УДК 004

Разработка пошаговой тактической клиент-серверной игры

Ульянов Егор Андреевич
Приамурский государственный университет имени Шолом-Алейхема
Студент

Аннотация
В данной статье рассматривается и описывается разработка пошаговой тактической клиент-серверной игры на компьютер. Игра будет разрабатываться на языке программирования C# с помощью игрового движка Unity 3D. Практическим результатом является разработанная игра.
Ключевые слова: Игра, C#, Unity 3D

Creation of a turn-based tactical client-server game

Ulianov Egor Andreevich
Sholom-Aleichem Priamursky State University
Student

Abstract
This article discusses and describes the development of a turn-based tactical client-server game on a computer. The game will be developed in the C # programming language using the Unity 3D game engine. The practical result is a developed game.
Keywords: Game, C#, Unity 3D

Компьютерные игры за время своего существования претерпели множество качественных изменений в части выразительности и технологичности. Наблюдается проникновение компьютерных игр в различные сферы деятельности – образование, искусство, спорт. В образовательной сфере, например, они могут служить инструментом для проведения лекций, экзаменов, профессиональной практики. Также игры могут помочь в таких вещах, как развитие пространственного мышления, улучшение зрительной реакции, снятие стресса, тренировка организаторских навыков.

Цель данной статьи рассмотреть возможности игрового движка Unity 3D в сетевых играх и создание пошаговой тактической клиент-серверной игры.

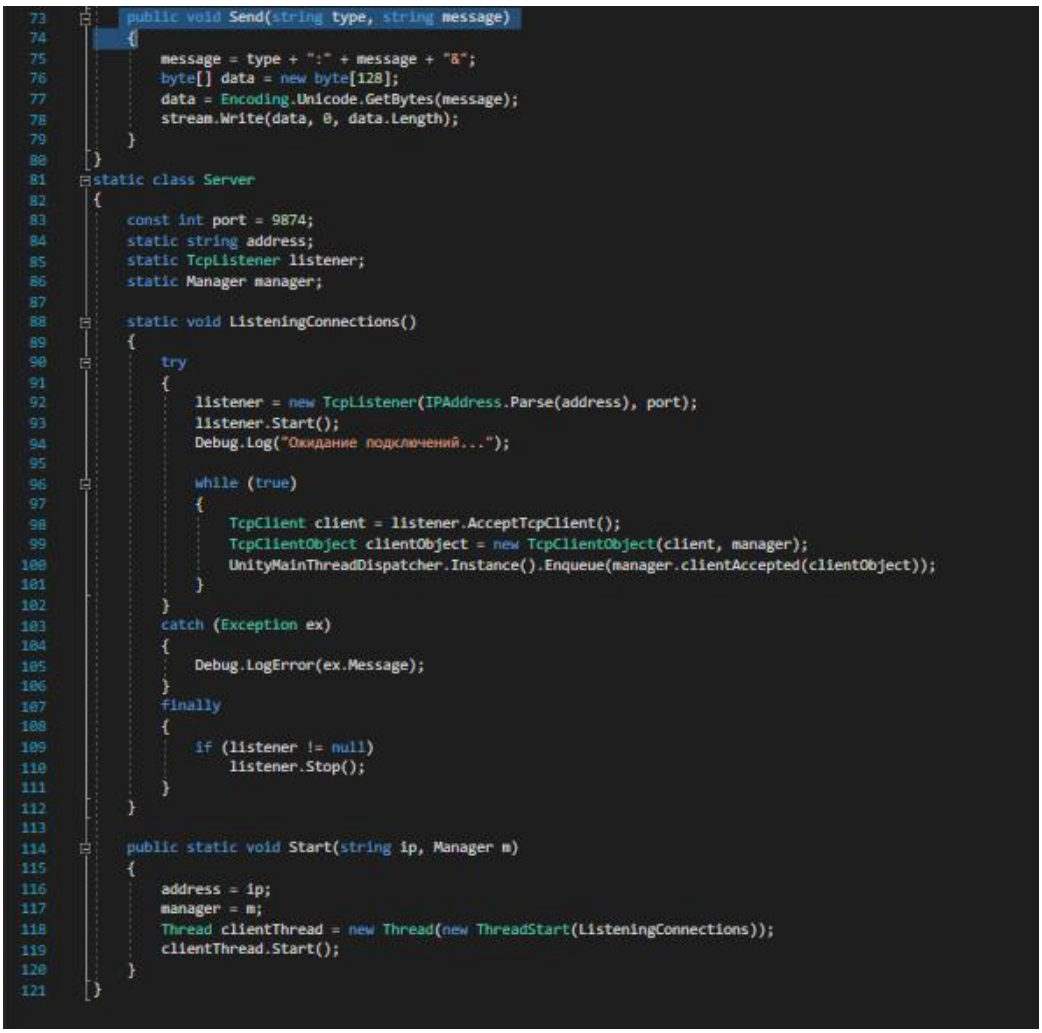
В.Д.Лукиных рассмотрел возможность создания компьютерной игры в жанре «Action/RPG» [1]. В данной работе Э.С. Окороков, Н.Н. Сивцев, Г.Ю. Протодьякова описали технологии разработки инди игр. Также рассмотрели понятия, что такое инди игры, и кто такие инди разработчики [2]. В.И. Макаров провел в своей статье анализ способов для создания

