

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему: **«Розробка навчальної гри «LanguageRPG» за допомогою Unity3D»**

Виконав: студент 4 курсу, групи ПД-41
спеціальності
121 Інженерія програмного забезпечення
(шифр і назва спеціальності/спеціалізації)

_____ Лацинський С.С.
(прізвище та ініціали)

Керівник _____ Трінтіна Н.А.
(прізвище та ініціали)

Рецензент _____
(прізвище та ініціали)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти -«Бакалавр»

Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

Негоденко О.В.

“ _____ ” _____ 2021 року

ЗАВДАННЯ НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТА

Лащинському Сергію Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка навчальної гри «LanguageRPG» за допомогою Unity3D»

Керівник роботи: Трінтіна Н.А., к.т.н., доцент кафедри ІІЗ

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом вищого навчального закладу від «12» березня 2021 року №65.

2. Строк подання студентом роботи «1» червня 2021 року

3. Вхідні дані до роботи

Методи розробки навчальних ігор;

Науково-технічна література з питань, пов'язаних з розробкою навчальних систем.

4. Зміст розрахунково-пояснювальної записки(перелік питань, які потрібно розробити).

4.1 Ігрофікація навчального процесу.

4.2 Аналіз використаних технологій.

4.3 Опис проектування системи.

4.4 Опис використаних технологій.

5. Перелік демонстраційного матеріалу (назва основних слайдів)

1. Актуальність проблеми.
2. Існуючі навчальні ігри та принципи їх роботи.
3. Принципи роботи навчальної гри.
4. Робота навчальної гри на конкретних прикладах.
5. Архітектура проекту.

6. Дата видачі завдання «19»квітня 2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір та вивчення джерел інформації з напряму дослідження	01.10.2020	Виконано
2	Вимоги до системи	05.11.2020	Виконано
3	Проектування системи	11.12.2020	Виконано
4	Розробка системи	30.02.2021	Виконано
5	Написання основної частини	11.04.2021	Виконано
6	Вступ, висновки, реферат	17.04.2021	Виконано
7	Розробка обов'язкових демонстраційних матеріалів	20.04.2021	Виконано
8	Попередній захист роботи		
9	Здача роботи		

Студент _____ Лацинський С.С.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Трінтіна Н.А.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина бакалаврської роботи 50 с., 49 рис., 17 джерел.

НАВЧАЛЬНА ГРА, ПРОГРАМУВАННЯ, UNITY, НАВЧАЛЬНИЙ ПРОЦЕС,
ДЕСКТОПНА ГРА.

Об'єкт дослідження – ігрофікація навчальної системи на основі навчальної гри.

Предмет дослідження – ігровий додаток для ігрофікації навчальної системи на основі декількох міні-ігор з відкритим світом.

Мета роботи – розробити сучасну навчальну гру, здібну покращувати навички користувача у декількох напрямках та для полегшення навчального процесу у сфері програмування.

Методи дослідження – емпіричні методи для дослідження та проведення експериментів і теоретичні методи для аналізу даних.

Загальною проблемою більшості сучасних навчальних систем є необхідність у постійному з'єднанні з інтернетом, що не дає змогу навчатися при його відсутності. Створений ігровий додаток не потребує з'єднання з мережею, тому може бути використаним будь-де за наявності комп'ютера на базі операційної системи Windows.

Додаток здатен навчати 4 різними мовами програмування, таким як Java, JavaScript, C#, PHP та мові розмітки HTML за допомогою влаштованих міні-ігор. В якості основи ігрового світу було розроблено відкриту карту, де гравець зможе переміщуватися без перешкод на шляху до своїх пригод.

Отже, було розроблено та описано навчальну гру, завданням якої є поліпшення знань учнів у сфері програмування та покращення їх навичок уваги та зосередженості.

Даний додаток може бути використано у навчальній сфері, яка потребує впровадження процесу ігрофікації.

Відомості про апробацію – було опубліковано 2 тезисів на наукових конференціях на тему дипломної роботи:

1. Назва конференції – Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в інфокомунікаційних технологія». Науковий напрямок – «Розробка Ігор». Тема доповіді – «Роль ігор у навчальному процесі».
2. Назва конференції – Всеукраїнська науково-технічна конференція «Сучасний стан та перспективи розвитку ІОТ». Науковий напрямок – «Сучасні інформаційні технології в Україні і світі». Тема доповіді – «Актуальність створення навчальної гри для вивчення програмування».

Галузь використання – навчальна система.

ЗМІСТ

ВСТУП.....	8
1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ	10
1.1. Розвиток інформаційних технологій у галузі ігрофікації навчальних процесів у комп'ютерних іграх.....	Ошибка! Закладка не определена.10
1.2. Аналіз існуючих навчальних інформаційних технологій у комп'ютерних ігрових системах.....	11
1.3. Класифікація навчальних ігор	12
Висновки до розділу	15
2. АНАЛІЗ ТА ПРОЕКТУВАННЯ СИСТЕМИ.....	16
2.1. Середовище розробки Unity, як засіб для розробки навчальних ігор	16
2.2. Проектування навчальної гри «LanguageRPG»	21
Висновки до розділу	31
3. РЕАЛІЗАЦІЯ НАВЧАЛЬНОЇ ГРИ	0
шибка! Закладка не определена.32	
3.1. Файлова структура проекту	32
3.2. Реалізація навчальної гри «LanguageRPG».....	Ошибка! Закладка не определена.33
3.3. Тестування навчальної гри «LanguageRPG».....	Ошибка! Закладка не определена.50
Висновки до розділу	55
ВИСНОВКИ	0
шибка! Закладка не определена.56	
Перелік посилань	58
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація).....	60

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

Рушій – ядро комп'ютерної програми для реалізації конкретної прикладної задачі, щоб відрізнити її від наповнення і зовнішнього вигляду конкретної програми .

Міні-гра – маленька окрема гра у грі.

Білд – готова до запуску версія програмного продукту.

Test Case – сукупність кроків, умов та параметрів, необхідних для перевірки програмного забезпечення.

Баг – дефект у програмному забезпеченні.

Квест – певне завдання у грі.

ВСТУП

Актуальність роботи. У сучасному світі нова система навчання у вигляді комп'ютерної ігрової системи відіграє надзвичайно важливу роль у сучасному житті суспільства. Сьогодні ігрофікація повсюдно використовується у сфері кінематографа, проектування, маркетингу й архітектурного дизайну. Насправді ігри можуть також допомагати і у навчанні, а не тільки носити розважальний характер. Приміром, гра Minecraft Education Edition. Ця гра була розроблена для спеціального використання в цілях навчання різним предметам у школі. Використовуючи гру, школярі вивчають такі предмети, як історію, хімію, географію, архітектуру, фізику, літературу та багато чого іншого. Під час уроків учні спостерігають за різноманітними природними процесами та беруть участь в історичних подіях, проводять експерименти та досліді. У грі також є аналог візуального програмування Scratch, що значно розширює можливості гравця та розвиває його креативність.

В наслідок появи навчальних ігор та їх масовізації, цифрові ігрові системи перейшли на інший рівень та стали необхідними у навчанні та розвагах як ніколи. Комп'ютерні ігри вже давно перестали бути просто звичайною дитячою забавкою. Безумовно, не можна заперечувати, що приблизно 95% дітей кожен день грають в ігри, проте даний факт не скасовує того, що середній вік гравців в цифрові відеоігри приблизно 30 років. Не існує грані віку для цієї розваги, також ігри можуть бути дуже корисними, наприклад, можуть відстрочити старіння головного мозку. У наш час, це явище настільки розповсюджено, як колись був перегляд цифрового телебачення, від якого більша частина людей на сьогодні відмовилась. Так і навчальні ігри поступово замінюють звичайні системи навчання.

Комп'ютерні ігри допомагають користувачеві здобути нові навички та розвинути ті, що він вже мав. Відеоігри можуть розвинути різні навички:

1. Соціальні навички. Нинішні ігри націлені на інтенсивну комунікативність. Багато гравців взаємодіють у віртуальних світах зі своїми друзями чи іншими людьми, граючи в командах чи змагаючись

один проти одного.

2. Когнітивні здібності. У прихильників екшн типу ігор поліпшується реакція, просторова уява, орієнтування в просторі.

Мета дослідження. Розробити навчальну гру для програмістів за допомогою середовища розробки Unity3D та дослідити найкращі засоби реалізації навчальної гри.

Практична цінність роботи. З моменту появи ігор, багато людей завжди вважали, що вони існують лише для розваг, але це не так: ігри можна використовувати і для навчального процесу. Іноді дітям дуже складно вивчати важкий матеріал по звичайним книгам, і тут до допомоги приходять ігри, які можуть зацікавити учнів, вони допомагають здобути не лише нові знання, а ще й можуть розвеселити. Основною різницею між комп'ютерними відеоіграми та стандартним навчанням – це відношення до помилок. У школі за помилки завжди карають у вигляді поганих оцінок, що лише знижує бажання учнів вчитися, але при цьому лише іноді вчителя хвалять за правильні відповіді або рішення. Тому школярі знають тільки те, що вони робили некоректно. Це призводить до того, що учні будуть сконцентровані лише на оцінках, а ні на самих знаннях.

Структура роботи. Дипломний проект (робота) складається зі вступу, аналізу предметної області і постановки задачі, аналізу системи, проектування системи, реалізації системи поділених на підрозділи, додатків та списку використаних джерел.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

1.1 Розвиток інформаційних технологій у галузі ігрофікації навчальних процесів у комп'ютерних іграх

Жодна сфера життєдіяльності людини не росте і не розвивається так широко, як комп'ютерна. І перед освітньою системою стоїть завдання використання сучасних освітніх технологій для створення необхідних компетенцій. Одним з найбільш суперечливих способів вирішення цієї ситуації є комп'ютерні ігри та ігрофікація освіти, яка широко використовується в шкільній освіті та інших сферах життя. Багато дослідників відзначають, що застосування гейміфікації в освіті позитивно впливає на мотивацію учнів, що в кінцевому підсумку підвищує їх результати навчання та їх бажання це робити.

На соціально-педагогічному рівні актуальність дослідження обумовлена тим, що застосування гейміфікації сприяє залученню студентів до процесу навчання.

На науково-теоретичному рівні актуальність полягає в недостатній теоретичній розробленості науково-методологічних підходів до використання елементів ігрофікації.

На науково-методичному рівні актуальність полягає в тому, що дослідники визнають практичну користь застосування гейміфікації в освіті, проте практично немає засобів та технік, які дозволяли б впровадити гейміфікацію у процес навчання.

Теорія гейміфікації в освіті полягає в тому, що учні будуть краще навчатися, коли їм також весело. Мало того, вони краще навчаються, коли у них є цілі та досягнення, до яких слід дійти, звичайно, таким чином, учні все сприймають із задоволенням та зацікавленістю.

Через захоплюючі особливості відеоігор, які можуть заінтригувати не тільки

дітей, а й дорослих, та можуть легко заволодіти їх увагою, цілком, що ми бачимо подібні результати залучення, коли ці засновані на іграх елементи застосовуються до навчальних матеріалів.

Гейміфікація у навчанні передбачає використання ігрових елементів, таких як набрання балів, змагання з боку однолітків, роботи в команді, таблиці результатів для стимулювання залучення, допомагає учням засвоїти нову інформацію та перевірити свої знання. Вона може застосовуватися до шкільних предметів, але також широко використовується в програмах та курсах для самонавчання, показуючи, що ефективність ігрофікації не припиняється, навіть коли люди стають дорослими.

1.2 Аналіз існуючих навчальних інформаційних технологій у комп'ютерних ігрових системах

Вчителі та батьки можуть інтегрувати гейміфікацію різними способами в незліченних предметних областях. Багато шкіл вже використовують додатки та навчальні ігри на комп'ютерах та планшетах. Зараз стає дуже популярним ігрове навчання, яке передбачає розробку студентами власних ігор або відтворення комерційних відеоігор.

Деякі приклади ігрових елементів, які можна використовувати для залучення та мотивації учнів, включають:

- Розповідь
- Миттєвий зворотний зв'язок
- «Навчання на естрадах» де складність поступово зростає
- Рівень майстерності (наприклад, зростає рівень)
- Показники прогресу (наприклад, через бали, значки, таблиці лідерів)

По видам діяльності, ігри можна розділити на фізичні, рухові, інтелектуальні, або розумові, а також соціальні, трудові та психологічні. У залежності від характеру навчального процесу виділяють такі групи ігор:

- тренувальні, навчальні, узагальнюючі та контролюючі;
- виховні, пізнавальні та розвиваючі;
- творчі, репродуктивні та продуктивні;
- комунікативні, психотехнічні, професійно орієнтовані та діагностичні.

Навчальна гра, в свою чергу, є формою навчальної та виховної діяльності, яка в процесі гри імітує різноманітні практичні ситуації, також рахується засобом початку активізації навчального процесу і може сприяти розумовому розвитку. За своєю суттю, навчальні комп'ютерні ігри схожі на дидактичні ігри, але організовані на більш високому рівні. У таких іграх, з одного боку, грає здійснює певні дії, які пов'язані з розв'язанням певних завдань. З іншої сторони, діяльність у віртуальному світі носить лише умовний характер, який може дозволити відволіктися від реальної ситуації, зняти з себе відповідальність та зменшити рівень стресу за скоєні помилки, які в реальному світі могли б привести до певних наслідків. Крім того, візуалізація і одночасний вплив на різні органи чуття активізують пізнавальну діяльність і сприяють легшому засвоєнню матеріалу.

1.3 Класифікація навчальних ігор

Що стосується навчальних комп'ютерних ігор, то тут теж можна виділити ключові характеристики.

За ознакою категорії навчання можна виділити ігри для освоєння знань та ігри для розвитку умінь. В іграх для освоєння знань в ігровому контексті пропонуються різні відомості, що дозволяє гравцеві підвищити свій рівень знань. Ігри для розвитку умінь можна розділити на ігри для розвитку формалізованих навичок і ігри для розвитку слабо формалізованих навичок. Ігри для розвитку

формалізованих навичок орієнтовані на освоєння технологій, які можна відтворити в ігровому світі. Ігри для розвитку слабо формалізованих навичок призначені для розвитку комунікаційних навичок.

За внутрішнім поданням (моделі) предметної області в грі виділені ігри на основі ситуаційного, імітаційного моделювання, і на основі логіко-формальної моделі. Ситуаційні моделі реалізуються через конкретні приклади (діалоги, візуалізація персонажів, середовища і т.д.). Найчастіше такий підхід реалізується в рольових іграх і пригодах. Процес проектування таких ігор включає розробку ігрового сюжету, віртуального середовища і способу її відображення. Імітаційні моделі знайшли відображення в іграх-симуляторах, оскільки моделі – реальні ситуації або технології управління тим чи іншим засобом. Проектування таких ігор включає розробку моделі процесу роботи у відповідній предметній області. Однак таку модель неможливо побудувати для всіх дисциплін.

Основним завданням проектувальників таких навчальних ігор є вибір і реалізація способів найбільш привабливого способу відтворення сценарію чи моделі. Ігри на основі формально-логічної моделі полягають в перевірці введених даних передбаченими шаблонами. Однак ситуаційні та імітаційні моделі надають розробникові більше можливостей для реалізації.

За ознакою домінування навчального або ігрового компоненту виділені два класи ігор: навчальна система з елементами гри, гра з елементами навчання. В існуючих навчальних іграх всіх категорій навчальний компонент істотно домінує над ігровим. В результаті аналізу в області розробки навчальних ігор і існуючих реалізацій подано класифікацію навчальних ігор за трьома ознаками: категорія предмета навчання, модель предметної області, організація взаємодії навчальної та ігрової компоненти. (Рисунок 1.1).

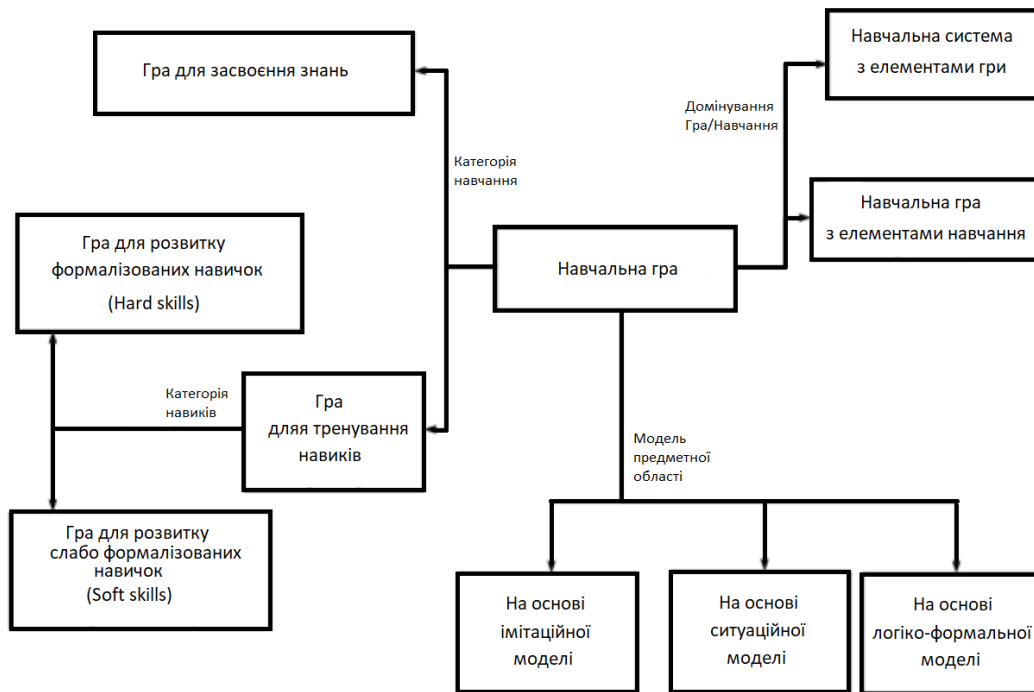


Рисунок 1.1 – Класифікація навчальних ігор

Ігри для навчання програмуванню майже завжди являють собою стратегії, в яких гравець має розробляти алгоритми для управління ігровими об'єктами. Такі ігри можна розділити на дві категорії – ігри для навчання основам програмування, наприклад, Colobot(рисунок 1.2) та його більш новіша версія SeeBot та ігри вже для більш досвідчених програмістів – RoboCode(рисунок 1.3).



Рисунок 1.2 – Навчальна гра Colobot



Рисунок 1.3 – Навчальна гра Robocode

Можна виділити чотири способи організації сценаріїв в навчальних іграх:

- навчальний сценарій;
- навчальний сценарій з елементами гри;
- незалежні ігровий і навчальний сценарії;
- ігровий сценарій з елементами навчання.

Висновки до розділу

Отже, провівши аналіз серед уже існуючих ігор, можна зробити висновок, що найкраще всього розробити навчальну гру, зосередивши увагу на навчальному процесу, а не на ігровому. Для дипломної роботи буде розроблено навчальну 3D гру «LanguageRPG» для покращення знань у сфері програмування, використовуючи навчальний сценарій з елементами гри.

2 АНАЛІЗ ТА ПРОЕКТУВАННЯ СИСТЕМИ

2.1 Середина розробки Unity, як засіб для розробки навчальних ігор

На сьогоднішній день у сфері ігрових движків існує дуже широкий вибір, що дозволяє знайти підходящий засіб розробки, на свій смак і колір, кожному. Відомі такі найпопулярніші, на даний момент, ігрові рушії: Unity, Unreal Engine 4, Phaser, Cry Engine, Corona SDK, GameSalad, LibGDX. Для свого проекту я буду використовувати середина розробки Unity, так як вона найпопулярніша та й дуже гнучка для застосування. Ігровий рушій Unity від Unity Technologies є провідним світовим рішенням для створення ігор. За даними PR-сторінки веб-сайту Unity, це рушій, який обирають понад 45% світових розробників. Це робить його на сьогоднішній день найбільш уживаним у світі. Це ж опитування повідомляє, що їх найближчий конкурент (Unreal Engine 4, або «UE4» від Epic Games) має трохи більше третини кількості користувачів – 17% частки ринку.

