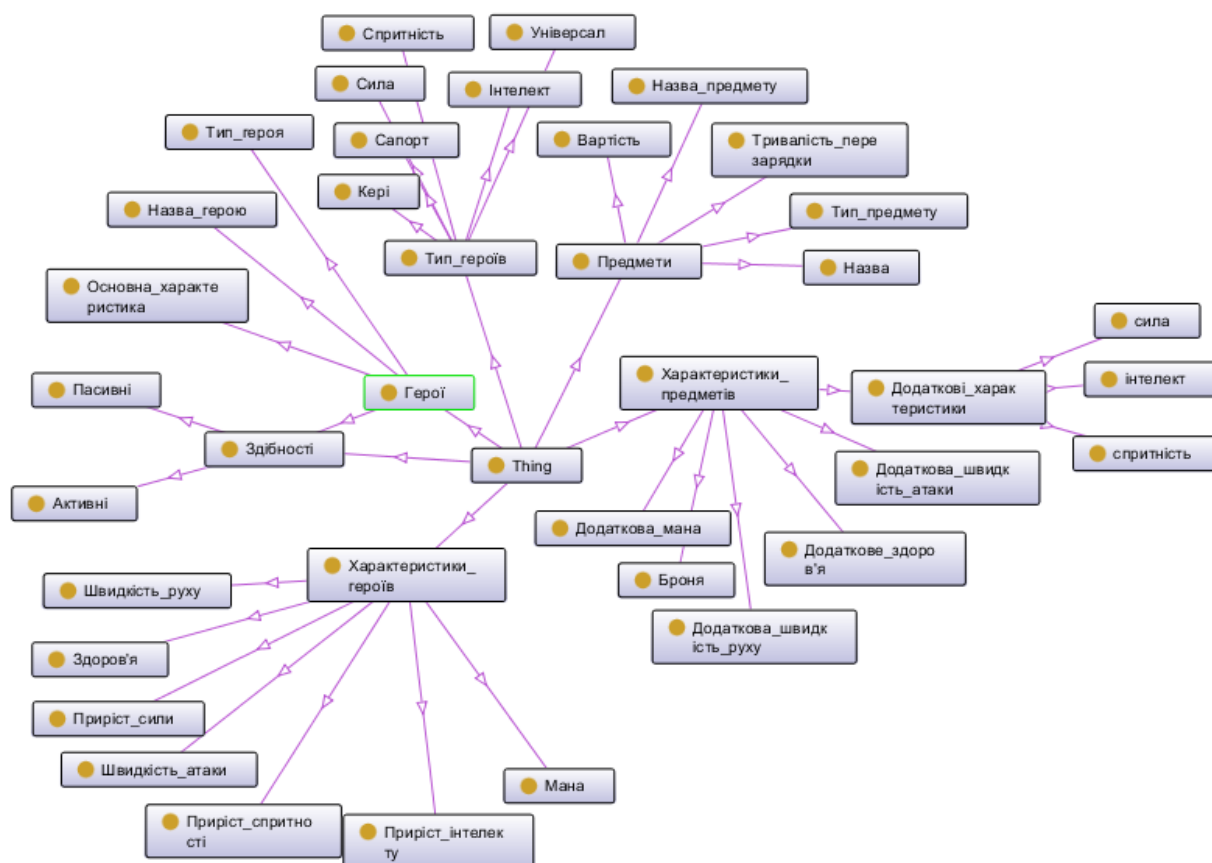


Рисунок Г.1 – Пошук за графом онтології

Для того щоб виконати запит SPARQL потрібно перейти до вкладки «SPARQL Query». Якщо вкладка не відображається, в меню Window переходимо по вкладці Tabs і виберемо «SPARQL Query». Для створення запитів на мові SPARQL спочатку треба в текстовому полі вкладки «SPARQL Query» задати простори імен, які будуть використовуватися. Для виконання запиту потрібно натиснути «Execute» і під кодом відобразиться результат у вигляді таблиці. На рисунку Г.2 подано результат цього запиту.

subject	object
спритність	Додаткові_характеристики
Активні	Здібності
Приріст_спритності	Характеристики_героїв
Швидкість_атаки	Характеристики_героїв
Мана	Характеристики_героїв
Універсальний_герой	Основна_характеристика
Поліпшений	Тип_предмету
Додаткове_здоров'я	Характеристики_предметів
Основний	Тип_предмету
Броня	Характеристики_предметів
Здоров'я	Характеристики_героїв
Тривалість_перезарядки_активної_здібності	Активні
Додаткові_характеристики	Характеристики_предметів
Приріст_сили	Характеристики_героїв

Рисунок Г.2 – Результат виконання SPARQL-запити