НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС «ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ» НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

**КУРСОВА РОБОТА**

з дисципліни

Програмування

на тему: Містобудівелний симулятор(гра)

Студента 1 курсу групи КП-23

Спеціальність 121 Інженерія програмного забезпечення.

Ващенко Денис Сергійович

Керівник асистент Погорелов В. В.

Національна оцінка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_\_\_\_\_\_Оцінка: ECTS \_\_\_\_\_

Члени комісії \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (підпис) (вчене звання, науковий ступінь, прізвище та ініціали) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (підпис) (вчене звання, науковий ступінь, прізвище та ініціали) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (підпис) (вчене звання, науковий ступінь, прізвище та ініціал Київ – 20XXрік

Київ – 2023 рік

ВСТУП

РОЗДІЛ 1 Постановка задачі

1.1 Огляд існуючих підходів до розв’язання поставленої задачі

1.2 Уточнена постановка задачі на розробку програмного забезпечення РОЗДІЛ 2 Розробка програмного продукту

2.1Метод розв’язку задачі

2.2 Алгоритм розв’язку задачі

РОЗДІЛ 3 Опис розробленого програмного продукту

3.1 Опис головних структур і змінних програми

3.2. Опис головних функцій програми

3.3. Опис інтерфейсу

3.4. Результати роботи програмного продукту.

ВИСНОВКИ

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

**Вступ**

Темою даної курсової роботи є містобудівельний симулятор створений в середовищі Unity.

Актуальність:

Містобудівельний симулятор є досить популярним жанром в ігровій індустрії. Він надає гравцю можливість відчути себе в шкурі мера міста, захоплює гравця відчуттям постійного розвитку та прогресу міста, надає різноманітні виклики гравцю, дозволяє гравцю проявити креативність та оригінальність при проектуванні власних міст.

Мета:  
Реалізувати закінчений програмний продукт, що дозволяє виконувати такі базові операції містобудівельного симулятора, як будівництво та проектування міста, симуляція економіки міста.

Завдання:

На основі аналізу літературних джерел, було вирішено, що для розробки даного програмного продукти найкраще підійде програмне середовище Unity, оскільки воно дозволяє використовувати 3д моделі та є досить простим у використанні. Складено структурний алгоритм роботи та реалізовано його у вигляді програмного продукту.

Практичне значення одержаних результатів:

Була створена працююча комп’ютерна гра, що має всі базові можливості містобудівельного симулятора.

Використане програмне забезпечення:

При виконанні роботи було використано таке програмне середовище:

Стандартне середовище розробки:

* Unity
* Microsoft Visual Studio
* Fbx converter

Операційна система:

* Windows 10

Веб браузер:

* Google Chrome

Для роботи з веб-сайтами:Unity Documentation, ScetchFab

Текстовий редактор:

* Microsoft Word для підготовки та оформлення курсової роботи

Структура роботи

Робота складається зі вступу, трьох розділів, висновків, додатків та списку використаних джерел

**Основна Частина**

ПОСТАНОВКА ЗАДАЧІ

Постановка задачі для розроблення містобудівельного симулятора:

Мета: Розробити містобудівельний симулятор, який надасть користувачам можливість експериментувати з містобудівними концепціями та створити місто з економічно успішною моделлю.

Завдання:

1. Створення віртуального середовища міста:
   * Розробка 3D-моделей будівель та ландшафту міста.
   * Відтворення деталей містобудівної обстановки.
2. Функціональні можливості для користувача:
   * Розробка інтуїтивного інтерфейсу користувача, що дозволяє легко взаємодіяти з містобудівельним середовищем.
   * Реалізація функцій створення, редагування та управління містом, включаючи зонування, розташування об'єктів, управління ресурсами тощо.
3. Аналіз та оцінка ефективності містобудівних рішень:
   * Введення системи оцінки, що дозволяє користувачам аналізувати та оцінювати вплив їх містобудівних рішень на різні аспекти, такі як транспортна потужність, екологічність, соціальна сфера тощо.