Unity – це потужний крос-платформний рушій та дуже зручне середовище розробки. Досить простий для початківця і достатньо функціональний для експерта. Unity може зацікавити кожного, тих хто хоче створювати двовимірні та тривимірні ігри та програми для мобільних пристроїв, настільних комп'ютерів, вебу та консолей. Завдяки архітектурі рушія існує можливість створювати ігри для будь-якої платформи та в разі потреби легко змінити її. Це дозволить створювати як настільні, так і мобільні додатки. Unity підтримує такі платформи як:

- Windows(desktop)
- Mac OS (desktop)
- Linux desktop
- iOS
- Android
- Xbox
- PS4

- Google Cardboard
- Oculus Rift
- Unity Web Player
- PS VR
- PS3
- AndroidTV
- WiiU
- SamsungTV

Unity завдяки своїй гнучкості, різним засобам реалізації та великій кількості документації та відео уроків підійде як професіоналам, так і звичайним користувачам. Також Unity повністю безкоштовна, поки дохід від додатку не перевищує 100 000 доларів США.

Одним із найголовніших плюсів Unity є можливість використовувати вже створені іншими людьми асети, що іноді дуже допомагає не тільки новачкам, а й досвідченим розробникам. Асети – це будь-який ресурс, який може використовувати гра. До них належать тривимірні моделі, матеріали, текстури, аудіо, сценарії та шрифти та інше. Окрім кількох простих об'єктів, таких як сфери та куби, розробник насправді не може створити більшість елементів у Unity. Їх потрібно створювати зовні із застосуванням програм 3D-моделювання та інструментів малювання, а потім імпортувати в Unity. На щастя, імпорт активів Unity є надійним та простим. Традиційно тривимірні ігрові рушії, як правило, дуже вибагливі і дуже детально ставляться до того, які файли їм надаються, змушуючи розробників ретельно конвертувати всі свої файли. Але не Unity. Він підтримує усі популярні формати файлів 3D, включаючи Maya, 3D Studio Max, Blender та FilmBox з усіма матеріалами та текстурами. Unity також підтримує всі поширені формати зображень, включаючи PNG, JPEG, TIFF і навіть файли PSD імпортовані безпосередньо з Photoshop. Що стосується звуку, Unity підтримує WAV та AIF, які ідеально підходять для звукових ефектів, а MP3 і OGG для музики.

Додаток Unity – 3D-середовище, придатне для розстановки рівнів,

створення меню, анімацій, написання сценаріїв та організації проектів. Інтерфейс користувача добре організований, і панелі можна повністю налаштувати, перетягуючи та змінюючи їх розміри. (Рисунок 2.1).

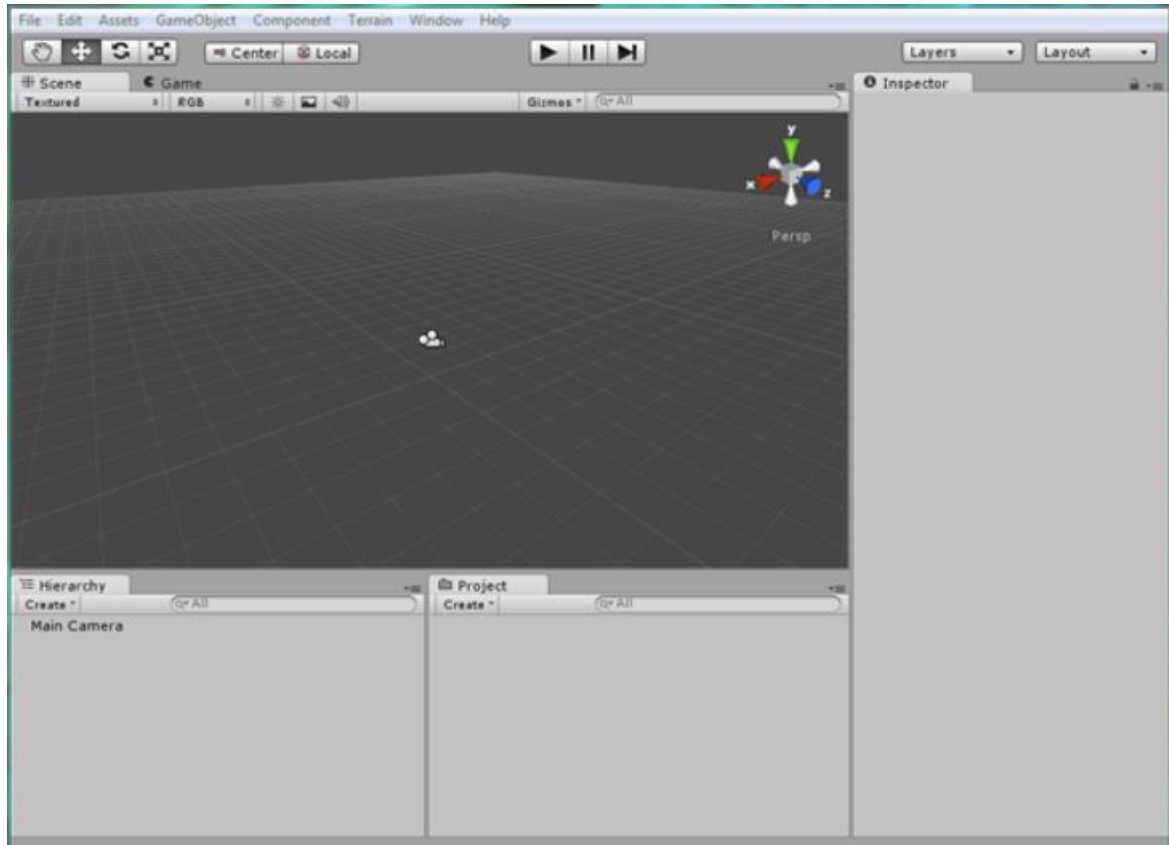


Рисунок 2.1 – Середина розробки Unity3D

На панелі проекту зберігаються всі асети проекту. Коли асети імпортуються, вони спочатку з'являться тут. Панель ієрархії – це місце, де відображаються всі елементи у сцені. Асети з панелі «Проект» можна перетягнути на панель «Ієрархія», щоб додати їх до поточної сцени. Панель інспектора дозволяє перевіряти та налаштовувати всі атрибути вибраного асету. Все, від його положення та обертання, до того, чи впливає на нього сила тяжіння чи має елемент мати тінь. Панель «Сцена» – це область перегляду у форматі 3D, де ви можете фізично впорядковувати об'єкти, переміщуючи їх у просторі. Розробник має можливість переміщатися по області перегляду, обертаючи та масштабуючи сцену.

Сцени – це місце, куди можна перетягнути асети проекту та розташувати їх, щоб зробити рівні. Панель Ієрархія представляє вміст вибраної сцени у форматі дерева. Хоча панель «Сцена» ідеально підходить для впорядкування об’єктів сцени у тривимірному просторі, «Ієрархія» – це місце, де розробник проводить більшу частину свого часу, фактично впорядковуючи свої сцени та підтримуючи їх у порядку. Коли створюється новий проект, Unity автоматично створює нову сцену. Сцени починаються лише з елемента Camera. Запустивши гру зараз, гравець не побачить нічого, крім кольору тла, щоб це змінити потрібно додати інші елементи. (Рисунок 2.2).

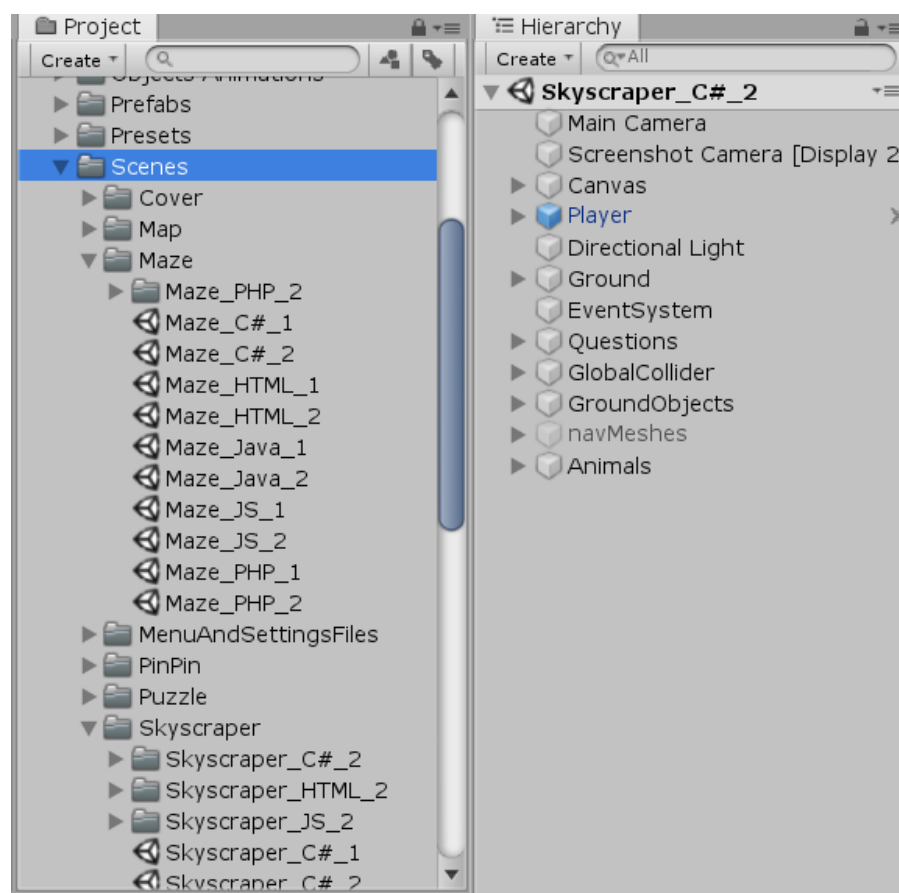


Рисунок 2.2 – Вікно Ієрархія та список асетів

Скрипти у Unity дозволяють брати асети на сцені та надавати їм життя. До одного об’єкта можна приєднати кілька скриптів, що дозволяє легко використовувати різноманітні дії. Раніше Unity підтримував три різні мови програмування; UnityScript, C# та Boo, але зараз доступною являється лише мова

програмування C#.

Всі нові створені скрипти мають такий формат:

```
using UnityEngine;
using System.Collections;
public class Name: MonoBehaviour {
    void Start () {
    }
    void Update () {
    }
}
```

Для того щоб візуалізувати ігрові елементи рушій використовує GameObject. У випадку з 3D грою це можуть бути геометричні фігури, 3D моделі і т.д. У 2D іграх використовуються Sprites, що являє собою звичайну картинку, щоб відобразити ігрові об'єкти. Також рушій має елементи UI, які використовуються для реалізації призначеного для користувача інтерфейсу і являються одним з ключових елементів в ході розробки комп'ютерної гри. Для відображення ігрових елементів на екрані гравця використовується елемент Camera.

Для того, щоб задавати логічні зв'язки і правила взаємодії всередині ігрового світу, Unity використовує скрипти, які описуються за допомогою високорівневої мови програмування C#. За допомогою скриптів можна задавати як порядок дій на сцені, так і правила за якими будуть взаємодіяти один з одним і з гравцем ігрові елементи. Також за допомогою скриптів ігровий движок дозволяє взаємодіяти з реляційними базами даних, такими як Microsoft SQL.

Отже, можна зробити висновок, що Unity являється найкращою середою розробки як для новачків, так і для досвідчених розробників, завдяки своїй документації та дружельобній навігації.

2.2 Проектування навчальної гри «LanguageRPG»

Перед тим як приступити безпосередньо до розробки гри, для більш зручної реалізації потрібно спроектувати різноманітні елементи гри, такі як:

- Жанр гри
- Засоби реалізації
- Елементи меню
- Система рівнів
- Система нагород
- Досягнення
- Налаштування
- Система взаємодії між персонажами

Жанр відеоігор – це класифікація, присвоєна відеогрі, що базується насамперед на її геймплеї, а не на візуальних чи розповідних ознаках. Жанр відеоігор, як правило, визначається набором ігрових елементів, які розглядаються незалежно від обстановки чи змісту ігрового світу.

Відеоігри можна поділити на такі жанри:

- Екшн – це ігри, які зосереджені на фізичних викликах, для подолання яких потрібна одночасна координація реакції та моторики. Усі дії зосереджуються навколо гравця, що контролює більшу частину дії. Більшість найдавніших відеоігор вважалися екшн-іграми, цей жанр можна по справжньому назвати прародичем всіх інших жанрів. Сьогодні це все ще головний жанр, що охоплює всі ігри, що пов'язані з важкими викликами та великою кількістю активних дій.
- Екшн-пригода – незважаючи на те, що екшн-пригодницькі ігри можна окремо розділити на екшн або пригодницькі, вони поєднують елементи цих двох компонентних жанрів, що, як правило, мають складні перешкоди, які необхідно подолати, використовуючи інструменти або інші різноманітні предмети, а також багато

невеликих перешкод майже постійно знаходяться на шляху, що надає іграм екшн елемент. Екшн-пригодницькі ігри, як правило, спрямовані на дослідження різноманітних локацій і зазвичай включають збір предметів, просте вирішення головоломок та боротьбу. «Action-adventure» став ярликом, який іноді прикріплюють до ігор, які не вписуються в інший добре відомий жанр. Через їх поширеність на відео ігрових приставках та відсутність типових пригодницьких ігор, екшн-пригодницькі ігри сучасні геймери часто називають «пригодницькими».

- Пригода – пригодницькі ігри були одними з найперших ігор, на відміну від пригодницьких фільмів, пригодницькі ігри не визначаються історією чи змістом. Швидше, пригода описує манеру гри без рефлексивних викликів чи дій. Зазвичай вони вимагають від гравця розгадування різноманітних головоломок, взаємодії з персонажами чи навколишнім середовищем, найчастіше неконфліктним способом. Він вважається твердо визначеним жанром і, як правило, виключає все, що включає екшн елементи, крім міні-ігор.
- RPG – рольові відеоігри беруть свій геймплей із традиційних настільних рольових ігор, як Dungeons & Dragons. Більшість з цих ігор відводить гравцеві роль персонажа, який під час гри зростає в силі та досвіді. Долаючи складні виклики та перемагаючи монстрів, гравець набирає бали досвіду, які відображають прогрес персонажа у вибраній професії або класі (наприклад, бій у рукопашному бою, володіння різним обладнанням або магією), і дозволяють гравцеві отримати нові здібності, як тільки встановлена сума балів була отримана. Багато рольових ігор містять відкритий світ, який зазвичай населений монстрами та повен різноманітних викликів. Це ігри, в яких гравці виконують ролі персонажів у вигаданій обстановці. Гравці несуть відповідальність за виконання цих ролей в рамках розповіді, або

шляхом буквальної акторської діяльності, або через процес структурованого прийняття рішень. Дії, що проводяться в багатьох іграх, досягають успіху або провалу згідно з офіційною системою правил та вказівок.

- Симулятор – відеоігри-симулятори – це різноманітна супер-категорія ігор, загалом розроблена для того, щоб точно імітувати аспекти реальної чи вигаданої реальності.
- Спорт – це відеоігри, що імітують спорт, наприклад, футбол, баскетбол, волейбол, гонки та інше.
- ММО – багатокористувацькі онлайн-ігри, які здатні одночасно підтримувати велику кількість гравців, де для гри інтернет є обов'язковим аспектом.
- Пісочниця – не є окремим жанром відеоігор, оскільки, як правило, описують особливості ігрового процесу, але часто ігри описуються як пісочниця або гра з відкритим світом, ніби це визначальний жанр. Це категорія ігор, які мають велику карту, по якій гравець може пересуватися без обмежень.

Перед тим як починати розробку гри, розробник визначається з основною концепцією гри і планує та створює різноманітні прототипи для реалізації. Розробка прототипу – ключовий процес для розробки та створення успішного продукту. Детальний план називається прототипом, а сам процес планування – прототипуванням. Процес прототипування не є обов'язковим, але це дуже полегшує життя розробнику, так як простіше спочатку створити прості прототипи та їх реалізовувати, чим вигадувати ідеї на ходу.

Для дипломного проекту буде розроблена гра з елементами навчання для розвитку формалізованих навиків – Hard skills, використовуючи жанр RPG, де більшість дій буде зосереджена на виконання завдань та взаємодію з персонажами для отримання нагород.

Проект буде розроблено за допомогою безкоштовної версії середі розробки Unity3D, для розробки 3D моделей буде використана програма Blender.

Blender – це безкоштовний набір програмних засобів для графіки з відкритим кодом, використовується для розробки анімаційних фільмів, ефектів, моделей із 3D-принтом, графіків руху, інтерактивних 3D-програм, віртуальної реальності та комп’ютерних ігор. Особливості Blender включають 3D-моделювання, розгортання ультрафіолетовим світлом, текстурування, редагування растрової графіки, такелаж та обшивку, імітацію рідини та диму, імітацію частинок, імітацію м’якого тіла, ліплення, анімацію, переміщення матчів, візуалізацію, графіку руху, редагування відео та інше. (Рисунок 2.3).

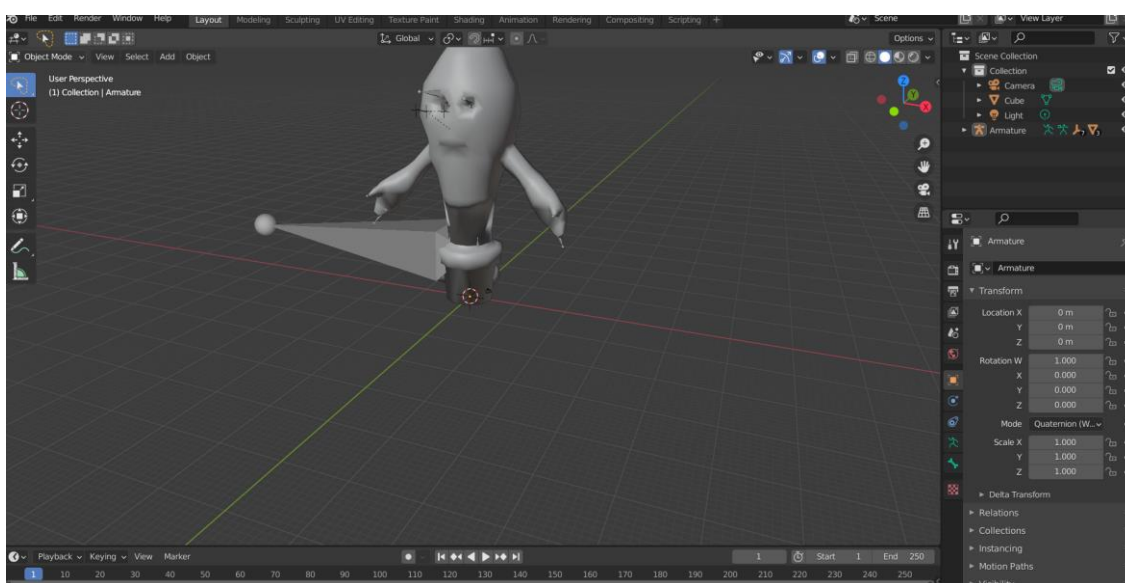


Рисунок 2.3 – Розробка 3D моделі за допомогою Blender

Наступним кроком буде проектування головного меню гри да його підпунктів. Для цієї цілі буде використаний веб ресурс MindMeister.