пользовательского интерфейса при разработке приложений [3]. Так же А.И. Долженко совместно с С.А. Глушенко провели анализ целесообразности разработки мобильного приложения для android устройств, а также разработали собственное приложение на android [4]. В статье Р.В. Мальчева и С.В. Кривошеева был выполнен анализ архитектурной системы ARM как аппаратной основы для создания симуляторов т/с [5].

Напишем код, который будет образовывать сервер на компьютере. Правила игры были запрограммированы с помощью языка программирования C# и IDEVisualStudio 2017. На одном устройстве работает сервер (и клиент), а на других - клиенты. Напишем соединительную часть сервера см. рисунок 1 и 2.

```
using System;
using System.Net.Sockets;
using System.Net;
using System.Text;
using System.Threading;
using UnityEngine;
using System.Collections;
using System.Collections.Generic;

public class TcpClientObject
{
    public TcpClient client;
    NetworkStream stream = null;
    static Manager manager;
    public TcpClientObject(TcpClient tcpClient, Manager m)
    {
        try
        {
            manager = m;
            client = tcpClient;
            Thread clientThread = new Thread(new ThreadStart(Reading));
            clientThread.Start();
        }
        catch (Exception ex)
        {
            Debug.LogError(ex.Message);
        }
    }

    public void Reading()
    {
        try
        {
            stream = client.GetStream();
            byte[] data = new byte[128];
            while (true)
            {

                StringBuilder builder = new StringBuilder();
                int bytes = 0;
                do
                {
                    bytes = stream.Read(data, 0, data.Length);
                    builder.Append(Encoding.Unicode.GetString(data, 0, bytes));
                }
                while (stream.DataAvailable);

                string message = builder.ToString();

                string[] messages = message.Split('&');

                foreach (string msg in messages)
                {
                    if (msg.Length > 0)
                    {
                        UnityMainThreadDispatcher.Instance().Enqueue(manager.OnClientMessage(this, msg));
                    }
                }
            }
        }
        catch (Exception ex)
        {
            Debug.LogError(ex.Message);
        }
        finally
        {
            if (stream != null)
                stream.Close();
            if (client != null)
                client.Close();
        }
    }
```

Рис. 1. Серверный код



```
    public void Send(string type, string message)
    {
        message = type + ":" + message + "&";
        byte[] data = new byte[128];
        data = Encoding.Unicode.GetBytes(message);
        stream.Write(data, 0, data.Length);
    }
}
static class Server
{
    const int port = 9874;
    static string address;
    static TcpListener listener;
    static Manager manager;

    static void ListeningConnections()
    {
        try
        {
            listener = new TcpListener(IPAddress.Parse(address), port);
            listener.Start();
            Debug.Log("Ожидание подключений...");

            while (true)
            {
                TcpClient client = listener.AcceptTcpClient();
                TcpClientObject clientObject = new TcpClientObject(client, manager);
                UnityMainThreadDispatcher.Instance().Enqueue(manager.clientAccepted(clientObject));
            }
        }
        catch (Exception ex)
        {
            Debug.LogError(ex.Message);
        }
        finally
        {
            if (listener != null)
                listener.Stop();
        }
    }

    public static void Start(string ip, Manager m)
    {
        address = ip;
        manager = m;
        Thread clientThread = new Thread(new ThreadStart(ListeningConnections));
        clientThread.Start();
    }
}
```

Рис.2. Серверный код (Продолжение)

Далее необходимо написать код клиента игры. Будем писать популярный сейчас “BattleRoyale” с видом сверху (рис.3-4).

```
using System;
using System.Net.Sockets;
using System.Text;
using UnityEngine;
using System.Threading;

static class Client
{
    const int port = 9874;
    static string address;

    static Manager manager;

    static NetworkStream stream;
    static TcpClient client;
    static void StartClient()
    {
        try
        {
            client = new TcpClient(address, port);
            stream = client.GetStream();

            Thread readingThread = new Thread(new ThreadStart(Reading));
            readingThread.Start();

        }
        catch (Exception ex)
        {
            Debug.LogError(ex.Message);
        }
    }

    static void Reading()
    {
        try
        {
            while (true)
            {
                byte[] data = new byte[128];
                StringBuilder builder = new StringBuilder();
                int bytes = 0;
                do
                {
                    bytes = stream.Read(data, 0, data.Length);
                    builder.Append(Encoding.Unicode.GetString(data, 0, bytes));
                }
                while (stream.DataAvailable);

                string message = builder.ToString();

                string[] messages = message.Split('&');

                foreach (string msg in messages)
                {
                    if (msg.Length > 0)
                    {
                        UnityMainThreadDispatcher.Instance().Enqueue(manager.OnServerMessage(msg));
                    }
                }
            }
        }
        catch (Exception ex)
        {
            Debug.LogError(ex.Message);
        }
        finally
        {
            client.Close();
        }
    }
```

