Київський національний університет технологій та дизайну

Лабараторна робота №5

Створення документації

Виконав студент БКІ-22

Савенко Ярослав Володимирович

Огризков Роман Олексійович

Крук Павло Миколайович

Романенко Михайло Вадимович

Київ 2024

**Лабораторна робота №5**

**Створення документації**

**Мета:** формування практичних навичок розробки документації на програмний продукт та інструкцій користувача.

**Порядок виконання:**

**1. Загальні положення**

• Назва проекту: “Змійка”;

• Тип гри: Аркада;

• Цільова платформа: ПК;

• Мова програмування: C#;

• Графіка: 2D;

• Інтерфейс користувача: Мінімалістичний (консольний), керування через клавіатуру.

**2. Опис процедури встановлення програми та її запуску**

Для встановлення гри нам необхідно завантажити Unity hub, так як на основі Unity реалізована гра та звичайно сам файл гри. Після чого треба в Unity відкрити файл проекту та встановити необхідну версію Unity для коректної роботи гри. Нижче буде наведено посилання на папку проекту.

**3. Перелік операцій, які може виконувати користувач з програмним продуктом.**

**1.Початок гри:**

Запуск гри через меню.

**2.Управління змійкою:**

Використання клавіш (стрілки або WASD) для переміщення змійки.

Зміна напрямку руху (вліво, вправо, вгору, вниз).

**3.Збір їжі:**

Зіткнення з об'єктами їжі для збільшення довжини змійки та набрання очок.

**4.Вихід з гри:**

Закриття гри через меню або натиснення кнопки "Вихід".

**4.** **Перелік дій, які виконує програмний продукт.**

**1.Ініціалізація гри:**

Завантаження початкових налаштувань та ресурсів (графіка, звуки).

Відображення стартового меню.

**2.Створення змійки:**

Генерація початкової змійки з заданою довжиною.

Встановлення початкової позиції змійки на ігровому полі.

**3.Генерація їжі:**

Випадкова генерація об'єктів їжі на ігровому полі.

Забезпечення, щоб їжа не з'являлася в позиціях, зайнятих змійкою.

**4.Обробка введення користувача:**

Визначення напрямку руху змійки на основі введення (клавіші, торкання).

Зміна напрямку руху змійки відповідно до введення.

**5. Рух змійки:**

Оновлення позиції змійки в залежності від напрямку.

Розширення змійки при зборі їжі.

**6.Перевірка зіткнень:**

Визначення зіткнень змійки з їжею, власним тілом.

Реакція на зіткнення (збільшення довжини змійки або закінчення гри).

**7.Відображення інформації:**

Оновлення інтерфейсу з відображенням рахунку, довжини змійки.

Відображення повідомлень про закінчення гри та можливість перезапуску.