MindMeister – це онлайн-програма для відображення думок, яка дозволяє своїм користувачам візуалізувати, ділитися та представляти свої думки. MindMeister забезпечує спосіб візуалізації інформації на розумових картах за допомогою моделювання, а також інструментів для полегшення співпраці в реальному часі, координації управління завданнями та створення презентацій. Також його використовують для проектування різних проектів та цілей. (Рисунок 2.4).

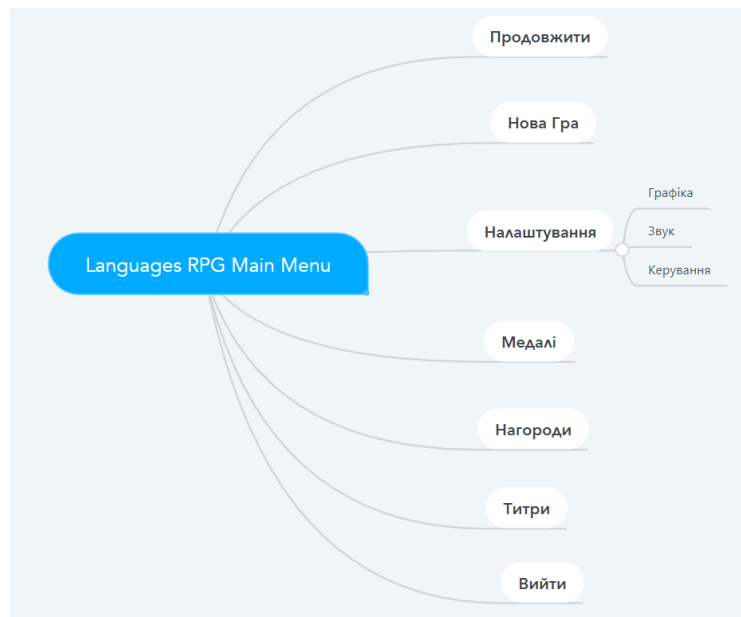


Рисунок 2.4 – Проектування головного меню гри за допомогою Mindmeister

Для прототипування можна використовувати не тільки таблиці, текст, графіки, а й графічні прототипи для полегшення подальшої роботи дизайнерів, тому як простіше буде розробити функціонал по вже вирішеному та обговореному дизайну, чим розробляти його вже під час розробки самої програми, на що буде витрачено занадто багато часу та сил. (Рисунок 2.5).

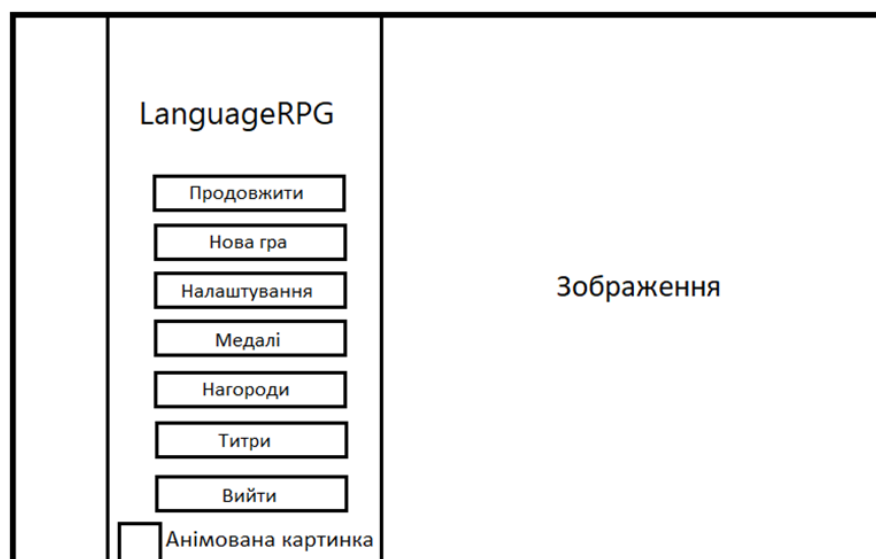


Рисунок 2.5 – Графічний прототип головного меню

Наступним кроком буде прототипування глобальної карти та системи рівнів

гри для неї, для цього буде використано звичайний безкоштовний інструмент для створення зображень, вбудований у Windows, Paint. Звісно, його функціонал не досягає рівня такої професійної програми, як Photoshop від компанії Adobe, але за допомогою нього можна створювати прості 2D малюнки та спроектувати простий список необхідних елементів та їх порядок.

В проєкті «LanguageRPG» буде присутня глобальна карта з 5 різними країнами програмування, в кожній з котрих, гравець буде зустрічати різних істот та отримувати квести та проходити різноманітні міні ігри. (Рисунок 2.6).

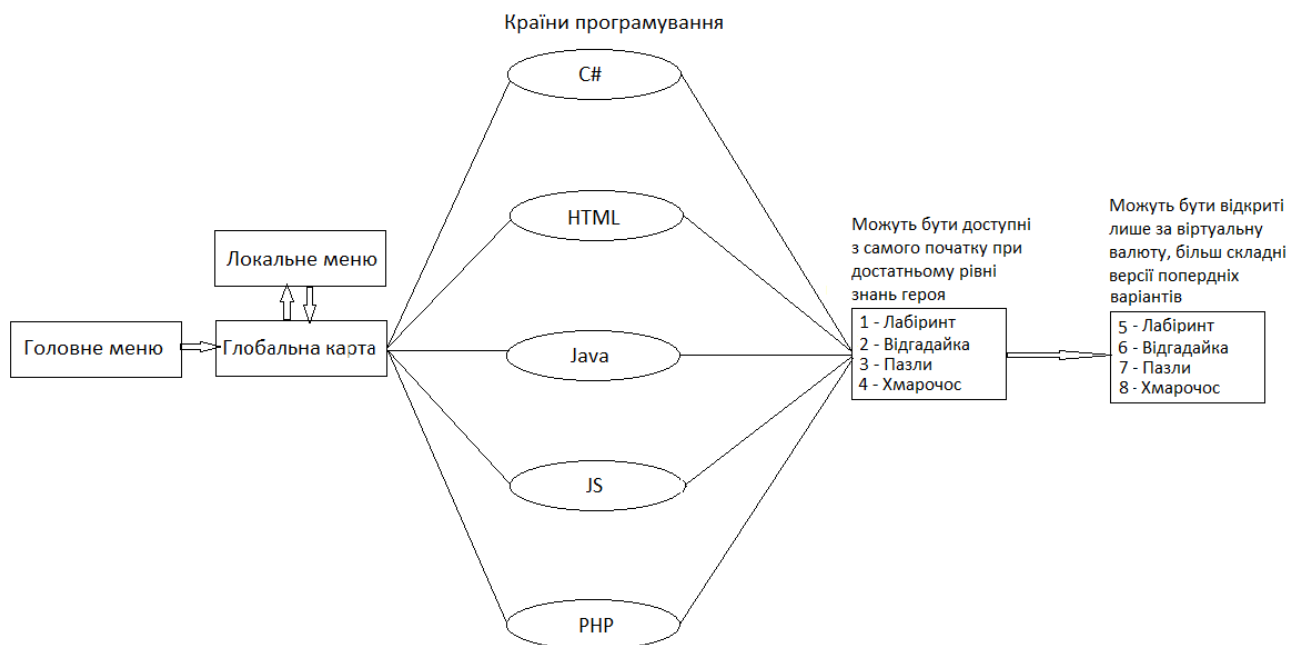


Рисунок 2.6 – Прототип системи рівнів гри «LanguageRPG»

Всього унікальних міні-ігор буде 4, всього в грі будуть присутні 40 різноманітних завдань, за кожне з котрих можна бути отримати віртуальну валюту та купити підручники по програмуванню у віртуальному магазині.

Глобальна карта буде відкритою зоною, тому гравець зможе пересуватися без всіляких перешкод. Перед практичною розробкою глобальної карти потрібно зробити простий графічний прототип для полегшення подальшого процесу. (Рисунок 2.7).

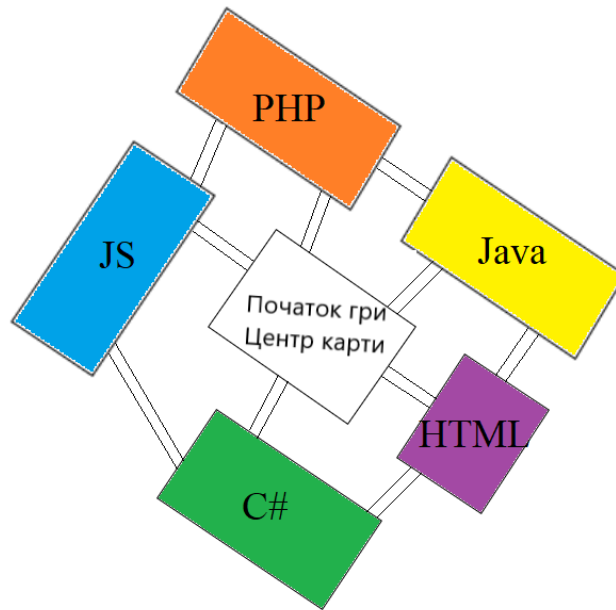


Рисунок 2.7 – Прототип глобальної карти гри «LanguageRPG»

У гравця буде присутній UI інтерфейс для полегшення навігації гри, будуть присутні такі меню як пауза, інформація о гравці, книжки та карта. (Рисунок 2.8).

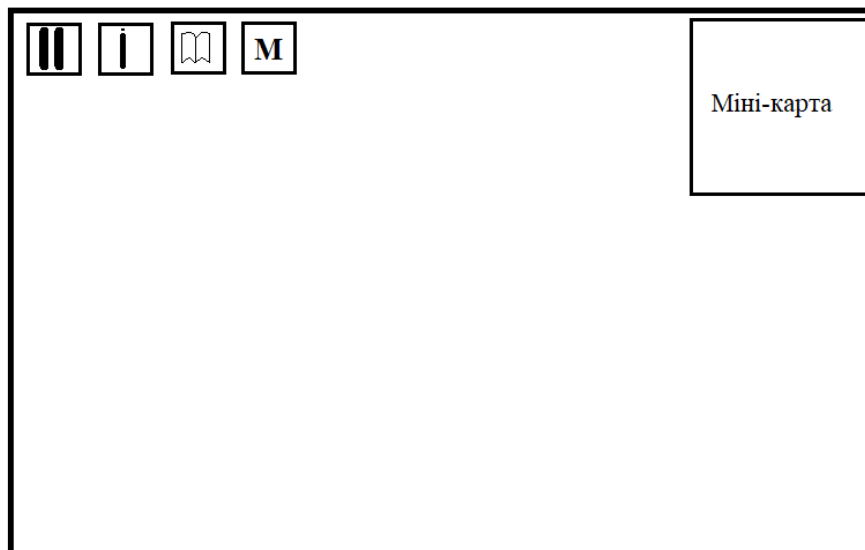


Рисунок 2.8 – Проектування інтерфейсу користувача

Після завершення квесту, на екрані з'явиться надпис про успішне завершення завдання, винагороду(віртуальна валюта), та певну кількість корисної та цікавої інформації про програмування. (Рисунок 2.9).

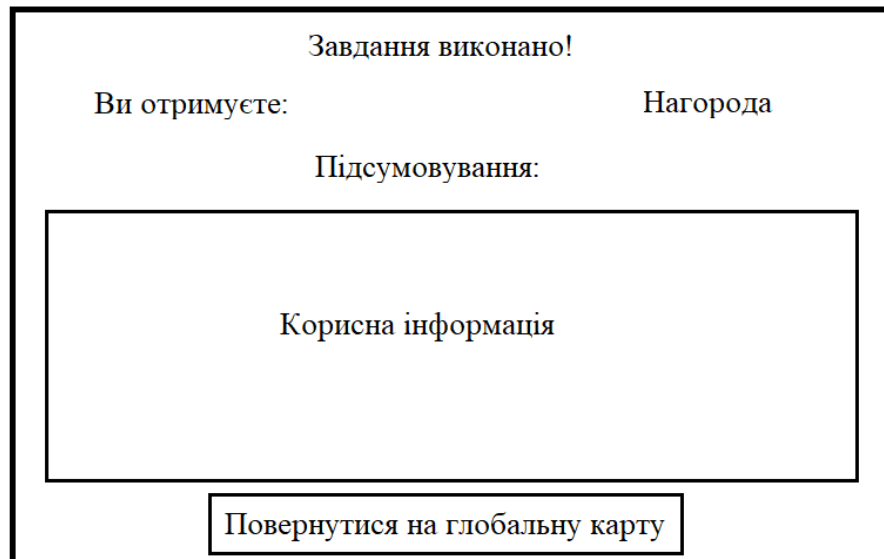


Рисунок 2.9 – Прототип відображення успішного завершення завдання.

Так як після завершення місії гравець отримає спеціальну нагороду – віртуальну валюту, її потрібно десь витратити. Для цієї цілі буде реалізовано магазин підручників, де гравець зможе придбавати різноманітні книжки для покращення своїх знань у сфері програмування. (Рисунок 2.10).

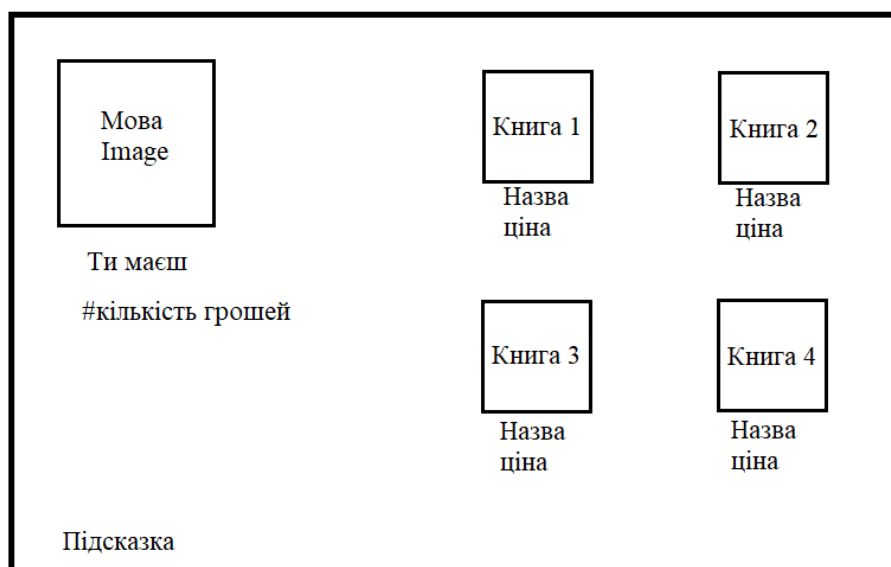


Рисунок 2.10 – Прототип магазину книжок

Також можна використовувати різноманітні UML-діаграми для подальшого планування функціоналу. UML – «Уніфікована Мова Моделювання» – це не

запатентована мова об'єктного моделювання та специфікації, яка використовується в програмній інженерії. UML включає стандартизовану графічну нотацію, яка може бути використана для створення абстрактної моделі системи. Для розробки UML моделей був використаний безкоштовний веб-сервіс App.creately.com.

Проходячи різноманітні ігри ніхто не хоче втрачати свій прогрес та починати все спочатку. Тому в проєкті буде реалізована система збереження, завдяки якій гравець зможе зберігати свій прогрес та повертатися до гри пізніше. (Рисунок 2.11).



Рисунок 2.11 – UML діаграма системи збереження

Гравець буде мати 5 балів, які він зможе розподілити на різні мови програмування. Від рівня знань буде залежати кількість доступних завдань на різних територіях, також гравець буде мати можливість придбавати різноманітні підручники по програмуванню в магазині книжок. (Рисунок 2.12).

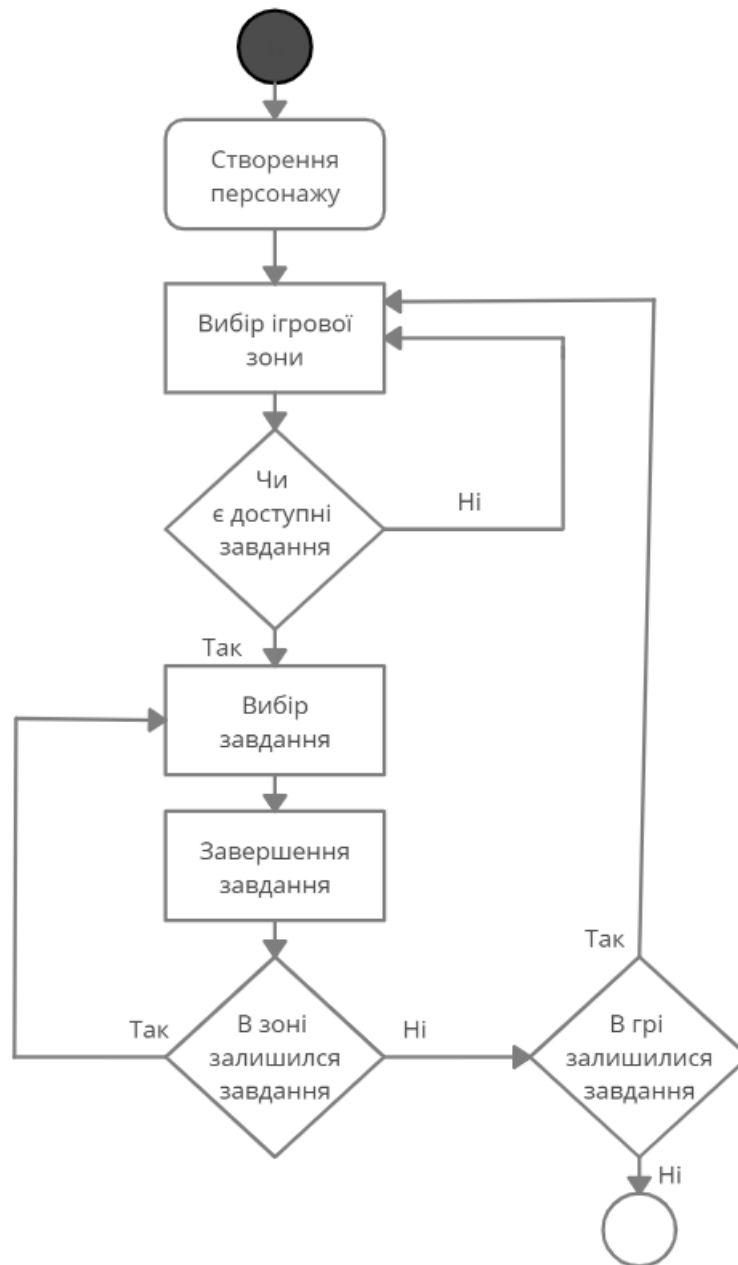


Рисунок 2.12 – Діаграма активності гри

Наступним важливим кроком буде розробити діаграму взаємодію гравця з функціоналом гри. Під час гри у гравця є можливість пересуватися та взаємодіяти з різними істотами та іншими речами, буде реалізована можливість відкривати карту для навігація та пошуку завдань та магазинів, під час паузи будуть доступні такі підменю як «Зберегти прогрес», «Головне меню» та налаштування звуку, що надає можливість кожному гравцю визначити приємний рівень звуку для різних елементів гри. (Рисунок 2.13).

