

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА
Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці**

ОСВІТНЬО ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

Форма навчання: денна

КВАЛІФІКАЦІЙНИЙ БАКАЛАВРСЬКИЙ ПРОЄКТ

на тему

«НАВЧАЛЬНА ІГРОВА СИСТЕМА ДЛЯ РОЗВИТКУ ДІТЕЙ»

здобувача Липського Микити Анатолійовича _____

Науковий керівник:

д. е. н., професор

Мозгаллі О.П.

**Робота допущена до захисту перед
екзаменаційною комісією з атестації
здобувачів вищої освіти**

завідувач кафедри:

к.е.н., доцент.

Тішков Б.О.

Київ 2024

Міністерство освіти і науки України
Київський національний економічний університет імені Вадима Гетьмана
Навчально-науковий інститут «Інститут інформаційних технологій в економіці»
Кафедра інформаційних систем в економіці

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «КОМП'ЮТЕРНІ НАУКИ»

галузь знань 12 «Інформаційні технології»

спеціальність 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ:

Завідувач кафедри

Тішков Б.О.

“ ” 2024 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

здобувача вищої освіти *Липського Микити Анатолійовича*

очної (денної) форми навчання

на підготовку кваліфікаційного бакалаврського проєкту
на тему: «Навчальна ігрова система для розвитку дітей»

Тему затверджено наказом ректора Університету від «11» березня 2024 р. № 529-ст.

Кваліфікаційний бакалаврський проєкт виконується на матеріалах
відкритих інтернет-джерел та інших публікацій

План кваліфікаційного бакалаврського проєкту

Розділ I. Характеристика та аналіз предметної галузі

Розділ II. Розробка вимог і моделювання навчальної ігрової системи

Розділ III. Проектування та реалізація компонентів навчальної ігрової системи

Об'єкт дослідження – навчальна ігрова система для розвитку дітей

Предмет дослідження – сукупність теоретичних, методичних і практичних підходів до технології створення навчальної ігрової системи для розвитку дітей.

Мета кваліфікаційного бакалаврського проєкту – розроблення навчальної ігрової системи для розвитку дітей, яка дозволить підвищити ефективність і мотивацію навчання дітей шляхом використання інтерактивних ігор, які роблять вивчення наук цікавим та захоплюючим.

**Конкретні завдання, які студент повинен виконати для
досягнення поставленої мети:**

У розділі I необхідно дати визначення і характеристику предметної галузі, навчальних ігор, особливу увагу слід приділити об'єктам та факторам, що впливають на навчання дітей, а також покращення сприйняття ними інформації через такі інтерактивні засоби.

Провести аналіз літературних джерел з метою визначення прогресивних технологій розв'язання задач, поставлених у кваліфікаційному бакалаврському проєкті.

У розділі II На основі проведеного аналізу необхідно розробити вимоги до інформаційної системи, які будуть коректними та актуальними з технологічної точки зору. Результат розробки вимог слід подати у формі діаграм і проєктної специфікації. Дотримуючись вимог стандартів виконати постановку та алгоритм розв'язання задачі створення навчальної гри для розвитку дітей. Розробити модель інформаційної системи відповідно до вимог, викладених в методичних вказівках кафедри.

У розділі III виконати проєктування таких компонентів системи: інформаційного, технічного та програмного забезпечення, а також продемонструвати результати виконання програмних модулів, зазначити можливість і доцільність їх впровадження на практиці, а також вказати особистий внесок у розроблення навчальної ігрової системи.

Завдання підготував

науковий керівник _____

Мозгаллі Ольга Петрівна

“11”__03__ 2024 р.

Завдання одержав

студент _____

Липський Микита Анатолійович

“11”__03__ 2024 р.

Відгук
про кваліфікаційний бакалаврський проєкт
здобувача навчально-наукового інституту
«Інститут інформаційних технологій в економіці»
освітньо-професійної програми
«Комп'ютерні науки»

Липського Микити Анатолійовича
на тему
«Навчальна ігрова система для розвитку дітей»

1. Актуальність теми. Тема кваліфікаційного бакалаврського проєкту є актуальною, адже в сучасному світі все більше уваги приділяється використанню ігрових методів навчання та розвитку дітей. Навчальні ігри можуть зробити процес навчання більш цікавим та захоплюючим, а також допомогти дітям краще засвоїти знання та навички.

2. Позитивні риси кваліфікаційного бакалаврського проєкту. В роботі було системно досліджено предметну область, зокрема принципи ігрового навчання дітей, архітектуру системи, методи та засоби розроблення навчальних ігрових систем. Автором було застосовано актуальні методи аналізу та синтезу інформації, а також програмного забезпечення для розроблення та тестування навчальної ігрової системи. Більш того, автор не лише розробив теоретичну модель навчальної ігрової системи, але й реалізував її на практиці, що дозволить покращити й урізноманітнити навчання дітей.

3. Наявність самостійних розробок автора: Кваліфікаційний бакалаврський проєкт виконаний автором самостійно із застосуванням сучасного інструментарію. Автор самостійно дослідив методи та засоби розроблення навчальних ігрових систем та принципи організації ігрового навчання; також ним було самостійно обрано програмні та технічні засоби й розроблено програмне забезпечення для реалізації навчальної ігрової системи. Крім того, автором було проведено первинне тестування навчальної ігрової системи та усунено виявлені помилки.

4. Цінність теоретичних висновків та практичних рекомендацій: Теоретичні висновки роботи можуть бути використані для розробки нових підходів до організації навчання дітей та підлітків. Розроблена навчальна ігрова система має потенціал для комерційного використання й забезпечення підвищення ефективності навчання дітей.

5. Наявність недоліків: Автор недостатньо детально описав психолого-педагогічні аспекти розробки ігрової системи, що може ускладнити її використання педагогами. Також автор не надав рекомендацій щодо комерціалізації навчальної ігрової системи, що може бути корисно для її

ширшого впровадження. Представлене проєктне рішення потребує подальшого вдосконалення й розвитку для навчання дітей.

6. Загальна оцінка кваліфікаційного бакалаврського проєкту та його допущення до захисту перед ЕК: Загалом, представлений кваліфікаційний бакалаврський проєкт відповідає встановленим вимогам методичних вказівок щодо структури, обсягу та змісту, та рекомендується до захисту з **позитивною оцінкою**, а Липському Дмитру Анатолійовичу може бути присвоєний освітньо-кваліфікаційний рівень «бакалавр» галузі знань 12 «Інформаційні технології» за спеціальністю 122 «Комп'ютерні науки».

Науковий керівник

(підпис)

д. е. н., Мозгаллі О.П.

“10” червня 2024 р.

Рецензія

на кваліфікаційний бакалаврський проєкт

Липського Микити Анатолійовича

на тему: «Навчальна ігрова система для розвитку дітей»

Актуальність теми кваліфікаційної бакалаврської роботи і доцільність її розроблення полягає у потребі удосконалення освітнього процесу за допомогою інтерактивної навчальної ігрової системи для розвитку дітей. У зв'язку із стрімким розвитком сучасних технологій та необхідністю постійного покращення освітнього процесу, створення таких систем є логічним розвитком подій.

Якість проведеного дослідження характеризується високим рівнем теоретичного обґрунтування та практичної реалізації. Автор даної роботи провів ретельний аналіз предметної області та існуючих програмних засобів, забезпечив ґрунтовну базу для розробки додатку, використав сучасні технології та підходи у проєктуванні та впровадженні ігрової системи.

Позитивні риси кваліфікаційного бакалаврського проєкту полягають у комплексному підході до реалізації навчальної ігрової системи, що є не типовим для подібного роду застосунків. Зазвичай ігри несуть суто розважальний характер, хоч сьогодні це вже і змінюється по трохи. Реалізація ігрового застосунку, що включає в себе методику інтерактивного ігрового навчання, значно полегшує процес подання матеріалу дітям. Також це зручний інструмент для вчителів, оскільки в перспективі цей застосунок може стати ледь не заміником класичним книгам і спрощує роботу, в області пояснення матеріал дітям, для вчителів. Окрім цього застосунок не потребує підключення до інтернет мережі, що унеможливорює витік даних і спрощує процес користування, оскільки не всі школи можуть бути оснащені інтернет з'єднанням.

Зауваження до кваліфікаційної бакалаврської роботи стосуються деяких аспектів її реалізації та перспектив розвитку. Незважаючи на загальну високу якість, робота є далекою від ідеалу, або впровадження її в закладах освіти, потрібно провести ще багато роботи над матеріал, що подається, а також над наявністю функціоналу для вчителів.

Практична значущість висновків і рекомендацій полягає у демонстрації перспективи подібних застосунків, а також демонстрації невичерпних можливостях ігор, в тому числі і в сфері освіти. Подібні проєкти в майбутньому спростять і покращать процес навчання. Вважаю, що робота виконана на високому рівні, має теоретичну та практичну значущість й заслуговує на оцінку «Відмінно».

Місце роботи та посада рецензента

(підпис, ПІБ)

Місце печатки організації, де працює рецензент

АНОТАЦІЯ

Кваліфікаційного бакалаврського проєкту студента 4 курсу

Навчально-наукового інституту

«Інститут інформаційних технологій в економіці»

Липського Микити Анатолійовича, виконаного на тему:

«Навчальна ігрова система для розвитку дітей»

Київ: кафедра інформаційних систем в економіці, 2024 р.

Кваліфікаційний бакалаврський проєкт присвячений створенню навчальної ігрової системи для розвитку дітей. Під час аналізу освітньої програми в школі, було виявлено, що школярів, в більшості своєму, не намагаються зацікавити у вивченні нових предметів. Особливо цей момент помітний, коли з'являються такі серйозні і нові, для дітей, науки як : біологія, хімія, фізика та ін. Це є проблемою, оскільки, як на мене, перехід від природознавства до таких важких предметів є занадто різким. Через дану проблему діти не стільки зацікавлені в предметі, скільки мають просто зрозуміти його за будь що, аби отримати задовільну оцінку. Цей проєкт, має на меті створити навчальну ігрову систему, яка б в ігровій формі могла зацікавити дитину в подальшому вивченні певної науки, шляхом проходження відповідних рівнів. В даному вигляді проєкт має продемонструвати потенціал подібних продуктів в навчальній сфері.

Кваліфікаційний бакалаврський проєкт складається з трьох розділів, пов'язаних між собою.

Розділ 1 розпочинається з характеристики предметної галузі та об'єкта дослідження. Далі проводиться аналіз завдань, які вирішуються в даній галузі. На завершення розділу наводиться перелік існуючих програмних та необхідних технічних засобів, що використовуються в цій предметній галузі.

У другому розділі описується процес проєктування системи, включаючи обґрунтування методу проєктування, розробку архітектури системи, формулювання завдань та розробку алгоритмів їх вирішення.

У третьому міститься інформаційне, технічне програмне, організаційне забезпечення та перелік комп'ютерних комплектуючих, необхідних для використання навчальної ігрової системи для розвитку дітей. Обґрунтований комплекс технічних засобів, а також програмне забезпечення, що використовується для створення системи. Вказані структури інформаційних масивів, які використовуються під час вирішення задачі

У висновках надаються рекомендації щодо доцільності розроблення та впровадження навчальної ігрової системи для розвитку дітей.

РЕФЕРАТ

Кваліфікаційний бакалаврський проєкт містить 86 сторінок, 2 таблиці, 39 рисунків, список використаних джерел з 26 найменувань, додатки.

«Навчальна ігрова система для розвитку дітей»

Об'єктом дослідження кваліфікаційного бакалаврського проєкту є навчальна ігрова система для розвитку дітей.

Предметом дослідження є сукупність теоретичних, методичних і практичних підходів до технології створення навчальної ігрової системи для розвитку дітей.

Мета і завдання дослідження. Основною метою кваліфікаційного бакалаврського проєкту є розроблення навчальної ігрової системи для розвитку дітей, яка дозволить підвищити ефективність і мотивацію навчання дітей шляхом використання інтерактивних ігор, які роблять вивчення наук цікавим та захоплюючим.

Відповідно до поставленої мети визначені такі завдання:

- дослідити предметну область,
- виконати аналіз існуючих систем пошуку інформації;
- визначити вимоги до системи: бізнес-вимоги, функціональні, нефункціональні
- розробити постановку задачі, описати її характеристику, вхідну, вихідну інформацію, спроектувати інформаційну модель задачі;
- розробити алгоритм функціонування системи,
- здійснити моделювання поведінки та структури системи.
- запроектувати інформаційне, технічне та програмне забезпечення.
- реалізувати запропоновані рішення у вигляді навчальної ігрової системи для розвитку дітей.

Теоретична, методична та практична значущість отриманих результатів.

Під час дослідження було проаналізовано предметну область, проаналізовано існуючі продукти схожого напрямлення, визначено перспективність навчальних ігрових платформ в освітньому процесі. Практичні результати дослідження полягають у створенні системи, яка покращить освітній процес та підвищить ефективність подачі матеріалу дітям, за допомогою інтерактивних засобів, та моментальної віддачі від отриманих знань, а також спростить процес сприйняття інформації завдяки візуальній складовій.

Рік виконання кваліфікаційного бакалаврського проєкту – 2024.

Рік захисту роботи – 2024.

Ключові слова: навчальна ігрова система, гра, користувач, гравець, ігровий

об'єкт

ЗМІСТ:

АНОТАЦІЯ.....	3
РЕФЕРАТ.....	9
ВСТУП.....	12
РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ	15
1.1. Характеристика предметної галузі та об'єкта дослідження	15
Перспективи навчальних ігрових систем.....	16
1.2. Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі.....	18
CodeCombat.....	18
Scratch	19
Популярність Scratch та CodeCombat.....	21
Перспективи розвитку та популярності навчальних комп'ютерних ігор	21
РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ НАВЧАЛЬНОЇ ІГРОВОЇ СИСТЕМИ.....	23
2.1. Аналіз і специфікація вимог до навчальної ігрової системи	23
Бізнес-вимоги до системи	23
Функціональні вимоги до системи	23
Нефункціональні вимоги до системи	25
2.2 Характеристика задачі.....	27
Вхідна інформація	28
Вихідна інформація	29
Функціонування системи	29
2.3 Моделювання поведінки системи	32
РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ НАВЧАЛЬНОЇ ІГРОВОЇ СИСТЕМИ	36

3.1. Інформаційне забезпечення.....	36
Процес інсталяції Unity.....	37
Процес інсталяції Visual Studio	39
Структура і компоненти рушія Unity	43
3.2 Технічне забезпечення	49
3.3 Програмне забезпечення	50
Системне програмне забезпечення.....	50
3.4 Результати реалізації навчальної ігрової системи	50
ВИСНОВКИ	54
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	56
ДОДАТКИ	59

ВСТУП

Сучасний світ перебуває в епоху стрімкого розвитку технологій, де комп'ютери відіграють ключову роль у всіх сферах життя. Одним із важливих напрямків їх використання є комп'ютерна навчальна гра. Це захоплююча та інноваційна форма освіти, яка здатна зробити процес навчання цікавішим та ефективнішим.

Комп'ютерні навчальні ігри стають все більш популярними серед вчителів, батьків та учнів, оскільки вони поєднують у собі елементи розваги та навчання. Ця форма освіти відкриває безмежні можливості для того, щоб залучити учнів до процесу вивчення різних предметів та навичок.

Комп'ютерні ігри сприяють розвитку креативності, логічного мислення та комунікативних навичок учнів. Вони дозволяють індивідуалізувати процес навчання, враховуючи потреби кожного учня, і надають можливість вчителю стежити за прогресом кожного учасника.

У наш час важко уявити сферу освіти без використання сучасних технологій, і комп'ютерні навчальні ігри стають не лише доповненням до традиційного навчання, але і ключовим інструментом у формуванні знань та вмінь нового покоління. У цьому контексті важливо розглядати як переваги, так і виклики, пов'язані з використанням комп'ютерних навчальних ігор, щоб максимально використовувати їхній потенціал для розвитку освітнього процесу.

Завдяки технологічним досягненням, комп'ютерні навчальні ігри можуть бути створені для різних рівнів складності та вікових груп, враховуючи індивідуальні особливості учнів. Вони сприяють активній участі та поглибленню знань через відтворення реальних ситуацій та завдань. Такий підхід дозволяє учням отримати практичні навички, які можна застосовувати в реальному житті.

Об'єктом дослідження є навчальна ігрова система для розвитку дітей. Предметом дослідження є сукупність теоретичних, методичних і практичних підходів до технології створення навчальної ігрової системи для розвитку дітей.

Основною метою кваліфікаційного бакалаврського проекту є розроблення навчальної ігрової системи для розвитку дітей, яка дозволить підвищити

ефективність і мотивацію навчання дітей шляхом використання інтерактивних ігор, які роблять вивчення наук цікавим та захоплюючим. Дана мета буде досягатись за допомогою реалізації певних завдань. Головними завданнями є розробити застосунок, який буде мати інтерактивні елементи, такі як пересування і взаємодія з ігровими об'єктами, а також з елементами невеликого сюжету, на який і покладено викладання певного навчального матеріалу.

Однією з головних переваг комп'ютерних навчальних ігор є їхня здатність створювати мотивацію для вивчення. Граючи, учні не лише засвоюють нові знання, але і розвивають творче мислення та вміння розв'язувати завдання. Це сприяє формуванню позитивного ставлення до навчання та стимулює самостійність.

З огляду на успіх комп'ютерних навчальних ігор і їхню важливу роль у сучасній освіті, очікується, що вони будуть і далі еволюціонувати і вдосконалюватися. Можливо, майбутнє комп'ютерних навчальних ігор полягатиме в більшій інтеграції з віртуальною реальністю та розширеною реальністю. Це відкриє нові можливості для імерсивного навчання, де учні зможуть взаємодіяти з віртуальними об'єктами та оточенням, що максимально наблизатиметься до реального життя.

Також, інтеграція штучного інтелекту може робити навчальні ігри більш адаптивними та індивідуалізованими, адже система може аналізувати успішність кожного учня та автоматично пристосовувати рівень складності завдань до їхніх потреб та можливостей.

Крім того, зростаюча доступність до мобільних пристроїв може сприяти більшій гнучкості у використанні комп'ютерних навчальних ігор, дозволяючи учням навчатися в будь-якому місці та часі, навіть поза межами класної кімнати.

З урахуванням цих тенденцій, можна стверджувати, що майбутнє комп'ютерних навчальних ігор буде спрямоване на ще більшу персоналізацію, інтеграцію новітніх технологій та постійне покращення якості освіти через цифрові засоби.

Ще одним можливим напрямком розвитку комп'ютерних навчальних ігор є їхнє більше активне використання для навчання практичних навичок і вмінь. Наприклад, ігри можуть бути спрямовані на вивчення конкретних професійних навичок, таких як програмування, дизайн, інженерія тощо. Це може стати особливо корисним у вищій освіті та професійній підготовці, де практичні навички мають вирішальне значення.

Також можливим є розвиток ігор, що сприяють розвитку міжкультурної та міжнаціональної співпраці та розуміння. Це може включати в себе ігри, що моделюють ситуації міжнародного співробітництва або взаємодії між різними культурами, щоб учні могли краще розуміти світ навколо себе та розвивати навички міжкультурного спілкування.

Загалом, майбутнє комп'ютерних навчальних ігор обіцяє бути різноманітним і захоплюючим, сприяючи не лише освіті, а й розвитку цифрової грамотності, творчого мислення та соціальних навичок учнів у всьому світі

РОЗДІЛ 1. ХАРАКТЕРИСТИКА ТА АНАЛІЗ ПРЕДМЕТНОЇ ГАЛУЗІ

1.1. Характеристика предметної галузі та об'єкта дослідження

Предметною областю даної роботи є навчальні комп'ютерні ігрові системи. В сучасних реаліях розвиток шкільної освіти з використанням ігор як платформи для навчання є закономірним перебігом подій і актуальним питанням. Ігри можуть стимулювати інтерес до навчання, покращувати мотивацію, розвивати креативність та критичне мислення.

Навчальні ігрові системи для розвитку дітей можна розділити на декілька категорій.

➤ За тематикою:

- **Наука.** Ці ігри знайомлять дітей з різними науковими концепціями, такими як фізика, хімія, біологія.
- **Історія.** Ці ігри розповідають про історичні події та дозволяють дітям подорожувати в часі.
- **Мистецтво та творчість.** Такі ігри розвивають творчі навички дітей, дозволяючи їм малювати, музикувати, писати історії тощо.
- **Мови.** Подібні ігри допомагають дітям вивчити нові мови або покращити свої знання вже вивчених.
- **Математика.** Ігри на цю тематику роблять математику більш цікавою та допомагають дітям розвинути математичні навички.

➤ За педагогічними методами:

- **Адаптивні** – складність гри автоматично підлаштовується під рівень знань і вмінь дитини.
- **Ігрифікація** – використовуються бали, нагороди, рейтинги та інші ігрові механіки для мотивації дітей.
- **Інтерактивність** – діти активно беруть участь у навчальному процесі, відповідаючи на запитання, вирішуючи завдання, роблячи вибір.

- **Персоналізація** – діти можуть налаштовувати ігрове середовище, створювати власних персонажів тощо.
- **Симуляція** – діти моделюють реальні процеси, що допомагає їм краще зрозуміти матеріал.
- **Візуалізація** – використовуються графіка, анімація, звук та відео для кращого засвоєння матеріалу.

Ігри, спрямовані на навчання, мають чітку мету, яка пов'язана з освітніми завданнями або розвитком конкретних навичок. Ці ігри захоплюють та мотивують дітей завдяки цікавому й динамічному процесу. Діти активно беруть участь у навчанні, отримуючи зворотний зв'язок та відчуваючи прогрес. Ігри вільні від насильства, дискримінації та інших шкідливих елементів.

Навчальні ігри роблять процес навчання приємним та ефективним, сприяючи кращому засвоєнню матеріалу. Ігри допомагають розвивати різноманітні навички, такі як мислення, творчість, спілкування та командна робота. Ігри роблять навчання цікавим та захоплюючим, мотивуючи дітей до подальшого пізнання.

Перспективи навчальних ігрових систем

Зростання доступності технологій, таких як смартфони, планшети та комп'ютери, робить доступ до навчальних ігор більш зручним. Розширення доступу до Інтернету дозволяє дітям грати в онлайн-ігри, які пропонують широкий спектр навчальних можливостей. Здешевлення цифрових пристроїв робить їх доступними для сімей з різним рівнем доходу.

Дослідження в галузі педагогічної психології та нейронауки доводять, що навчальні ігри можуть бути ефективним інструментом для навчання. Ігри допомагають дітям краще засвоювати інформацію, розвивати пам'ять, увагу, логічне мислення та інші когнітивні навички. Ігри мотивують дітей до навчання, роблячи його більш цікавим та захоплюючим.

Батьки та педагоги все більше визнають, що ігри не лише розважають, але й можуть бути цінним інструментом для навчання. Ігри сприймаються як інтерактивний та захоплюючий спосіб знайомства дітей з різними темами та

предметами. Зростає розуміння того, що ігри можуть розвивати важливі життєві навички, такі як співпраця, комунікація та вирішення проблем.

Ігрові технології постійно розвиваються, роблячи навчальні ігри більш інтерактивними, захоплюючими та реалістичними. Завдяки віртуальній та доповненій реальності ігри стають більш іммерсивними, що дозволяє дітям зануритися в процес навчання. Використання штучного інтелекту дає можливість персоналізувати ігри, адаптуючи їх до індивідуальних потреб та темпу навчання кожної дитини.

Навчальні ігри стають більш персоналізованими, що робить їх більш ефективними для кожної дитини. Алгоритми підбирають контент, рівень складності та темп гри відповідно до індивідуальних потреб та можливостей дитини. Це робить процес навчання більш захоплюючим та мотивує дітей до подальшого пізнання.

Гаміфікація – це використання ігрових елементів в неігрових контекстах, що робить навчання більш цікавим. Наприклад, у навчальні ігри можна додати бали, значки, рейтинги та інші ігрові механізми, які мотивують дітей досягати успіхів. Гаміфікація може допомогти дітям краще засвоїти інформацію, розвинути самодисципліну та наполегливість.

Зростання популярності онлайн-навчання призводить до збільшення попиту на навчальні ігри. Онлайн-платформи пропонують широкий спектр навчальних ігор, які можна використовувати як самостійно, так і в рамках онлайн-курсів. Ігри роблять онлайн-навчання більш інтерактивним та захоплюючим, що робить його більш ефективним для дітей.

Навчальні ігри стають все більш поширеним та ефективним інструментом для навчання завдяки поєднанню факторів, таких як зростання доступності технологій, підтримка досліджень, зміна ставлення до ігор, розвиток ігрових технологій, персоналізація, гаміфікація та зростання популярності онлайн-навчання. Важливо підбирати навчальні ігри відповідно до віку, інтересів та рівня розвитку дитини, а також звертати увагу на якість та чітку мету навчання.

1.2. Аналіз літературних джерел та практичного досвіду використання ІС і технологій в предметній галузі

Прикладів навчальних ігрових систем для розвитку дітей створено достатньо, але в даній роботі, для прикладу будуть розглянуті всього декілька популярних, а також розібрано їх педагогічний аспект і причину успіху.

CodeCombat

Розпочати розбір подібних успішних проєктів можна з CodeCombat[1], ось що про цей застосунок кажуть в інтернеті CodeCombat[2] – це навчальна відеогра для вивчення концепцій та мов програмування (див. рис. 1.1). Ця гра рекомендована для учнів віком 9-16 років. Учні вчаться набирати тексти на таких мовах програмування, як JavaScript, Python, HTML та CoffeeScript, а також вивчають основи комп'ютерних наук. CodeCombat складається з 11 розділів - трьох розділів з розробки ігор, двох розділів з веб-розробки та шести розділів з інформатики. Перший блок, "Інформатика 1", є безкоштовним для всіх учнів та вчителів.

В цьому проєкті користувачі грають у віртуальний світ, де вони розв'язують завдання та беруть участь у боях, використовуючи код. Завдання можуть бути адаптовані для різних рівнів вивчення, від початкового до просунутого. Є можливість взаємодії та змагання з іншими гравцями в режимі реального часу. А найголовніше, що для вчителів надаються інструменти для керування класом та відстеження прогресу учнів.

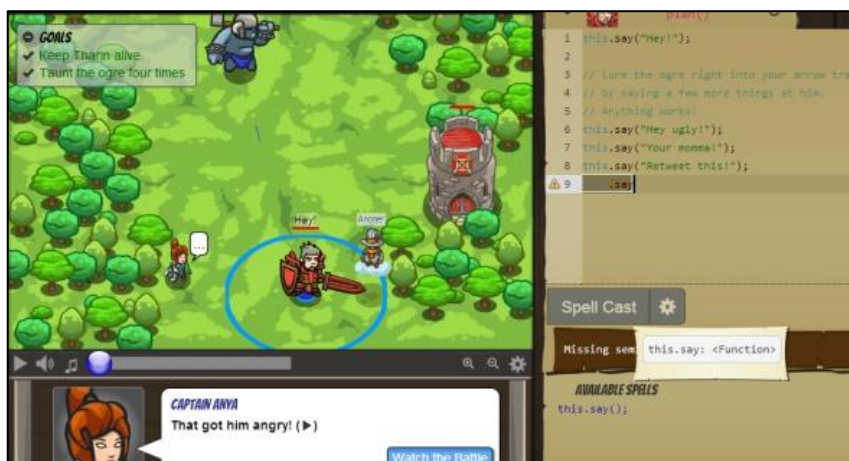


Рисунок 1.1 – CodeCombat

Педагогічний Аспект

CodeCombat не лише надає практичний досвід програмування, але також розвиває логічне мислення та стратегічне планування через ігровий сценарій. Це робить навчання програмування цікавим та мотиваційним для учнів, особливо для тих, хто тільки починає свій шлях у цій сфері.

