МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Крымский федеральный университет имени В.И. Вернадского»

Таврический колледж
(структурное подразделение)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

на тему:

РАЗРАБОТКА ВИДЕОИГРЫ ЖАНРА СТРАТЕГИЯ-ROGUELIKE

Обучающийся: Касьянов Артур Анатольевич
Специальность: 09.02.03 Программирование в компьютерных системах
Группа 4ПКС17
Научный руководитель: Смирнова Светлана Ивановна

Симферополь, 2023

Цель и задачи

Цель дипломной работы состоит в разработке видеоигры жанра стратегия-roguelike.

Для достижения цели были поставлены следующие задачи:

- исследовать видеоигры, жанра статегия, их историю создания;
- изучить игровой движок Unity и синтаксис использующегося в нем языка программирования;
- изучить интерактивную среду разработки;
- разработать дизайн игры;
- создать программную реализацию.

Язык программирования

Преимущества	Недостатки
Объектно-ориентированность	Низкая безопасность кода
Простой синтаксис конструкций	Низкая скорость
Количество готовых скриптов и библиотек	Необходимость в предустановке библиотек
Популярность	Слабое взаимодействие с железом

Рисунок 1 – Преимущества и недостатки C#

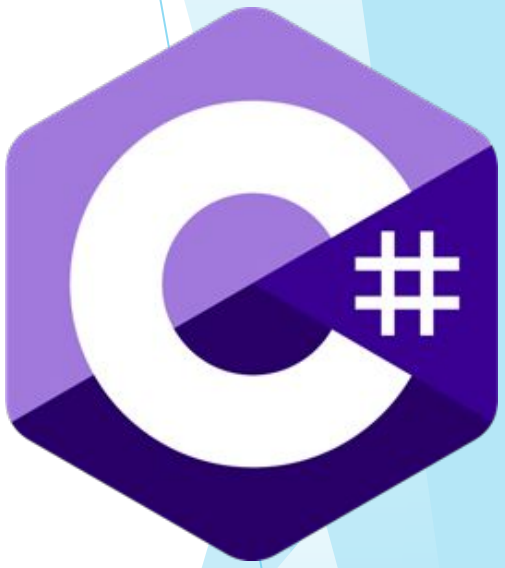

Рисунок 2 – Логотип C#

Среда разработки

Преимущества	Недостатки
Поддержка множества языков	Отсутствуют
Интуитивный стиль	
Оптимизация написания кода	
Возможность отладки	

Рисунок 3 – Преимущества и недостатки Visual Studio

Рисунок 4 – Логотип Visual Studio

Среда разработки

Преимущества	Недостатки
Наличие точной документации	Большой размер сформированных приложений
Мультиплатформенность	Сложность в создании масштабных сцен
Компонентно-ориентированная концепция	Отсутствие поддержки внешних ссылок

Рисунок 5 – Преимущества и недостатки Unity

Рисунок 6 – Логотип Unity

Концепт жанра Стратегия

Стратегическая компьютерная игра – один из основных жанров компьютерных игр, в котором игроку для победы необходимо применять стратегическое мышление.

Рисунок 7 – Логотип Europa Universalis 4

Концепт жанра Roguelike

Roguelike – жанр компьютерных игр. Который подразумевает случайную генерацию, необратимость смерти и разблокирование игрового контента в процессе игры.

Рисунок 8 – Логотип Noita

Изменение игрового времени

```
// 1 реальная секунда = 30 игровых минут
GameMinute = GameMinute + 30 * Time.deltaTime;
//смена часа
if (GameMinute >= 60.0f)...

//смена дня
if (GameHour >= 24.0f)...

//смена месяца
if (GameDay >= 31.0f)...

//смена года
if (GameMonth >= 13.0f)...
```

Рисунок 9 – Механика игрового времени

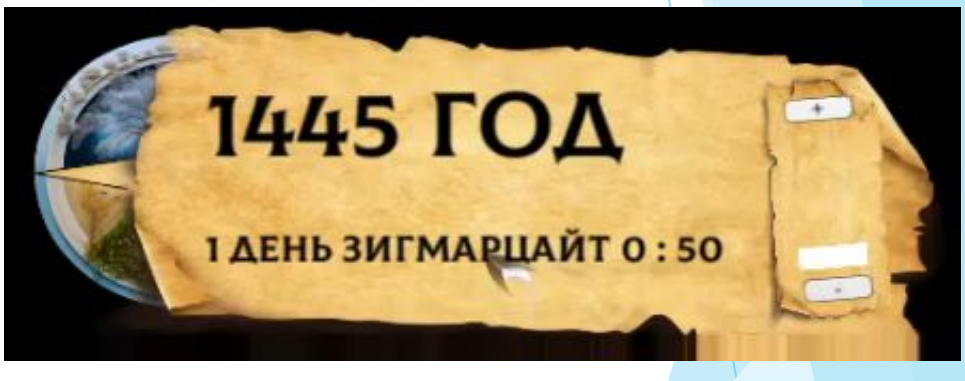

Рисунок 10 – Реализация в игре

Скорость игрового времени

```
//ускорение времени
public void time_up()...

//замедление времени
public void time_down()...

//пауза
public void pause()
{
    Time.timeScale = 0;
    pauseButton.SetActive(false);
    resumeButton.SetActive(true);
}

//возобновить
public void resume()
{
    Time.timeScale = GameTimeSpeed;
    resumeButton.SetActive(false);
    pauseButton.SetActive(true);
}
```