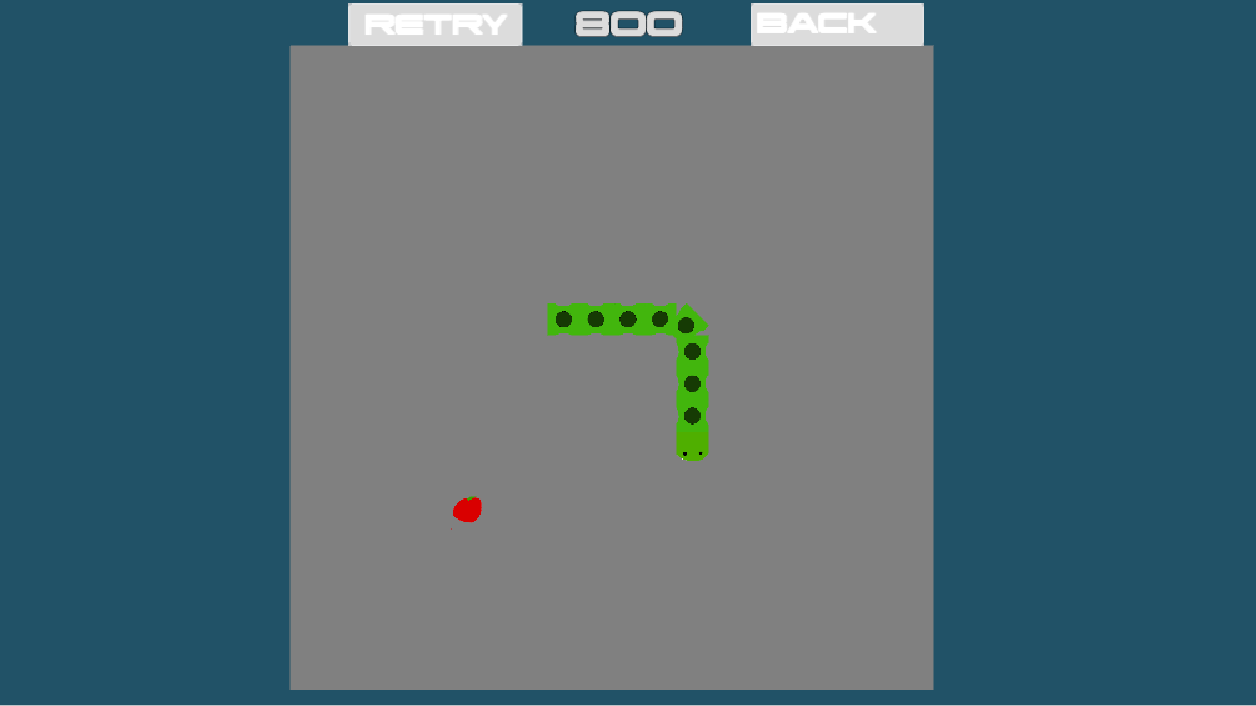
**8.Вихід з гри:**

Завершення гри та виведення на екран відповідних повідомлень.

**5. Найбільш ймовірний сценарій подій користувача зі грою**

Користувач запускає гру



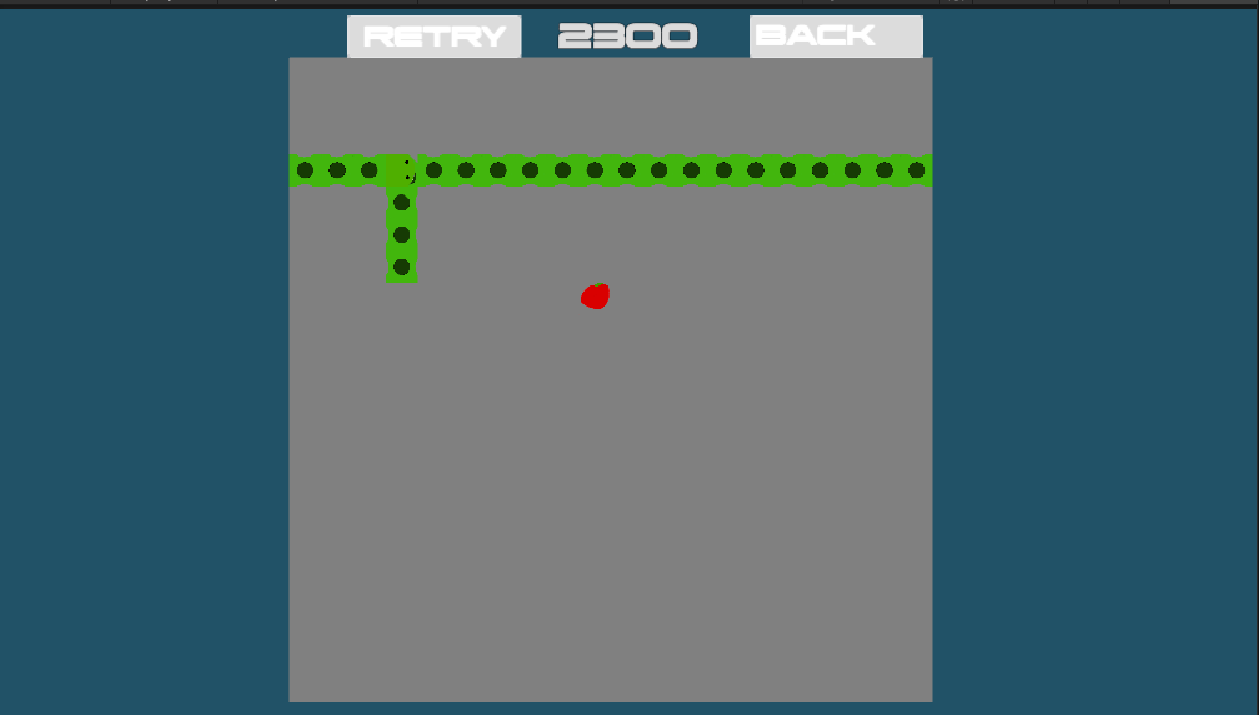


Користувачеві здається цей процес нудним і він хоче збільшити швидкість.

Для цього йому необхідно зайти в налаштування та перемістити повзунок на більше значення. Після чого запустити гру.