Причина успіху

Успіх даного проєкту зумовлений тим що він вирішує важливу проблему, а саме вивчення програмування дітьми. CodeCombat є гарним прикладом того як подати, здавалося б, важку тему під цікавим соусом і тим спонукати дітей до подальшого вивчення програмування. Завдяки простим та цікавим механікам представленим в проєкті, граючи, користувач навіть не помічає що навчається, але при цьому здобуває цінні знання та навички в сфері програмування.

Scratch

Іншим прикладом успішного проєкту в цій сфері буде Scratch[3] (див. рис. 1.2), який вже є в шкільній програмі України і викладається на уроках інформатики. Scratch[4] здобув широку популярність завдяки своєму унікальному підходу до навчання програмування, роблячи його доступним та захоплюючим для людей різного віку.

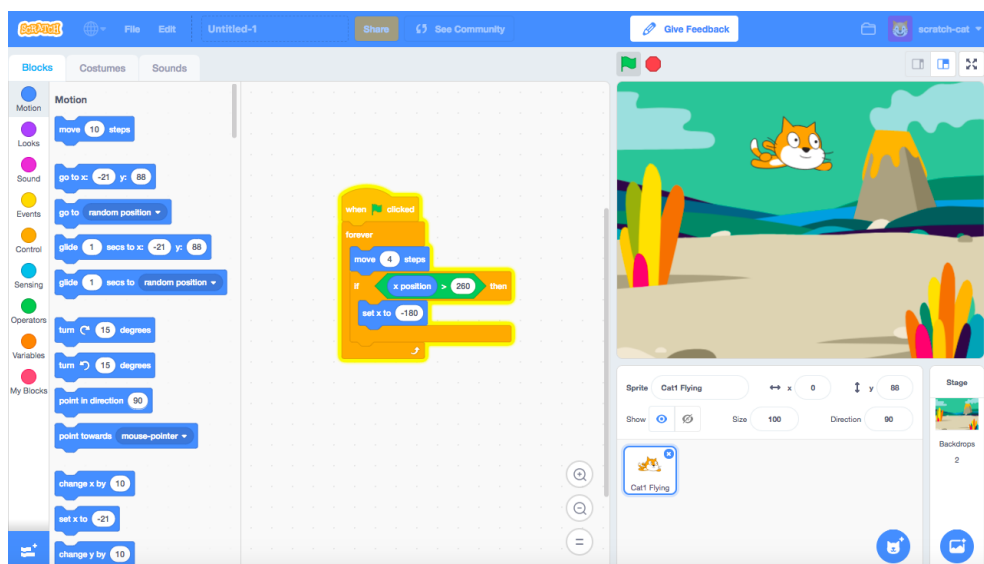


Рисунок 1.2 – Scratch

Педагогічний Аспект

Scratch використовується для навчання основам програмування та стимулює творчий підхід до розв'язання завдань. В освітньому контексті, Scratch допомагає:

- **Розвивати Логічне та Алгоритмічне Мислення.** Користувачі вчаться створювати послідовності операцій та взаємодіяти з алгоритмами.
- **Стимулювати Творчість.** Шляхом створення своїх проєктів користувачі розвивають творчий підхід та самовираження.
- **Підготовка до Програмування в Загальному.** Scratch служить вступом до програмування та може стати основою для подальших вивчення більш складних мов програмування.

Причини успіху

Scratch використовує **візуальні блоки коду**, які можна перетягувати та скидати, замість традиційного текстуального програмування. Це робить його надзвичайно доступним для початківців, адже не потребує знання мов програмування.

Scratch дозволяє користувачам не лише вивчати основи програмування, а й **створювати власні ігри**, анімації та інтерактивні історії. Цей аспект робить процес навчання цікавим та захоплюючим, а також дає можливість застосувати набуті знання на практиці.

Scratch має **велику та активну спільноту** користувачів, які готові допомогти одне одному з проєктами та поділитися своїми знаннями. Окрім цього, існує безліч онлайн-ресурсів, таких як посібники, уроки та відео, які допоможуть вам навчитися використовувати Scratch.

Завдяки цій комбінації доступності, захопливості та підтримки Scratch став чудовим інструментом для першого знайомства з програмуванням для дітей та дорослих.

Популярність Scratch та CodeCombat

Навчальні комп'ютерні ігри, такі як Scratch та CodeCombat, демонструють стрімке зростання популярності завдяки своїй здатності зробити навчання програмування цікавим та доступним.

Візуальні блоки коду, захоплюючий геймплей та можливість створення власних проєктів роблять ці платформи ідеальними інструментами для початківців.

Зростання популярності онлайн-спільнот та навчальних ресурсів сприяє доступності та ефективності навчання.

Очікується, що в найближчі роки тенденція до зростання популярності навчальних комп'ютерних ігор збережеться, адже вони пропонують інноваційний та захоплюючий підхід до освоєння навичок програмування.

Цей тренд може призвести до значних змін у сфері освіти, роблячи програмування доступнішим для людей різного віку та походження.

Перспективи розвитку та популярності навчальних комп'ютерних ігор

Навчальні комп'ютерні ігри, такі як Scratch та Code Combat, стають все більш популярними завдяки їхній здатності зробити навчання програмування цікавим та захоплюючим.

Доступність. Візуальні блоки коду, які можна перетягувати та скидати, роблять ці платформи доступними для людей з будь-яким рівнем досвіду, роблячи програмування більш інклюзивним.

Захоплюючий геймплей. Інтерактивний та захоплюючий геймплей мотивує користувачів продовжувати навчання, роблячи його більш ефективним.

Практичне застосування. Можливість створювати власні ігри, анімації та інтерактивні історії дозволяє користувачам застосовувати набуті знання на практиці, роблячи навчання більш змістовним.

Спільноти та ресурси. Зростання онлайн-спільнот та навчальних ресурсів сприяє доступності та ефективності навчання, надаючи користувачам підтримку та додаткові матеріали.

Інноваційний підхід. Навчальні комп'ютерні ігри пропонують інноваційний та захоплюючий підхід до освоєння навичок програмування, що може призвести до значних змін у сфері освіти.

Зміна освітньої парадигми. Навчальні комп'ютерні ігри можуть трансформувати традиційні методи навчання, роблячи їх більш інтерактивними, практичними та цікавими.

Підвищення навичок 21 століття. Навчання програмування за допомогою ігор може допомогти розвинути навички вирішення проблем, критичного мислення та співпраці, які є важливими для успіху в 21 столітті.

Навчальні комп'ютерні ігри мають потенціал[5] зіграти значну роль у формуванні майбутнього освіти та технологій, роблячи навчання більш доступним та захоплюючим для всіх. На прикладі двох успішних і вже популярних продуктів у цій галузі можна побачити перспективи для зміни форми навчання дітей, на більш ефективну і цікаву. Таким чином діти будуть саме пізнавати і вчитись, у цікавій для них формі, а не завчати параграфи не потрібної та нецікавої для них інформації.

РОЗДІЛ 2. РОЗРОБКА ВИМОГ І МОДЕЛЮВАННЯ НАВЧАЛЬНОЇ ІГРОВОЇ СИСТЕМИ

2.1. Аналіз і специфікація вимог до навчальної ігрової системи

Бізнес-вимоги до системи

Під час розробки навчальної ігрової системи для розвитку дітей, необхідно враховувати різні потреби та очікування, в тому числі вчителів, учнів, видавця та міністерства освіти.

Бізнес-вимоги:

- Створення партнерських програм з освітніми закладами для поширення системи
- Забезпечення захисту персональних даних учнів та вчителів, які використовують систему
- Можливість інтеграції з програмами навчання, що використовуються в закладах шкільної освіти
- Максимізувати комфорт використання для вчителів для спрощення процесу навчання

Ці бізнес-вимоги є ключовими для розробки та впровадження системи, яка задовольнить потреби всіх зацікавлених сторін та сприятиме подальшому розвитку навчальної ігрової системи, а відповідно і освіти.

Функціональні вимоги до системи

Функціональні вимоги навчальної ігрової системи для розвитку дітей (див.рис. 2.1) визначають ключові можливості, які мають бути реалізовані в системі для задоволення потреб користувачів.

Далі наведено детальний опис кожної функції, яка має бути в системі для ефективного функціонування програми та задоволення потреб.

- Movement of the character (пересування персонажа): ця функція передбачає пересування ігрового персонажа по сцені, відповідно до натиснення відповідних клавіш пересування з пристроїв вводу;

- Interaction with different objects (взаємодія з різними об'єктами): дана функція передбачає можливість для користувачів взаємодіяти з певними ігровими об'єктами, за допомогою кліка миші, або інших кнопок, в залежності від використаного пристрою введення. Це забезпечить більшу залученість користувача в процес гри (навчання через гру);

- A notification system for providing info to the user (система сповіщень, у вигляді діалогових вікон, для подання корисної інформації користувачеві): ця функція представляє собою появу діалогової вікон на екрані, де персонаж, простою мовою, доносить важливу інформації користувачеві;

- Collecting items (збирання предметів): ця функція надає користувачам можливість підбирати певні предмети, на певній відстані від ігрового персонажа;

- Change the language of the app (змінити мову застосунку): в налаштування навчальної ігрової системи можна змінити мову всього застосунку. Така функція дозволить полегшити інтеграцію іноземних учнів, або вийти системі на більш широкий рівень;

- Animations of the actions of the character (анімація дій персонажа): Ця функція передбачає наявність анімацій для кожної можливої дії користувача. Вона дозволить привернути більшу увагу користувача і покращити його досвід використання навчальної ігрової системи.

- Lack of access Internet (відсутність доступу до інтернет мережі). Це вимога, яка передбачає відсутність мережевого доступу, та забезпечення повної роботи застосунку без використання інтернету. Таким чином можна запобігти витоку даних в мережу і легше організувати процес навчання.

<input checked="" type="checkbox"/> FR1	Functional	Proposed	Low	Low
<input checked="" type="checkbox"/> FR2	Functional	Proposed	Medium	High
<input checked="" type="checkbox"/> FR3	Functional	Proposed	High	High
<input checked="" type="checkbox"/> FR4	Functional	Proposed	Low	Medium
<input checked="" type="checkbox"/> FR5	Functional	Proposed	High	Medium
<input checked="" type="checkbox"/> FR6	Functional	Proposed	Medium	High
<input checked="" type="checkbox"/> FR7	Functional	Proposed	Medium	High

Рисунок 2.1 – Таблиця специфікації функціональних вимог

Нефункціональні вимоги до системи

Далі наведено нефункціональні вимоги до навчальної ігрової системи для розвитку дітей (див.рис. 2.2). Ці вимоги встановлюються для забезпечення зручності, ефективності та надійності системи, а також для задоволення потреб зацікавлених сторін.

Дані аспекти є дуже важливими для реалізації системи та її розповсюдження, розробка та впровадження нефункціональних вимог є необхідним етапом в процесі розробки навчальної ігрової системи.

Localization (Локалізація): в системі має бути забезпечена можливість міняти локалізацію застосунку, для того аби краще інтегруватись в середовище і для більшого поширення системи. Це забезпечить кращий досвід користування системою.

Animations (Анімації): в застосунку мають бути присутні анімації персонажа, дій які він виконує, а також анімації оточенні і UI елементів. Це забезпечить краще залучення уваги гравця, а також покращить сприйняття отримуваної інформації.

Gameplay (Ігровий процес): В системі має бути реалізовано різні ігрові механіки для покращення процесу вивчення навчального матеріалу та кращого користувацького досвіду.

Security (Безпека): оскільки система передбачена для використання учнями 4-6 класів, то необхідно забезпечити належні безпеку, для безпечного навчання.

Ці нефункціональні вимоги спрямовані на забезпечення зручності, ефективності та надійності навчальної ігрової системи для розвитку дітей.

<input checked="" type="checkbox"/> NR1	Nonfunction...	Proposed	Low	Medium
<input checked="" type="checkbox"/> NR2	Nonfunction...	Proposed	High	Medium
<input checked="" type="checkbox"/> NR3	Nonfunction...	Proposed	Medium	High
<input checked="" type="checkbox"/> NR4	Nonfunction...	Proposed	Medium	High

Рисунок 2.2 – Таблиця специфікації нефункціональних вимог

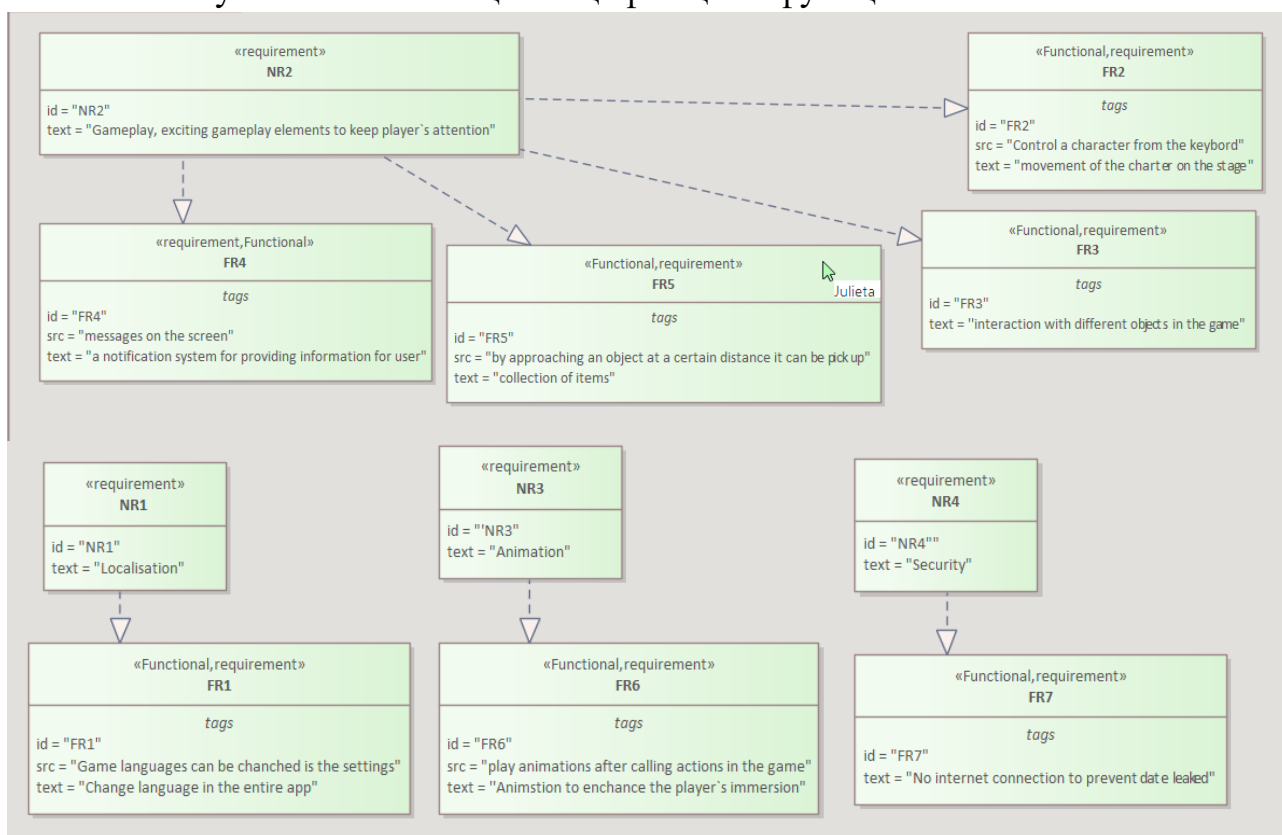


Рисунок 2.3 – Діаграма вимог з поданням ієрархії вимог

2.2 Характеристика задачі

Ця задача спрямована на полегшення навчання дітей, оскільки навчатись в ігровій формі, легше, викликає менше дискомфорту, а також краще мотивує дізнаватись нове

Основна мета – створення навчальної ігрової системи для мотивації школярів пізнавати цей світ, науки що його описують. Привити дітям асоціацію, що уроки це не нудно і нікому не потрібна, а що це цікаво і корисно. Технічно-економічна сутність це шляхом геймдизайну та використання педагогічних методик, полегшити навчання дітей.

Об'єкти управління в комп'ютерній навчальній грі — це ключові елементи, які гравець може взаємодіяти або контролювати в ході гри. Ці об'єкти визначаються з метою навчання конкретних концепцій, розвитку навичок або вирішення завдань. Розглянемо типові об'єкти управління у комп'ютерних навчальних іграх:

Завдання та задачі

Об'єкти управління можуть представляти конкретні завдання або задачі, які гравець повинен вирішити. Це може включати в себе розв'язання математичних завдань, логічних головоломок або інших викликів.

Персонаж

Гравець може управляти персонажем або аватаром, що представляє його в ігровому світі. Цей об'єкт може включати в себе власні характеристики та завдання.

Інструменти та обладнання

Управління інструментами або обладнанням дозволяє гравцеві використовувати різні ресурси чи знаряддя для досягнення конкретних цілей.

Елементи середовища

Важливими об'єктами управління можуть бути різні елементи ігрового середовища, такі як будівлі, території або інші об'єкти, які гравець може модифікувати чи використовувати для вивчення концепцій.

Загадки та головоломки

Спеціальні об'єкти управління можуть представляти загадки чи головоломки, які гравець повинен розв'язати для продовження гри.

Методи оцінки та повідомлення

Об'єкти управління включають систему оцінювання або повідомлення, щоб надавати гравцеві зворотний зв'язок щодо його успіху чи навіть прогресу в навчанні.

Ці об'єкти управління допомагають створювати динамічні та змістовні навчальні ігри, які сприяють активному вивченню та розвитку навичок у користувачів.

Вхідна інформація

Інтегруючи об'єкти управління в ігровий процес, ми можемо створити динамічні та змістовні ігри, які базуються на вхідних даних, що сприяють активному розвитку навичок користувачів.

Вхідна інформація для цього кваліфікаційного бакалаврського проєкту – це набір даних, які будуть надходити і використовуватись в процесі гри, за допомогою даної ПС, інформацію про вхідні дані можна переглянути у таблиці 2.1 (див. табл. 2.1).

Таблиця 2.1 – Перелік і опис вхідних даних повідомлень

№ з/п	Назва вхідного повідомлення	Форма подання	Термін і частота надходження	Джерело
1.	Натиснення клавіші (запит користувача на пересування)	Електронна форма	Кожен фрейм	Навчальна гра
2.	Персонаж входить в певну зону	Електронна форма	Кожен фрейм	Навчальна гра
3.	Натиснення клавіші «Esc»	Електронна форма	Кожен фрейм	Навчальна гра

№ з/п	Назва вхідного повідомлення	Форма подання	Термін і частота надходження	Джерело
4.	Взаємодія користувача з об'єктами	Електронна форма	Кожен фрейм	Навчальна гра

Вихідна інформація

Вихідна інформація буде використовуватися для візуального підтвердження дій користувача, для перегляду інформації про вихідні дані переглянемо таблицю 2.2 (див. табл. 2.2).

Таблиця 2.2 – Перелік і опис вихідних даних повідомлень

№ з/п	Назва вхідного повідомлення	Форма подання	Термін і частота надходження	Джерело
	Пересування персонажа по сцені	Електронна форма	Кожен фрейм	Користувач
	Вивід діалогового вікна на екран	Електронна форма	Кожен фрейм	Користувач
	Вивід меню паузи на екран	Електронна форма	Кожен фрейм	Користувач

Функціонування системи

Процес пересування в грі:

1. Користувач натискає клавішу що відповідає за рух в сторони (стрілки вліво вправо, або «A», «D»);
2. Система сприймає ці введення як значення, або 1 (вправо), або -1 (вліво), в залежності від сторони ;
3. В системі є змінна яка сприймає це значення, відповідно до цього значення, спрайт персонажа розвертається у потрібну сторону;

4. В системі є умова, якщо значення змінної яка відповідає за сприймання значень з клавіш пересування, не дорівнює нулю, то програється анімація ходьби.

5. Розрахування фізичної швидкості пересування ігрового об'єкта, тобто персонажа, здійснюється за допомогою даних про поточну позицію персонажа, по координатам x, y, а також з урахування параметру швидкості персонажа.

6. На екрані користувача відображається переміщення персонажа.

Процес розмови персонажа з гравцем:

1. Користувач при пересуванні перетинає певну зону;
2. При перетині зони система отримує інформації про те це;
3. Після отримання інформації діалогове вікно стає активним, себто видимим і дійсним в рамках ПС;

4. Запускається код виводу тексту в діалогове вікно за певним алгоритмом;

5. Запускається анімація для портрета персонажа поряд з діалоговим вікном;

6. На екрані користувача відображається діалогове вікно персонажа.

Процес взаємодії з об'єктами:

1. Користувач заходить в певну зону;
2. Система отримує про це інформацію і робить об'єкт клікабельним;
3. Об'єкт виділяється підсвіткою, аби було зрозуміло що він тепер клікабельний;

4. Користувач натискає лівою кнопкою миші по об'єкту;

5. Інформація про це надходить в систему, переключаячи відповідну змінну в положення «true»;

6. Запускається частина коду що відповідає за цей об'єкт;

7. Користувач бачить результат кліку на екрані свого монітору

Процес зупинки гри, пауза:

1. Користувач натискає кнопку зупинки гри, «Esc»

2. Система отримує інформацію про те що клавіша «Esc», була натиснута;
3. Активується меню паузи, тобто стає видимим і повністю функціональним;
4. Плин часу в грі зупиняється;
5. Користувачеві на екран виводиться меню паузи з трьома доступними опціями : «Продовжити», «Меню», «Вийти з гри»;
6. Користувач натискає клавішу «Esc», або вибирає опцію «Продовжити», тоді плин часу в грі повертається на попередній стан.

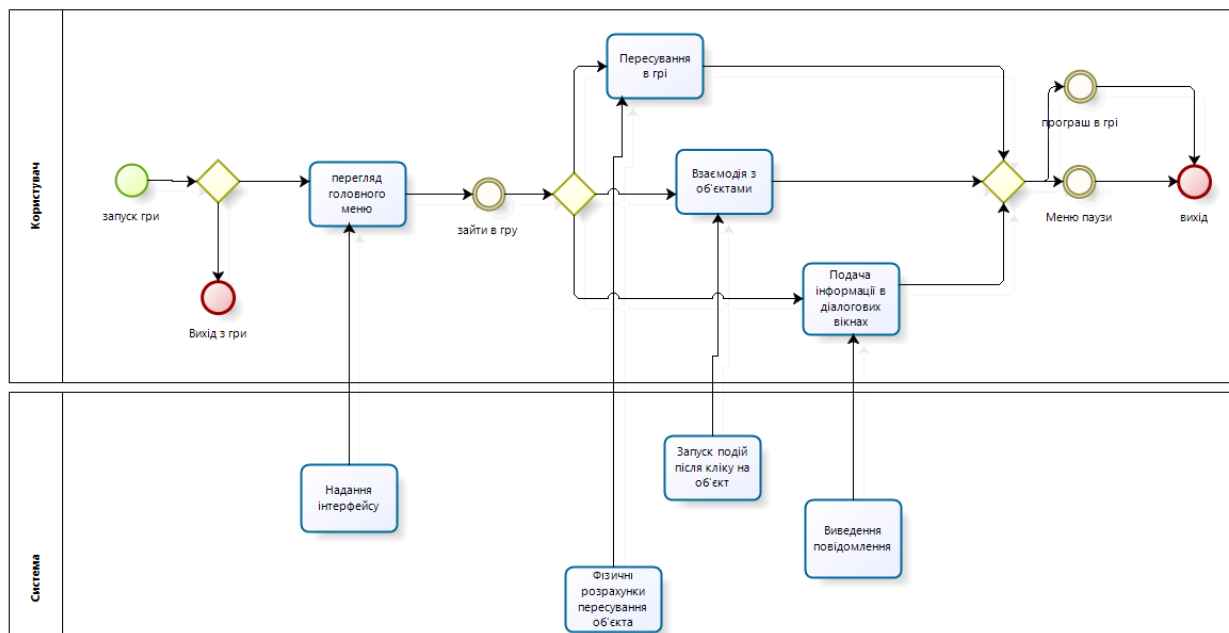


Рисунок 2.4 – Алгоритм функціонування системи

2.3 Моделювання поведінки системи

Діаграма прецедентів (див.рис. 2.5) є одним із найважливіших інструментів в системному аналізі, за допомогою якого можна визначити функціональність та її взаємодію з користувачами.

За допомогою діаграми прецедентів можна чітко визначити всі можливі взаємодії користувача і системи. Це допомагає зрозуміти потреби користувачів та вимоги до системи. Розробка даної діаграми сприяє збільшенню вірогідності збільшення шансів на успіх, подаючи послідовне та структуроване описання функцій системи й гарантуючи високу якість розробленого програмного забезпечення

Діаграма прецедентів навчальної ігрової системи для розвитку дітей описує різні можливі взаємодії, які може виконувати користувач.

Користувач системи має можливість виконувати такі дії:

- Пересування персонажем: цей прецедент вказує на те що користувача може пересуватись ігровим персонажем по сцені, задля виконання поставлених завдань.
- Взаємодія з ігровими об'єктами: цей прецедент дозволяє користувачеві натискати на певні ігрові, клікабельні, об'єкти, тим самим викликати якусь подію, або виконувати поставленні завдання.
- Отримання пояснень/підказок: цей прецедент дозволяє користувачеві, при взаємодії з ігровим світом отримувати пояснення у вигляді тексту в діалоговому вікні, таким чином, у користувача буде складатись враження, наче це йому пояснює сам персонаж з гри. Цей прецедент робить гру більш живою, а також дозволяє подавати пояснення у зручній і зрозумілій формі.
- Зупинка гри: цей прецедент вказує на можливість зупинки гри, користувачем, в будь який момент гри. Даний прецедент необхідний через причини зручності ігрового процесу, а також покращення досвіду користування системою.