Рисунок 11 – Механика изменения скорости игрового

Рисунок 12 – Реализация в игре

Панель ресурсов

```
//РЕСУРСЫ
public static float[] Resources = { 1000, 40, 40, 40, 400, 1000 }; //сами ресурсы

float[] Difference = { 0, 0, 0, 0, 0, 0 }; //общие расходы
float gnomsReproduction; //размножение гномов

public Text[] TextResourses; //ресурсы в панеле
public Text[] TextResourses2; //ресурсы в вкладке экономика
public Text FreeGnomsUI; //свободные гномы

float[] Production = { 0, 0, 0, 0, 0 };//общее производство
float[] Consuming = { 0, 0, 0, 0, 0 };//потребление
public float[] baseProduction; //базовое производство крепости(рогилик мб?)
float[] buildingProduction = { 0, 0, 0, 0, 0 }; // производство зданиями
```

Рисунок 13 – Механика панели ресурсов

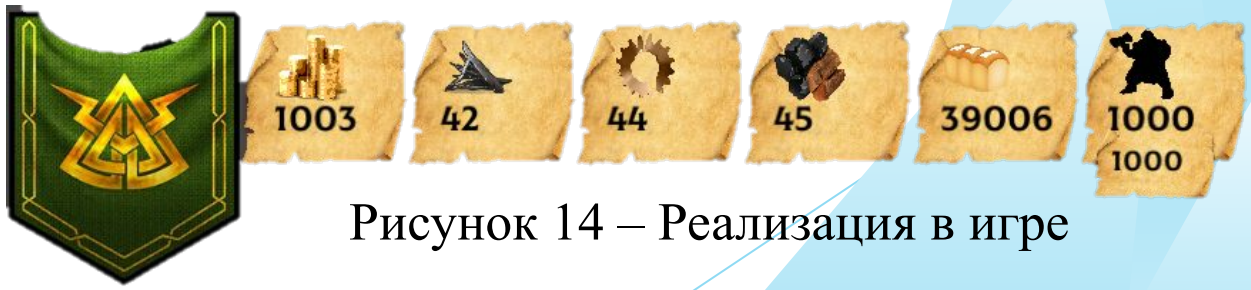

Рисунок 14 – Реализация в игре

Панель строительства

```
//разница
Difference[i] = Production[i] - Consuming[i];

//отображение ресов в интерфейсе
if (Difference[i] >= 0)
{
    DifferenceUI[i].color = Color.green; plus[i] = "+";
}
else
{
    DifferenceUI[i].color = Color.red; plus[i] = "";
}
DifferenceUI[i].text = plus[i] + Difference[i].ToString();
```

Рисунок 17 – Механика панели строительства

Рисунок 18 – Реализация в игре

Панель указов

```
Consuming[4] = Resources[5] * ModifierScript.foodMaintenanceModifier * (1/ModifierScript.moodModifier) * ModifierScript.seasonFoodModifier;

//производство зданиями
if (MaxWorkerGnoms[i] != 0)
{
    buildingProduction[i] = (BuildingsScript.Buildings[0, i] * CurrentWorkerGnoms[i] / MaxWorkerGnoms[i] * ModifierScript.TechnologyModifier[i]) * ModifierScript.workEfficiency;
}
else { buildingProduction[i] = 0; }
```

Рисунок 19 – Механика панели указов

Рисунок 20 – Реализация в игре

Панель войск

```
static public float[,] ArmyCharacteristics =
{
    // горняки, гномьи воины, гномьи воины тж, длиннобороды, длиннобороды тж, молотобойцы, железоломы, мстители, егеря, забияки, громобои, железные ящеры, пушки, местеметы, органные орудия, огненные пушки
    {1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3 },                      //ряд                           0
    {100, 100, 100, 80, 80, 80, 60, 80, 80, 80, 80, 28, 4, 4, 4, 4 },       //численность                   1
    {30, 40, 40, 50, 50, 50, 50, 80, 30, 30, 30, 50, 250, 250, 250, 250 }, //ОЗ на единицу                 2
    {20, 30, 30, 40, 40, 50, 70, 10, 10, 15, 15, 90, 10, 10, 10, 10 },      //броня                         3
    {30, 30, 35, 40, 45, 60, 50, 70, 30, 30, 30, 20, 10, 10, 10, 10 },      //атака в ближнем бою           4
    {20, 40, 30, 50, 40, 30, 70, 50, 20, 30, 20, 60, 10, 10, 10, 10 },      //защита в ближнем бою          5
    {1, 3, 4, 8, 8, 15, 10, 20, 1, 1, 1, 5, 20, 20, 20, 20 },               //сила оружия                   6
    {0, 0, 0, 0, 0, 0, 0, 0, 60, 60, 80, 90, 70, 70, 80, 80 },              //точность в дальнем бою        7
    {0, 0, 0, 0, 0, 0, 0, 0, 5, 5, 10, 50, 400, 300, 500, 750 },            //сила выстрелов                8

    };
```