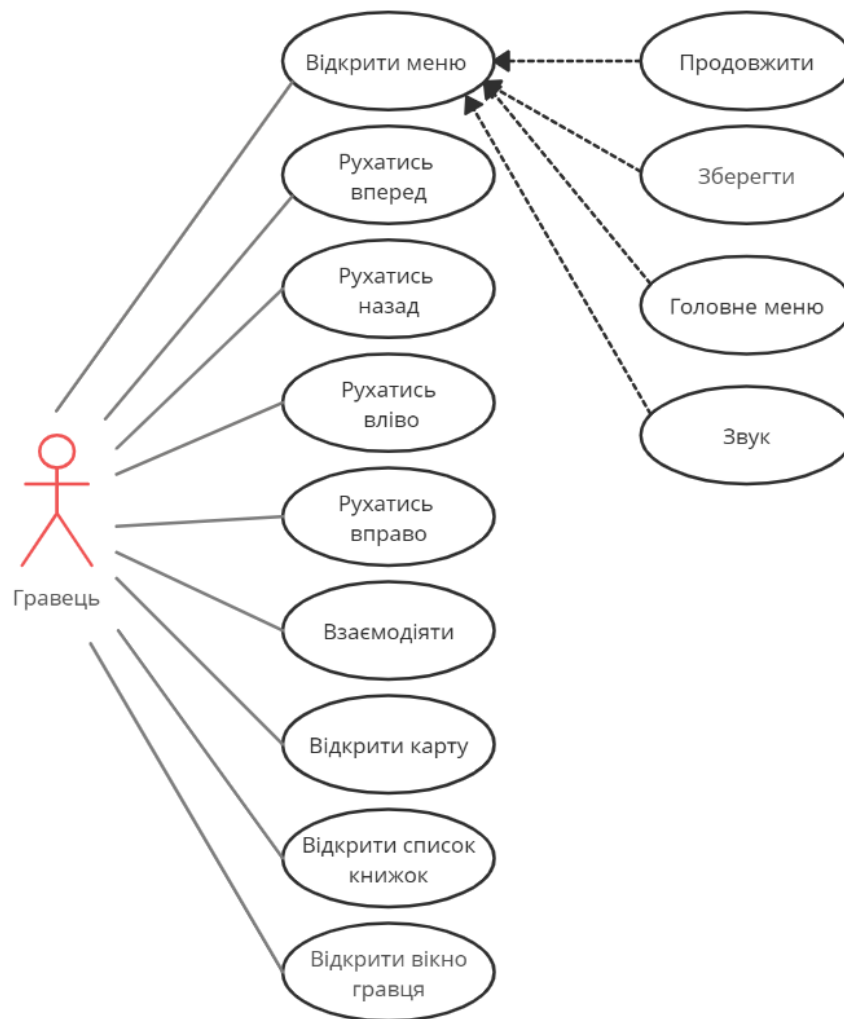


Рисунок 2.13 – Діаграма варіантів використання в ігровій сцені

Отже, після проектування та проектування основного функціонала та змісту гри та визначення основних цілей розробник може приступити до розробки проекту за допомогою середовища розробки Unity.

Висновки до розділу

Отже, у розділі було розібрано основний функціонал використаного у проекті програмного забезпечення та розроблено прототипи та діаграми для основного функціоналу проекту.

3 РЕАЛІЗАЦІЯ НАВЧАЛЬНОЇ ГРИ

3.1 Файлова структура проекту

Перед тим як приступити до розробки проекту, потрібно систематизувати розташування різних ігрових елементів по різних папкам для більш зручної навігації. (Рисунок 3.1).

Розроблений проект буде містити каталоги, в яких зберігаються:

- шаблони об'єктів;
- ігрові сцени;
- скрипти;
- зображення персонажів, фонів;
- анімації персонажів;
- звуки;
- шрифти для тексту.

В директорії Prefabs знаходяться готові шаблони ігрових об'єктів, за допомогою яких можна скласти ігровий рівень після його прототипування та проектування. Директорія Scenes містить всі сцени гри, які будуть доступні для користувача. В папці Scripts знаходяться усі скрипти проекту.

В каталозі Models містяться три підкаталоги:

- NPCs містить у собі усі 3D моделі істот, окрім моделей для гравця.
- Player містить у собі 3D моделі головних персонажів, анімації для них, та текстури для моделей.
- Objects містить у собі всі інші 3D моделі, наприклад, такі, як елементи рівнів.

Menu містить у собі елементи для меню гри.

Sounds містить всі звуки, які присутні у грі.

Controllers містить усі анімації доступні у грі.

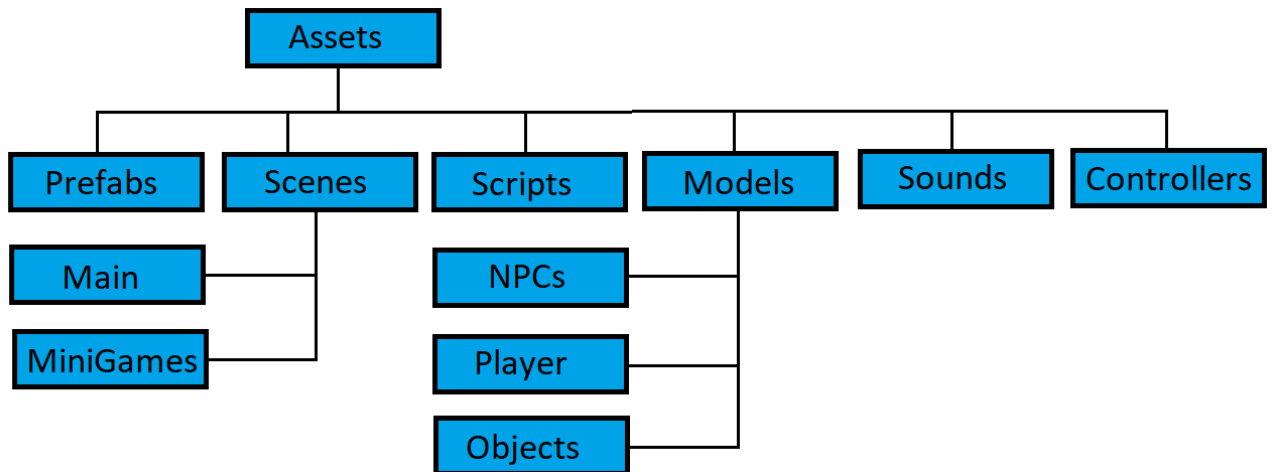


Рисунок 3.1 – Файлова структура проекту «LanguageRPG»

3.2 Реалізація навчальної гри «LanguageRPG»

Після побудови планування та прототипування основних елементів гри, наступним кроком буде реалізація системи. Для розробки проекту буде використано Unity3D, для розробки 3D моделей буде використано графічний редактор Blender та безкоштовний продукт від Microsoft – Visual Studio для написання скриптів.

Перш за все, потрібно зайти в Unity та створити новий проект та зайти в нього. Зайшовши в Unity, можна побачити, що перша сцена вже створена, але на ній присутня лише камера. Подальше потрібно заповнити сцену об'єктами.

Найпершою створеною сценою буде головне меню гри, без якого неможливо обійтися. Після розташування асетів та тексту, отримаємо графічно-готову сцену. (Рисунок 3.2).

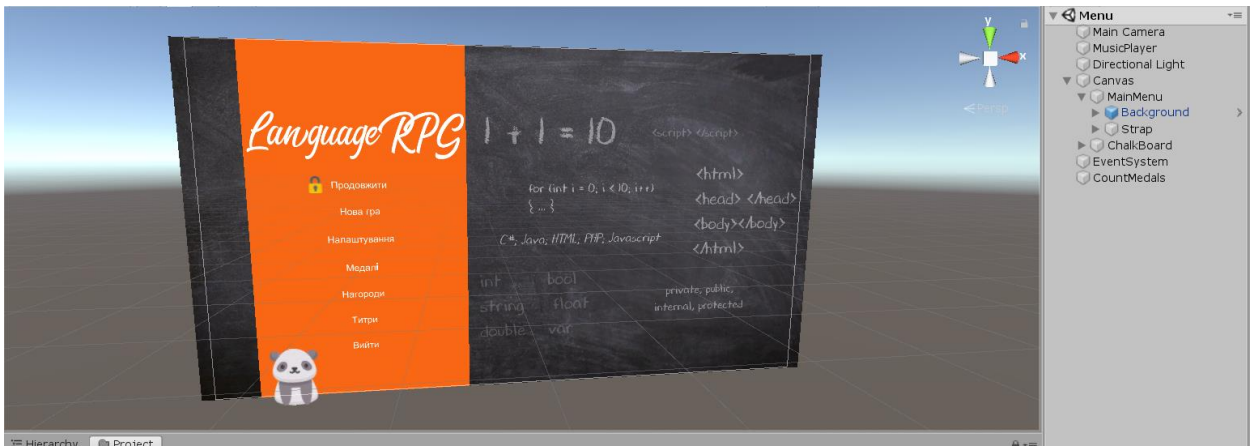


Рисунок 3.2 – Сцена головного меню

Після цього переходимо до потрібних об'єктів в підменю Ієрархія та додаємо певні властивості до кожного елемента. Після нажаття кнопки Add Component можна додати потрібні елементи. Наприклад, Physics додає об'єкту фізичні властивості, а New Script створює новий скрипт. (Рисунок 3.3).

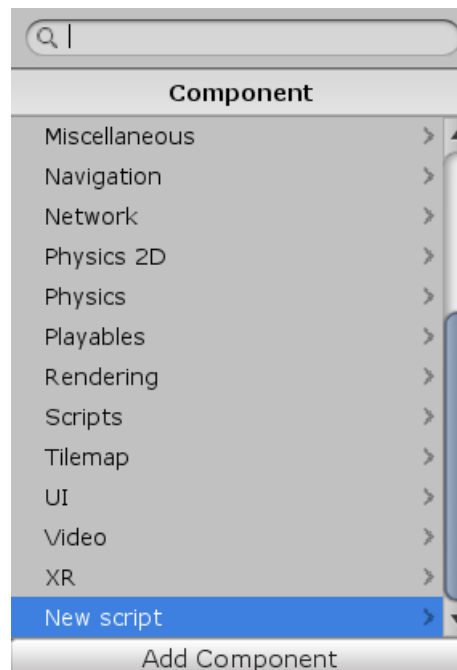


Рисунок 3.3 – Додавання нових властивостей до об'єкту

Після нажаття кнопки Play можна побачити готову сцену. (Рисунок 3.4).

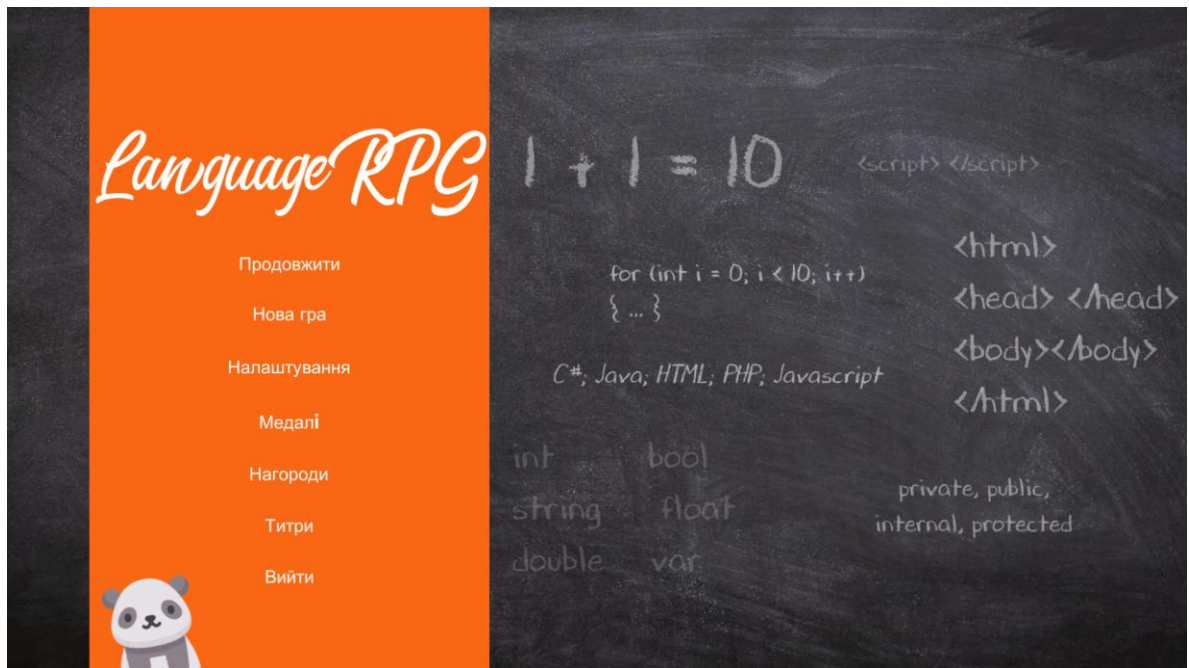


Рисунок 3.4 – Головна сцена Меню в режимі Play

Після додавання скриптів для пунктів меню, потрібно створити нові сцени по такому ж принципу. Першою була розроблена сцена «Медалі» – пройшовши всі 8 завдань певної країни програмування гравець отримає медалі, які свідчать о його досягненнях, кнопка назад повертає гравця на попередню сцену – «Меню». (Рисунок 3.5).

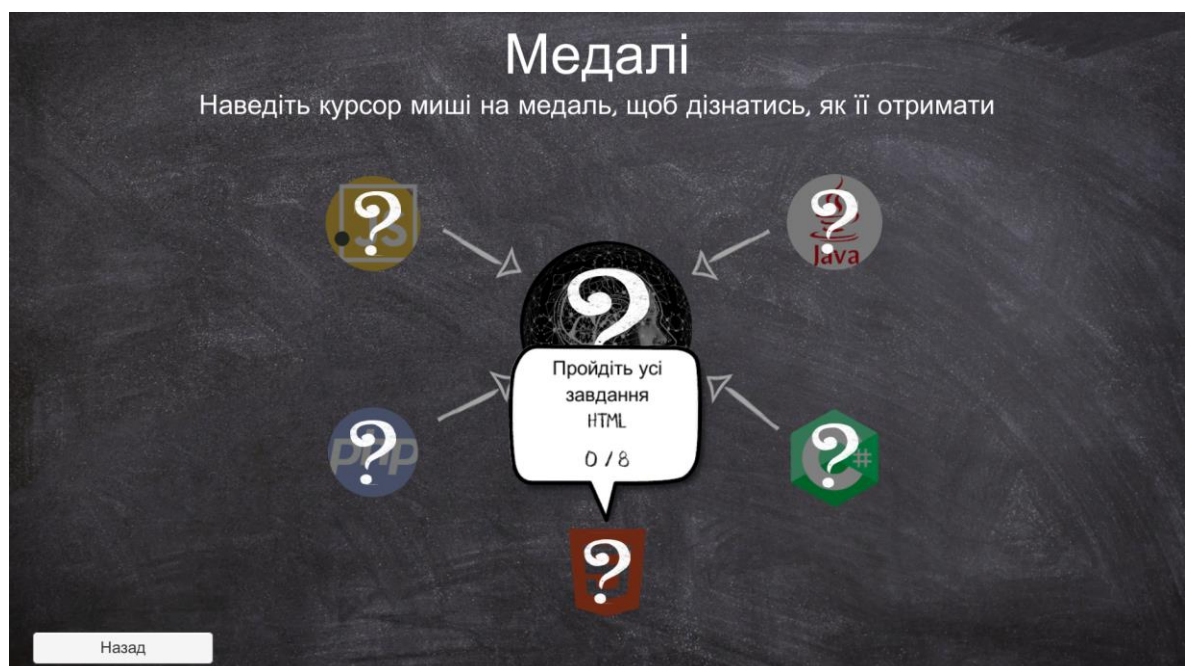


Рисунок 3.5 – Сцена Медалі

Далі було реалізовано певну систему нагород за отримані медалі для стимулу гравця. (Рисунок 3.6). Отримавши певну кількість медалей буде відкрито певні додаткові бонуси:

- 2 медалі – постать лицаря(відкриває нову модель для гравця до попередніх трьох);
- Додатковий бал при створенні нового персонажу, за допомогою якого, гравець зможе пройти більше квестів за один раз;
- Додаткова персоналізація лицаря(гравець зможе змінювати зовнішній вигляд моделі лицаря);
- Додаткова корисна інформація(буде отримана корисне посилання на веб сайт, де можна знайти дуже багато корисної інформації по програмуванню).

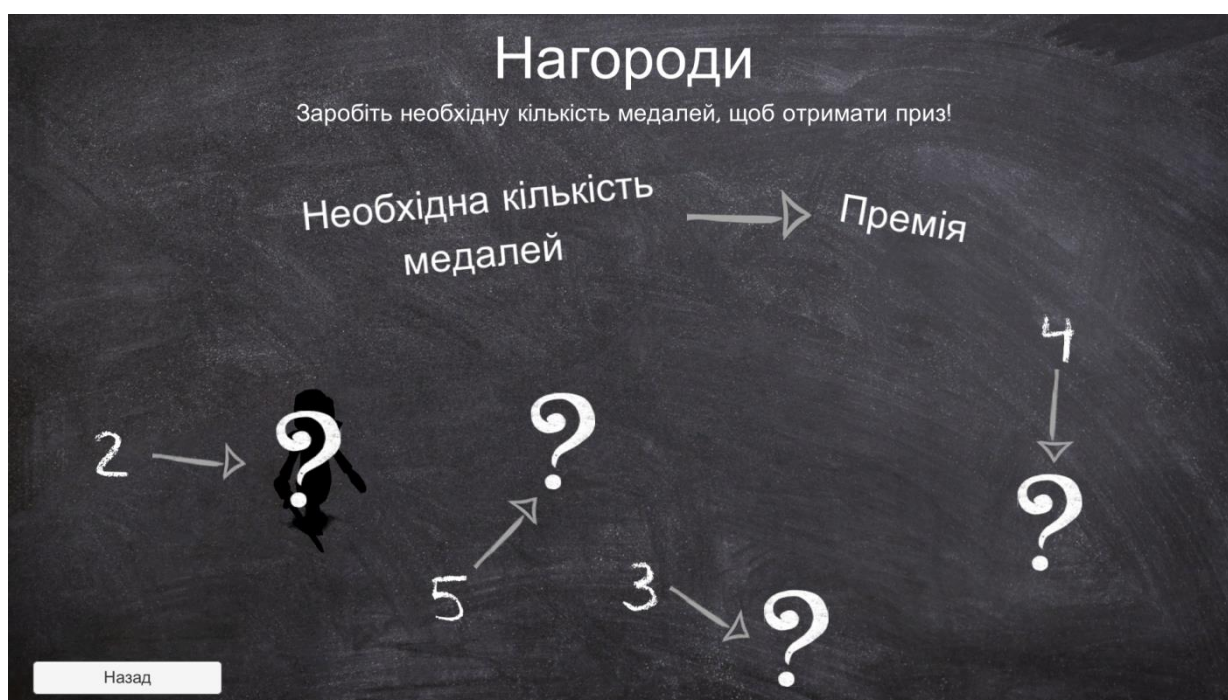


Рисунок 3.6 – Сцена Нагороди

Після цього було розроблено меню «Налаштування», де гравець може змінювати налаштування до комфортних йому значень. В меню присутні такі пункти, як графіка – де гравець може змінювати налаштування графіки, звук, де можна змінювати гучність різних джерел звуку та керування, де прописано весь

функціонал кнопок. (Рисунок 3.7).