Рушій на якому працює система має певні відповідні функції, які допомагають покращити досвід користувача:

- Прийняття введених користувачем значень: для рушія був написаний окремий код менеджер, який слідкує за тим що користувач натискає і відповідно реагує;
- Опрацювання даних: цей прецедент є наступним кроком після прийняття введених значень, після прийняття значень, окремий код і можливості рушія обробляють введенні значення для того аби відповідно відреагувати (пересунути персонажа в сторону на певну відстань і так далі);

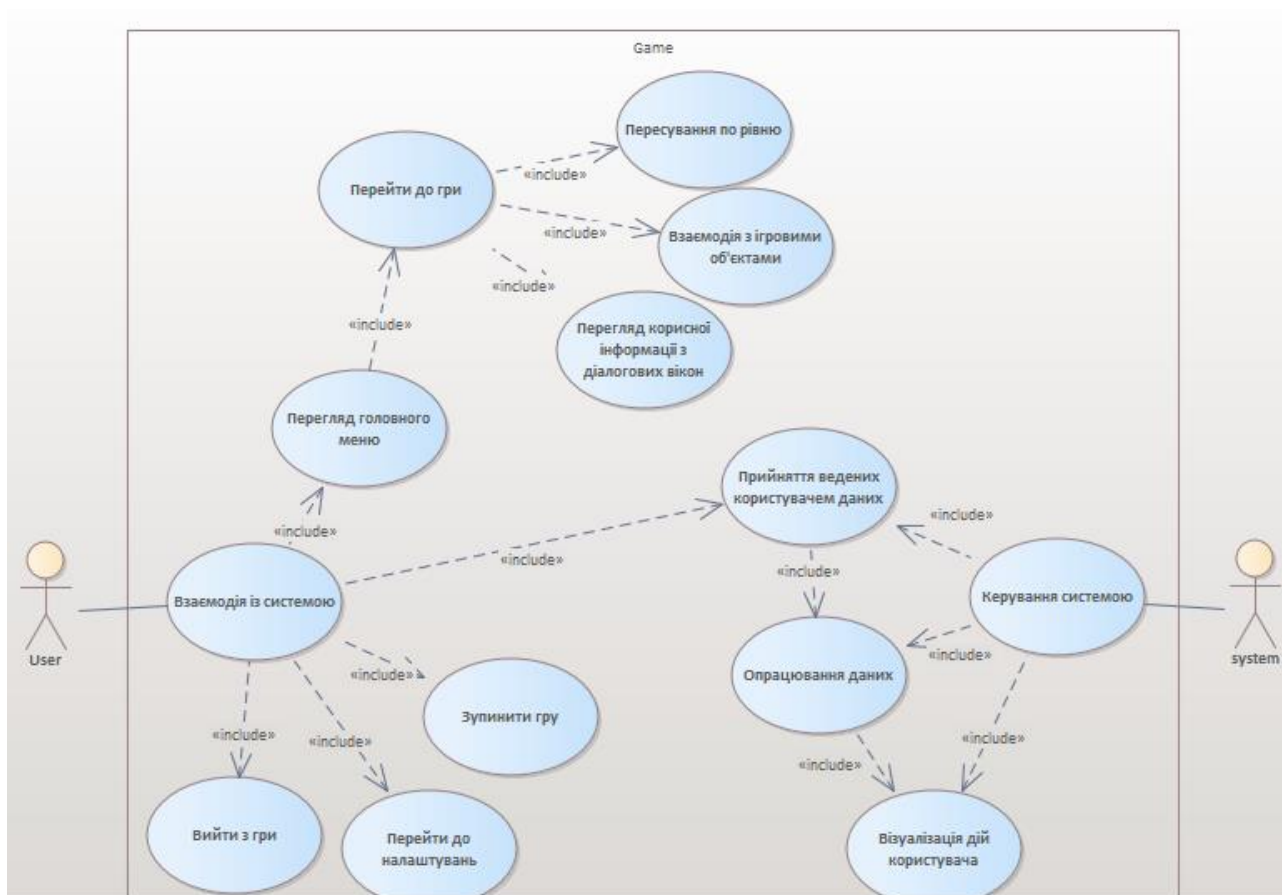


Рисунок 2.5 – Діаграма прецедентів

1. Користувач пересувається персонажем по сцені гри:

- 1) Користувач натискає відповідні клавіші пересування, коли ігровий персонаж знаходиться на сцені гри;

2) Система приймає це введення у вигляді відповідних значень;

3) Після отримання відповідних значень система, за допомогою формули, оброблює введенні дані, та визначає параметри дії, після чого.

4) Після визначення параметрів відповідної дії, ця дія відбувається на сцені гри, а також викликається її анімація.

Альтернативний сценарій:

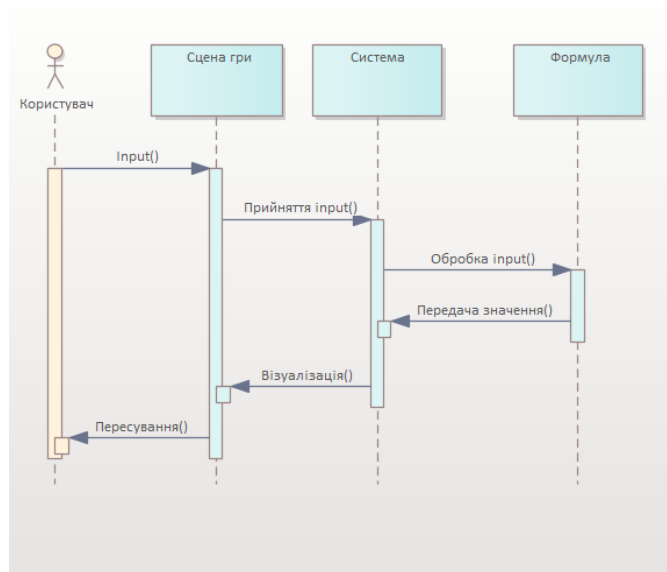


Рисунок 2.6 – Діаграма послідовності прецеденту «Пересування»

2. Користувач натискає на клікабельний ігровий об’єкт:

- 1) Користувач натискає на ігровий об’єкт на сцені гри;
- 2) Даний клік по об’єкту фіксується event системою;
- 3) Після отримання інформації про те по якому предмету було здійснено клік, event система міняє відповідні параметри;
- 4) Зміну параметрів здійснює сама система, вона обробляє відповідний запит і здійснює зміну значень об’єктів;
- 5) В подальшому ці змінені значення через event систему запускають в дію відповідні коди і відбувається візуалізація результату кліку на сцені гри.

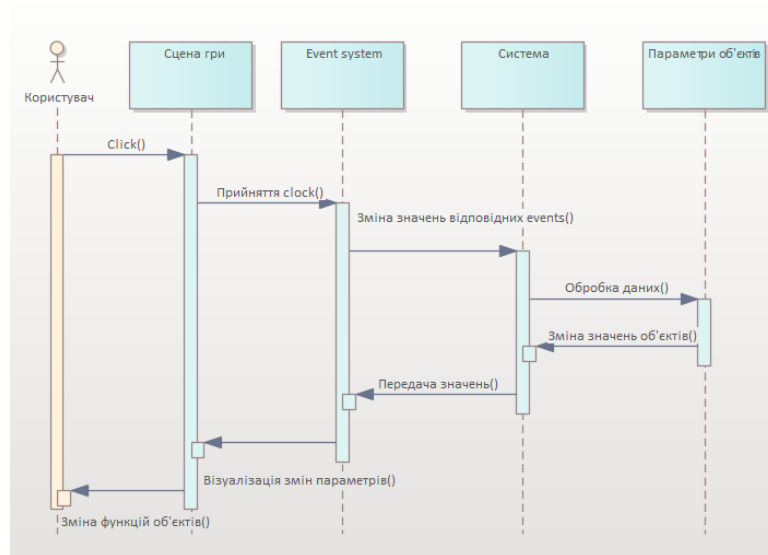


Рисунок 2.7 – Діаграма послідовності прецеденту «натиснення»

3. Користувач отримує повідомлення у вигляді діалогового вікна:

- 1) користувач пересувається по сцені гри;
- 2) колайдер (межі) персонажа перетинається з колайдером відповідної зони;
- 3) Після цього в систему передаються дані про перетин меж певної зони;
- 4) Після отримання даних система запускає подію діалог, яка в свою чергу візуалізує діалогове вікно із текстом на екрані користувача.

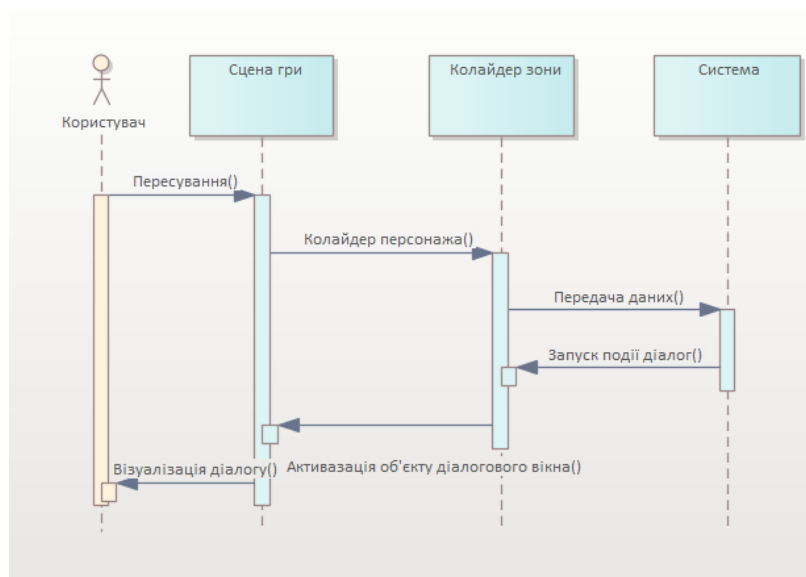


Рисунок 2.8 – Діаграма послідовності прецеденту «Отримання повідомлення»

РОЗДІЛ 3. ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ КОМПОНЕНТІВ НАВЧАЛЬНОЇ ІГРОВОЇ СИСТЕМИ

3.1. Інформаційне забезпечення

Для розробки даного кваліфікаційного бакалаврського проєкту буде використано такі види програм з розробки програмних продуктів як :

- Unity
- Visual Studio

Unity[6] – це платформа для розробки ігор та інтерактивних додатків, яка має широкий спектр можливостей:

1. **Розробка ігор.** Unity надає потужні інструменти для створення як 2D, так і 3D ігор для різних платформ, включаючи ПК, консолі, мобільні пристрої та веб.
2. **Кросплатформеність.** За допомогою Unity можна створювати ігри та додатки, які працюють на різних платформах, таких як iOS, Android, Windows, macOS, Xbox, PlayStation та інші.
3. **Візуальний редагувальний інтерфейс.** Unity має зручний візуальний інтерфейс, що дозволяє створювати та редагувати ігрові сцени, анімації, ефекти та інші елементи без потреби програмування.
4. **Мови програмування.** Unity підтримує різні мови програмування, такі як C#, JavaScript, Boo, що дозволяє розробникам вибирати ту, з якою вони найбільш зручно працюють.
5. **Активна спільнота та ресурси.** Unity має велику та активну спільноту розробників, а також безліч ресурсів, таких як уроки, документація та форуми, що сприяє навчанню та підтримці користувачів.
6. **Фізика та анімація.** Unity має вбудовану систему фізики та анімації, що дозволяє реалістично моделювати рух об'єктів та створювати живу анімацію персонажів.
7. **Артистичні можливості.** Unity має потужні інструменти для створення ефектів, освітлення, текстур та іншого артистичного контенту, що допомагає реалізувати бажаний стиль гри чи додатку.

Unity[9] використовується для створення ігор, віртуальної реальності, додатків для навчання та тренування, архітектурного візуалізації, інтерактивних історій, симуляцій та багато іншого.

Процес інсталяції Unity

Щоб встановити рушій, необхідно зайти на офіційний сайт (див.рис. 3.1) і натиснути на кнопку “Тарифи і ціни” (Plans and pricing). Після цього користувача переносить на сторінку, необхідно зайти в розділ “Студент і ентузіаст” (Student and hobbyist) та обирати варіант Personal. Це повноцінна безкоштовна версія Unity. Нею можна користуватися, поки виручка від ігор, зроблених на Unity, не перевищує 100 тисяч доларів за останні 12 місяців.

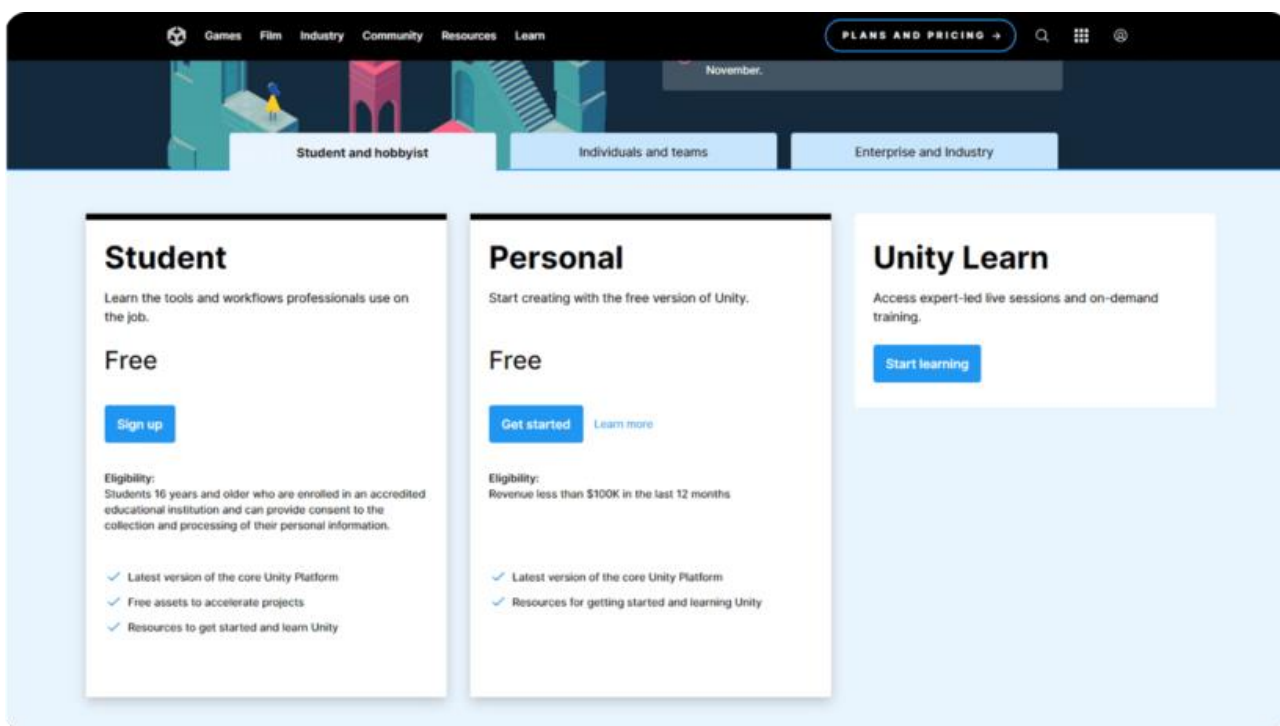


Рисунок 3.1 – Екран вибору версії UnityHub

Далі необхідно дотримуватись інструкцій. Спочатку треба завантажити та встановити програму «Unity Hub». З її допомогою можна завантажити різні версії Unity, створювати і відкривати проекти.

Під час першого запуску «Unity Hub» запропонує встановити найактуальнішу версію Unity. Встановити потрібну версію Unity після цього можна в розділі «Installs» (див. рис. 3.2).

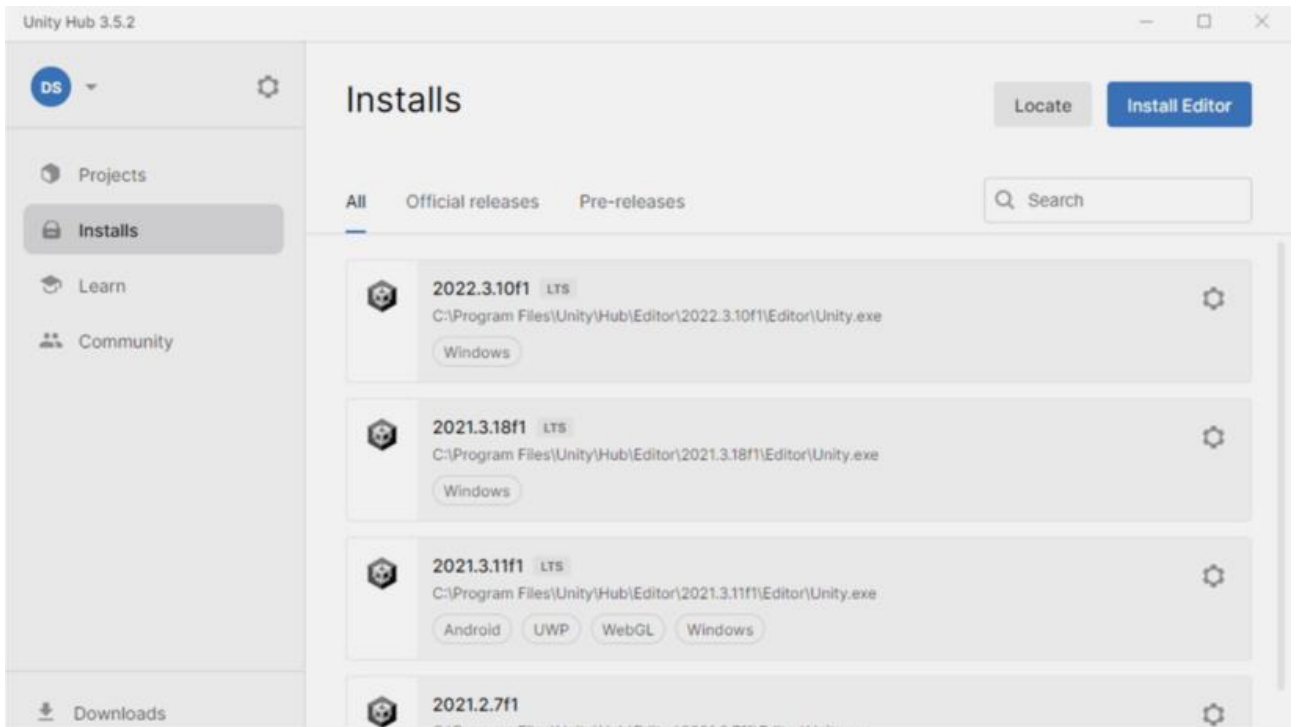


Рисунок 3.2 – Екран інсталяцій Unity версій

Створити новий проєкт можна через розділ «Projects». Після натиснення на кнопку «New project» користувача переносить в розділ, де можна вибрати, як буде налаштований редактор за замовчуванням (див. рис. 3.3).

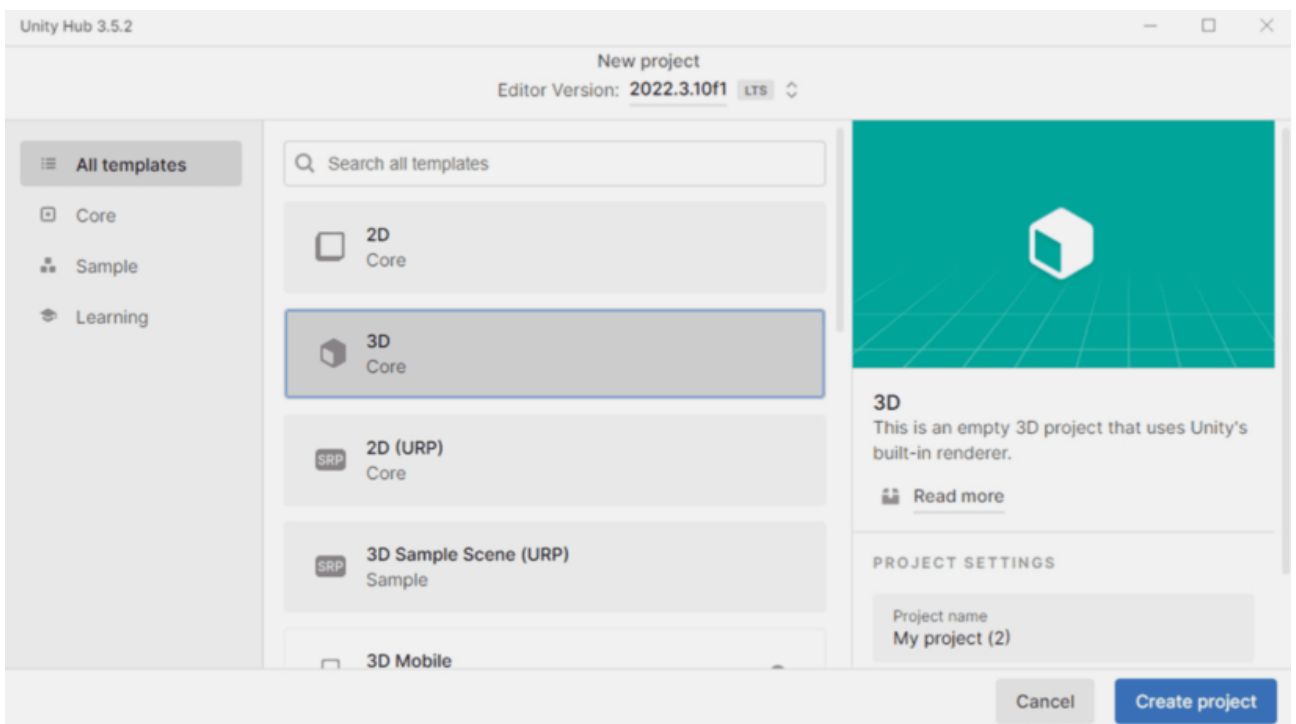


Рисунок 3.3 – Вибір передвстановлених налаштувань проєкту

Технічно всі ігри в Unity тривимірні. Тому навіть якщо почати проєкт у режимі 2D, а за кілька місяців розробки прийняти рішення про перехід в 3D

формат, то це можна зробити всередині самого редактора рушія, без перестворення проєкту.

Процес інсталяції Visual Studio

Спочатку необхідно зайти на офіційний сайт Visual Studio[7] та скачати необхідну версію (див.рис. 3.4).

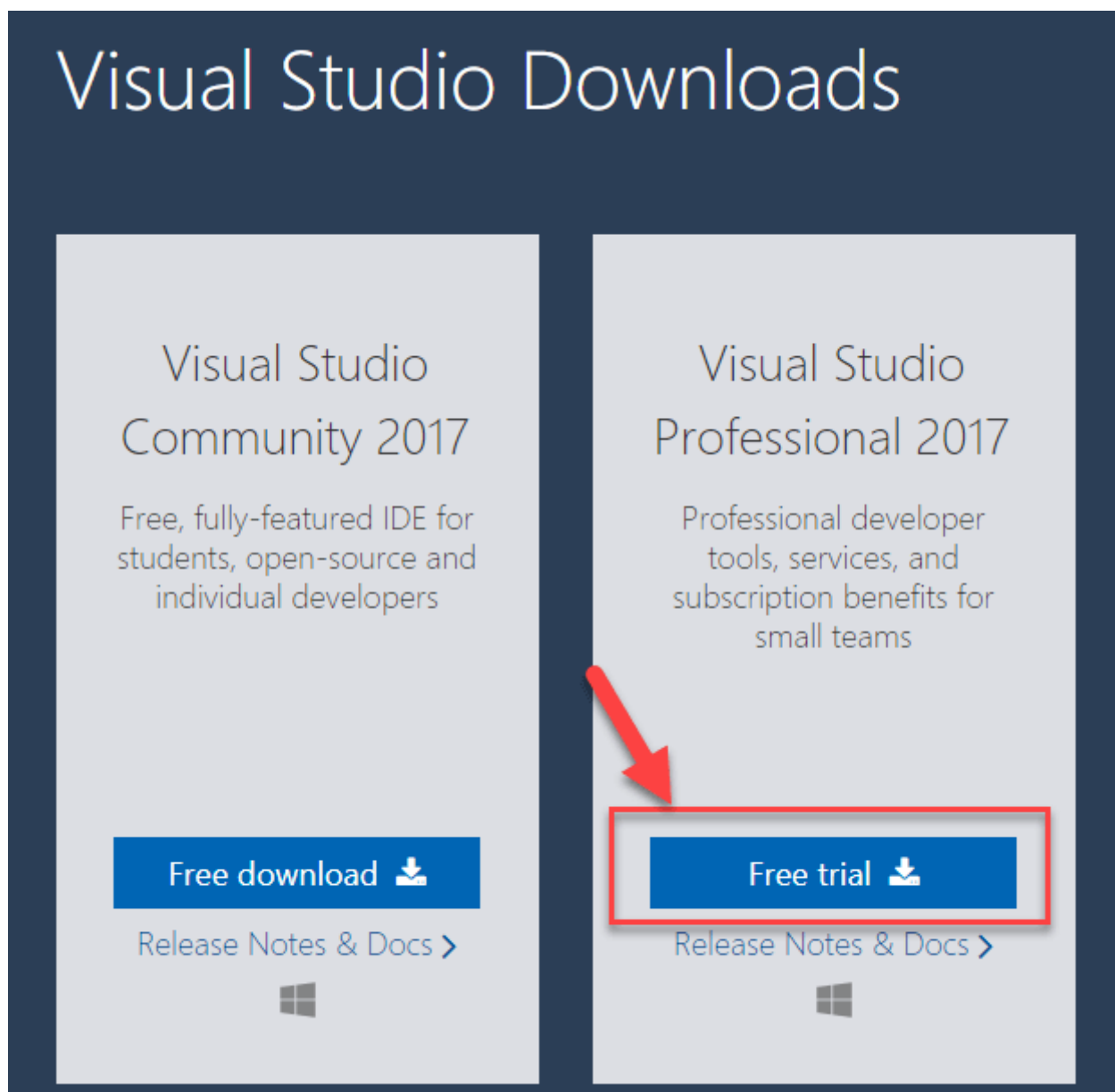


Рисунок 3.4 – Вибір версії редактора до завантаження

Далі необхідно натиснути на завантажений файл ехе, на екрані що з'явився треба натиснути «Продовжити», для початку установки Visual Studio (див. рис. 3.5).