Рисунок 21 – Механика панели войск

Рисунок 22 – Реализация в игре

Панель торговли

```
public void trade_confirm_value()
{
    tradeconfirm = 0;
    if ((Country == 6) || (Country == 7))
    {
        return;
    }
    else
    {
        for (int i = 0; i <= 4; i++)
        {
            tradeconfirm += trade[0, i] * tradeCost[i] * tradeCostModif[i, Country] - trade[1, i] * tradeCost[i] * tradeCostModif[i, Country];

        }
    }

    if (tradeconfirm > 0) { TradeConfirmUI.color = Color.green; }
    else { TradeConfirmUI.color = Color.red; }

    TradeConfirmUI.text = tradeconfirm.ToString();
}
```

Рисунок 23 – Механика панели торговли

Рисунок 24 – Реализация в игре

Сражение

```
690     static public void Battle()
691     {
692         for (int a = 0; a <= 19; a++)
693         {
694             //битва
695 
696             if (EnemyArmyScript.enemybattle[0, a] != 0)...
859 
860             if (battle[0, a] != 0)...
1024 
1025            if (EnemyArmyScript.enemybattle[1, a] != 0)...
1188 
1189            if (battle[1, a] != 0)...
1353 
1354            if (EnemyArmyScript.enemybattle[2, a] != 0)...
1517 
1518            if (battle[2, a] != 0)...
1683 
1684            battleArmyCheck = 0;
1685            enemybattleArmyCheck = 0;
1686            battleArmyCheck1row = 0;
1687            battleArmyCheck2row = 0;
1688            enemybattleArmyCheck1row = 0;
1689            enemybattleArmyCheck2row = 0;
1690 
1691            for (int r = 0; r <= 19; r++)...
1702 
1703            if (battleArmyCheck1row == 0)...
1710 
1711            if (battleArmyCheck2row == 0 && battleArmyCheck1row == 0)...
1718 
1719            for (int r = 0; r <= 15; r++)...
1723 
1724            for (int r = 0; r <= 19; r++)...
1736 
1737            if (enemybattleArmyCheck1row == 0)...
1744 
1745            if (enemybattleArmyCheck2row == 0)...
1752 
1753 
1754            for (int r = 0; r <= 15; r++)...
```

Рисунок 25 – Механика сражений

Рисунок 26 – Реализация в игре

События

```
for (int i = 0; i <= 2; i++)
{
    if (TimeAndResources.randomEvent == i)
    {
        Event[i].Invoke();

        for (int j = 0; j <= 3; j++)
        {
            Event_UI_Text[j].text = TextButtonEvent[j];
            Event_UI_Text_Main.text = TextMainEvent;
            Event_UI_Text_Main_Name.text = TextMainEventName;
        }

        for (int y = 0, k = 0, j = 0; y <= 23; y++)
        {
            Event_UI[y].text = ResourcesEvent[j, k].ToString();
            if (k <= 4) { k++; } else { k = 0; j++; }
        }

        EventPanel.SetActive(true);
        TimeAndResources.randomEvent = 101;
    }
}
```

Рисунок 27 – Механика событий

Рисунок 28 – Реализация в игре

Заключение

В ходе работы было сформировано представление о том, что такое игра жанра Стратегия и какие основные элементы ей присущи. Были исследованы существующие решения по реализации различных игровых модулей, таких как:. Рассмотрены различные инструментальные возможности и особенности программного обеспечения для разработки.

Все поставленные задачи были выполнены, в соответствии с этим достигнута цель выпускной квалификационной работы.

Перспективой исследования можно считать развитие приложения путем добавления новых механик.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«Крымский федеральный университет имени В.И. Вернадского»

Таврический колледж
(структурное подразделение)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

на тему:

РАЗРАБОТКА ВИДЕОИГРЫ ЖАНРА СТРАТЕГИЯ-ROGUELIKE

Обучающийся: Касьянов Артур Анатольевич
Специальность: 09.02.03 Программирование в компьютерных системах
Группа 4ПКС17
Научный руководитель: Смирнова Светлана Ивановна

Симферополь, 2023