Гравець зазнає поразки після того як більшим швидкість, бо нею не дуже легко грати. І він закінчує свою ігрову сесію.



**6. Перелік та детальний опис основних функцій та алгоритмів роботи**

Найголовнішими скриптами є Snake, GameHandler, LevelGrid  
Коротко пройдемось по них:

**Snake:**

Цей скрипт реалізує логіку гри «Змійка», контролюючи її рух, обробляючи введення від користувача, перевіряючи зіткнення з їжею та власним тілом. Основні функції включають:

* **Керування рухом**: Змійка рухається по сітці відповідно до напрямку, обраного гравцем.
* **Збільшення розміру**: Коли змійка з'їдає їжу, її розмір збільшується.
* **Перевірка на смерть**: Якщо голова змійки потрапляє в ту ж позицію, що й будь-яка частина тіла, змійка гине.

Скрипт також контролює швидкість руху змійки та візуалізує її частини тіла відповідно до позицій у грі.

**GameHandler :**

Цей скрипт управляє загальною логікою гри, зокрема налаштуванням ігрового поля, веденням рахунку та налаштуванням камери. Основні функції:

**Ініціалізація:** Під час старту гри скрипт налаштовує розмір ігрового поля, з якого береться розмір з налаштувань гри, і ініціалізує сітку (LevelGrid), в якій буде рухатися змійка.

**Керування рахунком:** Ведеться облік очок, які гравець отримує за досягнення (наприклад, за їжу). Є методи для отримання рахунку та його збільшення.

**Налаштування фону та камери:** Скрипт відповідальний за пропорційне масштабування фону та коригування позиції камери, щоб забезпечити правильне відображення ігрового поля.

**Повернення на попередній екран:** Метод для завантаження попередньої сцени, що дозволяє гравцеві повернутися назад у меню.

Таким чином, скрипт «GameHandler» відповідає за організацію ігрового середовища та інтеграцію різних елементів гри.

**LevelGrid**

Цей скрипт управляє логікою ігрового поля, зокрема розміщенням їжі та перевіркою, чи змійка її з'їла. Основні функції:

**Конструктор:** Ініціалізує розміри ігрового поля (width та height).

**Налаштування:** Метод Setup приймає об'єкт змійки і викликає метод для генерації їжі на полі.

**Генерація їжі:** Метод SpawnFood випадковим чином вибирає позицію для їжі на полі, уникаючи зіткнень з позиціями змійки. Створює об'єкт їжі з відповідним спрайтом.

**Перевірка споживання їжі:** Метод TrySnakeEatFood перевіряє, чи змійка з'їла їжу (порівнюючи її позицію з позицією їжі). Якщо так, їжа знищується, генерується нова, і рахунок збільшується.

**Валідація позицій:** Метод ValidateGridPosition перевіряє, чи знаходиться позиція змійки в межах поля, і, якщо це не так, "переносить" її на інший край (роблячи поле безшовним).

Отже, скрипт LevelGrid відповідає за управління ігровим полем, генерацію їжі та обробку взаємодії змійки з їжею.

**7. Список використаної літератури**

1. Unity. Unity Documentation. [Електронний ресурс]. Режим доступу:

<https://docs.unity3d.com/ru/2019.4/Manual/Overview2D.html>

2. Simple 2D Game in Unity: Snake (Unity Tutorial for Beginners)

<https://www.youtube.com/watch?v=Iz22-o7l6bc&list=PLzDRvYVwl53ucaUs0YGfyyKXdgqh5OtiK&ab_channel=CodeMonkey>

**8.Перелік розробників проекту**

Савенко Ярослав Володимирович

Огризков Роман Олексійович

Крук Павло Миколайович

Романенко Михайло Вадимович

**Висновок:** У ході розробки гри "Змійка" на платформі Unity було успішно реалізовано базову механіку класичної аркадної гри, включаючи створення змійки, керування її рухом, генерацію їжі та перевірку зіткнень. Гра дозволяє користувачеві виконувати такі операції, як керування змійкою, збір їжі, перезапуск після програшу. Програмний продукт обробляє введення від користувача, оновлює стан змійки та ігрового поля, а також забезпечує зберігання результатів.

Розроблена гра є функціональним ігровим додатком, що демонструє базові принципи роботи з Unity, а також навички програмування в C#. Реалізація гри дала змогу детально ознайомитися з процесом розробки ігор, включаючи обробку подій, керування об'єктами та створення інтерфейсу користувача.