Основні вимоги до містобудівельного симулятору:

1. Основні функції: Розробити містобудівельний симулятор, який дозволяє створювати та редагувати міста, розміщувати будівлі та дороги, а також здійснювати базове управління містом.
2. Графіка: Забезпечити просту, але чітку і зрозумілу графіку для візуалізації міста та його елементів. Вона може бути двовимірною або простою тривимірною графікою.
3. Інтерфейс користувача: Розробити простий інтерфейс, що дозволяє легко взаємодіяти з містобудівельними елементами. Наприклад, додайте кнопки або меню для розміщення будівель та доріг, зміни масштабу міста та зміни режимів перегляду.
4. Базові функції управління: Додати основні функції управління містом, наприклад, можливість встановлювати тарифи на комунальні послуги, керувати бюджетом міста та додавати нові будівлі.
5. Збереження та завантаження: Додати можливість зберігати створені міста та завантажувати їх знову для подальшого редагування.
6. Документація та пояснення: Включити документацію, яка описує функціональні можливості вашого симулятора, а також надайте інструкції щодо його використання.
7. Тестування та відлагодження: Впевнитися, що ваш симулятор працює правильно, провівши достатню кількість тестів та виправивши будь-які пом

Результати розроблення містобудівельного симулятора повинні надати користувачам можливість вивчати та експериментувати з різними містобудівними концепціями, розуміти принципи планування та управління містами.

Для проектування даної гри можна була використати консольний інтерфейс та графічний інтерфейс(такий як Unity або Unreal Engine). Перевагами першого способу є головним чином більш проста реалізація та менша кількість необхідних ресурсів. Перевагами другого способу є візуально приємніший та зрозуміліший інтерфейс, але натомість він складніший в реалізації, потребує додаткових знань в обраному програмному середовищі(в даному випадку Unity) та потребує використання значної кількості додаткових матеріалів таких, як 3д-моделі та текстури.

РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

Метод розв’язку задачі

Для реалізації цієї роботи було обрано середовище Unity, оскільки воно забезпечує зручну та зрозумілу взаємодію з користувачем та є досить легким для реалізації такого виду проектів. У цьому середовищі була розроблена структура об’єктів та класів, що взаємодіють між собою. Всього було реалізовано 21 клас та інтерфейс, що забезпечують роботу гри. Створено абстрактний клас Building та похідні від нього класи для реалізації всіх будівель, що використовуються програмою. Створено:

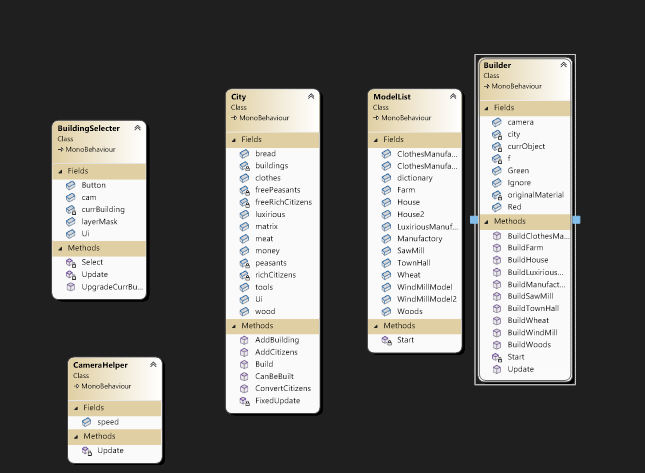
* клас Builder, що реалізує будівництво
* клас City, що реалізує економіку міста та містить всі об’єкти, з якими оперує програма

Також створені допоміжні класи:

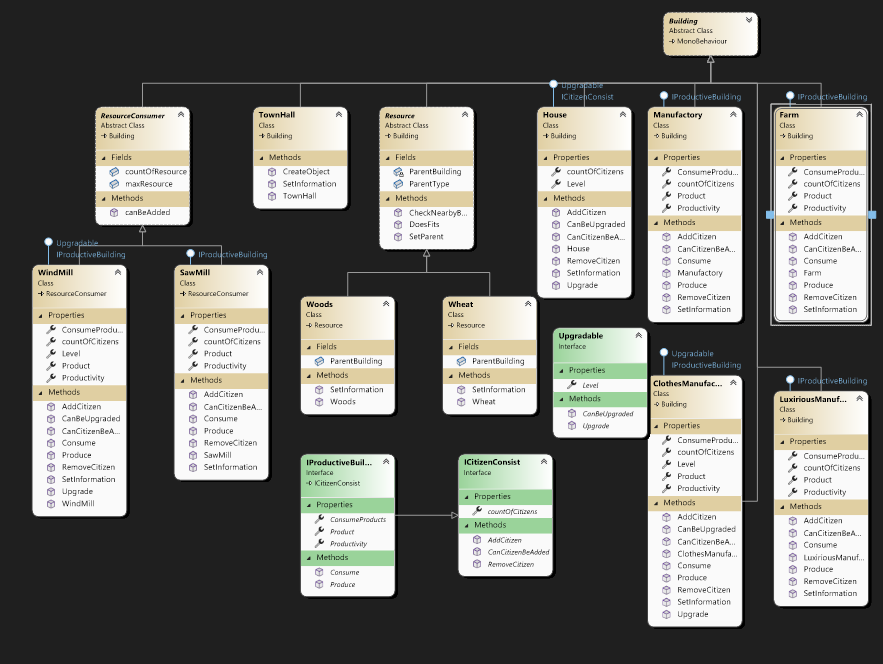
* Camera Helper(переміщення камери)
* BuildingSelecter(відображення інформації про будівлю)
* ModelList(набір всіх 3д моделей, що використовуються)