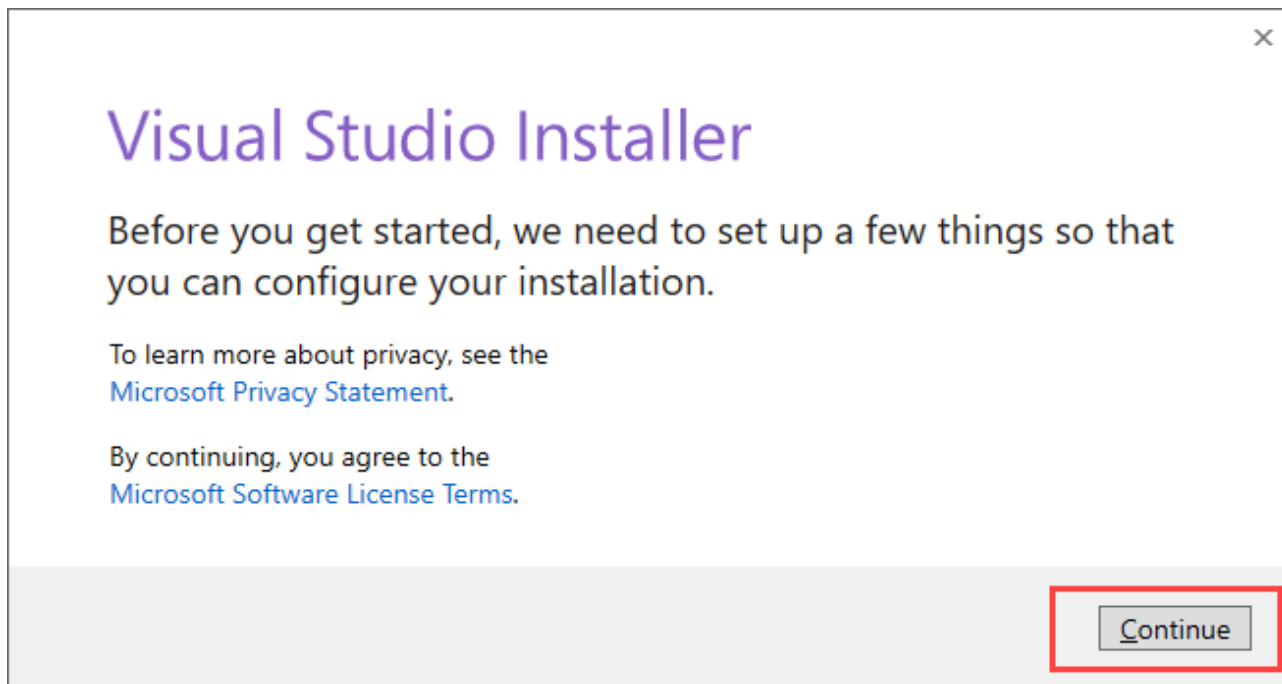


Рисунок 3.5 – Вікно встановлення Visual Studio

На наступному екрані встановлення необхідно вибрати версію редактора, яка буде встановлена, а після натиснути кнопку «Установити» (Instal) (див.рис. 3.6)

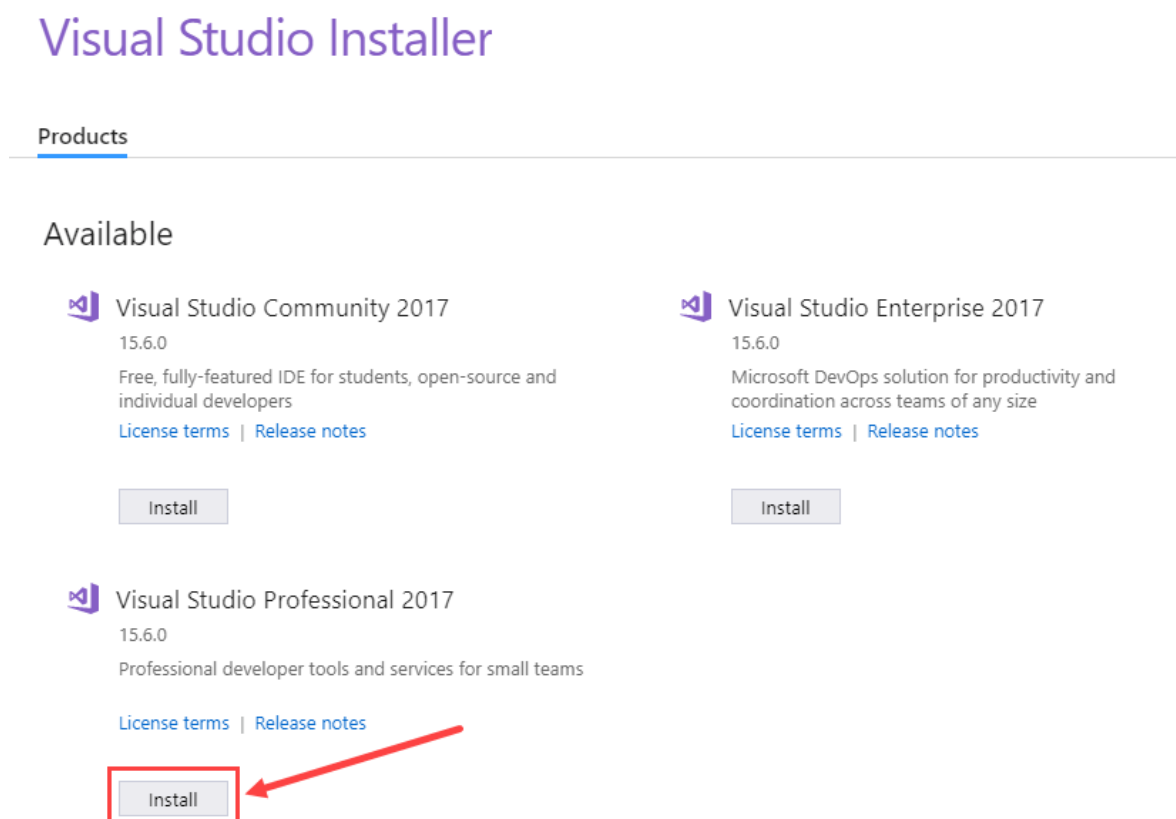


Рисунок 3.6 – Екран вибору версії редактора

На наступному екрані треба вибрати технології які мають бути встановлені у редактор, на думку користувача (див. рис. 3.7). Пізніше, за бажанням, можна докочати необхідні пакети.

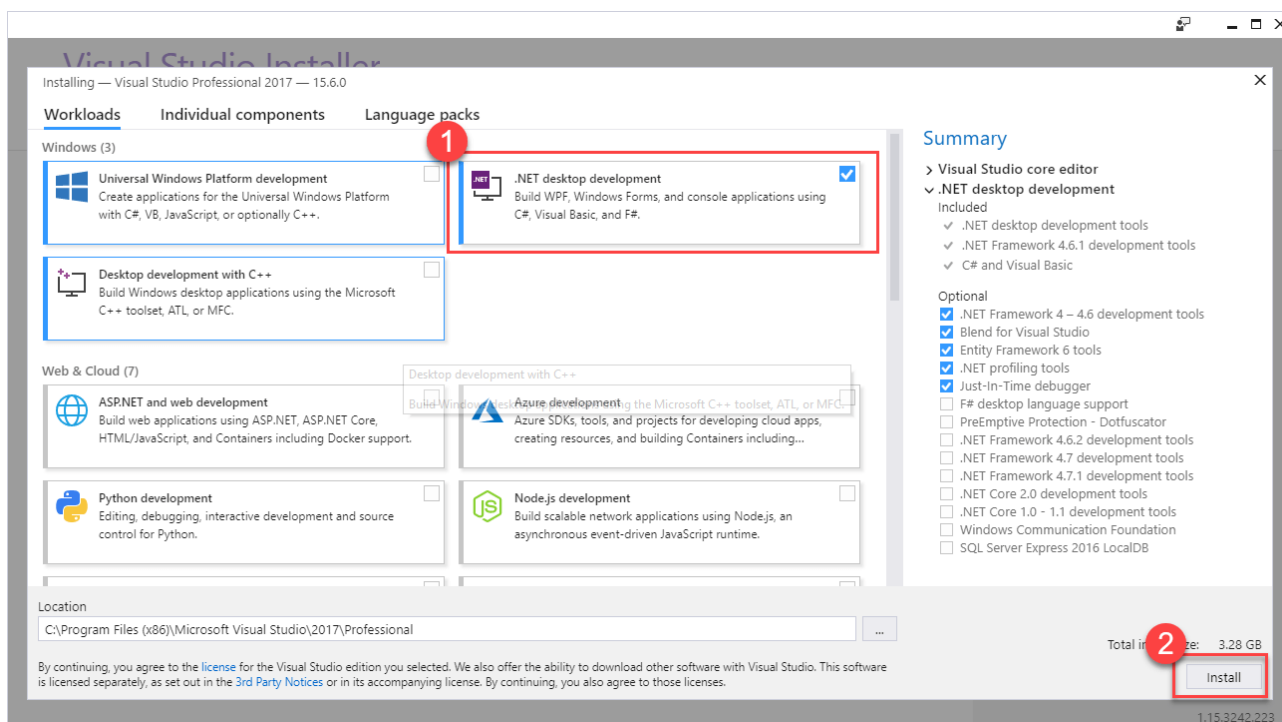


Рисунок 3.7 – Вибір технологій для редактора

Після вибору пакетів буде розпочато інсталювання редактору. По завершенню існталятор попросить користувача перезавантажити машину для коректної роботи редактора (див. рис. 3.8).

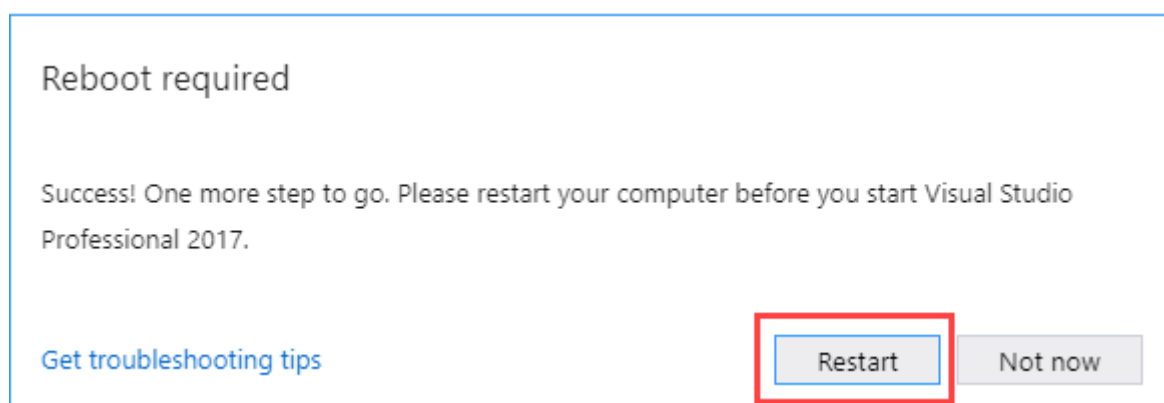


Рисунок 3.8 – Повідомлення про прохання перезавантажити комп'ютер

Після перезавантаження повторний запуск редактора пропонує вибрати тему (див.рис. 3.9), після чого з'являється можливість в повній мірі користуватись редактором.

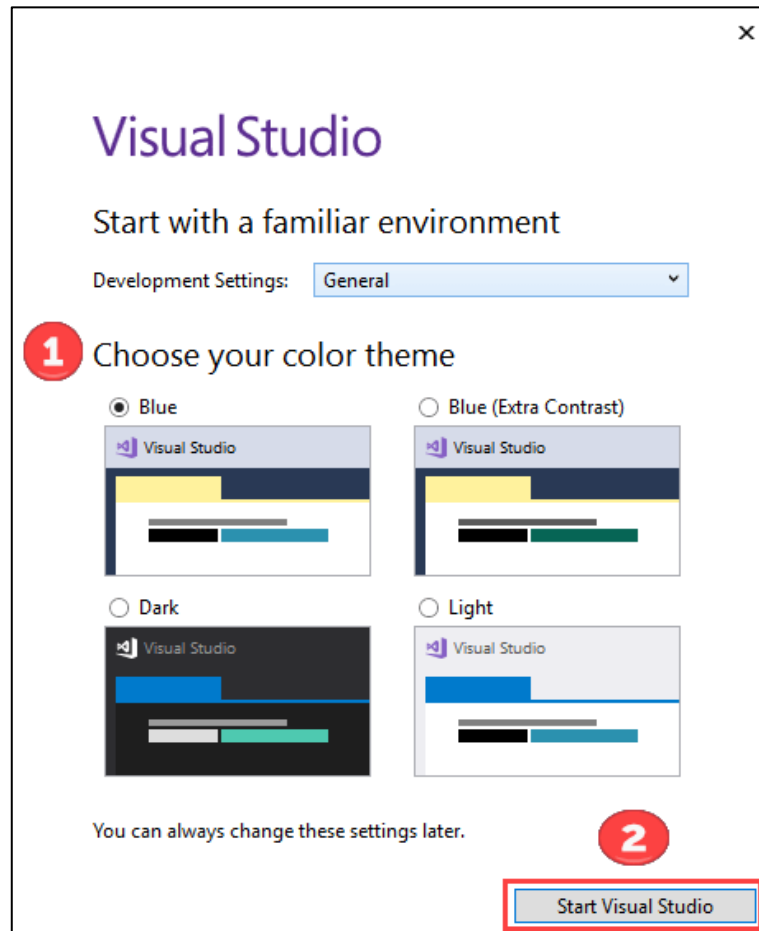


Рисунок 3.9 – Вибір кольорової теми для редактора

В редакторі натиснувши «File» можна створити новий проєкт і працювати з кодом. В даному проєкті Visual Studio використовується в якості редактора коду.

Ключові особливості Visual Studio[8]:

- Створення додатків будь-якого типу – Visual Studio IDE можна використовувати для створення програми будь-якого типу. (Веб-програма або Windows Додаток на основі форм).
- Налаштування програм на льоту – Програми можна тестувати в міру їх створення. IDE дозволяє запускати програму в будь-який момент часу в процесі розробки. Таким чином, розробник може перевірити будь-які помилки під час самої фази розробки.
- Розширення – IDE має можливість інсталивати сторонні розширення. Прикладом може бути Subversion, який використовується для керування

репозиторієм вихідного коду. Subversion використовується для завантаження коду в центральне сховище. Це робиться для того, щоб завжди існувала копія коду. Visual Studio має засоби для інтеграції з таким програмним забезпеченням із самої IDE.

Структура і компоненти рушія Unity

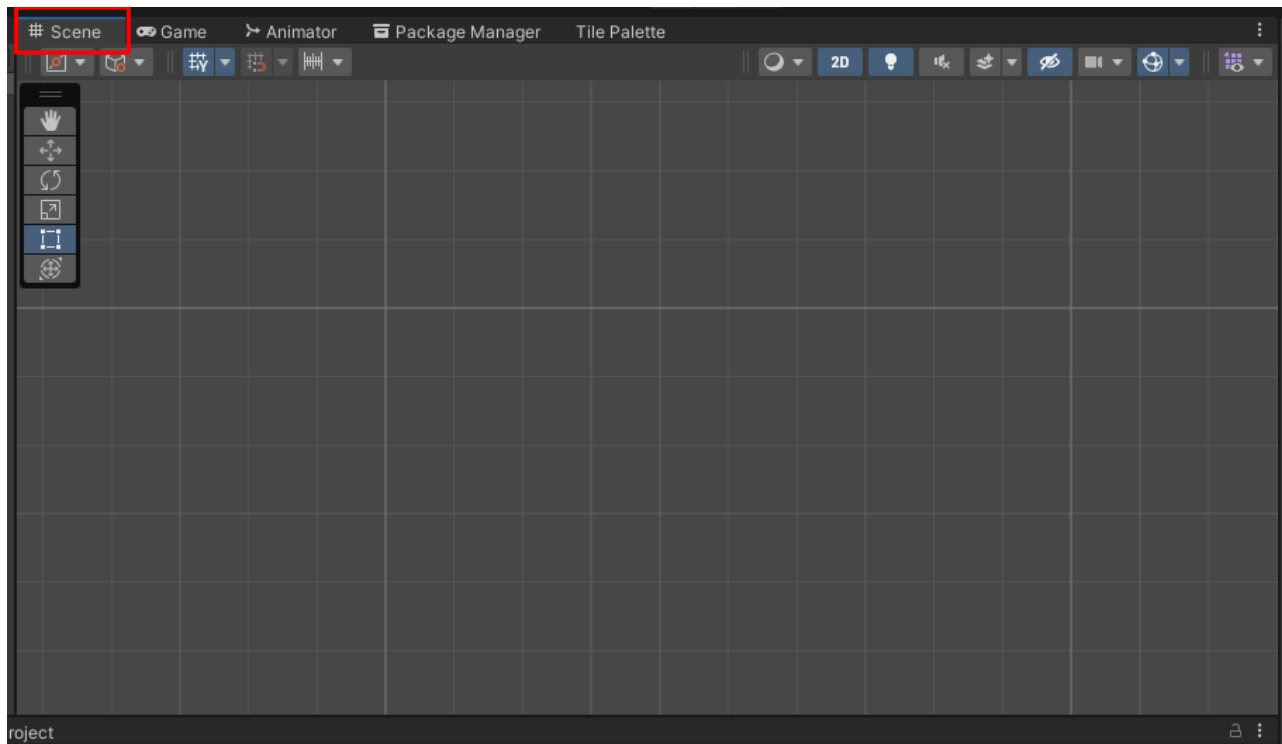


Рисунок 3.11 – Вкладка "Scene" в Unity

Вкладка "Scene"[10] в Unity (див.рис. 3.11) дозволяє переглядати і редагувати тривимірні та двовимірні сцени. Ви можете переміщувати камеру для огляду сцени з різних ракурсів. Інструменти трансформації дозволяють переміщувати, обертати і змінювати розмір об'єктів. Ви можете додавати, видаляти і редагувати об'єкти, змінювати їхні параметри через інспектор. Unity пропонує різні режими перегляду та варіанти відображення об'єктів. Навігація сценою здійснюється за допомогою комбінацій клавіш і миші. Клік правою кнопкою миші відкриває контекстне меню для швидкого доступу до функцій. Гіздо і інші інструменти допомагають розуміти розташування і взаємодію об'єктів.

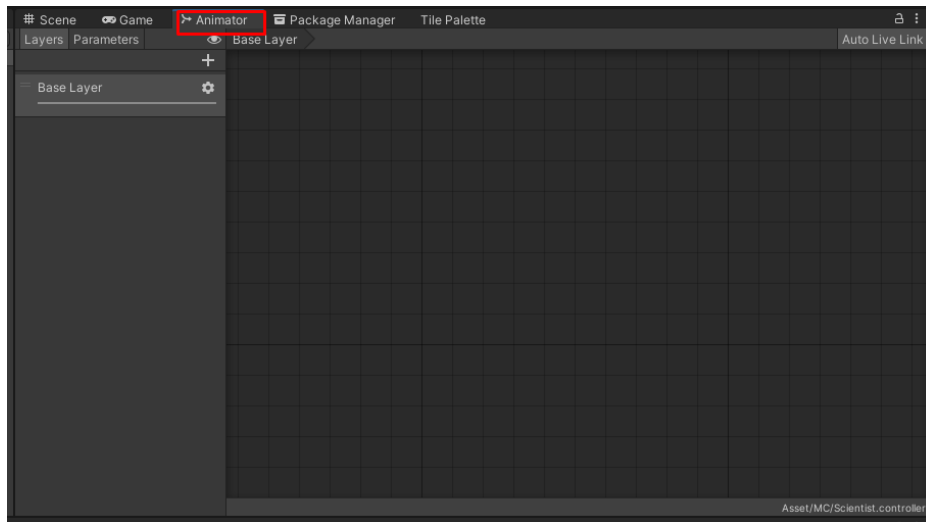


Рисунок 3.12 – Вкладка "Animator" в Unity

Вкладка "Animator"[11] в Unity (див.рис. 3.12) призначена для створення, налаштування і управління анімаційними поведінками об'єктів у грі. Вона дозволяє створювати різні анімаційні стани та налаштовувати їх переходи, використовуючи параметри анімації і зображуючи їх у вигляді державної машини. Крім того, ви можете використовувати Blend Trees для змішування анімацій, а також додавати підстани і редагувати анімаційні параметри прямо у вкладці "Animator".

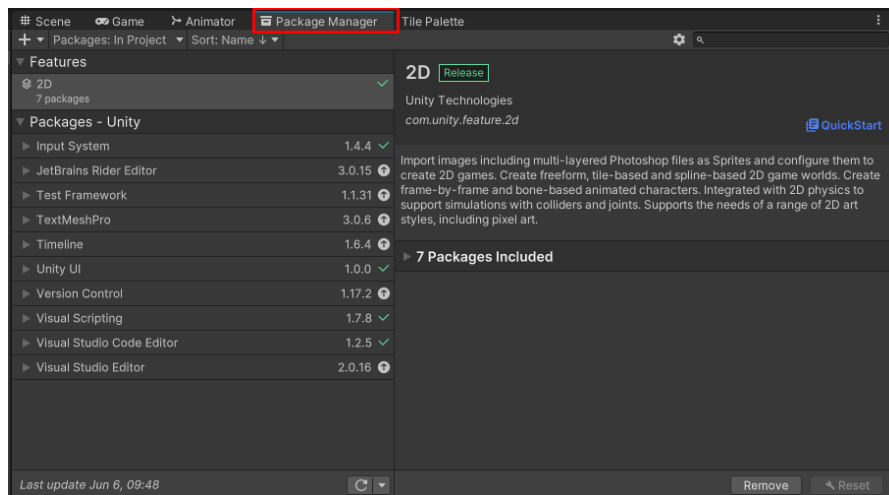


Рисунок 3.13 – Вкладка "Package Manager" в Unity

Вкладка "Package Manager"[12] в Unity (див.рис. 3.13) це інструмент для керування розширеннями (пакетами), які додають функціональність до вашого проєкту, дозволяючи швидко встановлювати, оновлювати та видаляти різні ресурси, такі як плагіни, бібліотеки, залежності та інші розширення. Ви можете

шукати пакети у магазині Unity або встановлювати їх з власних джерел, а також керувати версіями пакетів для кращої сумісності та стабільності вашого проєкту.

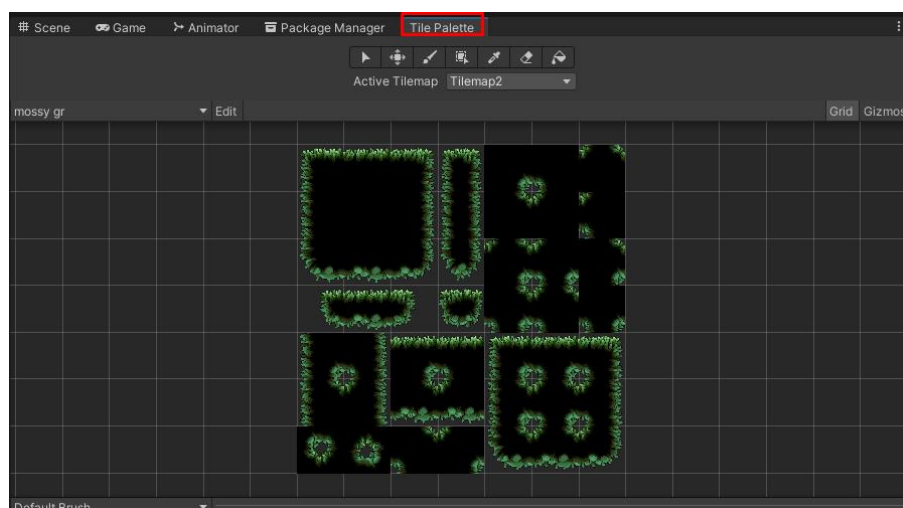


Рисунок 3.14 – Вкладка "Tile Palette" в Unity

Вкладка "Tile Palette"[13] в Unity (див.рис. 3.14) – це інструмент для створення та редагування ігрових карт за допомогою плиткових (тайлових) систем. Вона надає можливість вибирати, розміщувати та редагувати тайли (плитки) на ігровому полі, що дозволяє швидко створювати різноманітні ландшафти, рівні та інші ігрові об'єкти. Крім того, ви можете створювати власні набори тайлів, використовувати автотайли для автоматичного заповнення певних областей та інші корисні функції для роботи з тайловими картами.

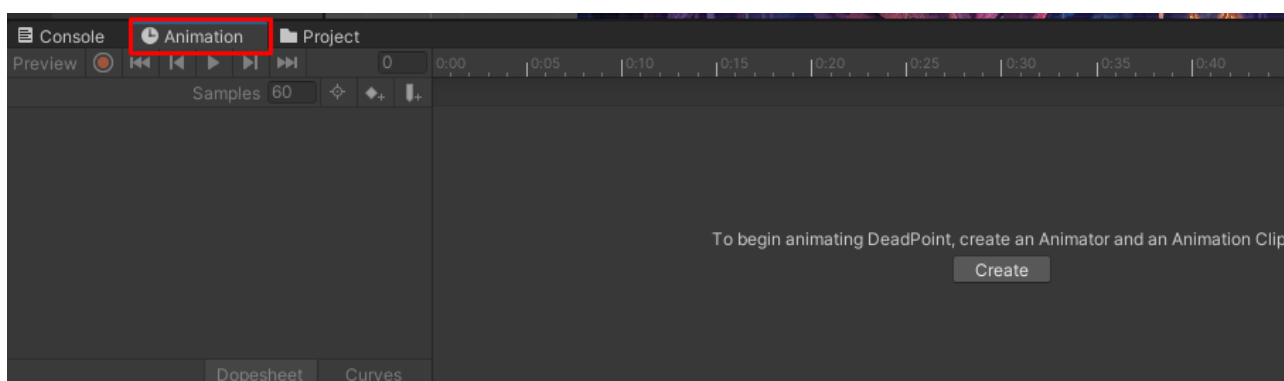


Рисунок 3.15 – Вкладка "Animation" в Unity

Вкладка "Animation"[14] в Unity (див.рис. 3.15) дозволяє створювати, редагувати та управляти анімаціями об'єктів у вашій грі чи програмі. Вона надає інтерфейс для створення ключових кадрів, налаштування кривих інтерполяції, а також для керування параметрами анімації, такими як швидкість відтворення та петлі. Ви можете створювати різні анімаційні кліпи для різних дій об'єктів і потім

комбінувати їх у комплексні анімаційні системи за допомогою вкладки "Animator".

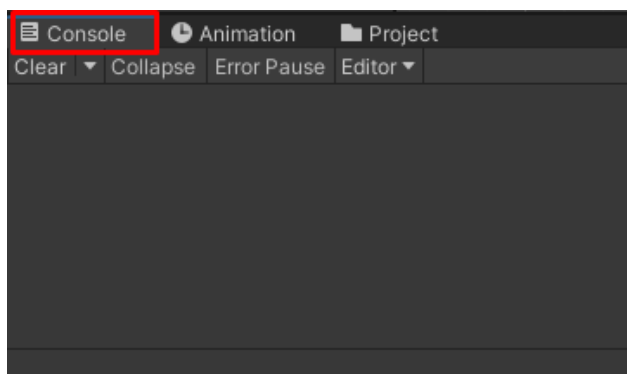


Рисунок 3.16 – Вкладка "Console" в Unity

Вкладка "Console"[15] в Unity (див.рис. 3.16) відображає вихідну інформацію, повідомлення про помилки, попередження та інші повідомлення системи під час роботи вашої гри чи програми. Вона дозволяє вам відслідковувати та виправляти помилки, а також збирати інформацію для налагодження та вдосконалення вашого проекту. Крім того, ви можете використовувати консоль для виведення власних повідомлень у режимі розробки.

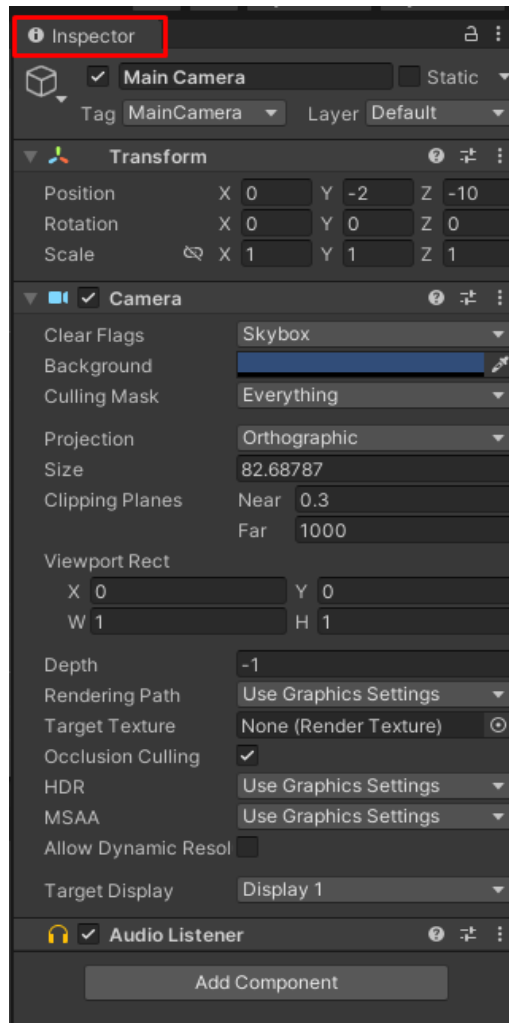


Рисунок 3.17 – Вкладка "Inspector" в Unity

Вкладка "Inspector"[16] в Unity (див.рис. 3.17) відображає детальну інформацію про вибраний об'єкт на сцені чи в ієрархії проекту. Вона дозволяє вам переглядати та редагувати параметри об'єкта, такі як його розмір, положення, обертання, матеріал, скрипти та інші властивості, а також додається доступ до компонентів, які додаються до об'єкта.