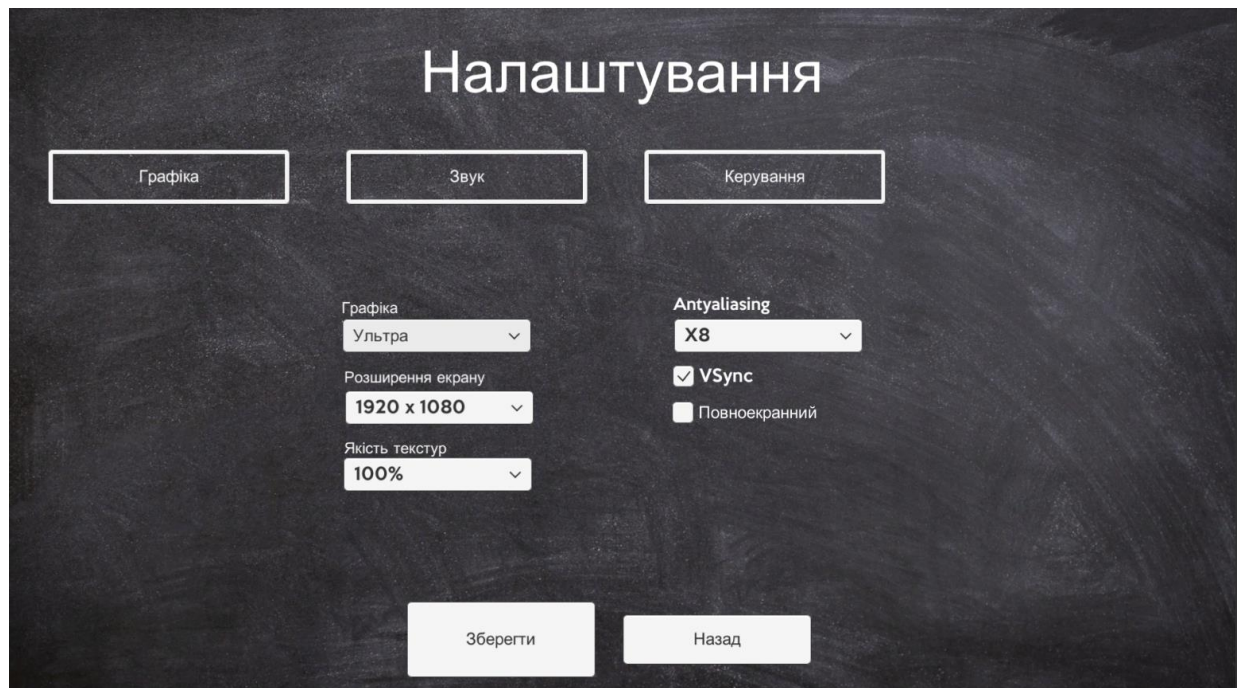


Рисунок 3.7 – Налаштування гри

Сцена «Титри» відображає інформацію про проект: ким було розроблено, для чого, також зліва від тексту відображаються картинки з гри, що робить титри більш цікавими для користувача. (Рисунок 3.8).

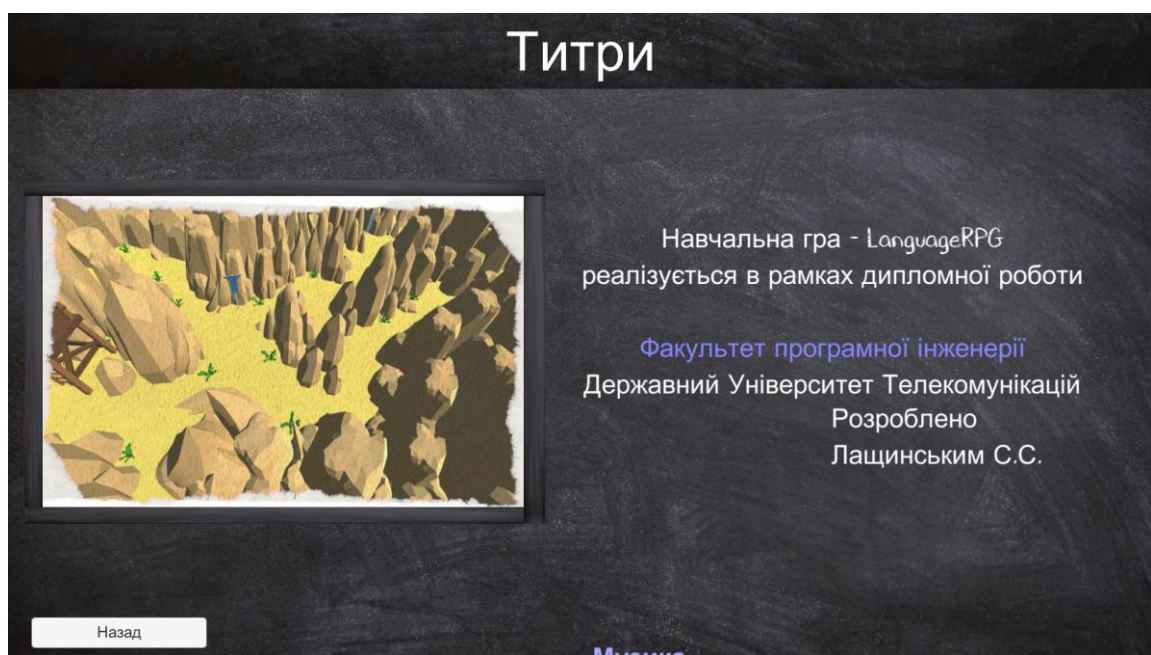


Рисунок 3.8 – Сцена Титри

В кінці розробки підпунктів меню була розроблена найголовніша сцена – створення персонажу, де гравець може змінити ім'я для головного героя, яке буде відображатися під час діалогів з істотами, вибір моделі та може розподілити бали навичок.



Рисунок 3.9 – Сцена створення персонажу

Тепер потрібно створити персонажа для гравця, для цього перетягуємо ассет на нову сцену та додаємо потрібні компоненти, такі як Rigidbody(цей компонент надає об'єкту фізичні властивості), скрипти, та інше. (Рисунок 3.10). Після додавання усіх компонентів та коректного розміщення на глобальній карті – ігровий персонаж – готовий для використання.

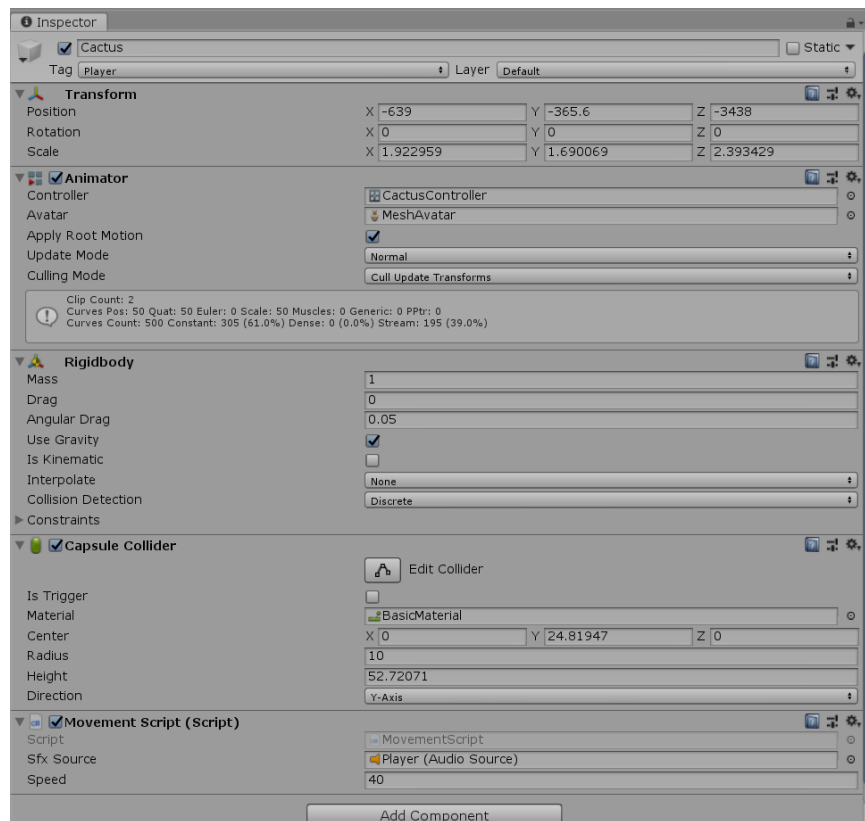


Рисунок 3.10 – Асет персонажу у Інспекторі

Для того, щоб кожен раз не створювати персонажа заново, можна використовувати Prefab, це допомагає зберігати об'єкт зі всіма його властивостями. Створити Prefab досить просто, достатньо просто перетягнути створений об'єкт до списку асетів. За допомогою Prefab-ів можна дуже прискорити то полегшити розробку гри. (Рисунок 3.11).

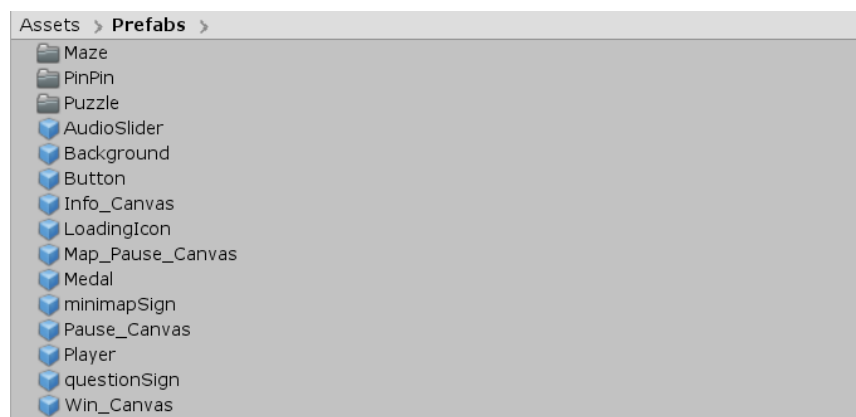


Рисунок 3.11 – Створені Prefab-и

Після того, як гравець натискає кнопку «Грати», його переносить на головну та найбільшу сцену гри «Мар», де і відбуваються усі дії гри. Після переходу на цю сцену, гравець одразу отримує корисні підказки про функціонал гри. (Рисунок 3.12).

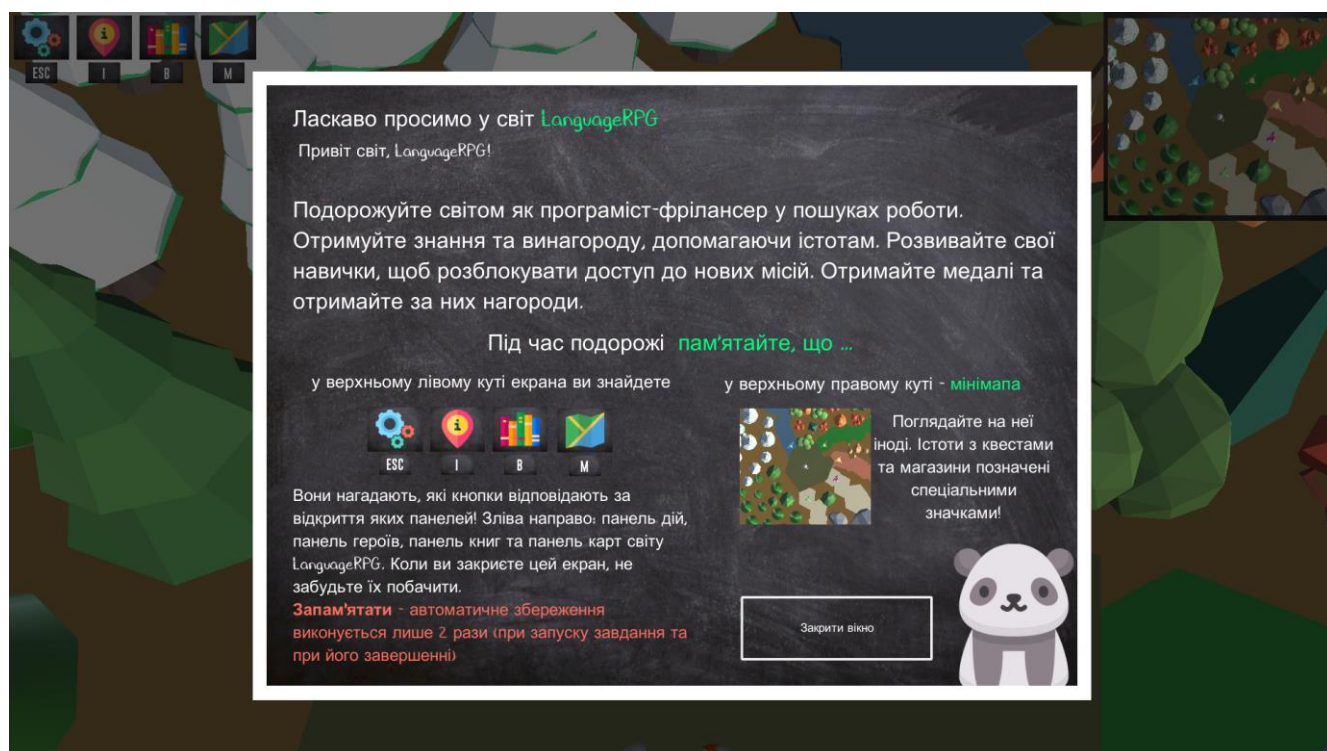


Рисунок 3.12 – Підказка при переході на сцену Мар

Також в сцені присутні інші підказки. Наприклад, до книгарень було додано колайдер з тригером, який при взаємодії з гравцем виводить на екран підказку про книгарні. (Рисунок 3.13). Зайшовши до книгарні, гравець отримує графічний список книжок тієї країни, де розташована книгарня.

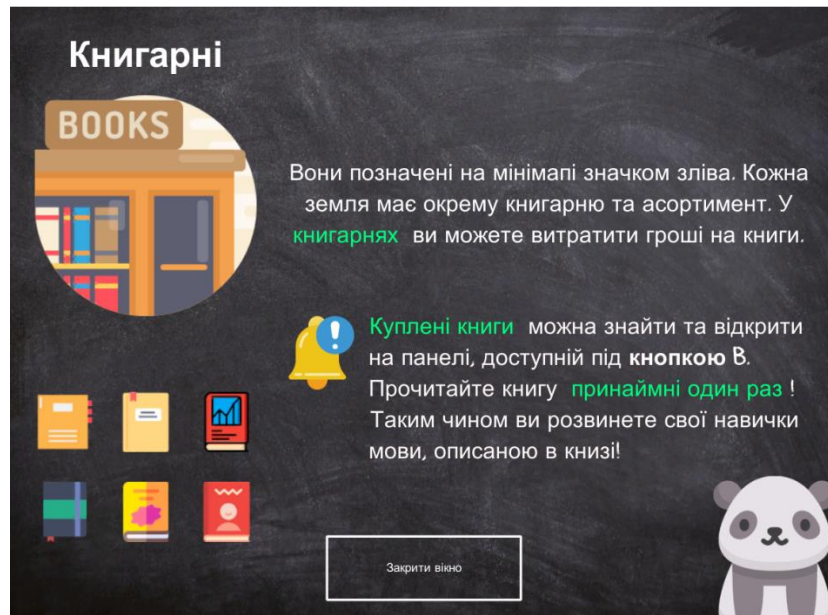


Рисунок 3.13 – Підказка при взаємодії з тригером книгарні

Після придбання книжки, її можна буде прочитати у спеціальному меню, яке викликається за допомогою клавіші В. (Рисунок 3.14). Після нажаття «читати» буде відкрита книга, де гравець може знайти багато корисної інформації. (Рисунок 3.15).



Рисунок 3.14 – Графічний список книжок

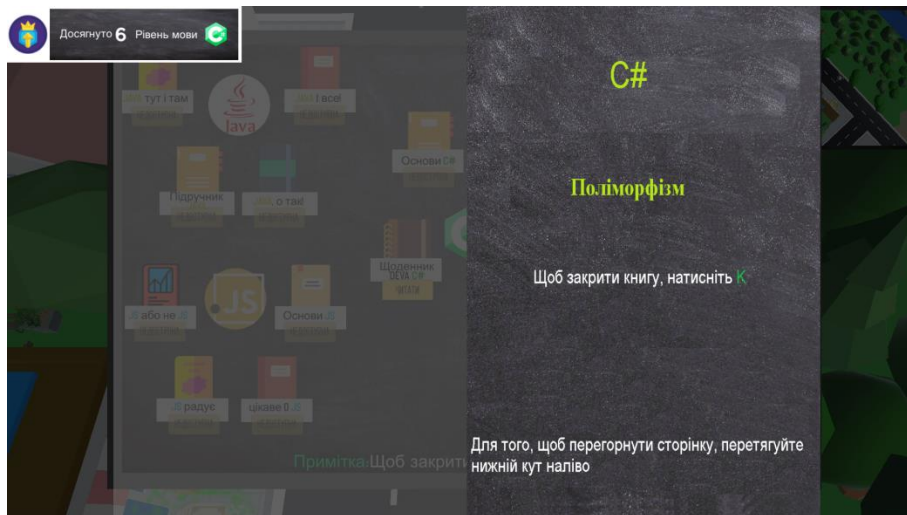


Рисунок 3.15 – Відкрита книга

Також була реалізована взаємодія між книжками та характеристиками головного героя. При створенні персонажу є можливість розподілити 5 балів навиків, що відкриває приблизно 4 місії із 40, доступних у грі, для того, щоб відкрити інші потрібно придбавати книжки, при прочитанні яких – збільшуються характеристики головного героя, що відкриває нові квести. Така система допомагає збільшити бажання користувачів читати книжки про програмування, тому що, якщо у гравця не було б такої потреби, то багато хто би не читав книжки. (Рисунок 3.16).

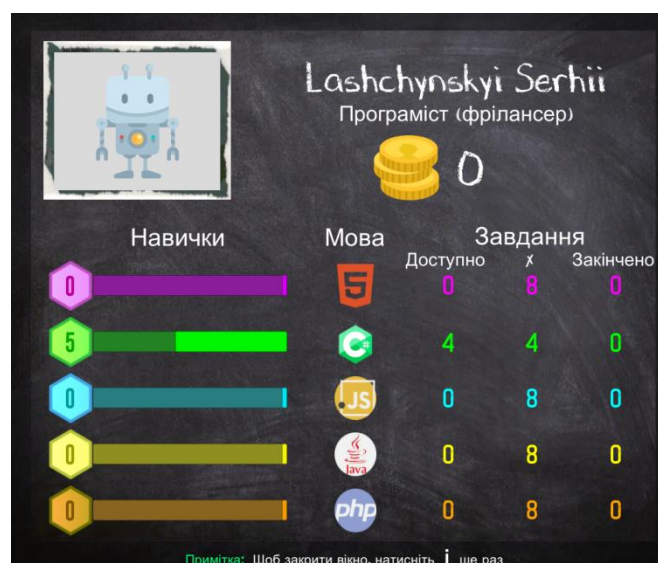


Рисунок 3.16 – Характеристики гравця

Після реалізації системи книжок та характеристик головного героя, потрібно реалізувати систему отримання віртуальної валюти – монет, за які і можна буде придбати необхідні книжки. Валюту можна отримати за проходження різних місій, кількість отриманих монет буде залежати від кількості помилок. Для початку місії потрібно знайти істоту зі знаком оклику та натиснути F. Після цього буде відкрито діалог з істотою. (Рисунок 3.17).

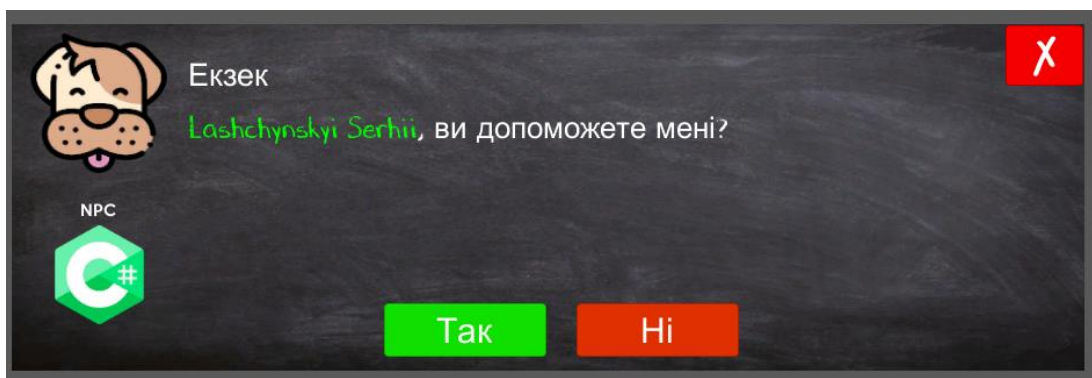


Рисунок 3.17 – Кінець діалогу з істотою

Кожна міні-гра була розроблена як окрема сцена. Після завантаження кожної сцени з міні-грою перед гравцем спочатку відкривається текст з підказками та описом певної міні-гри. (Рисунок 3.18).