Наведемо структуру класів:

Класи, що не пов’язані безпосередньо з класом Building:



Структура класів, що походять від Building:



ОПИС РОЗРОБЛЕНОГО ПРОГРАМНОГО ПРОДУКТУ

Розглянемо кожний клас окремо:

1. Абстрактний клас Builder: Реалізує такі методи:

CreateObject() : створює 3д модель будівлі

MoveObject(position) : переміщує модель на вказану позицію

Визначає такі класи та змінні:

SetInformation() – повертає основну інформацію про будівлю

Name – назва будівлі

Length, Width – розміри будівлі

City – Клас, до якого прикріплена будівля

Model – модель будівлі

Money, wood, tool – ресурси необхідна для побудови

1. Абстрактний клас Resource – наслідується від Building

Використовується такими будівлями, як ліс та поле

Реалізує такі методи:

CheckNearByBuilding() – визначає чи є поблизу будівлі до яких можна прикріпитися

DoesFits(building) – визначає чи підходить вказана будівля для прикріплення

SetParent() – прикріплює будівлю

Також визначає такі змінні:

ParentType – вид будівлі, до якої можна прикріпитися

ParentBuilding – будівля, до якої прикріплений даний ресурс

1. Абстрактний клас ResourceConsumer – наслідується від Building

Реалізує будівлю, до якої прикріплюються інші будівлі класу Resource

Реалізує такі методи:

canBeAdded() – визначає чи вистачає місця, щоб прикріпити будівлю

Також визначає такі змінні:

CountOfResource – кількість прикріплених будівель

maxResource – максимальна кількість прикріплених будівель

1. Інтерфейс Upgradable – визначає, що будівля може бути покращена

Визначає такі методи та змінні:

Upgrade() – покращити будівлю

CanBeUpgraded() – визначає, чи може будівля бути покращеною

Level – поточний рівень будівлі

1. Інтерфейс ICitizenConsist – визначає, що будівля містить жителів:

Визначає такі методи та змінні:

AddCitizen() – додати мешканця

RemoveCitizen() – видалити мешканця

CanCitizenBeAdded() – визначає чи можна додати мешканця

1. Інтерфейс IProductiveBuilding – визначає, що будівля виробляє ресурси:

Визначає такі методи та змінні:

ConsumeProducts – список ресурсів, що споживає будівля

Product – продукт, що виробляється

Productivity – продуктивність будівлі

Produce() – виробити продукт

Consume() – спожити ресурс

1. Клас WindMill – наслідує:

ResourceConsumer

Upgradable

IProductiveBuidling

Реалізує такі методи та змінні:

Конструктор WindMill()

1. Клас Wheat – наслідує:

Resource

Реалізує такі методи та змінні:

Конструктор Wheatl()

1. Клас Woods – наслідує:

Resource

Реалізує такі методи та змінні:

Конструктор Woods()

1. Клас House – наслідує:

Building

Upgradable

ICitizenConsist

Реалізує такі методи та змінні:

Конструктор House()

1. Клас SawMill – наслідує:

ResourceConsumer

IProductiveBuidling

Реалізує такі методи та змінні:

Конструктор SawMill()

1. Клас WindMill – наслідує:

Building

IProductiveBuidling

Реалізує такі методи та змінні:

Конструктор Farm()

1. Клас TawnHall – наслідує:

Building

Реалізує такі методи та змінні:

Конструктор TawnHall()

1. Клас Manufactory – наслідує:

Building

IProductiveBuilding

Реалізує такі методи та змінні:

Конструктор Manufactory()

1. Клас ClothesManufactory – наслідує:

Building

Upgradable

IProductiveBuilding

Реалізує такі методи та змінні:

Конструктор ClothesManufactory ()

1. Клас LuxiriousManufactory – наслідує:

Building

IProductiveBuilding

Реалізує такі методи та змінні:

Конструктор LuxiriousManufactory()

ОПИС ІНТЕРФЕЙСУ:



Внизу ми бачимо панель, де знаходяться всі будинки, що ми можемо побудувати

Справа внизу - панель з описом будівлі, що міняється в залежності від типу будівлі

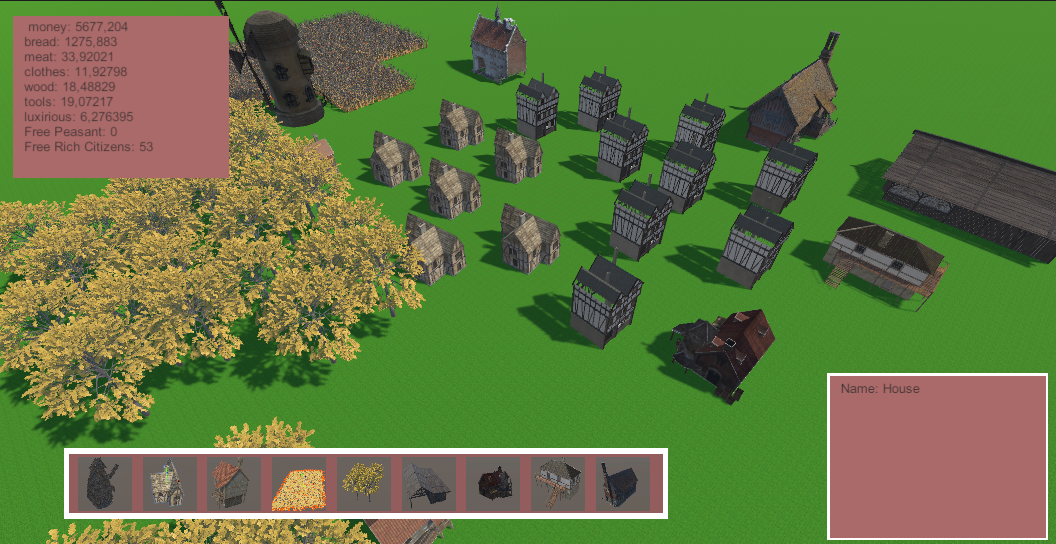
Зліва зверху перелік ресурсів, що наявні на даний момент

Натиснувши на будівлю в нижній будівлі, ми перейдемо в режим будівництва. Якщо модель, можна побудувати в даному місці, то модель стане зеленою, інакше – червоним. Натиснувши ЛКМ- ми побудуємо будівлю, ПКМ – відмінимо будівництво.

Ресурси в лівй верхній панелі змінюються динамічно

Якщо будівлю можна покращити, то справа знизу з’явиться кнопка Upgrade, при натисканні на яку, модель будинку зміниться разом з деякими параметрами шляхом виклику методу Upgrade()

РЕЗУЛЬТАТИ РОБОТИ ПРОГРАМИ



Місто, яке можна створити, використовуючи функціонал даного програмного продукту.

Висновок

В результаті виконання курсової роботи було розроблено містобудівельний симулятор, який дозволяє створювати та управляти віртуальними містами. Симулятор має простий інтерфейс користувача, що дозволяє легко взаємодіяти з містобудівельними елементами, такими як будівлі. Базові функції управління містом, такі як керування ресурсами

Графіка симулятора є простою, але чіткою і зрозумілою, забезпечуючи візуалізацію міста та його елементів.

В процесі виконання курсової роботи було проведено тестування та відлагодження симулятора для забезпечення його правильної роботи. Була створена документація, яка детально описує функціональні можливості симулятора, а також надані інструкції щодо його використання.

У результаті, ця курсова робота дала мені можливість ознайомитися з процесом розробки містобудівельного симулятора та набути практичних навичок у створенні програмного забезпечення та ігр за допомогою Unity.