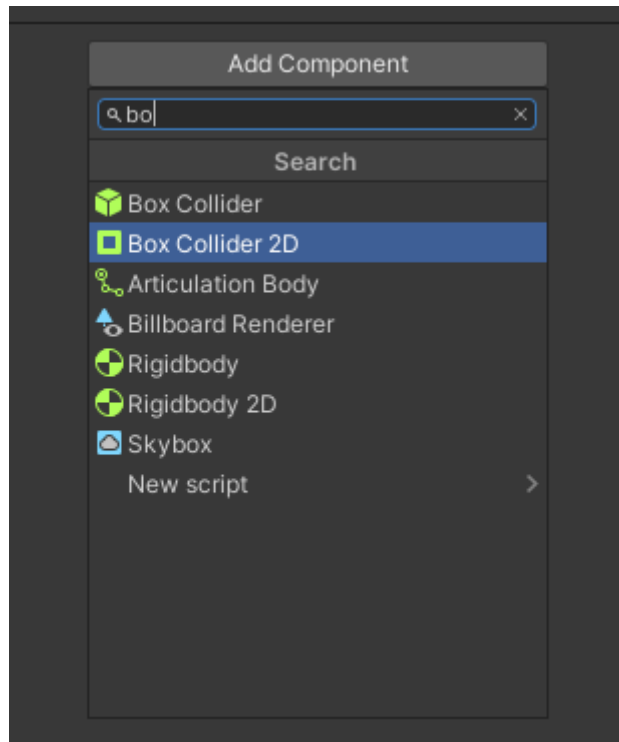


Рисунок 3.18 – Опція "Add Component" в Unity

Опція "Add Component"[17] в Unity (див.рис. 3.18) дозволяє додавати нові компоненти до обраних об'єктів у вашому проєкті. Вона відкриває доступ до широкого спектру компонентів, які надають різноманітні функції, такі як фізика, анімація, звук, штучний інтелект та інше, що допомагає реалізувати потрібні вам можливості у грі чи програмі.

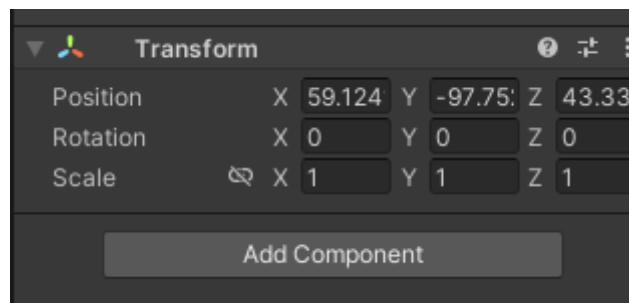


Рисунок 3.19 – Елемент " Transform " в Unity

Елемент "Transform"[18] в Unity (див.рис. 3.19) є основним компонентом, який визначає положення, обертання та масштаб об'єкта в 3D або 2D просторі. Він дозволяє вам визначати точне розташування об'єкта у світі гри, а також його орієнтацію та розмір. Transform також використовується для управління ієрархією об'єктів у сцені Unity.

3.2 Технічне забезпечення

Технічне забезпечення — це сукупність апаратних засобів, програмного забезпечення, мережесих ресурсів, інженерних систем та послуг з підтримки та обслуговування, які необхідні для функціонування та реалізації певних задач і процесів в організації чи проєкті.

Технічне забезпечення для навчальної комп'ютерної гри для дітей:

1. Апаратне забезпечення:

- Комп'ютери. Пристрої, на яких діти гратимуть у гру. Вони повинні мати достатню продуктивність для плавної роботи гри (мінімальні характеристики процесору IntelCore™ I5-3250; оперативна пам'ять: Мінімум 4 GB RAM; відеокарта: інтегрована Intel HD Graphics 520, або дискретна NVIDIA GeForce GT 710; Місце на диску близько 500 MB).
- Периферійні пристрої. Клавіатура, комп'ютерна миша, колонки.

2. Програмне забезпечення:

- Ігровий двигун. Платформа для розробки та запуску гри (Unity).
- Навчальні модулі. Програмні компоненти, які містять навчальні матеріали, вправи та ігрові елементи.

3. Інженерні системи:

- Електроживлення. Надійне електропостачання з резервними джерелами живлення для уникнення перерв у роботі.
- Охолодження та вентиляція. Системи для підтримання оптимальної температури та запобігання перегріву обладнання.

4. Підтримка та обслуговування:

- Технічна підтримка. Спеціаліст для вирішення технічних проблем та допомоги користувачам.
- Оновлення програмного забезпечення. Регулярні оновлення гри для додавання нових функцій, виправлення помилок та забезпечення безпеки.

3.3 Програмне забезпечення

Стандартний набір технологій і структура програмного забезпечення, які використовуються для реалізації навчальної комп'ютерної гри на платформі Unity без мережевого підключення, ґрунтуються на використанні середовища розробки Unity для створення гри та мови програмування C# для реалізації ігрової логіки.

Щодо операційної системи, для запуску цієї навчальної комп'ютерної гри може використовуватися будь-яка версія Windows, починаючи з Windows 7 і вище.

Оскільки це гра, інтерфейс користувача реалізований безпосередньо в Unity, використовуючи графічні ресурси та інші елементи інтерфейсу, що надаються цим середовищем розробки.

Системне програмне забезпечення.

До складу системного програмного забезпечення входять:

- Операційна система Windows 10, або Windows 11
- Драйвери пристроїв (Програми, що дозволяють операційній системі взаємодіяти з апаратним забезпеченням, наприклад, драйвери для відеокарт)

Всі пакети програми та компоненти які використовувались в кваліфікаційному бакалаврському проєкті наведено в повному об'ємі в додатках (див додаток А-Т).

3.4 Результати реалізації навчальної ігрової системи

Результатом моєї кваліфікаційної бакалаврської роботи є готовий прототип, або ж демо версія навчальної ігрової системи, для розвитку дітей. Даний кваліфікаційний бакалаврський проєкт не є повністю завершеним, оскільки в перспективі він може постійно поновлюватись новими предметами, новими темами уроків, новими методиками навчання і тому подібне.

В кваліфікаційному бакалаврському проєкті було розглянуто вже існуючі приклади навчальних ігрових систем та реалізовано схожий за метою продукт. Були реалізовані наступні рішення.

Пересування персонажа, що здійснюється за допомогою звичних користувачеві клавіш, а також має в своєму арсеналі переміщення по всім 4 осям. Пересування по сцені є однією з основних механік ігрової системи, яка допомагає взаємодіяти з ігровим оточенням.

Взаємодія з оточенням за допомогою об'єктів на які можна натиснути. На ігровій сцені розміщені клікабельні об'єкти, або ж об'єкти на які користувач може натиснути. Після натиснення користувачем на такий об'єкт, спрацьовує система подій, яка запускає низку процесів. На різних об'єктах різні результати після натиснення, на одних просто пояснення через діалогове вікно, на інших поява якихось нових об'єктів і тому подібне. Дана механіка дозволяє зробити ігровий процес більш інтерактивним та цікавим для користувача, тим самим тримаючи його увагу на навчальному матеріалі, який надається на рівні.

Пояснення матеріалу користувачеві за допомогою діалогових вікон. В процесі проходження рівня, користувач буде бачити діалогові вікна, за допомогою яких, персонаж, наче веде діалог з гравцем. Подібні вікна використовуються для інформування користувача з приводу ігрового процесу, або надаються таким чином підказки для проходження рівня, а також вони використовуються для того аби зробити процес гри більш цікавим та живим, оскільки діалогові репліки прописані простою мовою, з жартами та легкою атмосферою. За допомогою подібного прийому, користувач не буде відчувати тиску, який часто відчувається при навчальному процесі з вчителем. Часто вчителі говорять по книзі, або важкими термінами, вважаючи що школярі мають їх розуміти. Але це створює лише ефект прірви, коли діти не розуміють все що каже їм дорослий і в них складається відчуття що вчитель веде себе зверхньо. Це звісно не стосується всіх вчителів, але подібні прийоми дозволять користувачам відчувати більш дружню атмосферу під час вивчення матеріалу, що може сприяти ефективнішому освітньому процесу.

Цікавий і невеликий ігровий сюжет для демонстрації потенціалу кваліфікаційного бакалаврського проєкту. Оскільки навчальна ігрова система для розвитку дітей несе просвітницький характер, то вона не має бути безцільна, але разом з тим проходження рівня має бути цікавим та захопливим. Також, не мало важливою деталлю є те, що подібна ігрова система не має викликати ігрової залежності у користувача, а тому, було прийнято рішення, робити сюжет рівня цікавим, але коротким, таким щоб тримав користувача в матеріалі, але щоб не викликав залежність.

Однією з причин, через яку ігрова система є ефективним способом навчання, це моментальна віддача. Користувач бачить результат своїх дій моментально. Частіше за все, після виконання якоїсь роботи, або після вивчення певного матеріалу, необхідний час. Час поки вчитель перевірить роботу, час коли класові дадуть перевірочну роботу і тому подібне. А дуже часто стається і таке, що матеріал який дитина вивчила, так ніяк і не перевірявся. Кожна людина має відчувати віддачу, від свого прогресу, у дітей це має бути в більш короткий час, оскільки вони все ще не розуміються значимість знань, які в них вкладають. Саме тому, ігрові системи, які тут же і дають віддачу користувачеві, є ефективними. Гравцю банально хочеться далі виконувати завдання, йому починає подобатись процес, оскільки він тут поробив різні завдання, почитав трохи, пройшов рівень, тут же отримує оцінку, в будь якому вигляді, своїм знанням.

Даний проєкт є прототипом та демонстрацією можливостей і перспективи навчальних комп'ютерних ігор. В ньому реалізовані всі основні механіки, а також я легко поміняти, або модернізувати, в залежності від теми уроку та матеріалів, оскільки вся логіка прописана, то є можливість змінити тему уроку яка б підходила під створенні умови та замінити на інший матеріал.

Реалізована навчальна ігрова система для розвитку дітей має невеликий, але потрібний функціонал. В системі присутні такі можливості :

- Головне меню, з якого і починається гра і з якого можна перейти до самого проходження рівня

- Наповнена сцена гри, яка має дизайн, що змушує користувача проходити гру з певними зусиллями

- Пересування персонажа

- Об'єкти на які користувач може натиснути

- Діалогові вікна, з поясненнями матеріалу, або підказками до проходження рівня.

- Можливість зупинити гру

- Здоров'я персонажа, оскільки в грі присутній елемент з пастками, щоб користувачеві було цікавіше проходити рівень, то в системі також реалізований і функціонал здоров'я з індикатором що змінюється, в залежності від кількості здоров'я.

Весь пророблений функціонал, разом в сукупності, створює прототип навчальної ігрової системи для розвитку дітей. В системі відсутні насилля та жорстокість. В грі є елемент нанесення шкоди персонажу, шляхом його зіткнення з пастками, але відсутня така як «смерть» персонажа. Система з очками здоров'я, була реалізована для, так званого, підігріву цікавості користувача, щоб користувач відчував певний виклик в проходженні гри. Разом з тим, коли позначка здоров'я персонажа досягає нуля, гра не закінчується, персонажа переносить на початок проходження рівня з пастками, а також з'являється діалогове вікно, в якому персонаж закликає гравця спробувати ще раз, а рівень здоров'я знову поповнюється. Такий елемент і не є жорстоким і вмотивовує гравця проходити рівень далі.

Приклад проходження рівня, для розуміння того, як влаштована система, та навчальний елемент в даному кваліфікаційному бакалаврському проєкті продемонстровано в додатку У.

ВИСНОВКИ

У ході виконання кваліфікаційного бакалаврського проєкту було проведено дослідження та проаналізовано існуючі платформи для навчання дітей у формі комп'ютерної гри і реалізовано власний прототип для демонстрації можливостей і перспектив в цій сфері.

Також був проведений аналіз реалізованого продукту і доцільність використання саме даних технологій в цьому кваліфікаційному бакалаврському проєкті. Розроблена інформаційна система надає найголовніше – поєднання навчального елемента з комп'ютерними технологіями та використанням ігрового моменту як способу полегшення сприйняття матеріалу.

Було реалізовано навчальну ігрову систему для розвитку дітей, за допомогою продукту Unity, де будувалась гра та Visual Studio, де прописувалась логіка проєкту.

Також було визначено характеристики, переваги і педагогічні аспекти існуючих продуктів схожого напрямлення, окрім цього було визначено необхідні характеристики системного забезпечення даної роботи.

За допомогою розробленого продукту, можна в майбутньому спрощувати навчання дітей інтегруючи такі ІС в освітню систему. За допомогою мови C# було реалізовано логіку та механіку гри.

Розроблена зручна і зрозуміла, для сучасного покоління, система взаємодії з ігровим оточенням. Всі основні можливості обраної ІС було реалізовано в розробленому прототипі.

Наведена вище розробка навчальної ігрової системи для розвитку дітей, дасть змогу спростити навчання для школярів, покращить сприйняття ними інформації та може вмотивувати їх до подальшого вивчення представлених, в процесі проходження гри, наук. Розробка пропонує користувачеві зрозумілий та звичний спосіб взаємодії з ігровим оточенням, інформування та вивчення навчальних матеріалів в спрощеному вигляді і за допомогою діалогу персонажа з гравцем, а також за допомогою візуальних об'єктів. Дослідження показують,

що включення подібних освітніх ігрових систем до навчальної програми може значно підвищити залученість і успішність учнів.

Використання ігрових методів навчання не тільки робить процес навчання цікавішим, але й допомагає дітям краще засвоювати матеріал через інтерактивний діалог та емоційну активність.

У майбутньому планується подальший розвиток прототипу, зокрема додавання нових навчальних модулів, розширення його функціональності та оптимізація поведінки системи на різних платформах.

Також планується проводити додаткові дослідження для оцінки ефективності розробленої системи при її використанні в реальних освітніх умовах.

Таким чином, розроблена навчальна ігрова система для розвитку дітей має великий потенціал для вдосконалення навчального процесу і стає важливим інструментом в арсеналі сучасного вчителя, допомагаючи дітям не тільки вчитися, але й покращувати процес навчання.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

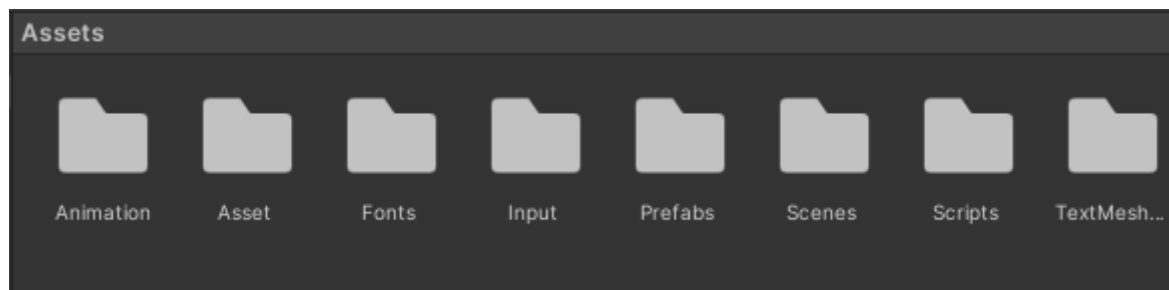
1. CodeCombat [Електронний ресурс] – Режим доступу до ресурсу: <https://codecombat.com>. Дата звернення: 15.03.2024.
2. CodeCombat wiki [Електронний ресурс] – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/CodeCombat>. Дата звернення: 15.03.2024.
3. Scratch [Електронний ресурс] – Режим доступу до ресурсу: <https://scratch.mit.edu>. Дата звернення: 15.03.2024.
4. Scratch for education [Електронний ресурс] – Режим доступу до ресурсу: <https://cobie.io/why-use-scratch-for-education/>. Дата звернення: 15.03.2024.
5. Семчук С. І. КОМП'ЮТЕРНІ ІГРИ ТА ЇХ ВИКОРИСТАННЯ В ПЕДАГОГІЧНОМУ ПРОЦЕСІ ДНЗ / С. І. Семчук. Дата звернення: 17.04.2024.
6. Unity [Електронний ресурс] – Режим доступу до ресурсу: <https://unity.com>. Дата звернення: 10.02.2024.
7. Visual Studio [Електронний ресурс] – Режим доступу до ресурсу: <https://visualstudio.microsoft.com>. Дата звернення: 15.02.2024.
8. About Visual Studio [Електронний ресурс] – Режим доступу до ресурсу: <https://learn.microsoft.com/en-us/visualstudio/ide/?view=vs-2022>. Дата звернення: 15.02.2024.
9. Why Unity ? [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/@juris.savos/a-few-reasons-to-learn-unity-in-2024-45cbc981db64>. Дата звернення: 10.02.2024.
10. Creating scenes unity [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.unity3d.com/560/Documentation/Manual/CreatingScenes.html>. Дата звернення: 15.03.2024.
11. Animator unity [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.unity3d.com/Manual/class-Animator.html> . Дата звернення: 15.03.2024.

12. Packages unity [Електронний ресурс] – Режим доступу до ресурсу:
<https://docs.unity3d.com/Manual/Packages.html>. Дата звернення:
15.03.2024.
13. Tilemap palette unity [Електронний ресурс] – Режим доступу до ресурсу:
<https://docs.unity3d.com/Manual/Tilemap-Palette.html>. Дата звернення:
15.03.2024.
14. Animation unity [Електронний ресурс] – Режим доступу до ресурсу:
<https://docs.unity3d.com/Manual/AnimationSection.html>. Дата звернення:
15.03.2024.
15. Console unity [Електронний ресурс] – Режим доступу до ресурсу:
<https://docs.unity3d.com/Manual/Console.html>. Дата звернення:
15.03.2024.
16. Inspector unity [Електронний ресурс] – Режим доступу до ресурсу:
<https://docs.unity3d.com/Manual/UsingTheInspector.html>. Дата звернення:
15.03.2024.
17. Components unity [Електронний ресурс] – Режим доступу до ресурсу:
<https://docs.unity3d.com/Manual/UsingComponents.html>. Дата звернення:
15.03.2024.
18. Transform unity [Електронний ресурс] – Режим доступу до ресурсу:
<https://docs.unity3d.com/Manual/class-Transform.html>. Дата звернення:
15.03.2024.
19. Assetstore for unity [Електронний ресурс] – Режим доступу до ресурсу:
<https://assetstore.unity.com/2d>. Дата звернення: 15.03.2024.
20. Assets for games [Електронний ресурс] – Режим доступу до ресурсу:
<https://itch.io>. Дата звернення: 15.03.2024.
21. Навчальні ігри [Електронний ресурс] – Режим доступу до ресурсу:
<https://babyplus.ua/ua/pazly-i-razvivayushchie-igry-igry-dlya-obucheniya>.
Дата звернення: 05.04.2024.

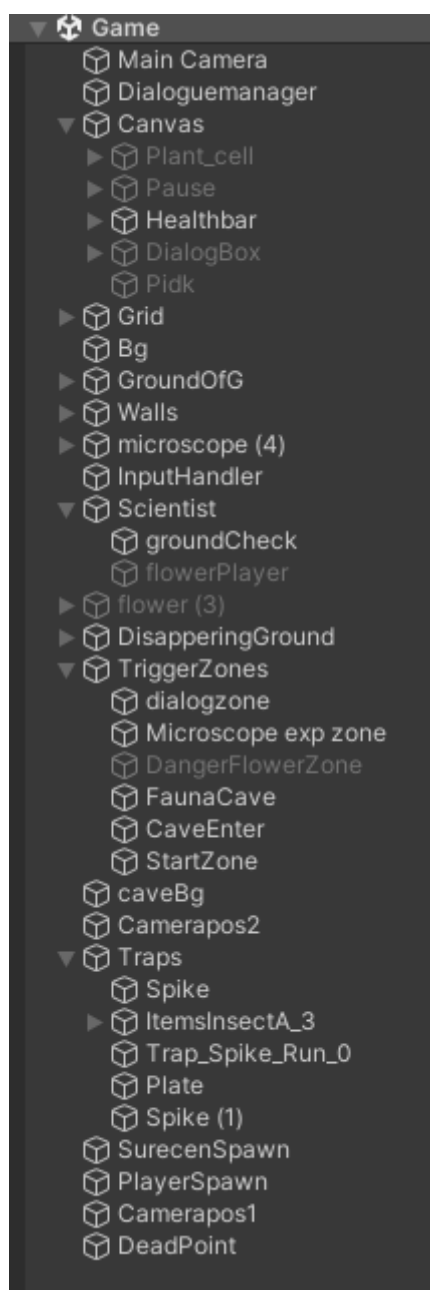
22. Вплив навчальних ігор для дітей [Електронний ресурс] – Режим доступу до ресурсу: <https://nus.org.ua/view/yak-navchannya-na-osnovi-igor-rozvyva-navychky-21-go-stolittya/>. Дата звернення: 05.04.2024.
23. Educational games and toys for kids: Are they effective? [Електронний ресурс] – Режим доступу до ресурсу: <https://parentingscience.com/educational-games-for-kids/>. Дата звернення: 02.04.2024.
24. Application of the Educational Game to Enhance Student Learning [Електронний ресурс] – Режим доступу до ресурсу: <https://www.frontiersin.org/articles/10.3389/feduc.2021.623793/full>. Дата звернення: 02.04.2024.
25. Gene Luen Yang. SECRET CODERS / Gene Luen Yang., 2015. Дата звернення: 02.05.2024.
26. Norton T. Learning C# by developing games in Unity / Terry Norton. Дата звернення: 10.05.2024.

ДОДАТКИ

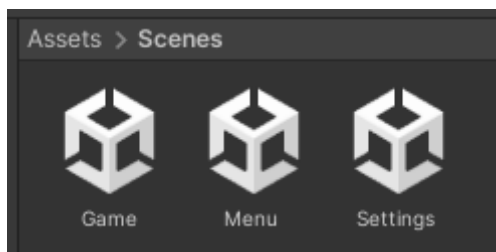
Додаток А. Всі наявні в проєкті компоненти



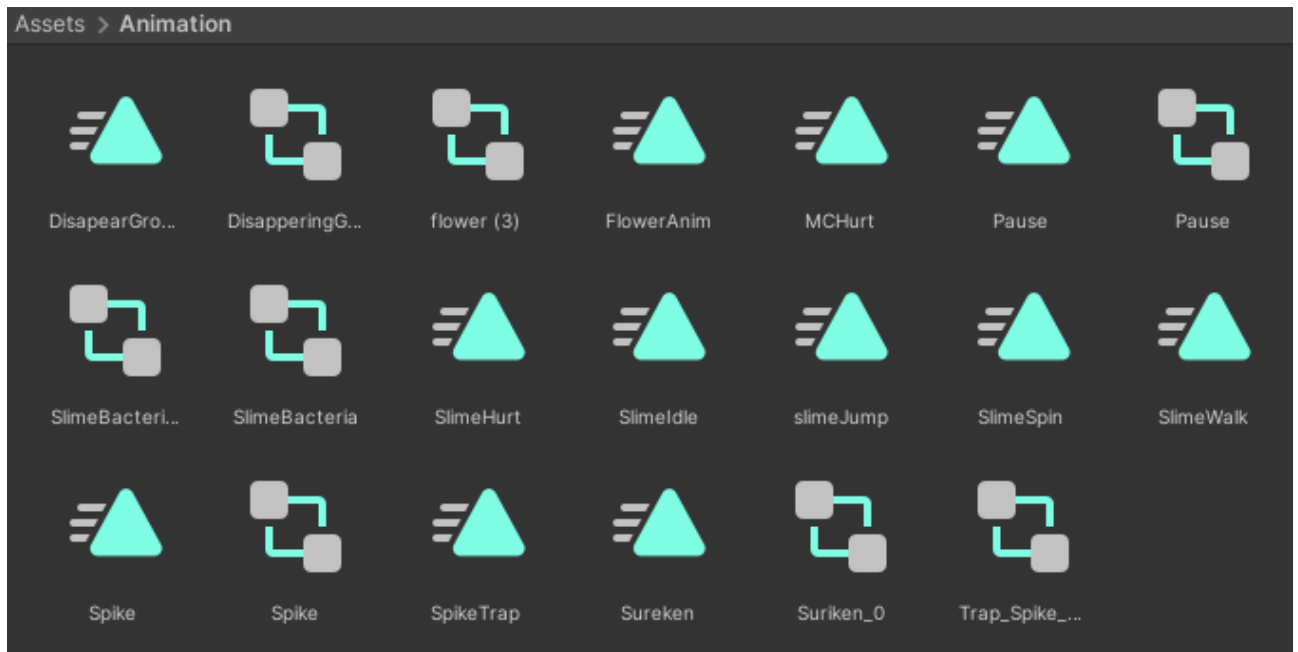
Додаток Б. Ієрархія ігрових об'єктів в проєкті