Рис. 3. Код клиента

```
    public static void Send(string type, string message)
    {
        message = type + ":" + message + "&";
        byte[] data = Encoding.Unicode.GetBytes(message);
        stream.Write(data, 0, data.Length);
    }

    public static void Start(string ip, Manager m)
    {
        address = ip;
        manager = m;
        StartClient();
    }
}
```

Рис. 4. Продолжение кода

После того как были написаны клиент и сервер, осталось добавить персонажей, усилители и карту (рис.5-7).

Рис. 5. Персонаж и противник

Рис. 6. Усилители

Рис. 7. Карта

Теперь игра готова и на рисунках ниже представлена ее реализация и “gameplay”. Первый игрок, создает сервер на своем компьютере см. рисунок 8.

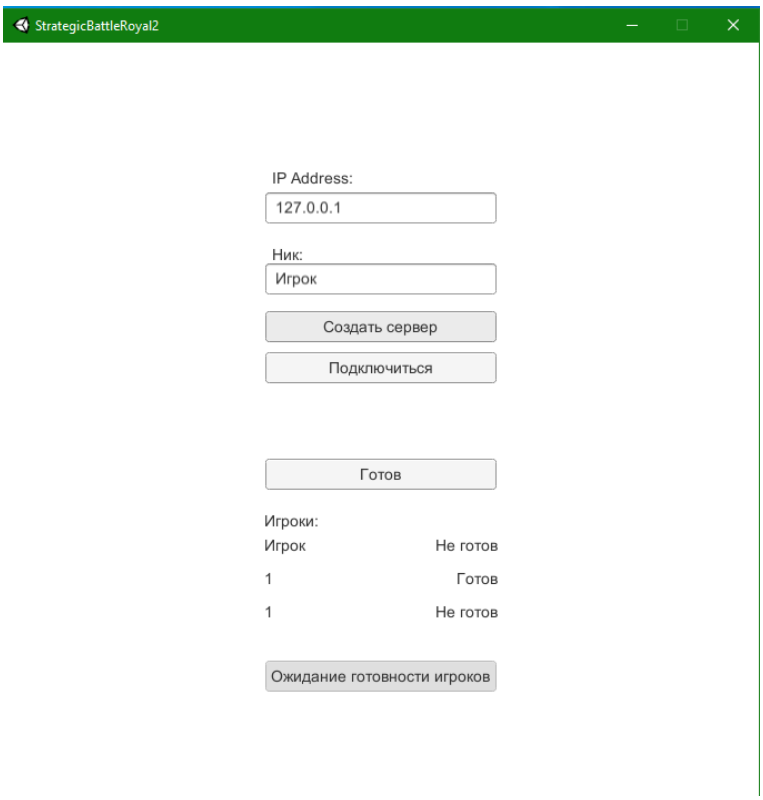


Рис. 8. Создание сервера

Остальные игроки вводят IP-адрес сервера, свои “nickname” и нажимают подключиться см. рисунок 9 и 10.