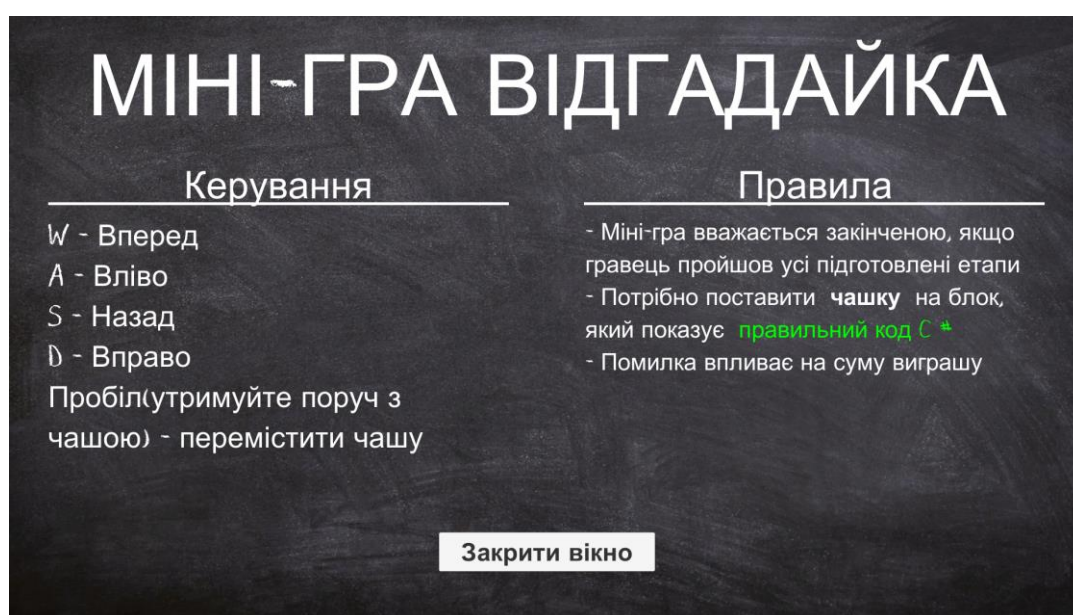


Рисунок 3.18 – Підказки до міні-гри «Відгадайка»

Після закриття підказки, гравець отримує контроль над своїм персонажем, Першою реалізованою міні грою була міні-гра «Відгадайка», де гравцю потрібно переміщувати чашу за допомогою клавіши «Пробіл», після розташування чаші на правильному фрагменті коду, гравець має встати на кнопку чек, якщо відповідь правильна – тоді завантажуються наступне завдання, якщо ні – то програється певний звуковий сигнал, і гравець має обрати правильну відповідь. (Рисунок 3.19).

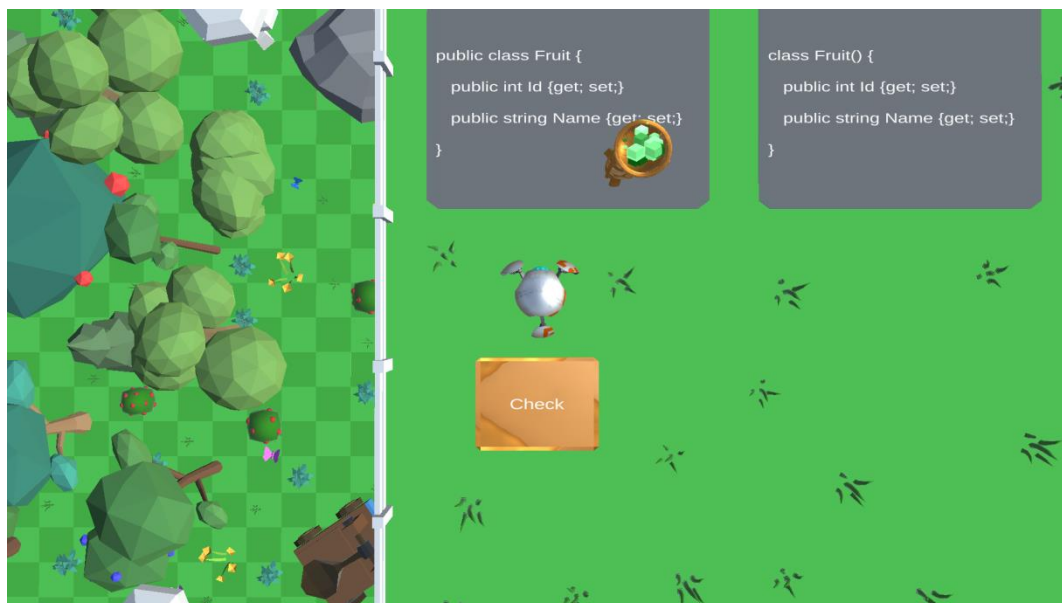


Рисунок 3.19 – Міні-гра «Відгадайка»

Після проходження усіх рівнів міні-гри перед гравцем відкривається вікно завершення міні-гри, де написано кількість отриманих монет та певну кількість корисної інформації. Нагорода залежить від кількості зроблених помилок. (Рисунок 3.20).

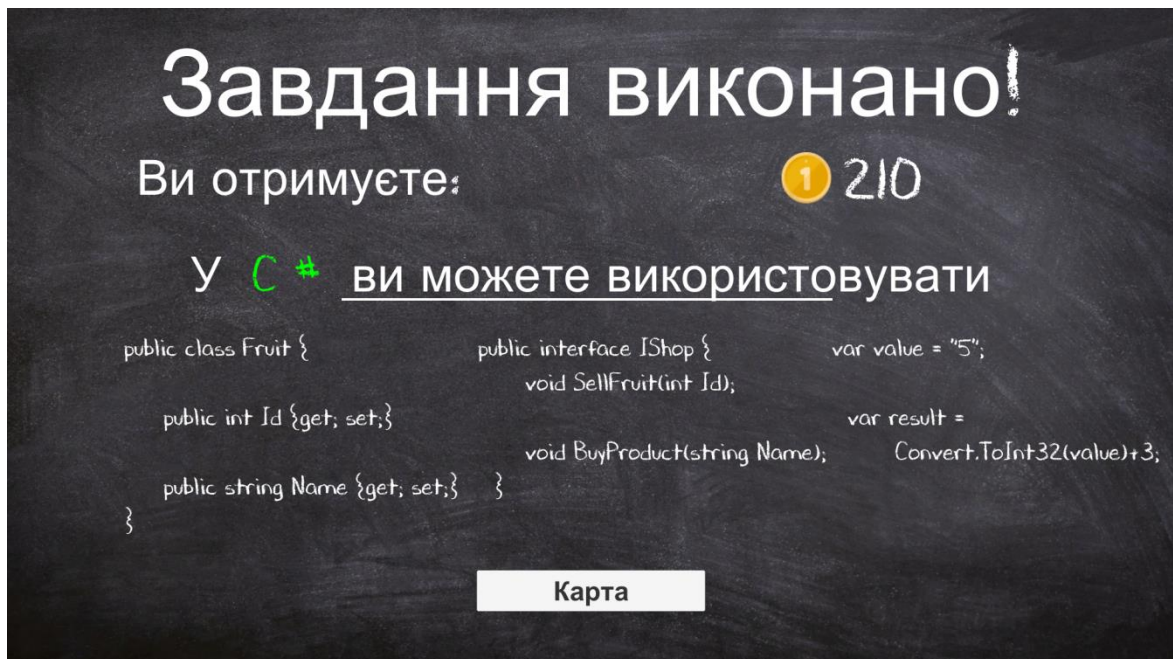


Рисунок 3.20 – Переможний екран міні-гри «Відгадайка»

Наступною реалізованою міні-грою стала міні-гра «Хмарочос». В міні-грі є питання та 4 відповіді, цю міні-гру можна назвати звичайним тестом, після відповіді на кожне питання гравець спускається по один рівень нижче, після проходження усіх рівнів сцени, гравець отримає нагороду та корисну інформацію. (Рисунок 3.21).

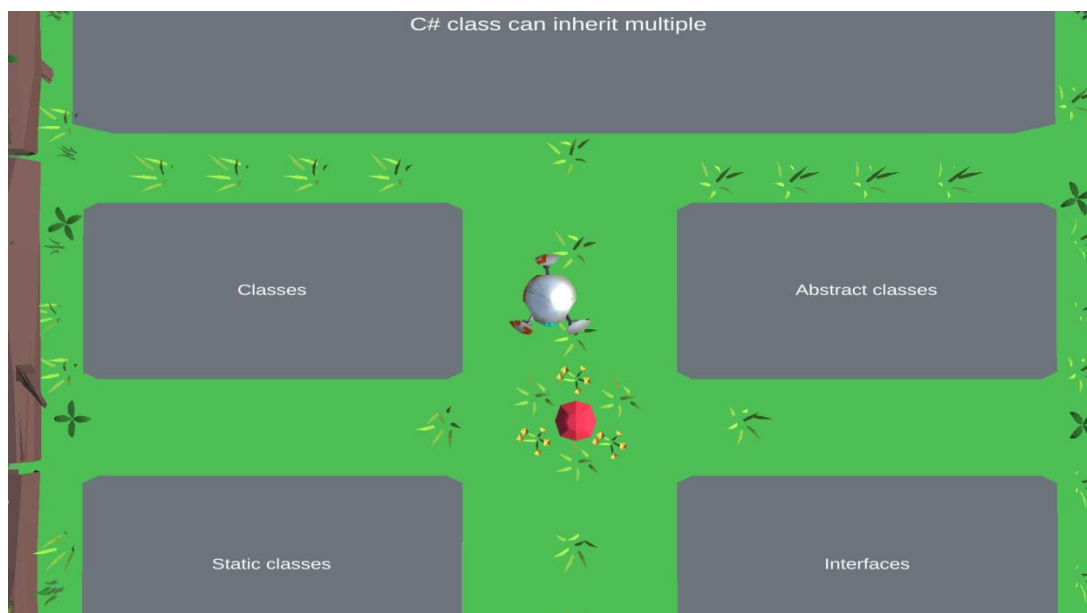


Рисунок 3.21 – Міні-гра «Хмарочос»

Залишилося реалізувати ще дві міні-гри. Наступна гра це «Пазли», де гравцю потрібно розмістити кубики з елементами коду у правильному порядку, в гравця є необмежена кількість часу та спроб, щоб це зробити. Нагорода завжди фіксована під час цієї міні-гри. Ця гра допомагає учню зрозуміти структуру коду. (Рисунок 3.22).

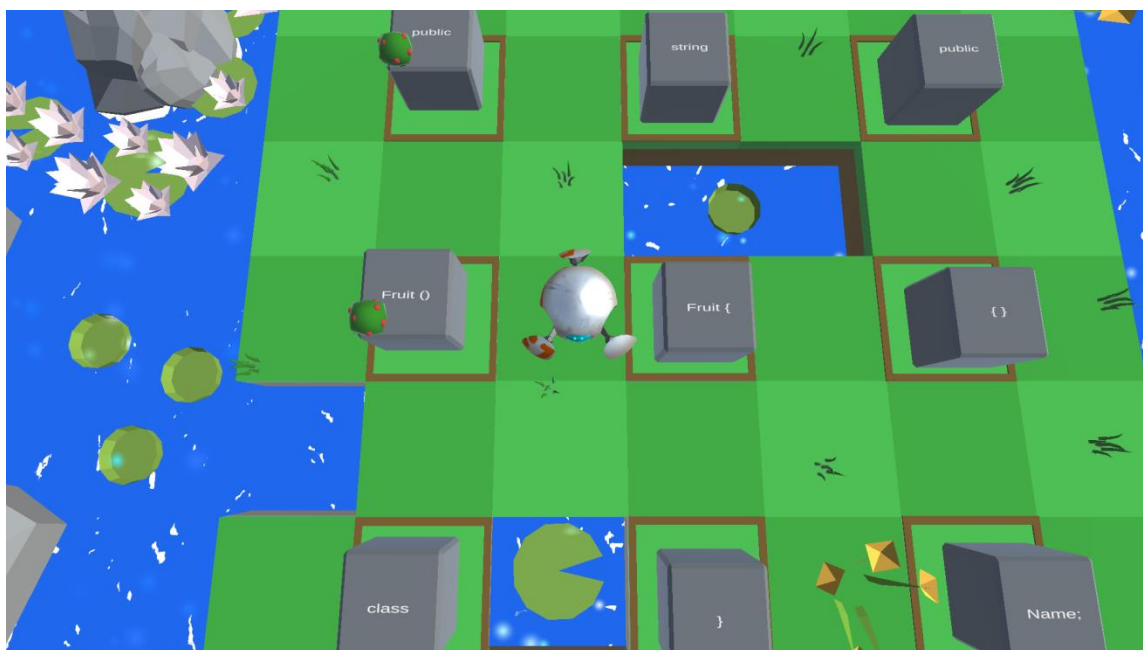


Рисунок 3.22 – Міна-гра «Пазли»

Останньою розробленою міні-грою був «Лабіринт», суть цієї гри полягає у пошуках елементів коду, гравець має уникати блоки з багами, ця міні-гра не вчить програмуванню, але вона навчає увазі та терпінню(можна пройти взявши блок багу чи шукати обхідні шляхи), а ці показники дуже важливі у сфері ІТ. Кількість зібраних багів впливає на нагороду, зібравши занадто багато, гравець має почати рівень знову. У верхньому правому куті відображено кількість знайдених елементів коду та багів. (Рисунок 3.23). Гра закінчується, коли зібрано усі елементи коду, зазвичай, це 6 частин. Після завершення міні-гри гравець також отримає нагороду та певну кількість корисної інформації.

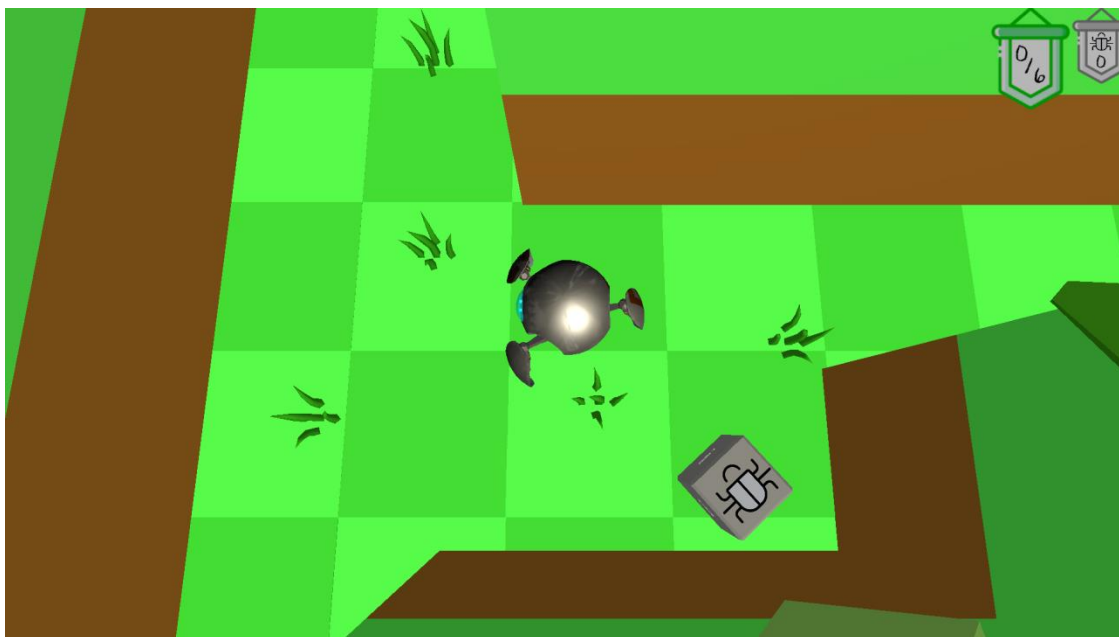


Рисунок 3.23 – Міна-гра «Лабіринт»

Після розробки звичайних 4 сцен міні-ігор для кожної країни програмування, було розроблено нові 4 версії. Єдина різниця між цими сценами – це збільшений рівень складності, наприклад, більш складні питання по програмуванню. В більш складній версії міні-гри «Лабіринт» гравець залишився без глобального освітлення, він має лише ліхтарик, що збільшує важкість знаходження правильних елементів коду. (Рисунок 3.24).

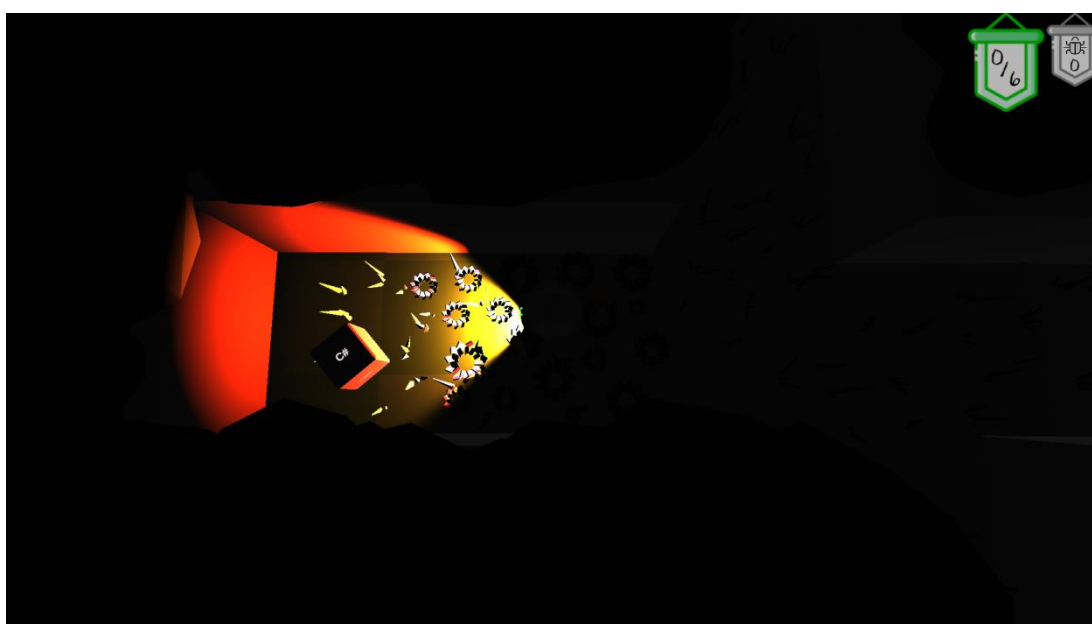


Рисунок 3.24 – Ускладнена версія міна-гри «Лабіринт»

Після закінчення розробки основної частини гри, залишилися реалізувати додаткові елементи. Першим пунктом було розроблення повідомлення про закінчення активних завдань – якщо гравець пройшов усі доступні завдання, він має почати нову гру для допомоги істотам у інших країнах програмування. (Рисунок 3.25).