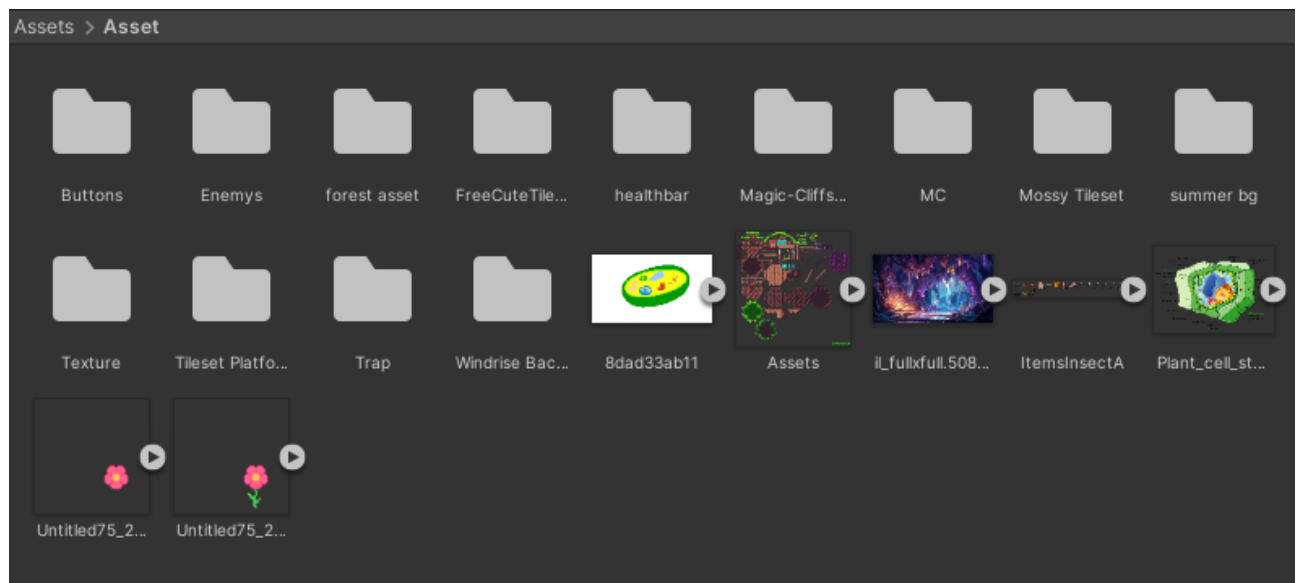
Додаток В. Всі сцени в проєкті



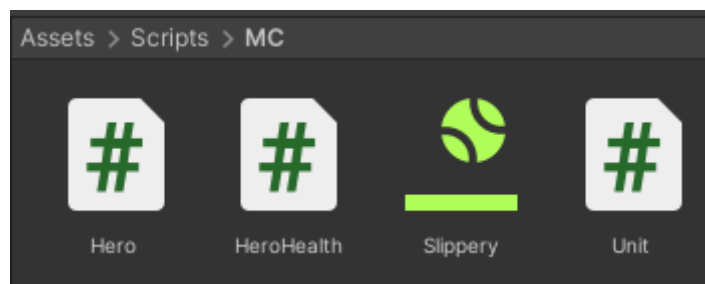
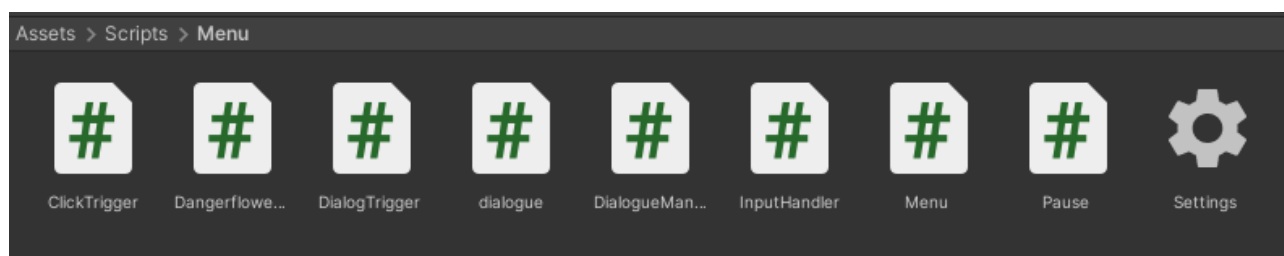
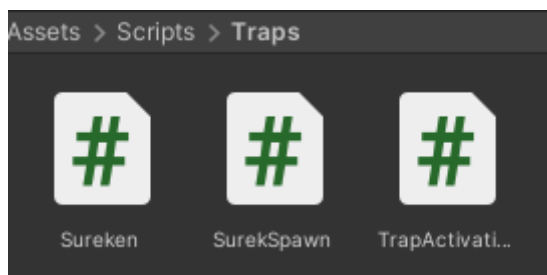
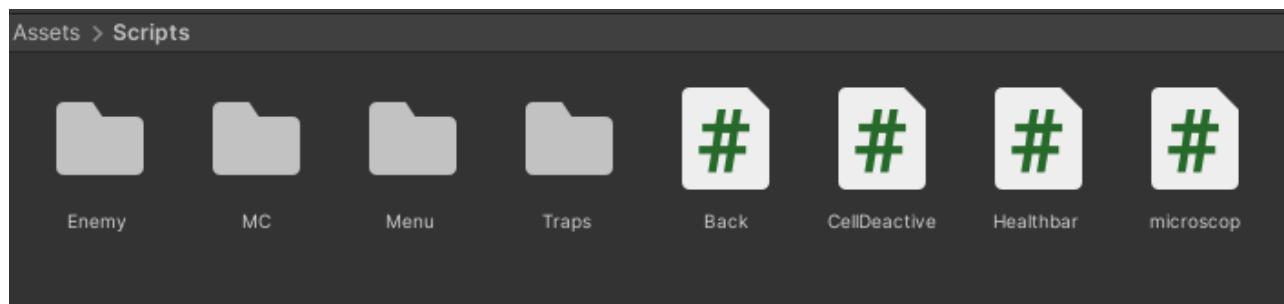
Додаток Г. Анімації наявні в навчальній ігровій системі



Додаток Д. Всі асети, що використовуються в проєкті



Додаток Е. Всі файли кодів в проєкті



Додаток Ж. Система подій та перевірки натиснення на ігрові об'єкти

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.InputSystem;
using UnityEngine.Events;
using UnityEngine.VisualScripting;

public class InputHandler : MonoBehaviour
{
    private Camera _maincamera;

    public UnityEvent microclick;
    public UnityEvent flowerclick;
    public UnityEvent Endgame;

    public GameObject player;
    public GameObject microscope;
    public GameObject flower;
    public GameObject flowerPlayer;
    public GameObject camerapos2;
    public GameObject FaunaCave;
    public GameObject Cell;

    private void Update()
    {
        float distance = Vector3.Distance(player.transform.position, microscope.transform.position);
        if (player.transform.position.y <= -207)
        {
            _maincamera.transform.position = camerapos2.transform.position;
        }
    }

    private void Awake()
    {
        _maincamera = Camera.main;
    }

    public void OnClickMicroscope(InputAction.CallbackContext context)
    {
        if (!context.started) return;

        var rayHit = Physics2D.GetRayIntersection(_maincamera.ScreenPointToRay(Mouse.current.position.ReadValue()));
        if (!rayHit.collider) return;

        float distance = Vector3.Distance(player.transform.position, microscope.transform.position);

        if (rayHit.collider.gameObject.tag == "Microscope" && distance < 30f)
        {
            microclick.Invoke();

            if (!flowerPlayer.activeSelf)
            {
                flower.SetActive(true);
            }
            else
            {
                Endgame.Invoke();
            }
        }
    }
}
```

```

    }
}
public void OnClickFlower(InputAction.CallbackContext context)
{
    if (!context.started) return;

    var rayHit = Physics2D.GetRayIntersection(_maincamera.ScreenPointToRay(Mouse.current.position.ReadValue()));
    if (!rayHit.collider) return;

    float distance = Vector3.Distance(player.transform.position, flower.transform.position);

    if (rayHit.collider.gameObject.tag == "flower" && distance < 30f)
    {
        flowerclick.Invoke();
    }
}
public void OnClickFaunaCave(InputAction.CallbackContext context)
{
    if (!context.started) return;

    var rayHit = Physics2D.GetRayIntersection(_maincamera.ScreenPointToRay(Mouse.current.position.ReadValue()));
    if (!rayHit.collider) return;

    float distance = Vector3.Distance(player.transform.position, FaunaCave.transform.position);

    if (rayHit.collider.gameObject.tag == "FaunaCave" && distance < 30f)
    {
        FaunaCave.GetComponent<ClickTrigger>().Trigger();
    }
}
}

```

Додаток II. Код для пересування персонажа

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.EventSystems;

public class Hero : Unit
{
    private Rigidbody2D rb;

    public Transform groundCheck;
    public LayerMask groundMask;
    private float moveHorizontal;
    private bool isFacingRight;
    public bool flamer = false;
    public Animator animator;

    void Start ()
    {
        rb = GetComponent<Rigidbody2D>();
        speed = 40f;
        jumpforce = 185f;
    }

    void Update()
    {
        moveHorizontal = Input.GetAxisRaw("Horizontal");
    }
    private bool isGrounded()
    {
        return Physics2D.OverlapCircle(groundCheck.position, 0.2f, groundMask);
    }
    void FixedUpdate()
    {
        rb.velocity = new Vector2(moveHorizontal * speed, rb.velocity.y);
        animator.SetFloat("speed", Mathf.Abs(moveHorizontal));

        if (Input.GetButton("Jump") && isGrounded())
        {
            rb.velocity = new Vector2(rb.velocity.x, jumpforce);
        }

        if (isGrounded())
        {
            animator.SetBool("IsJumping", false);
        }
        else if (!isGrounded())
        {
            animator.SetBool("IsJumping", true);
        }

        if (moveHorizontal > 0f && !isFacingRight)
        {
            Flip();
        }
        if (moveHorizontal < 0f && isFacingRight)
        {
            Flip();
        }
    }
    private void Flip()
    {
        isFacingRight = !isFacingRight;
        transform.Rotate(0f,180f,0f);
    }
}
```

Додаток К. Код для здоров'я персонажа

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;

public class HeroHealth : MonoBehaviour
{
    public int Maxhealth = 4;
    public int currenthealth;
    public Healthbar healthBar;
    public Animator animator;
    public GameObject deadPointCave;
    public ClickTrigger trigger;

    void Start()
    {
        currenthealth = Maxhealth;
        healthBar.SetMaxHealth(Maxhealth);
    }

    public void TakeDamage(int damage)
    {
        //Take damage MC
        animator.SetTrigger("Hurt");
        currenthealth -= damage;
        healthBar.SetHealth(currenthealth);

        if (currenthealth <= 0)
        {
            gameObject.transform.position = deadPointCave.transform.position;
            trigger.Trigger();
            currenthealth = Maxhealth;
        }
    }
}
```

Додаток Л. Код для діалогового менеджера проєкту

```
using System;
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using TMPro;

public class DialogueManager : MonoBehaviour
{
    public Queue <string> sentences;
    public TextMeshProUGUI dialogueText;
    public GameObject dialogueBox;

    void Start()
    {
        sentences = new Queue <string>();
    }
    private void Update()
    {
        if (Input.GetKeyDown(KeyCode.F))
        {
            NextLine();
        }
    }
    public void StartDial(dialogue dialog)
    {
        sentences.Clear();
        foreach (string sentence in dialog.sentences)
        {
            sentences.Enqueue(sentence);
        }
        NextLine();
    }

    public void NextLine()
    {
        if (sentences.Count == 0)
        {
            dialogueBox.SetActive(false);
            Time.timeScale = 1f;
            return;
        }
        string sentence = sentences.Dequeue();
        dialogueText.text = sentence;
    }
}
```

Додаток М. Код для створення класу діалог, прописування реплік

```
using System.Collections;  
using System.Collections.Generic;  
using UnityEngine;
```

```
[System.Serializable]  
public class dialogue  
{  
    public string[] sentences;  
}
```

Додаток Н. Код для реагування діалогового вікна на тригер

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class DialogTrigger : MonoBehaviour
{
    public GameObject dial;
    public Dialogue dialog;
    public void TriggerDialogue()
    {
        FindObjectOfType<DialogueManager>().StartDial(dialog);
    }

    void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.gameObject.tag == "Hero")
        {
            dial.SetActive(true);
            Time.timeScale = 0f;
            TriggerDialogue();
            Destroy(gameObject);
        }
    }
}
```

Додаток П. Код для індикатора здоров'я персонажа

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;

public class Healthbar : MonoBehaviour
{
    public Slider slider;
    public Gradient gradient;
    public Image fill;

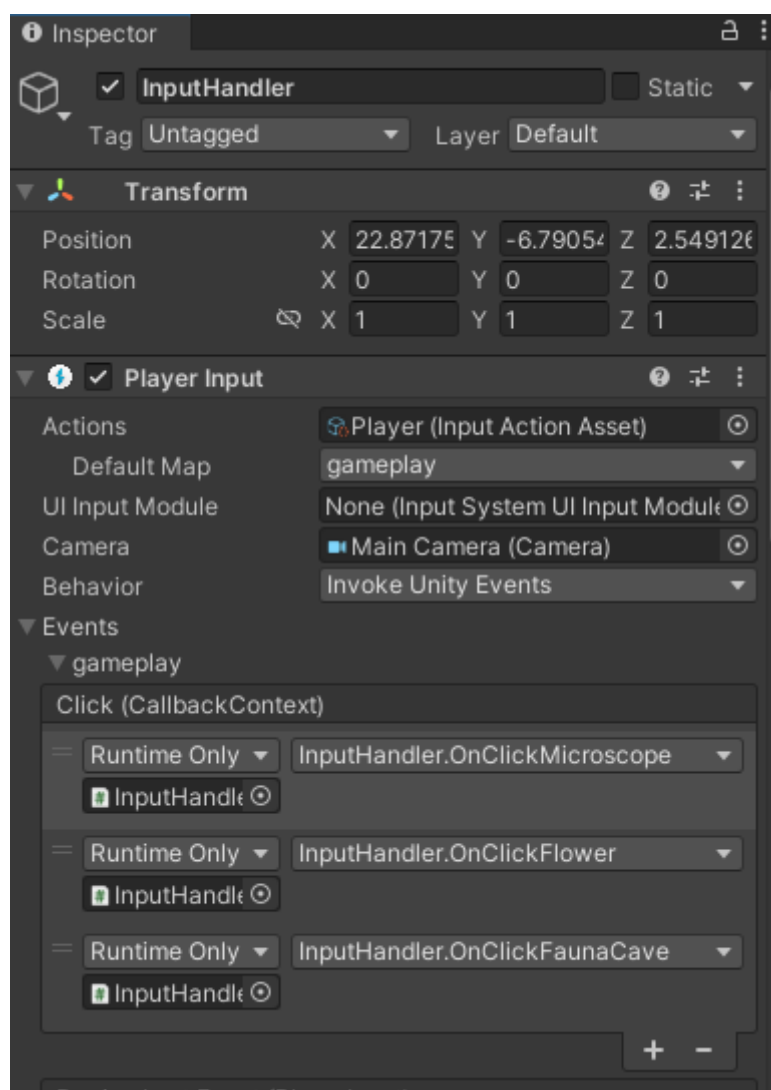
    public void SetMaxHealth(int health)
    {
        slider.maxValue = health;
        slider.value = health;

        fill.color = gradient.Evaluate(1f);
    }

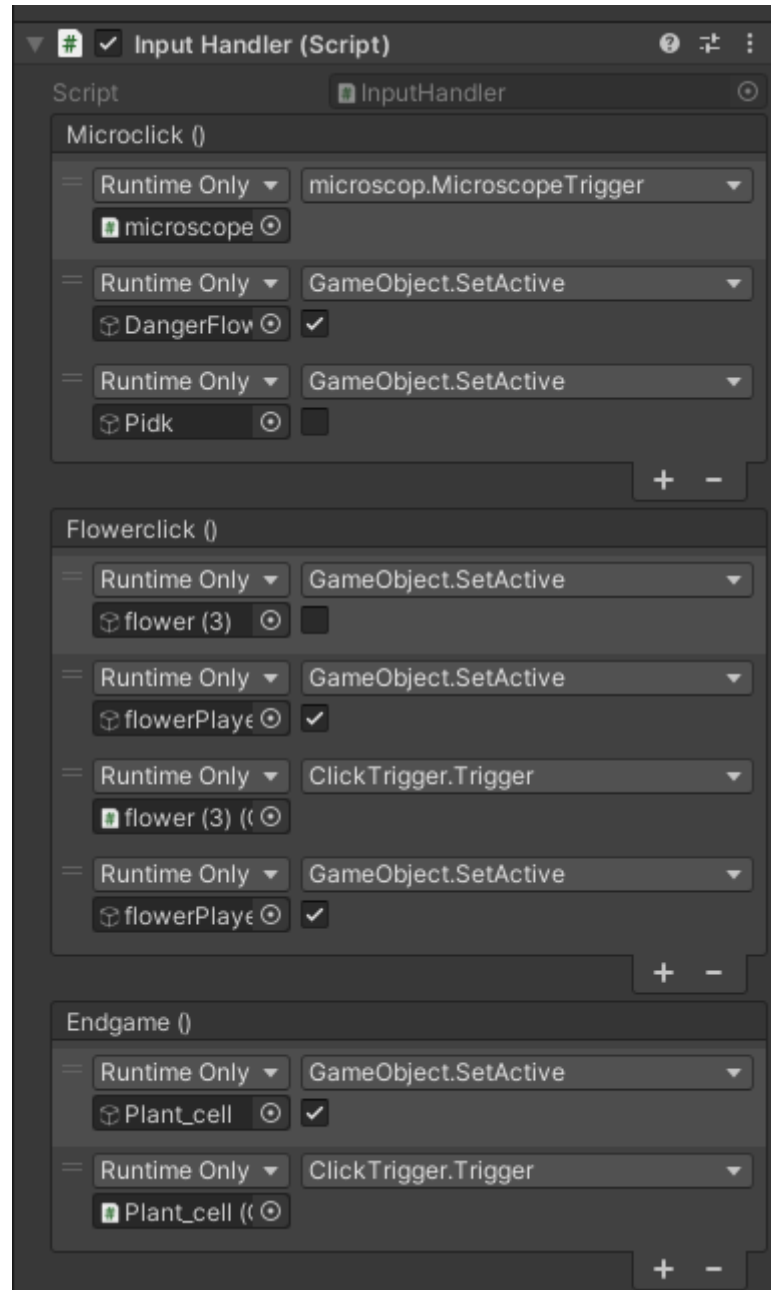
    public void SetHealth(int health)
    {
        slider.value = health;