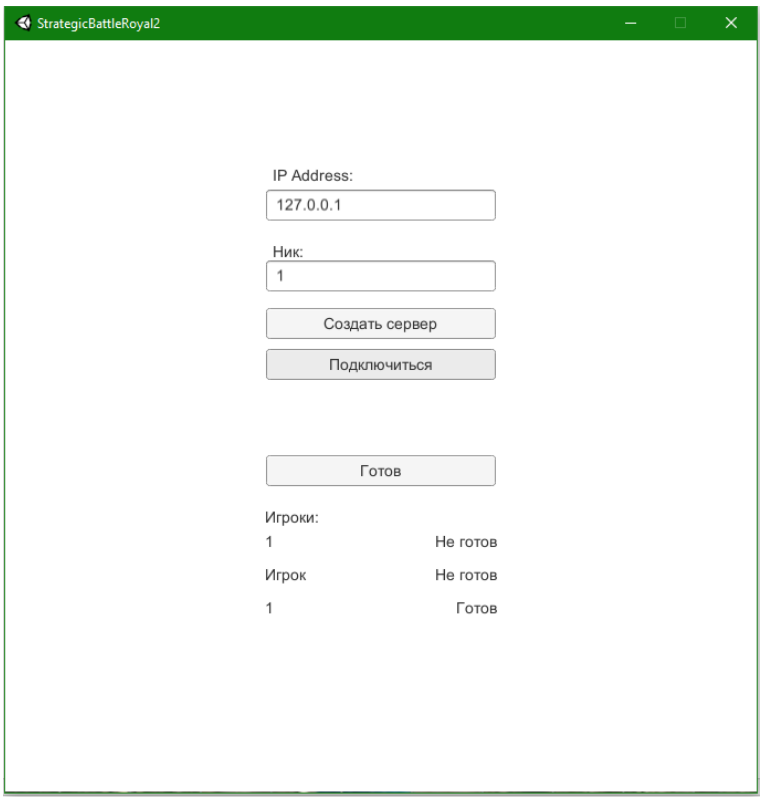


Рис. 9. Подключение игроков

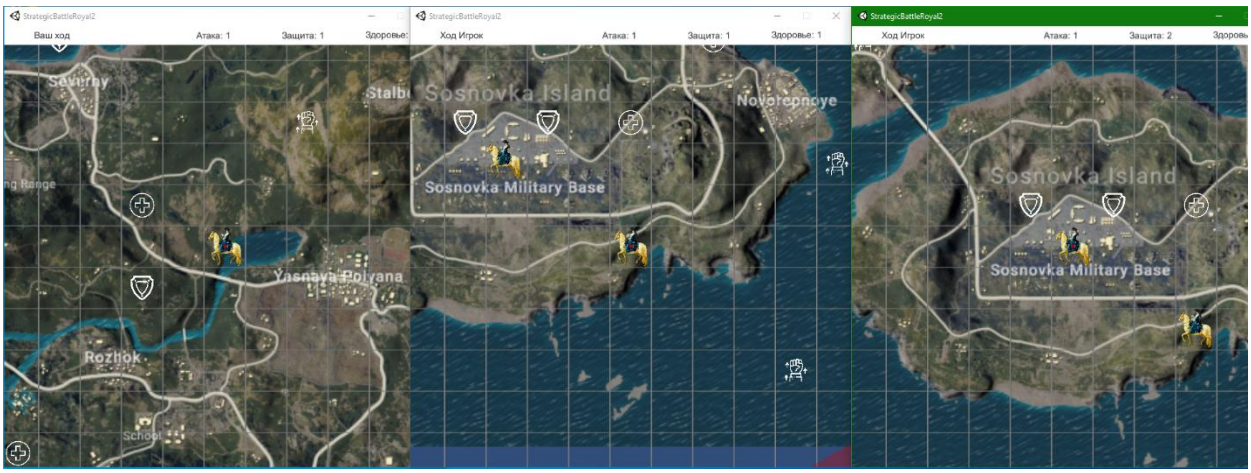


Рис. 10. Игра с вида хоста и двух игроков

Были проанализированы существующие аналоги и методы разработки, а также выбрана среда разработки. Для реализации поставленной задачи отлично подошла разработка с помощью Visual Studio и языка программирования C#. Такой выбор заметно упростил разработку проекта, так как в интернете имеется достаточное кол-во документации. Во время создания игры был полученный ценный опыт работы с этим средством разработки.

В итоге была разработана и протестирована игра по схеме «Клиент – Сервер», что позволяет пользователям взаимодействовать в игре с разных компьютеров в локальных или онлайн сетях. Такая возможность очень важна в наше время, потому что позволяет играть с разными людьми, с разных уголков мира. Созданная игра имеет потенциал к развитию, а именно: добавление новых функций; улучшение интерфейса; увеличение количества контента; более точная настройка баланса игроков.

Библиографический список

1. Лукиных В.Д. Разработка компьютерной игры в жанре action/rpg "the third circle" // Труды Международного симпозиума «Надежность и качество». 2019. №3. С. 12-14.
2. Окороков Э.С., Сивцев Н.Н., Протодьякова Г.Ю. Разработка инди игр // Автоматика. Вычислительная техника. 2018. №9. С. 6-10.
3. Макаров В.И. Особенности разработки пользовательского интерфейса для android-приложений в среде разработки androidstudio// Современные научные исследования и инновации. 2017. № 7-5 (43). С. 47-55.
4. Долженко А.И., Глушенко С.А. Разработка мобильного приложения для товарищества собственников жилья (ТСЖ) на платформе android // Труды Международного симпозиума «Надежность и качество». 2014. №5. С. 14-20.
5. Мальчева Р.В., Кривошеева С.В. Разработка симуляторов транспортных средств с использованием операционной системы android // Автоматика. Вычислительная техника. 2012. №1. С. 24-30.