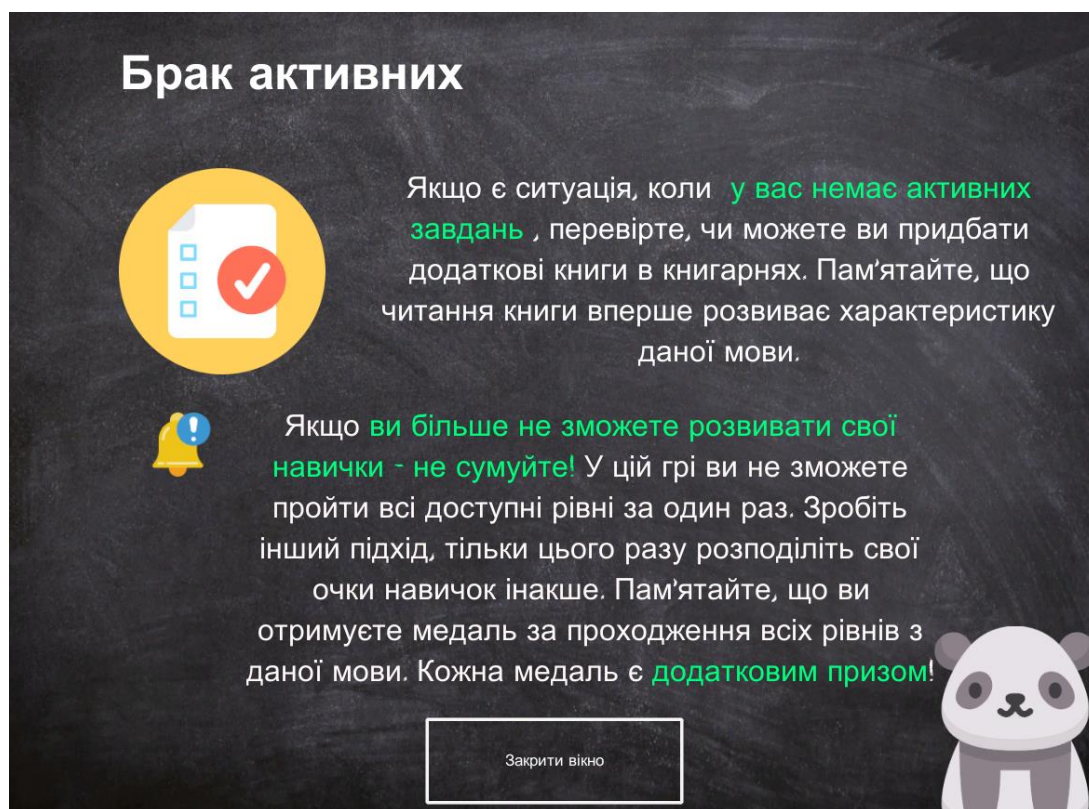


Рисунок 3.25 – Повідомлення про закінчення активних завдань

Якщо гравцю вдасться пройти за один раз усі 8 завдань однієї з країн програмування він отримає нагороду – медаль. (Рисунок 3.26 та 3.27). Також, за певну кількість медалей можна буде отримати нагороди.

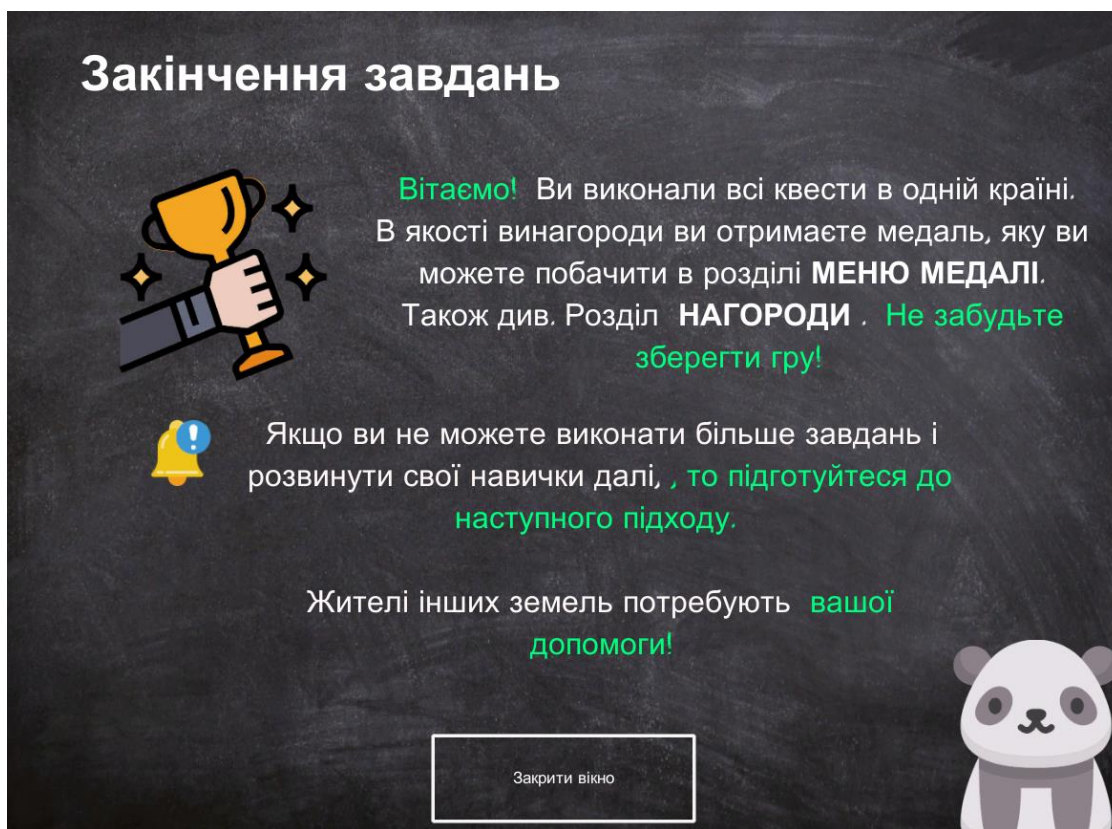


Рисунок 3.26 – Отримання медалі після проходження усіх завдань

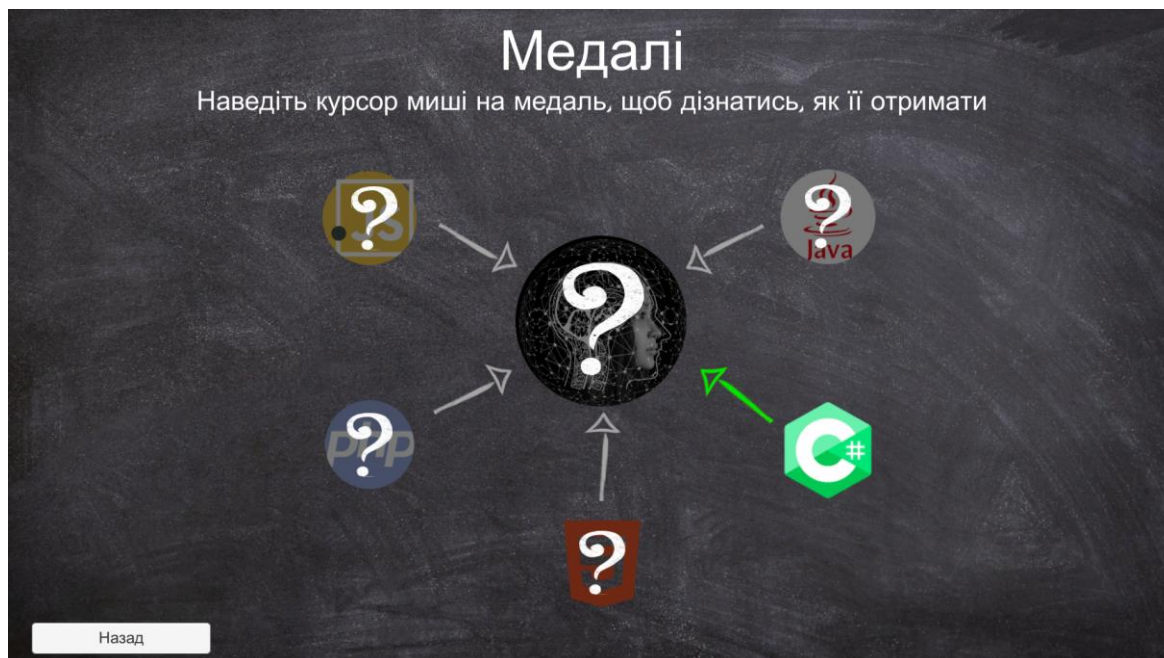


Рисунок 3.27 – Відображення отриманої медалі

Після завершення розробки усіх сцен та переходів між ними потрібно «збудувати» білд гри для подальшого його використання у тестуванні. Для

побудови білда потрібно натиснути File та обрати Build settings. Після цього потрібно додати усі сцени у порядку їх розташування, обрати платформу та вибрати шлях, куди буде скопільовано білд. (Рисунок 3.28).

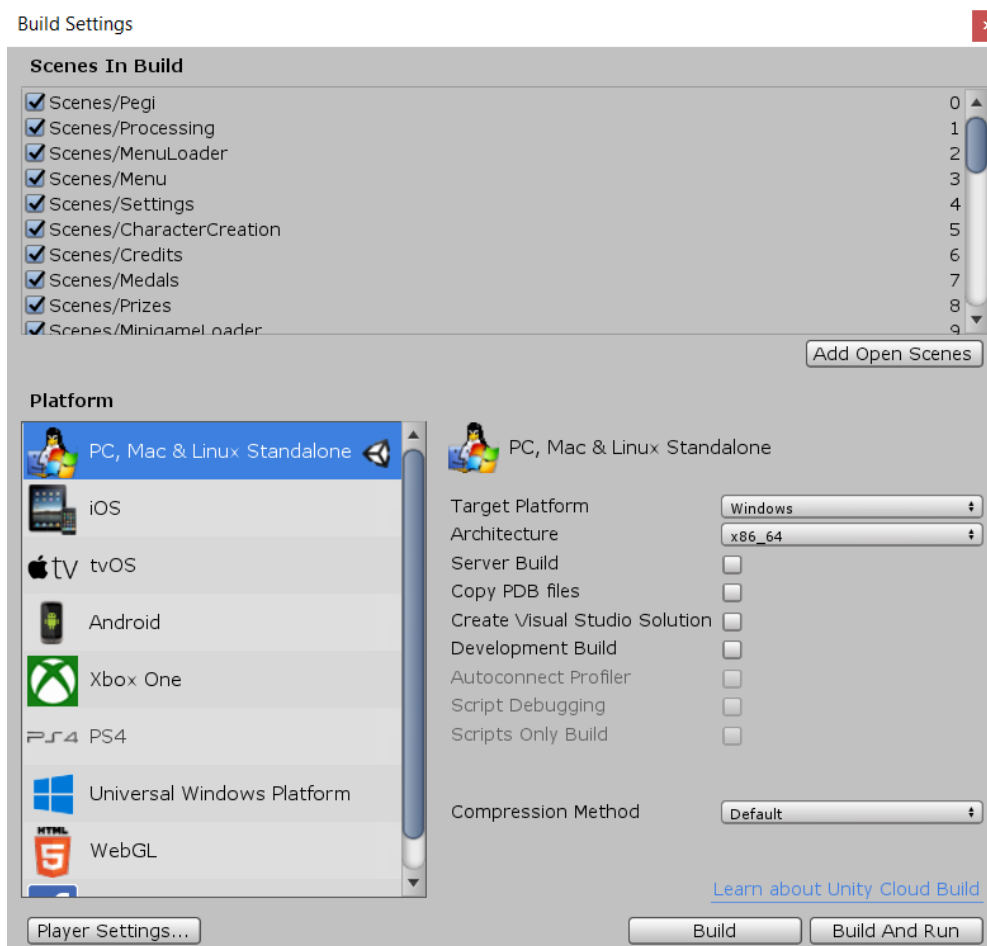


Рисунок 3.28 – Створення білда проекту

3.3 Тестування навчальної гри «LanguageRPG»

Після побудови робочого білду потрібно приступити до найважливішого кроку при розробці проекту – тестуванню або контролю якості.

Контроль якості (Quality Control) – це сукупність процедур, що використовуються для перевірки програмного продукту, щоб він відповідав своїй документації, цілям та був зручний до використання.

Контроль якості програмного забезпечення стосується як визначених функціональних вимог, так і нефункціональних вимог, таких як продуктивність,

зручність використання, швидкість роботи та інше. Це також стосується здатності програмного забезпечення добре працювати в непередбачуваних сценаріях(negative cases) та утримувати відносно низький рівень дефектів.

Баг – це помилка чи неправильна роботи програмного забезпечення, якщо програма працює правильно, але не відповідає документації чи вимог – це також баг.

Для створення та трекінгу баг репортів було використано безкоштовний сервіс Jira. Jira – продукт для відстеження процесу розробки програмного забезпечення, розроблений Atlassian, який дозволяє відстежувати помилки та підтримує гнучке управління проектами. Після реєстрації аккаунту користувач має створити та обрати проект, після цього можна приступити до створення баг репортів. (Рисунок 3.29).

Після знаходження дефекту у грі потрібно створити новий баг репорт, для цього потрібно натиснути кнопку create та заповнити усі поля:

- Project – обираємо проект, де було знайдено дефект;
- Issue Type – завжди вибираємо bug, якщо Jira використовується лише для трекінгу дефектів;
- Summary – короткий опис проблеми, звичайно, одне речення, текст записаний у квадратні дужки – це теги, які допомагають знайти дефекти дуже швидко;
- Description – детальний опис проблеми, де потрібно вказати усі відомі факти, щоб розробнику було легше зрозуміти та вирішити проблему;
- Priority – важливість проблему, поділяється на 4 рівні: Lowest, Low, Medium, High;
- Labels – схожі за функціоналом на теги, використовуються для певних задач, або коли занадто багато тегів;
- Attachments – можна додати різні файли, наприклад, картинки чи відео;
- Assignee – до кого буде прив'язано баг, ця людина може просте відповідати за нього чи бути розробником, який буде фіксувати цей баг.

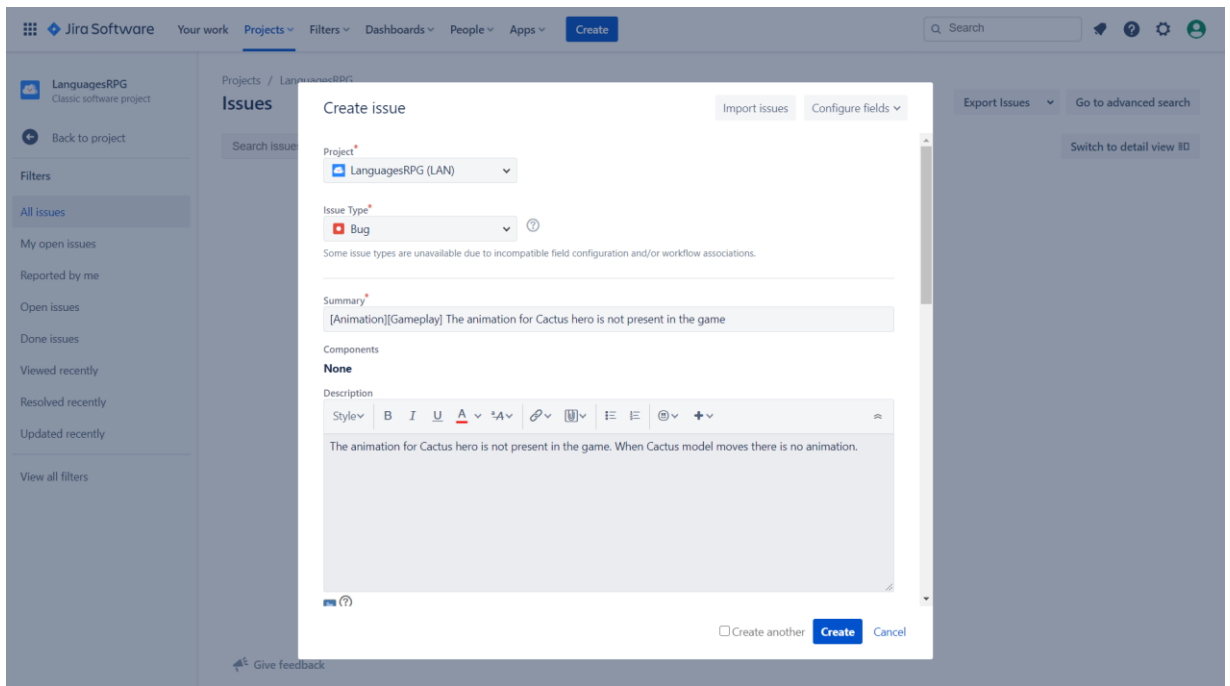


Рисунок 3.29 – Створення баг репорту за допомогою Jira

Після натискання кнопки Create буде створено оформлений раніше баг репорт і він з'явиться у списку багів проекту, де його можна подальше відслідковувати. (Рисунок 3.30).

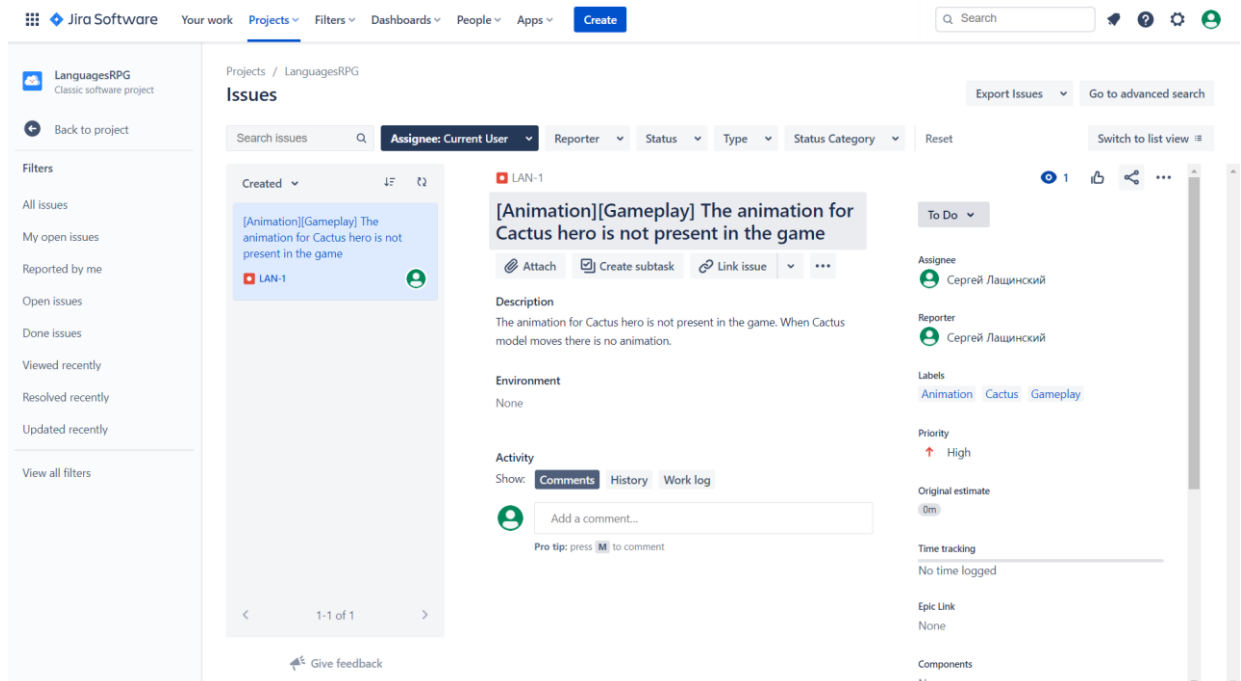


Рисунок 3.30 – Створений баг репорт за допомогою сервісу Jira

Останнім кроком у розробці буде створення набору тест кейсів, за допомогою яких можна буде перевірити рівень готовності проекту. Для цієї цілі було використано платний сервіс TestRail.

TestRail – це веб-інструмент управління тестами, який використовується тестувальниками, розробниками та іншими зацікавленими сторонами для управління, відстеження та організації спроб тестування програмного забезпечення. Він дотримується концепції централізованого управління тестами, яке допомагає у легкому спілкуванні та дозволяє швидко розробити завдання серед команди контролю якості та іншими зацікавленими сторонами.