        fill.color = gradient.Evaluate(slider.normalizedValue);
    }
}
```

Додаток Р. Виставлені функції в інспекторі на системі подій на клік



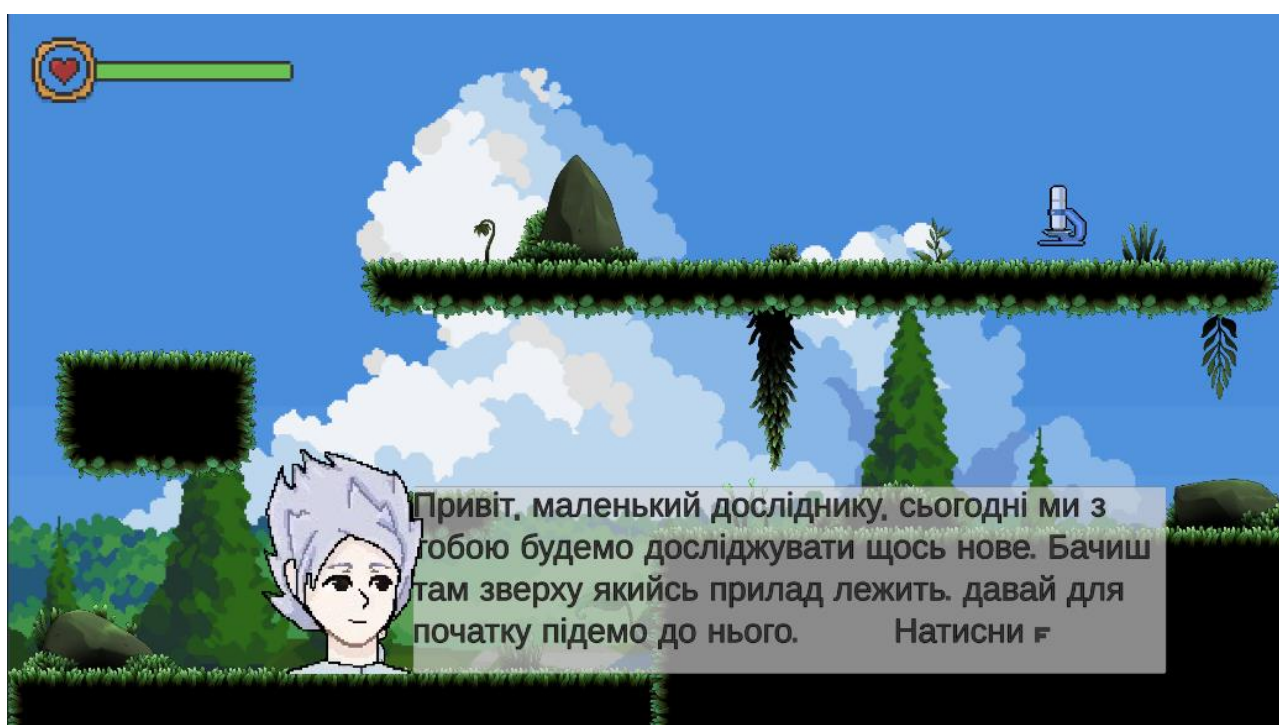
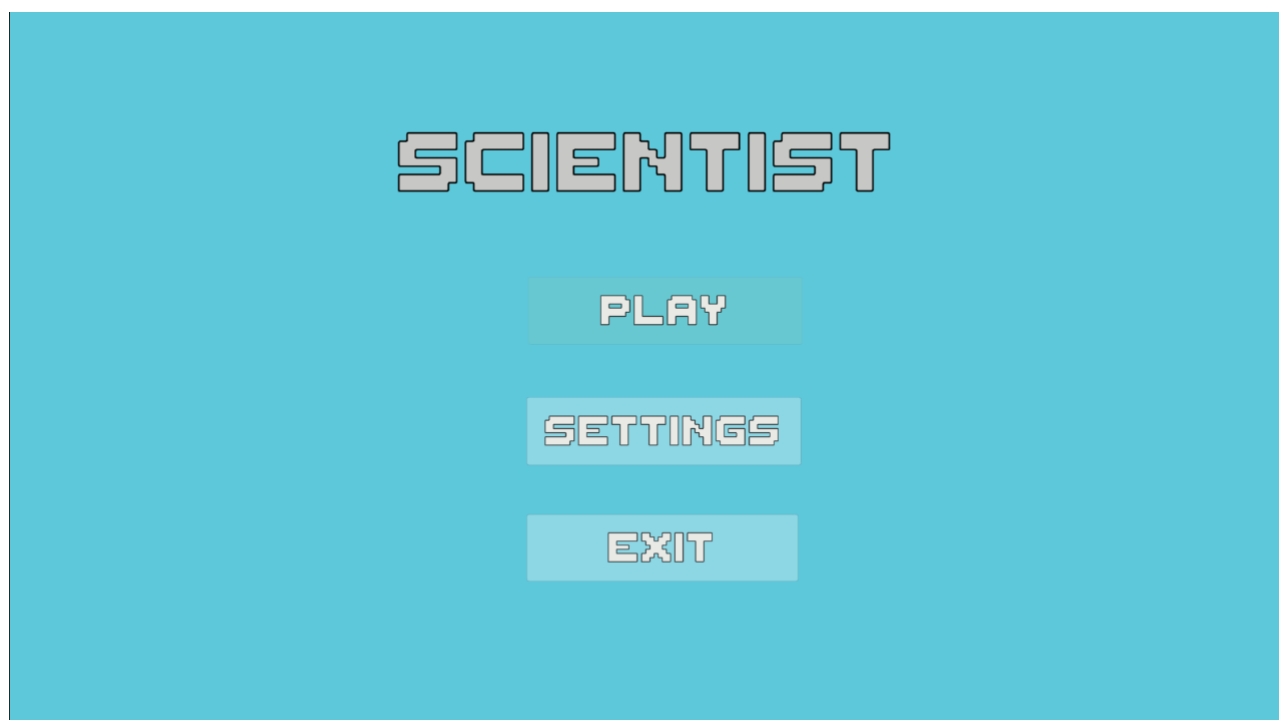
Додаток С. Виставлені функції подій на різні види кліків

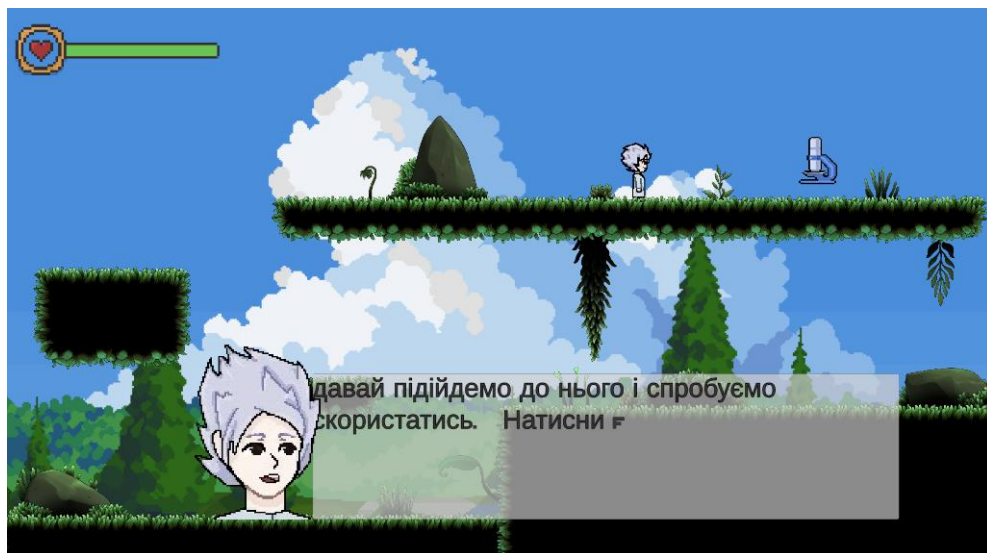
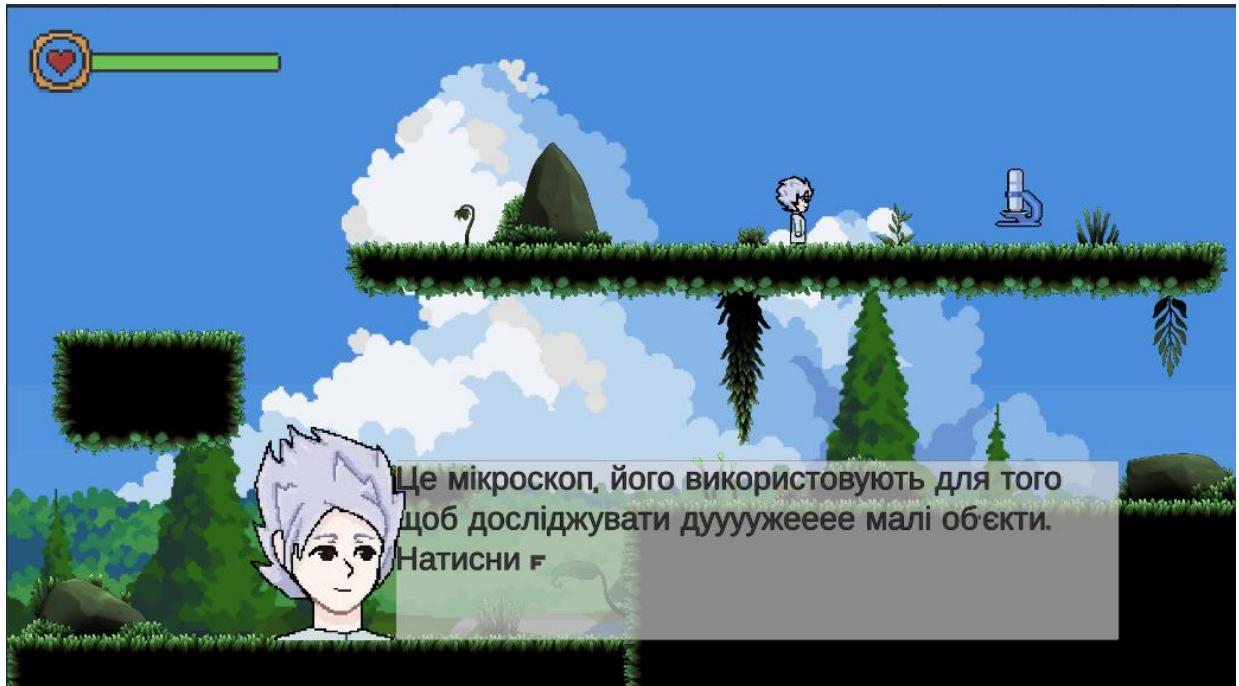


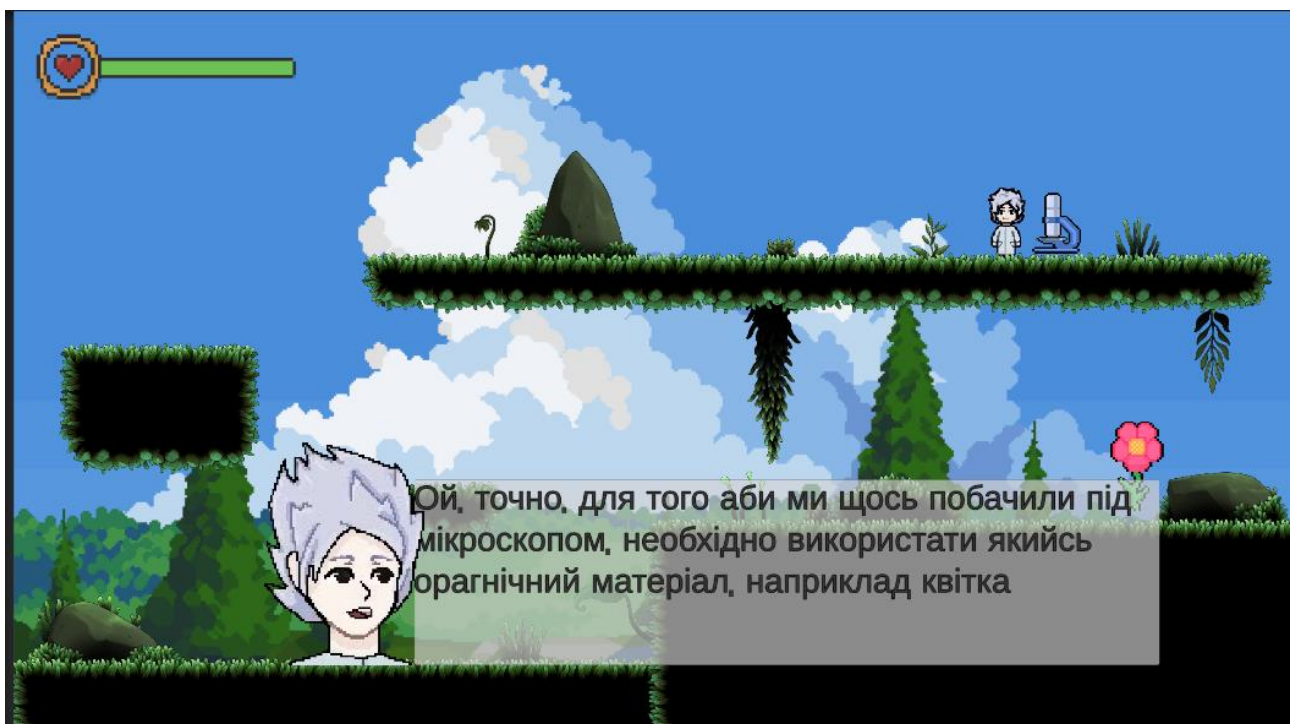
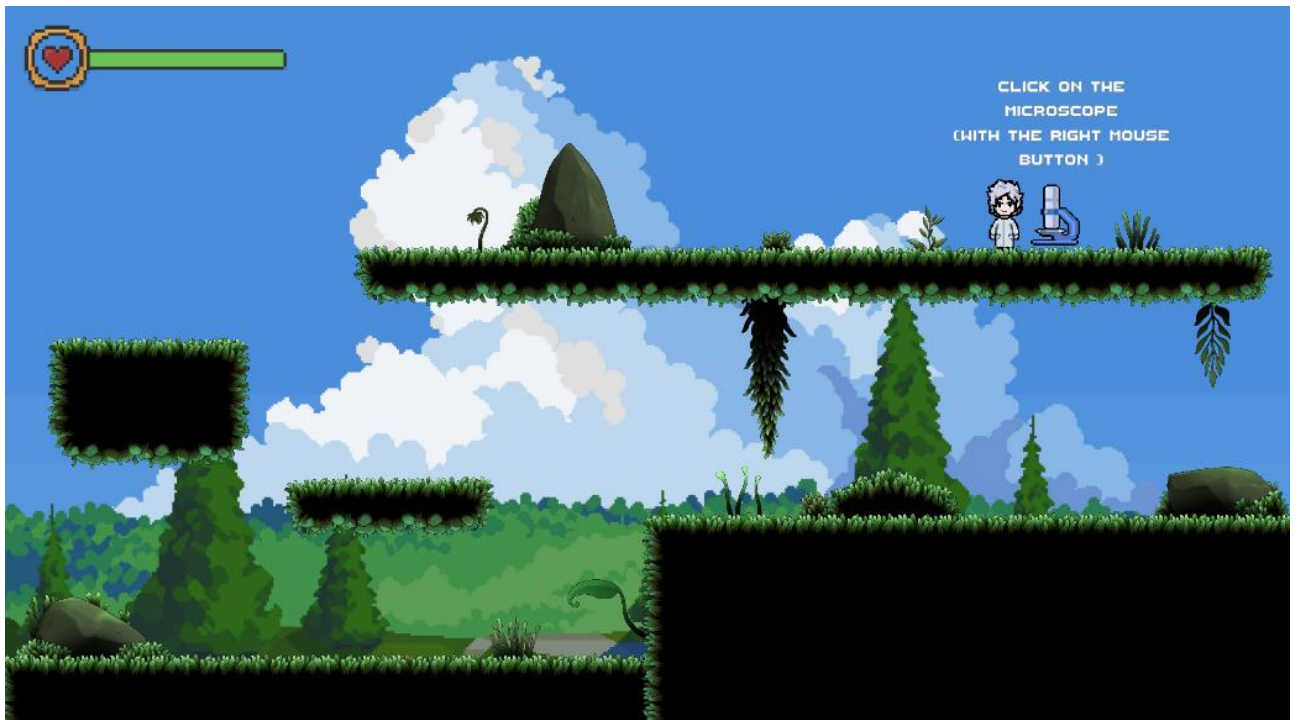
Додаток Т. Ігрові об'єкти, на які посилається система подій

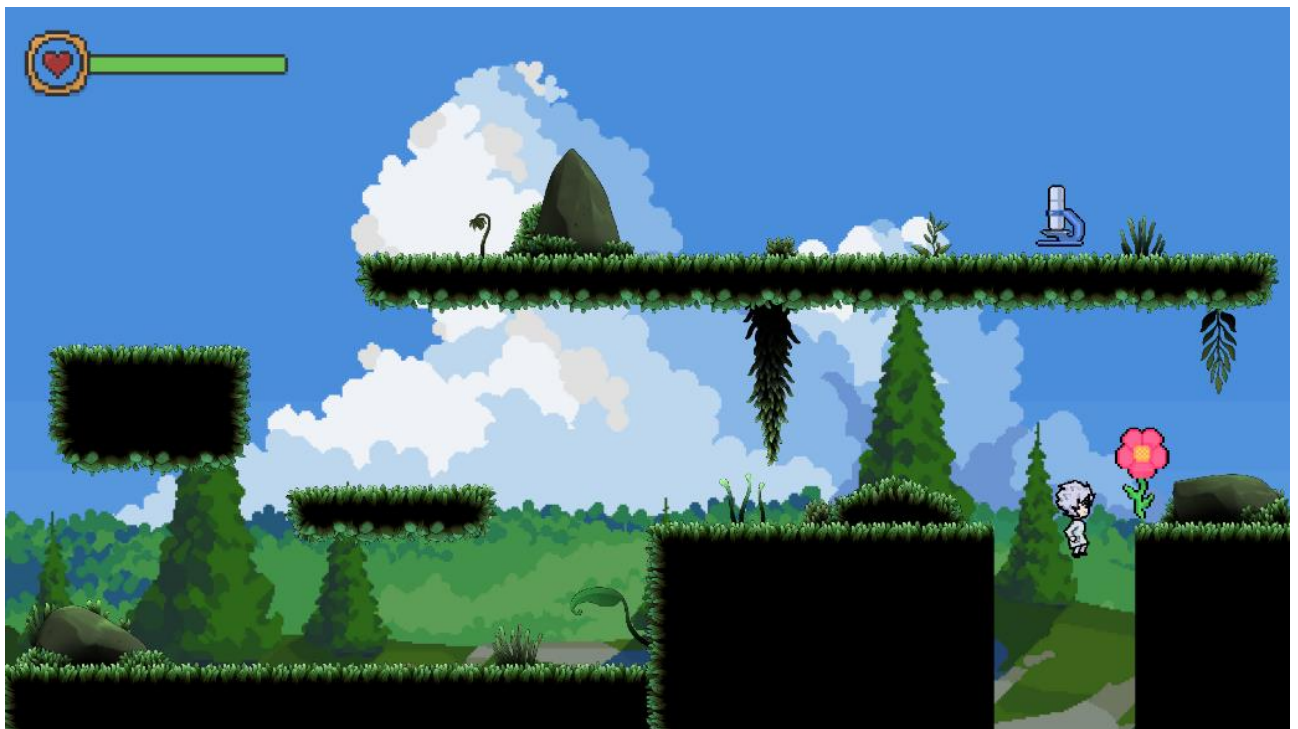
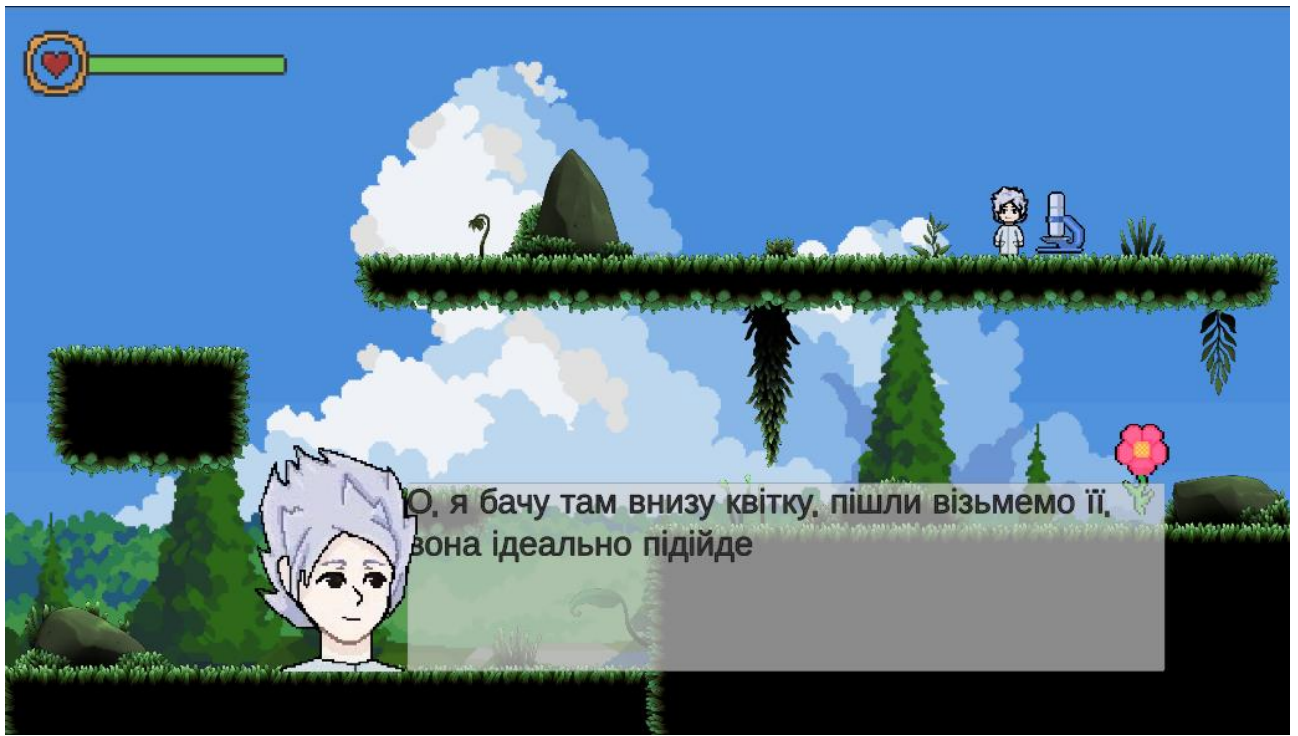
Player	Scientist
Microscope	microscope (4)
Flower	flower (3)
Flower Player	flowerPlayer
Cameras 2	Cameras2
Fauna Cave	FaunaCave
Cell	Plant_cell

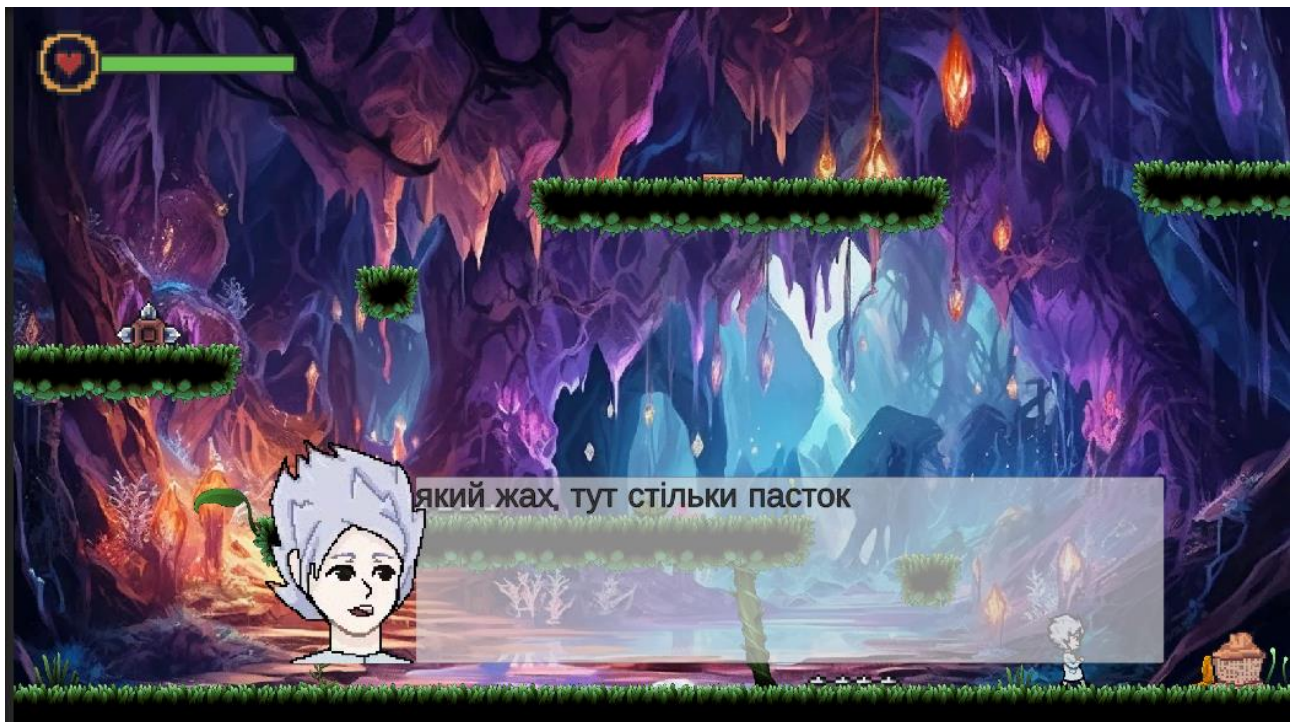
Додаток У. Проходження рівня

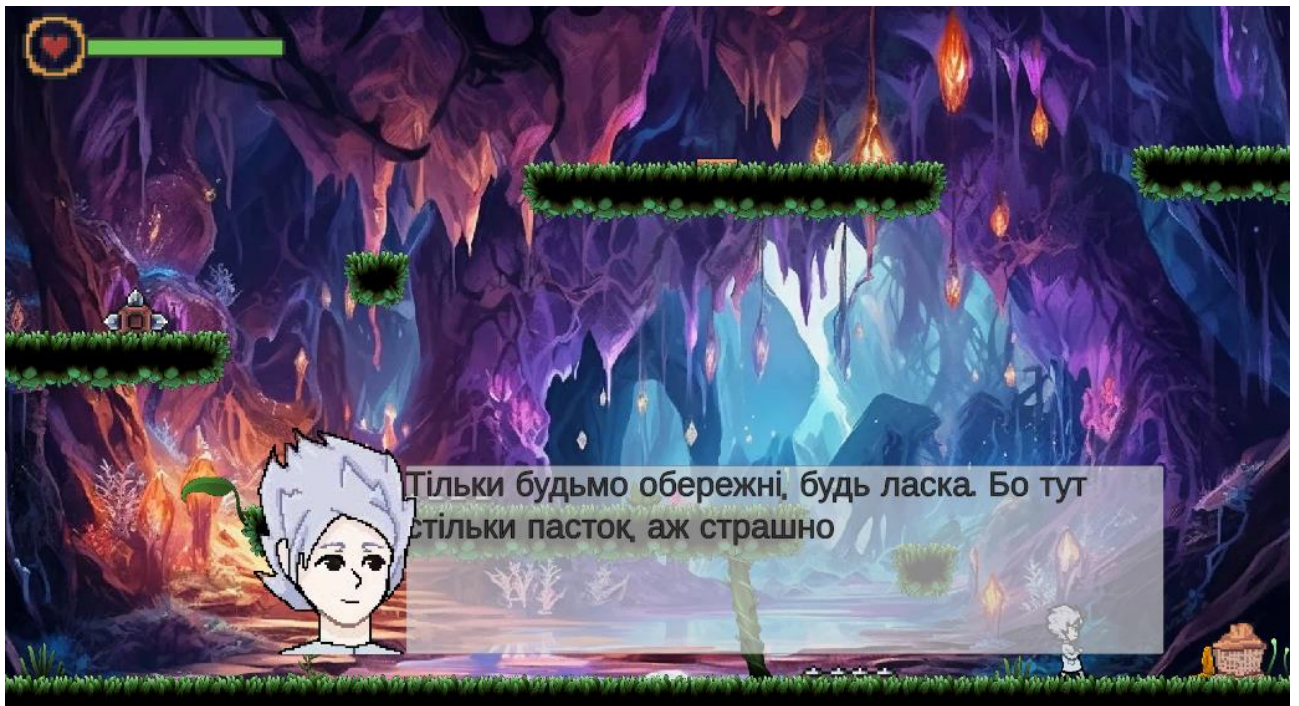
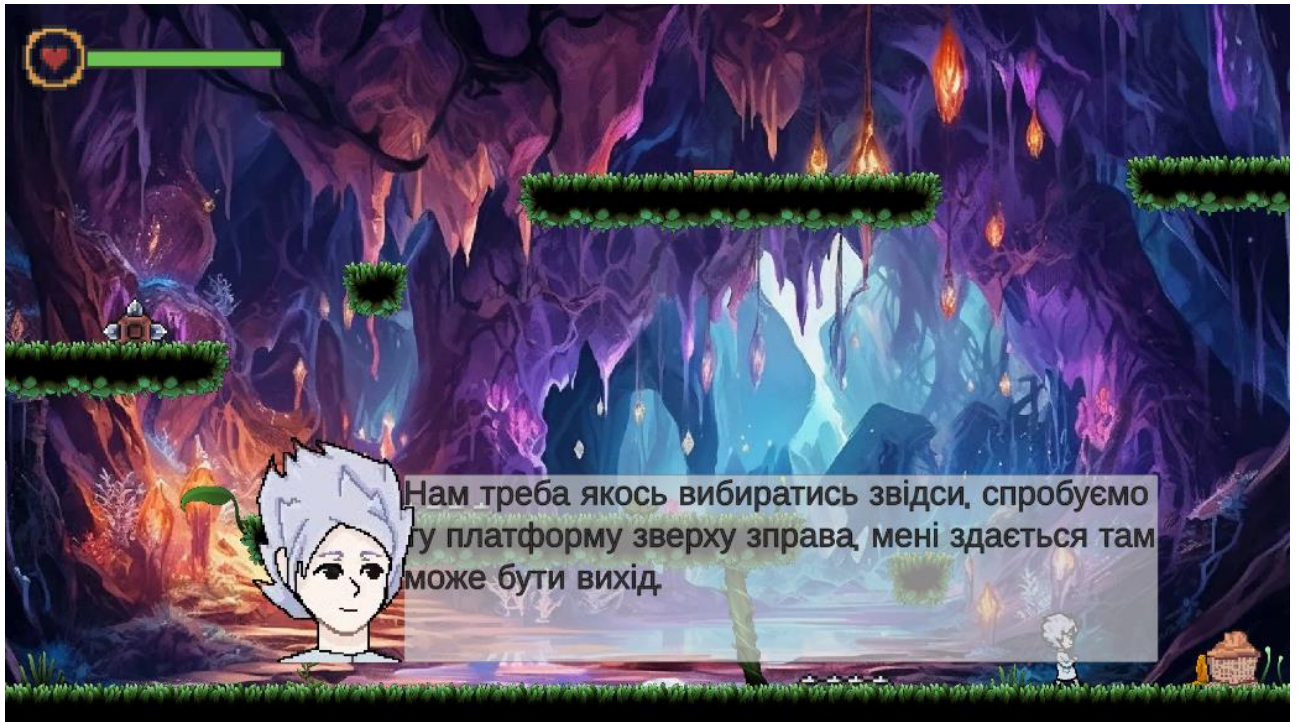


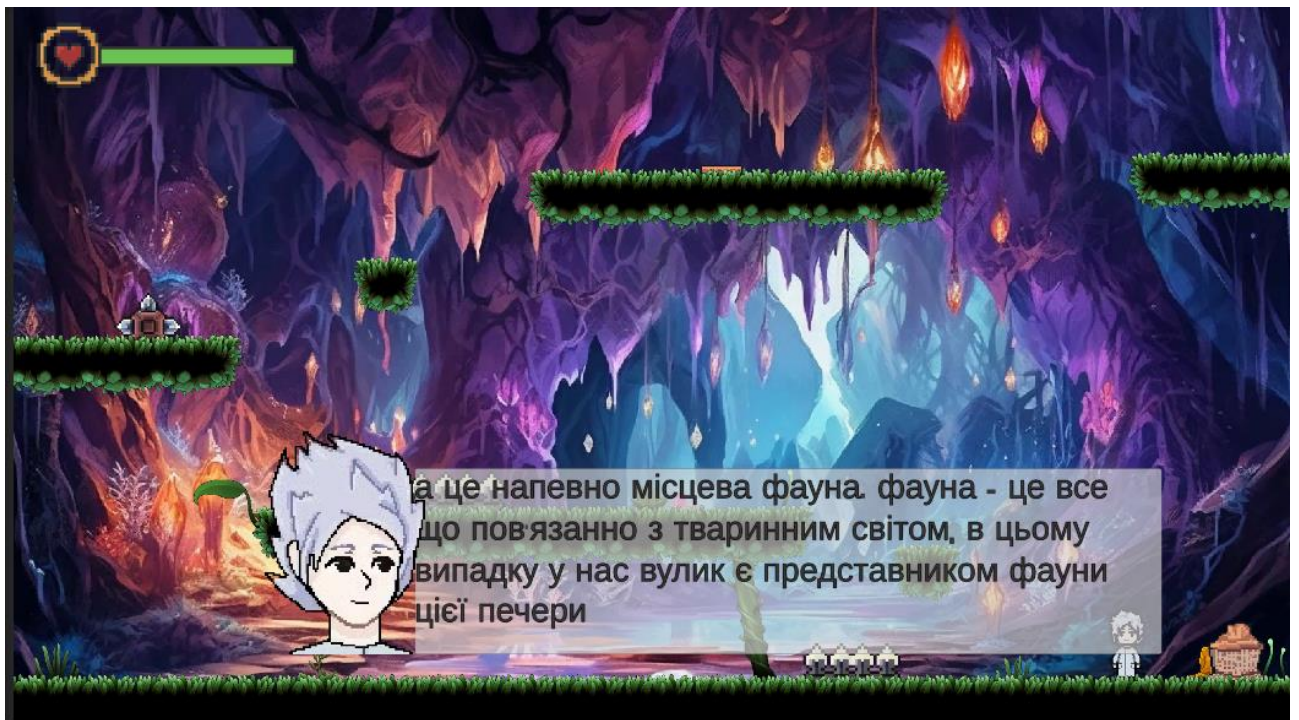
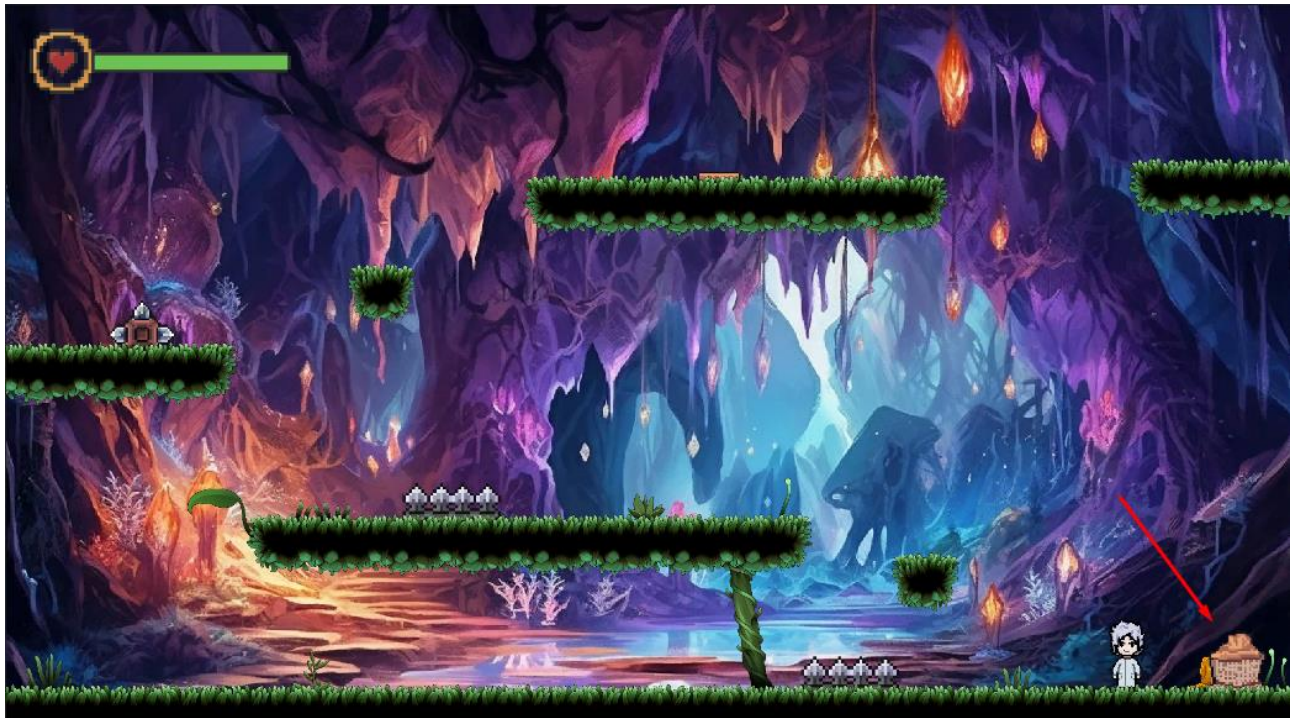




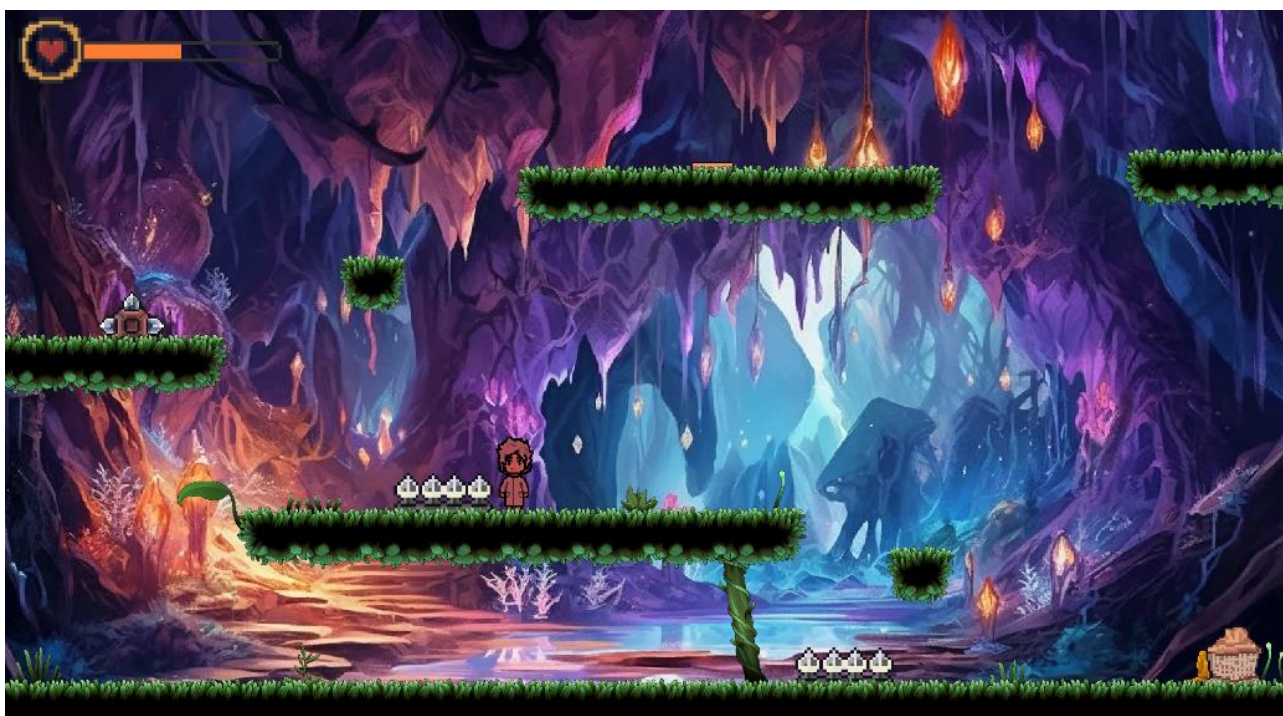
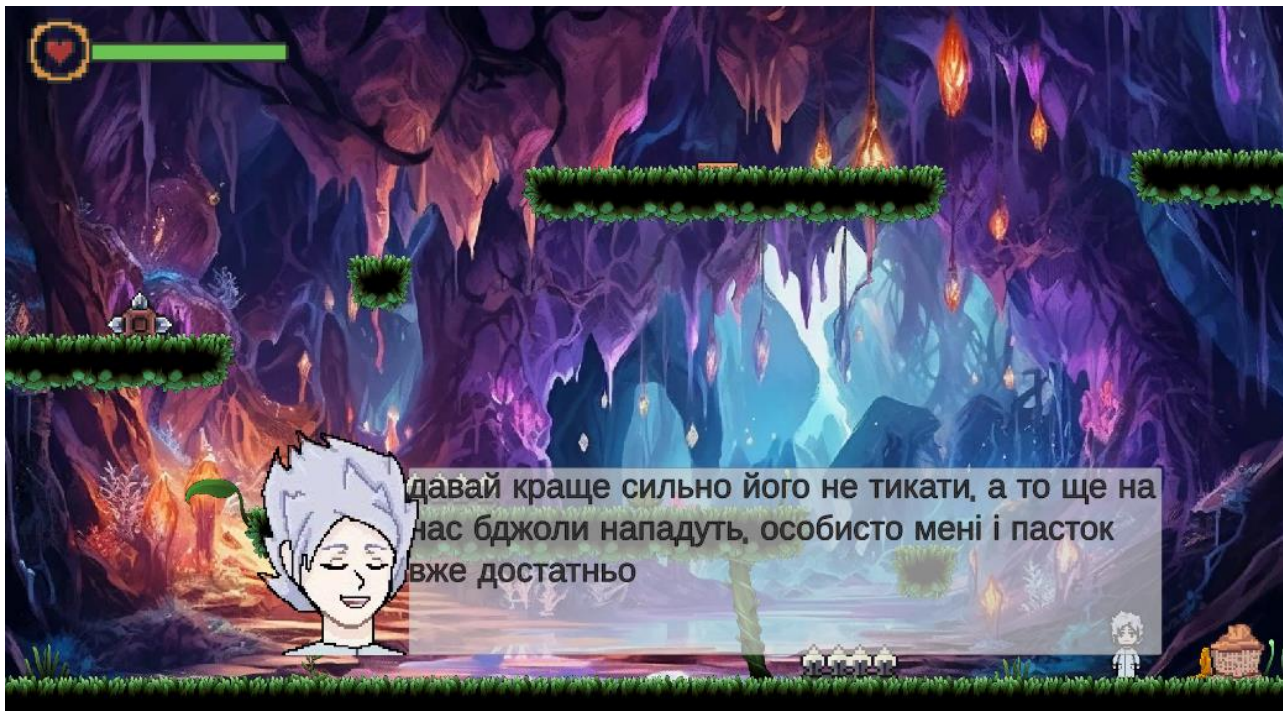


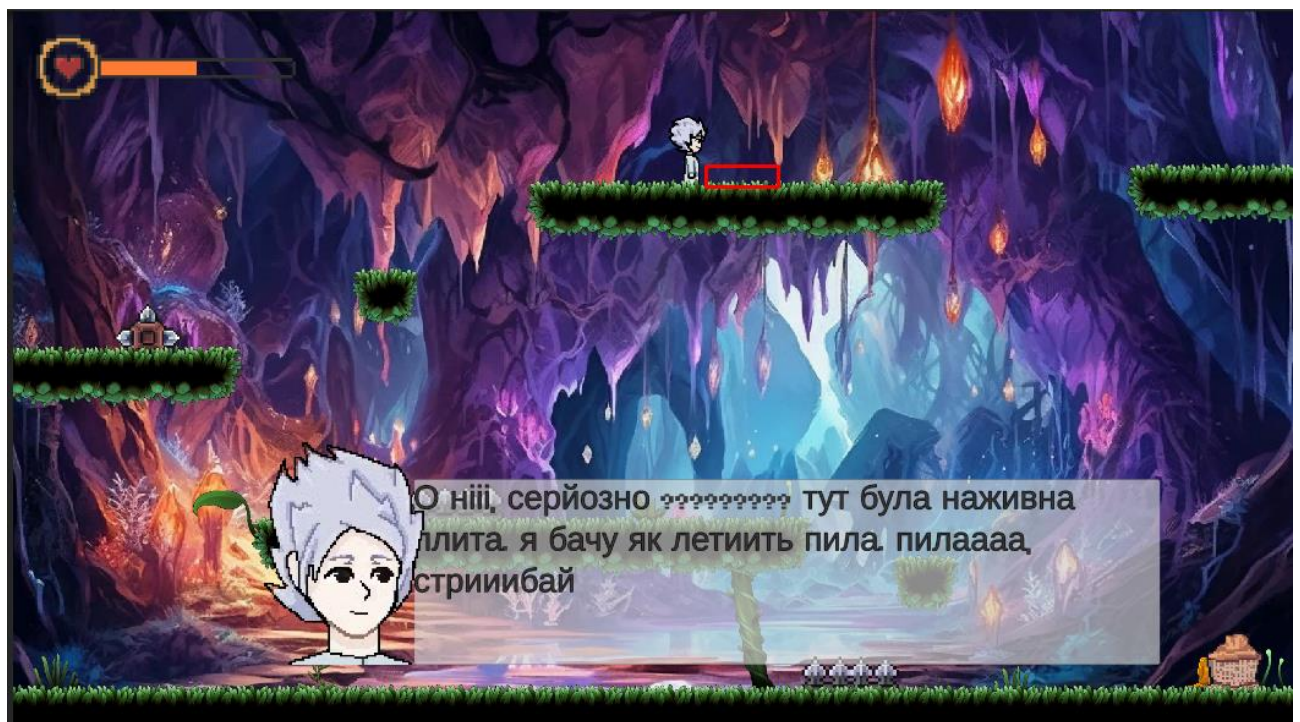


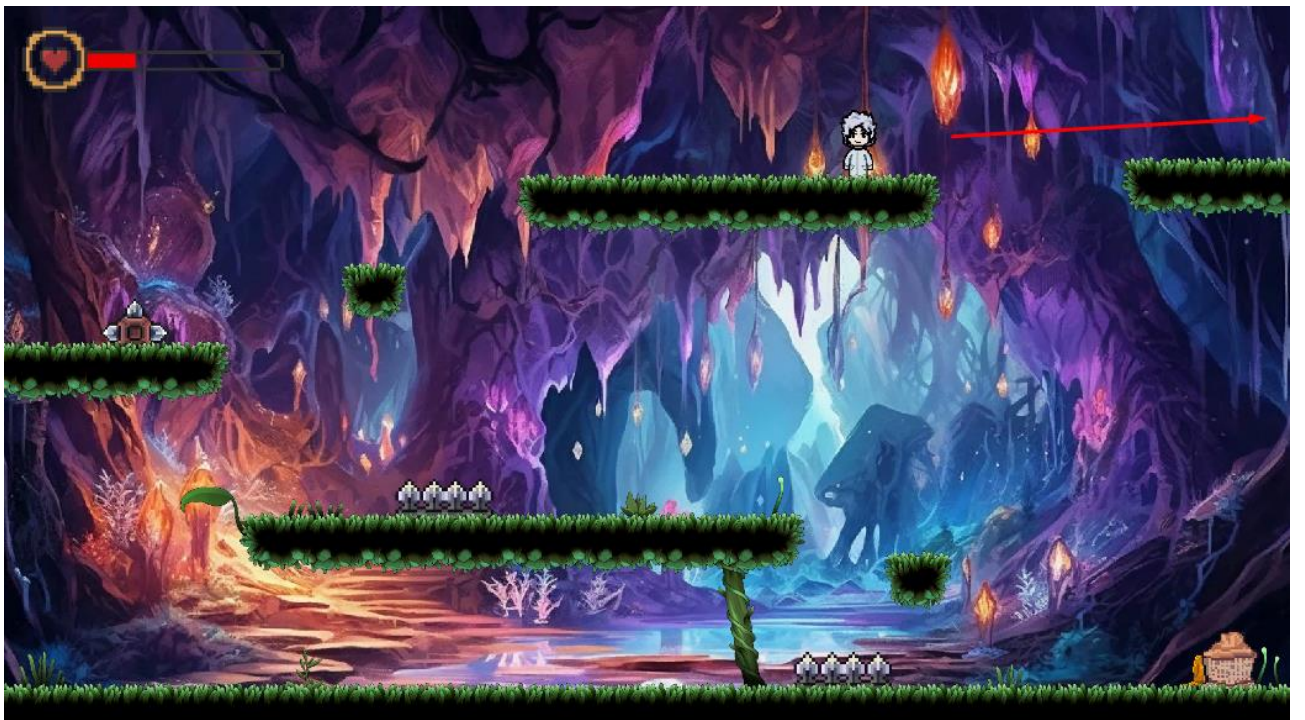


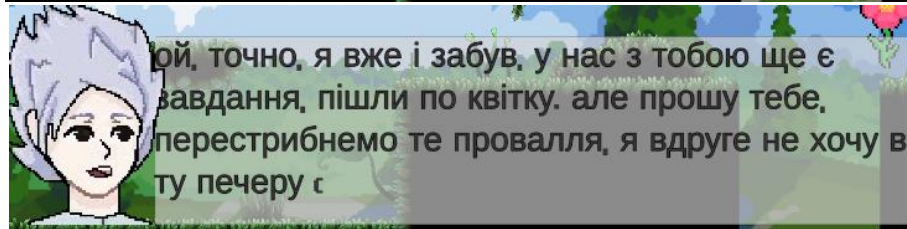
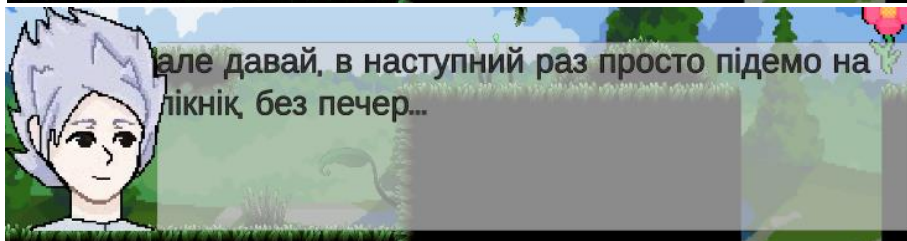
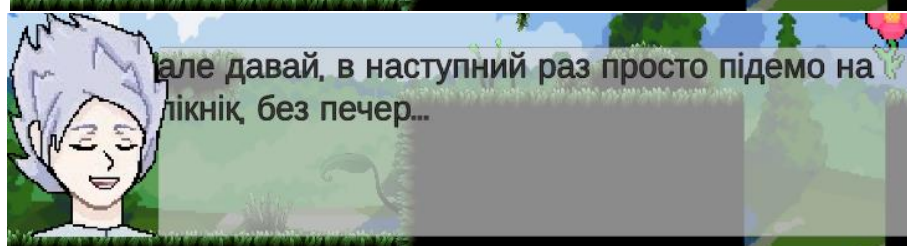
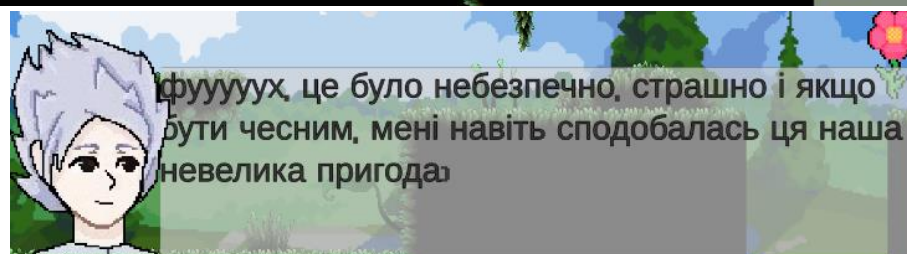
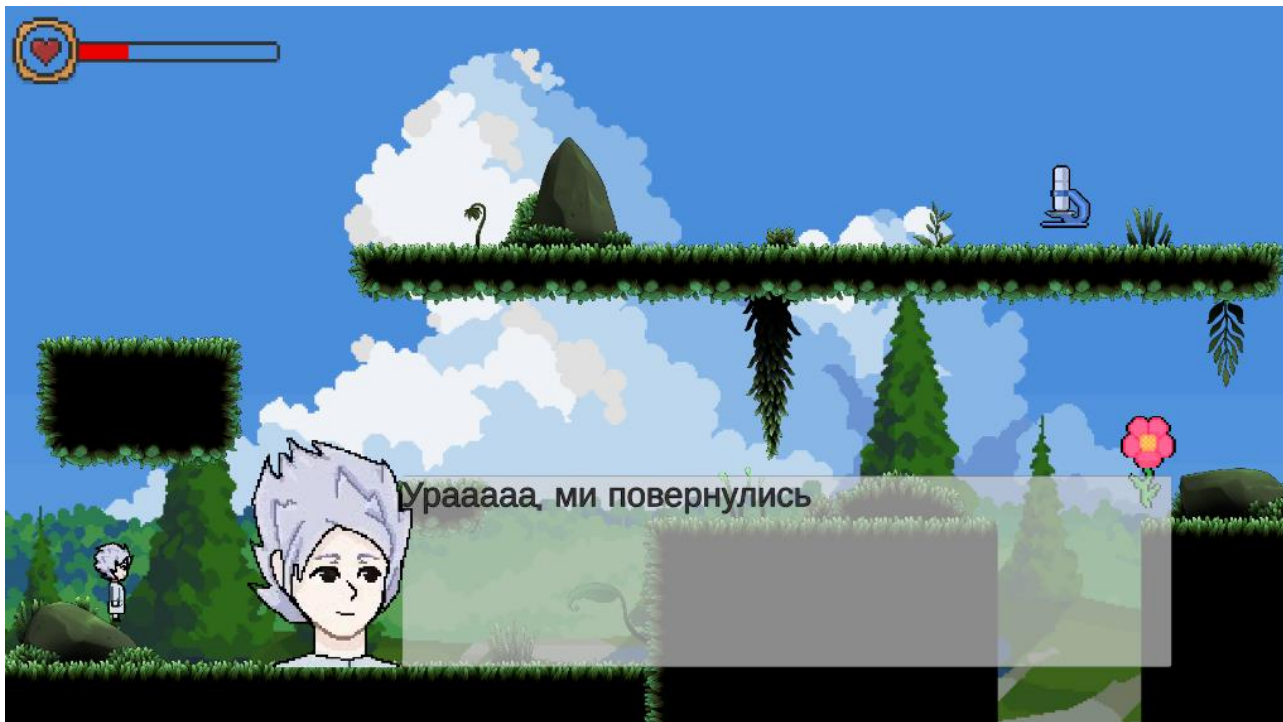


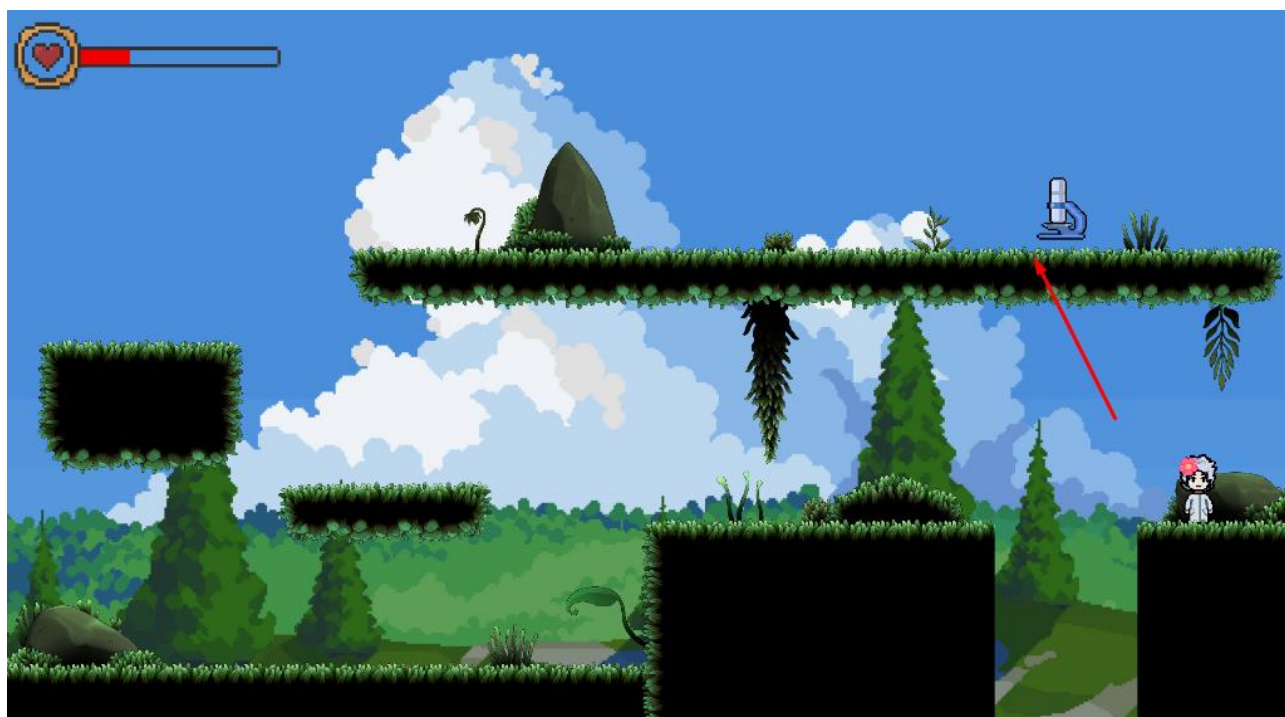
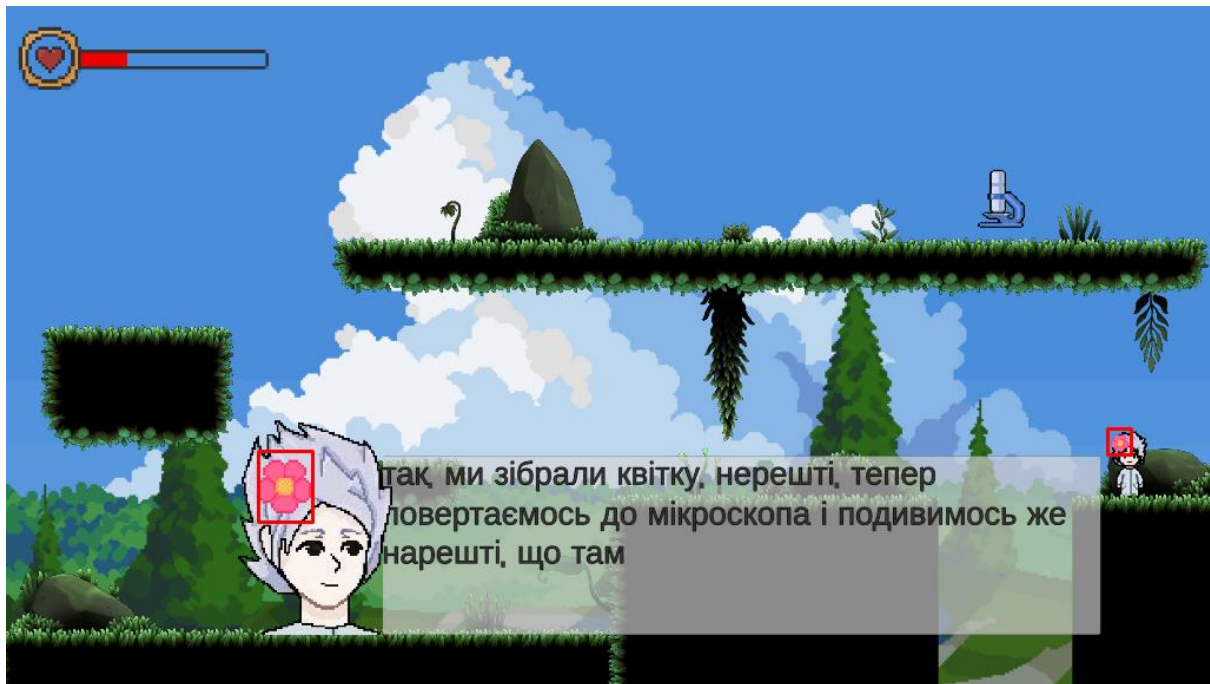
а це напевно місцева фауна. фауна - це все
що пов'язанно з тваринним світом, в цьому
випадку у нас вулик є представником фауни
цієї печери

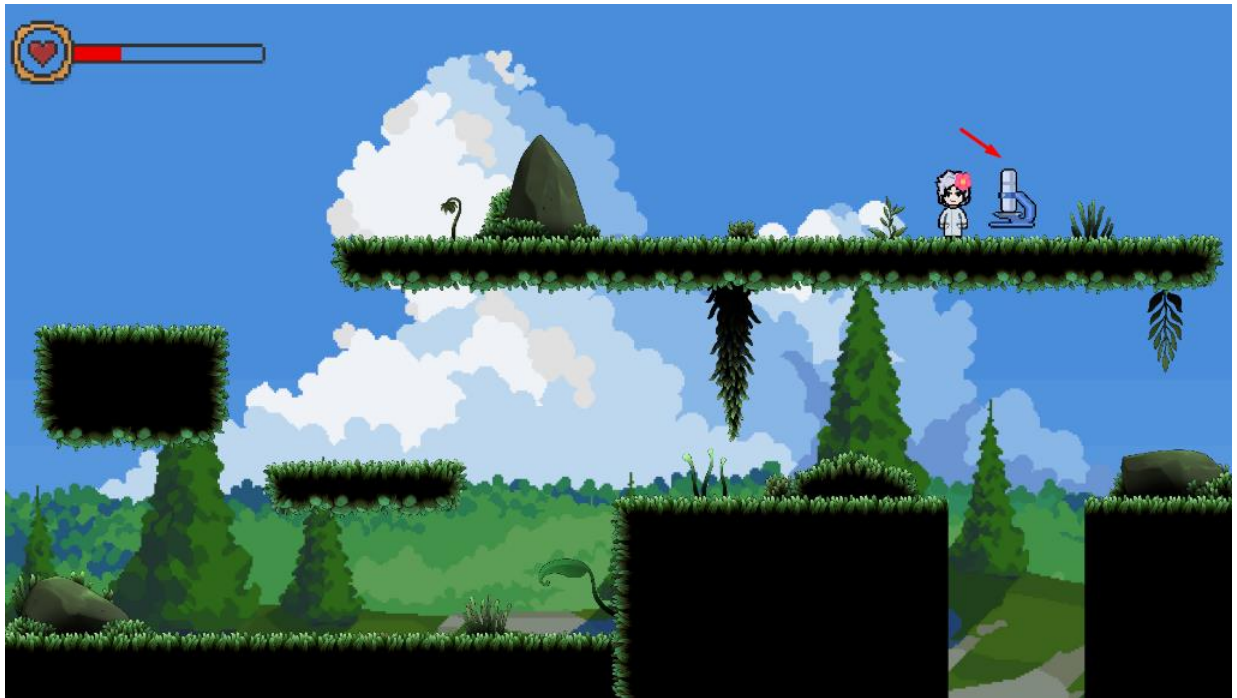












Плазмодесми
 Плазматична мембрана
 Клітинна стінка
 Цитоскелет
 Маленькі мембранні везикули
 Гладкий ендоплазматичний ретикулум
 Рибосоми
 Ядро
 Ядерна поря
 Ядерна оболонка
 Ядерце
 Шорсткий ендоплазматичний ретикулум
 Апарат Гольджі
 Везикули Гольджі
 Галоплазма
 Пероксисома
 Мітохондрія
 Тонопласт
 Вакуоля
 Вакуоля
 Мембрани тилакоїдів
 Зерна крохмалю
 Хлоропласт
 Плазмодесми
 Плазматична мембрана
 Клітинна стінка

Ти перед собою бачиш клітину, всі живі організми складаються з таких клітин