Після реєстрація облікового запису потрібно створити проект та набір тест кейсів для нього. (Рисунок 3.31). Для створення тест кейсу потрібно заповнити наступні поля:

- Title – назва тест кейсу;
- Section – назва набору тест кейсів;
- Type – вид тест кейсу;
- Priority – пріоритет;
- Steps – кроки виконання;
- Expected Result – очікуваний результат.

Рисунок 3.31 – Створення тест кейсу

Після створення повного набору тест кейсів можна приступити до тестування, після запуску тесту потрібно проставляти результат для кожного тест кейсу. (Рисунок 3.32). Потрібно заповнити такі поля як:

- Status – результат проведеного тесту;
- Comment – коментар;
- Assign to – до кого прикріпити тест кейс;
- Version – версія продукту, використаної для тестування;
- Defects – посилання на виявленні дефекти.

Після заповнення усіх потрібних полів та натискання кнопки «Add Test Case» було створений новий тест кейс.

Рисунок 3.32 – Заповнення результатів тесту

Закінчивши заповнювати всі пункт набору тест кейсів, можна побачити загальну картину готовності продукту. (Рисунок 3.33). Всі тест кейси мають статус «Passed», це означає, що не було знайдено ніяких критичних багів під час фінального тестування, і це значить, що проект готовий для використання іншими користувачами.

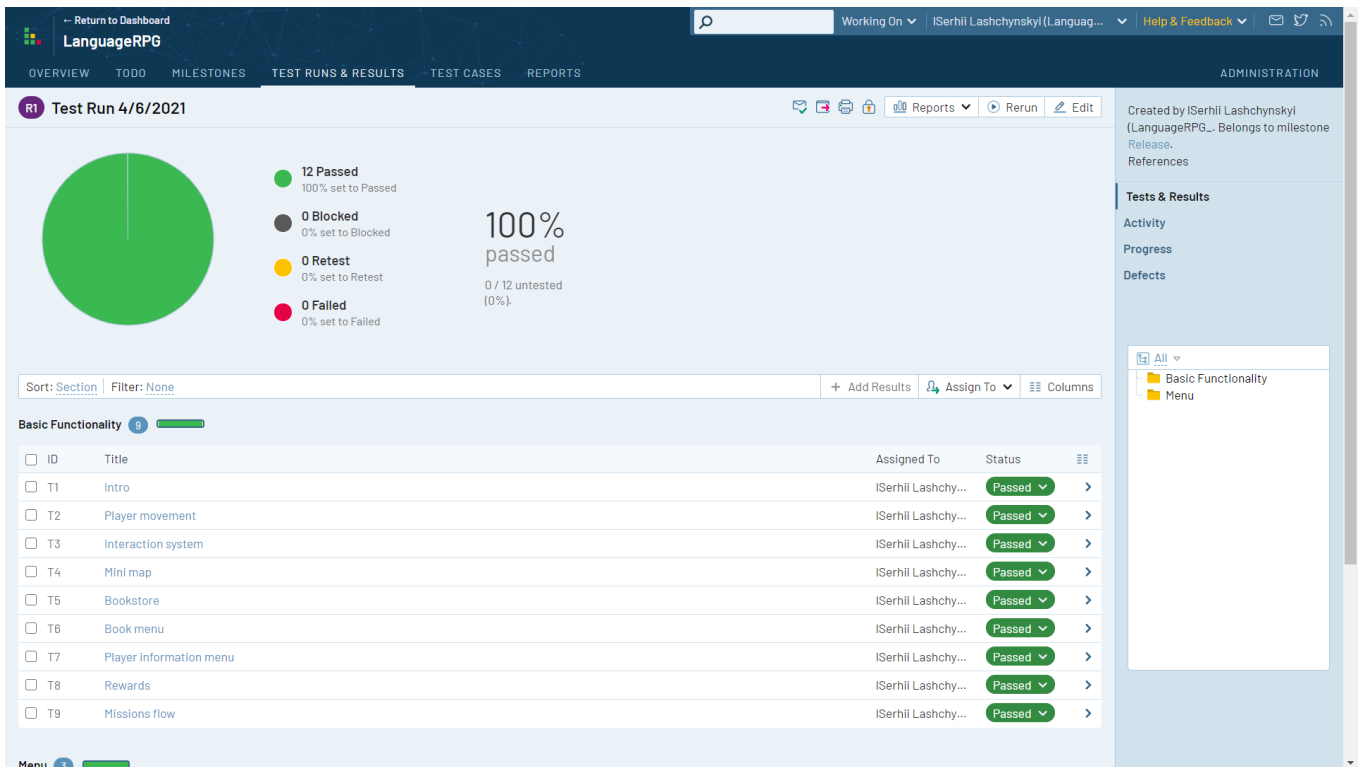


Рисунок 3.33 – Результати останнього тестування

Висновки до розділу

У розділі було проведено описання процесу розробки та проаналізовано розроблений проект. Після створення робочого білда проекту було проведено тестування навчальної гри «LanguageRPG». Результати останнього тестування показали, що в під час проведеного тестування не було знайдено критичних помилок і проект готовий для використання.

ВИСНОВКИ

Метою написання дипломного проекту була розробка навчальної гри для допомоги людям, які хочуть стати програмістами або лише тільки встали на цей шлях. Створення цієї гри надало змогу детальніше розглянути роботу навчальних елементів, логіку їх роботи та також стало міцним високотехнологічним фундаментом для подальшого розвитку цього жанру і для побудови схожих за тематикою систем. Соціально-практична значущість дипломної роботи полягає в тому, що дана гра надає можливість зрозуміти прості базові принципи та поняття програмування. Також гра допомагає розвинути такі обов'язково потрібні у програмуванні навички, як увага, терпіння та зосередженість.

Розробка даного проекту дозволила побачити важливість та ефективність ігрофікації навчального процесу. Навчальна гра допомагає розвинути навички у декількох напрямках у сфері програмування, зрозуміти базові ази таких мов як JavaScript, Java, C#, PHP та HTML. Навчальна система дозволяє розвивати не тільки практичні навички, а й такі здібності як увага та витримка.

У першому розділі дипломної роботи було проведено дослідження питання актуальності навчальних ігор в сучасній системі навчання, зроблено теоретичний аналіз існуючих методів ігрофікації, та розібрано систему класифікації навчальних ігор, поділеної на 3 різні категорії, таких як категорія навчання, домінування грального елемента чи навчального та модель предметної області. Визначено актуальність та значущість ігрових систем на навчальних заняттях. Виходячи з отриманих результатів, було отримано вимоги до створення додатку. В кінці першого розділу проводиться порівняння існуючих аналогів.

У другому розділі було проведено аналіз використаних для реалізації проекту технологій та зроблено прототипування ігрових процесів та елементів програми у вигляді графіків та схем.

У третьому розділі було описано реалізацію проекту за допомогою середовища розробки Unity3D та побудову десктопного додатку. Було проведено аналіз проробленої роботи та якості продукту. Відбулося тестування остаточної версії

програмного продукту за допомогою Web-сервісів Jira та TestRail та наведено приклади процесу тестування навчальної гри «LanguageRPG».

Отже, в результаті розробки була отримана навчальна система з елементами гри, структура якої дозволяє їй мати можливість бути нескінченно модифікованою новими елементами та міні-іграми, працювати та надавати нові знання користувачеві без з'єднання з мережею інтернет.

На тему дипломної роботи було опубліковано 2 тезисів на наукових конференціях. Тема першого тезису – «Роль ігор у навчальному процесі». У першому тезисі була розглянута проблема ігрофікації навчального процесу та її плюси. Тема другого тезису – «Актуальність створення навчальної гри для вивчення програмування». У цьому тезисі розглядалося цілеспрямованість створення ігрового навчального додатку «LanguageRPG» та було досліджено його основні плюси.

Апробація результатів. Основні положення і результати бакалаврської роботи доповідались і обговорювались на науково-практичній конференціях:

1. Лашинський С.С. Роль ігор у навчальному процесі / Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в інфокомунікаційних технологія».
2. Лашинський С.С. Актуальність створення навчальної гри для вивчення програмування / Всеукраїнська науково-технічна конференція «Сучасний стан та перспективи розвитку ІОТ».

Перелік посилань

1. Игрофикация [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: <https://ru.wikipedia.org/wiki/Игрофикация>
2. Gamification in [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: <https://www.trueeducationpartnerships.com/schools/gamification-in-education/>
3. Як навчання на основі ігор розвиває навички 21-го століття [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: <https://nus.org.ua/view/yak-navchannya-na-osnovi-igor-rozvyva-navychky-21-go-stolittya/>
4. Геймификация в образовании: виды, компоненты, примеры [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: <https://vuz24.ru/news/fakty-i-sobytiya/gejmifikaciya-v-obrazovanii-vidy-komponenty-primery>
5. Види дидактичних ігор [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: <http://studentam.net.ua/content/view/3611/97/>
6. Colobot и Сееbot [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: <https://habr.com/ru/post/59708/>
7. Unity (рушій гри) [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: [https://uk.wikipedia.org/wiki/Unity_\(рушій_гри\)](https://uk.wikipedia.org/wiki/Unity_(рушій_гри))
8. Мультиплатформенность Unity [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: <https://unity.com/ru/features/multiplatform>
9. Myths and Facts of the Unity Game Engine [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: <https://blog.theknightsofunity.com/myths-and-facts-of-unity-game-engine/>
10. List of video game genres [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: https://en.wikipedia.org/wiki/List_of_video_game_genres
11. Blender [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: <https://uk.wikipedia.org/wiki/Blender>

12. MindMeister [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: <https://uk.wikipedia.org/wiki/MindMeister>

13. Приемы при проектировании архитектуры игр [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: <https://habr.com/ru/post/255561/>

14. Что такое ассеты Unity, где их скачать и как добавить [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: https://skillbox.ru/media/gamedev/что_такое_asset_unity/

15. Software quality control [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: https://en.wikipedia.org/wiki/Software_quality_control

16. Jira [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: [https://en.wikipedia.org/wiki/Jira_\(software\)](https://en.wikipedia.org/wiki/Jira_(software))

17. TestRail – Introduction [Електронний ресурс]: [Веб-сайт]. – Електронні дані. – Режим доступу: https://www.tutorialspoint.com/testrail/testrail_introduction.html

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка навчальної гри «LanguageRPG» за допомогою Unity3D

Виконав студент 4 курсу
групи ПД-41
Лацинський Сергій Сергійович
Керівник роботи
Трінтіна Наталія Альбертівна

Київ – 2020

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

- **Мета роботи** – розробити сучасну навчальну гру, здібну покращувати навички користувача у декількох напрямках та для полегшення навчального процесу у сфері програмування.
- **Об'єкт дослідження** – ігрофікація навчальної системи на основі навчальної гри.
- **Предмет дослідження** – ігровий додаток для ігрофікації навчальної системи на основі декількох міні-ігор з відкритим світом.

2

Аналіз аналогів

Переваги розробленої навчальної системи:

Відсутність потреби у постійному підключенні до мережі інтернет;
Різноманітність ігрового світу;
Кількість доступних знань;
Реіграбельність.

Недоліки розробленої навчальної системи:

Низький рівень складності завдань;
Відсутність можливості писати код у грі.

3

ТЕХНІЧНІ ЗАВДАННЯ

Навчальна система на основі ігрового додатку розробленого за допомогою середи розробки Unity3D.

- Графічний функціонал
- Звуковий функціонал
- Внутрішньоігровий функціонал
- Інтерфейс користувача
- Система нагород

4

Спеціалізовані технології

Ігровий Рушій – Unity3D

3D редактор – Blender

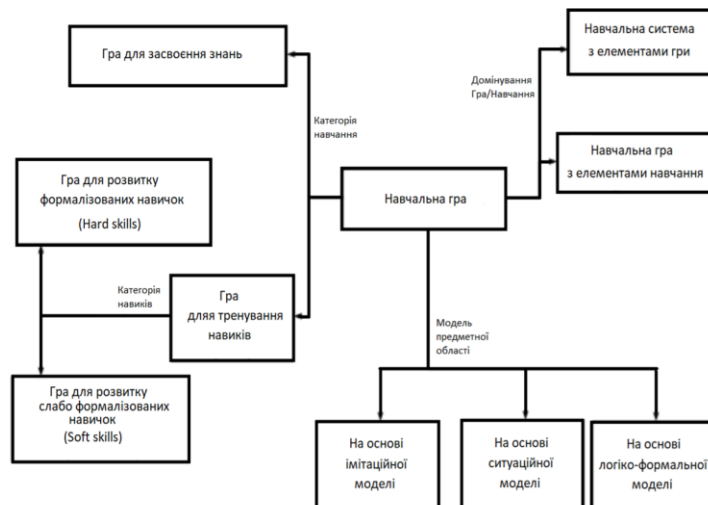
Мова реалізації – C#

Редактор коду – Microsoft Visual Studio



5

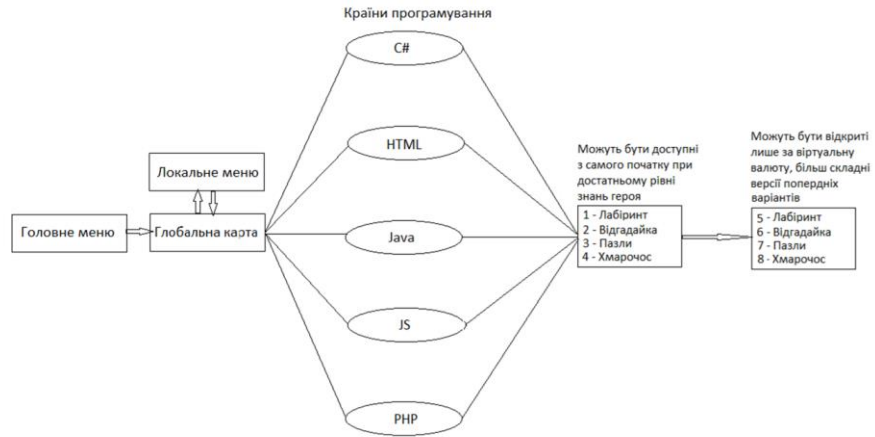
Класифікація навчальних ігор



6

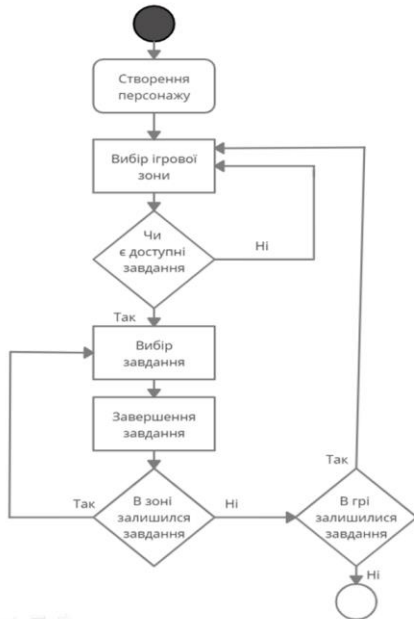
Прототипи

Система рівнів

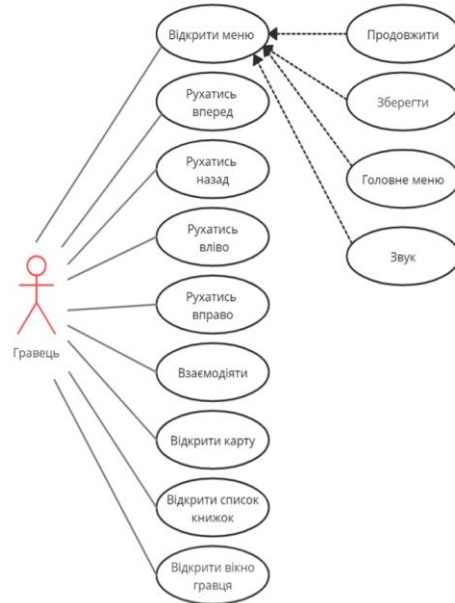


7

Діаграма активності гри

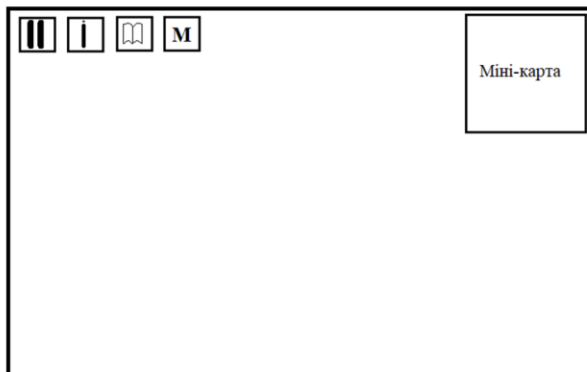


Діаграма варіантів використання



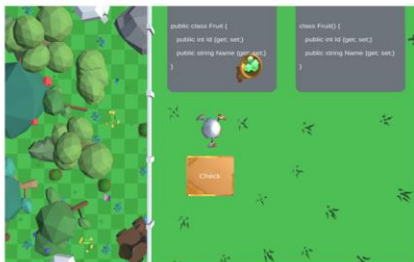
8

Проектування інтерфейсу користувача



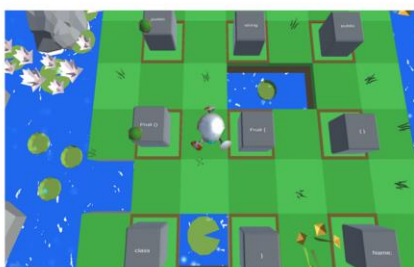
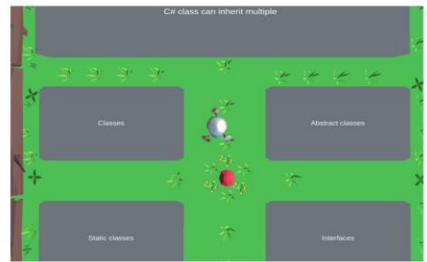
9

Практичне використання



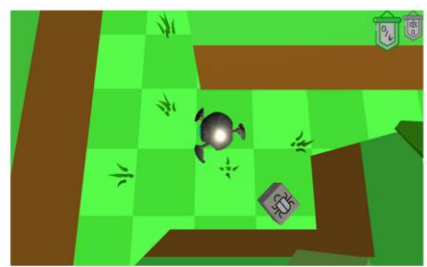
Відгадайка

Хмарочос



Пазли

Лабіринт



10

Наукова новизна та практична значимість

Новизна дослідження полягає в реалізації новітньої навчальної 3D гри для допомоги у вивченні програмування та перевірки знань за допомогою безкоштовної середовища розробки Unity3D.

Практична значимість дослідження полягає в реалізації навчальної гри, здібної частково замінити стандартну систему навчання, збільшуючи інтерес учнів до навчання.



АПРОБАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

- Лашинський С.С. Роль ігор у навчальному процесі / Всеукраїнська науково-технічна конференція «Застосування програмного забезпечення в інфокомунікаційних технологія» : Розробка Ігор. Збірник тез. 12.02.2021, ДУТ, м Київ – С. 33.
- Лашинський С.С. Актуальність створення навчальної гри для вивчення програмування / Всеукраїнська науково-технічна конференція «Сучасний стан та перспективи розвитку ІОТ» : Сучасні інформаційні технології в Україні і світі. Збірник тез. 09.04.2021, ДУТ, м Київ – С. 113.

ВИСНОВКИ

- Розглянуто основні принципи навчальних ігор
- Розглянуто та проаналізовано важливість ігрофікації навчального процесу
- Досліджено класифікацію навчальних ігор
- Було зроблено прототипування ігрового додатку
- На основі прототипу була розроблена навчальна гра «LanguageRPG» на основі декількох міні-ігор у відкритому світі за допомогою Unity3D
- Був проведений аналіз якості та тестування готового продукту на помилки

ДЯКУЮ ЗА УВАГУ!