Компьютерная академия “ШАГ”

Кафедра Разработки программного обеспечения

КУРСОВОЙ ПРОЕКТ

BrpgCenter

Студента группы \_\_\_\_\_\_\_\_\_

Павлова Е.И.

Научный руководитель:

Скидан О.С.

(Астана)2018

Аннотация

Документация, предназначенная для курсового проекта на тему “Платформа .NET”. В данной документации содержится следующая информация:

-техническое задание содержащие в себе все то что было необходимо выполнить и что было бы желательно выполнить;

-технологии, применяемые для воссоздания данного приложения и их краткое описание;

-информацию о структуре разрабатываемой системы где наглядно рассмотрены все связи между компонентами и их задачи;

-разработка алгоритмов. В котором показан общий алгоритм взаимодействия компонентов системы и алгоритм взаимодействия с пользователем. Тут выделены основные операции, которые выполняет система и потоки данных, которые курсируют в системе;

-описание базы данных для системы;

-структура программного проекта. Содержит диаграммы классов class diagramm (описание классов, связей, примененных паттернов), диаграммы последовательности (sequence diagram), активности (activity diagram), use-case диаграммы, другие диаграммы UML;

-описание того как проходила разработка интерфейса. Содержит описание и макеты интерфейса для системы;

-описание процесса создания инсталлятора;

-руководство пользователя. Описание готового продукта с рисунками;

-вывод о проделанной работе, в котором рассказывается с какими проблемами я сталкивался и как их решал по мере подготовки данного проекта;

-список использованной литературы.

BrpgCenter это приложение в котором я вижу смысл и предполагаю что такое приложение необходимо рынку на сегодняшний день и я надеюсь у меня получилось его воссоздать в полной мере.

Содержание

Введение ...................................................................................................... 4

Техническое задание .................................................................................. 5

Выбор технологии реализации проекта ................................................... 6

Разработка структуры системы ................................................................. 7

Разработка алгоритмов функционирования системы ............................. 9

Разработка базы данных для системы....................................................... 14

Разработка интерфейса системы................................................................ 16

Руководство пользователя.......................................................................... 21

Выводы......................................................................................................... 24

Список использованной литературы ........................................................ 25

Введение

Проект BrpgCenter является сборкой всех самых важных и нужных инструментов для настольных ролевых игр, в нее вошли такие инструменты как:

-Удобное создание персонажа;

-Наличие литературы по системам ролевых игр;

-Онлайн комната поддерживающая:

-Текстовый чат;

-Холст для рисования;

-Хранилище файлов;

-Возможность просмотреть и отредактировать персонажей других игроков.

Данный проект является актуальным в данный момент на рынке так как если представленные мною функции и реализованы, то все по отдельности что приводит к затруднениям во время проведения игры. Так как присутствует необходимость переключатся между всеми используемыми программами и ориентироваться в этом практически невозможно из-за чего опыт игры становится не таким интересным, а отыгрываемая история не такой красочной и захватывающей.

Техническое задание

*Обязательные требования*

Создать приложение BrpgCenter имеющее следующие функции:

1)Реализация онлайн комнаты для игры в настольные ролевые игры, имеющая следующие функции и инструменты:

-Видеть персонажей игроков;

-Обмениваться файлами внутри комнаты;

-Хранение карт мира;

-Растровый графический редактор с особенностью обмена данными по сети интернет между игроками;

-Возможность получить и автоматически отправить остальным игрокам результаты броска игральных костей.

2)Удобное создание персонажа;

3)Полезные ссылки на литературу по системам ролевых игр, имеющие в себе правила и подсказки для ведения правильной игры;

4)База данных с ранее созданными персонажами;

5)База данных с комнатами в которых находился игрок.

*Желательные требования*

1)Базы данных с возможными значениями полей персонажа;

2)Возможность смена языка.

*Системные требования*

OC: Windows XP или более новая версия.

Процессор: Dual Core processor

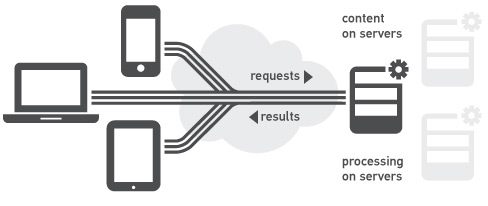
Оперативная память: 2 GB ОЗУ

Сеть: Широкополосное подключение к интернету

Место на жестком диске: 150 MB

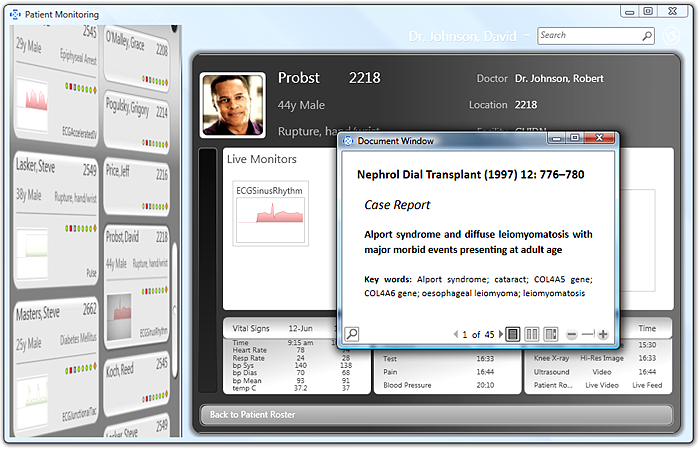
Выбор технологий для реализации проекта

Сетевое программирование. В области компьютеризации понятие программирования сетевых задач или иначе называемого сетевого программирования (англ. network programming), довольно сильно схожего с понятиями программирование сокетов и клиент-серверное программирование, включает в себя написание компьютерных программ, взаимодействующих с другими программами посредством компьютерной сети.



Программа или процесс, инициирующие установление связи, называются клиентским процессом, а программа, ожидающая инициации связи, называется серверным процессом. Клиентский и серверный процессы вместе образуют распределенную систему. Связь между клиентским и серверным процессами может быть или на основе соединений (как например, TCP-протокол, устанавливающий виртуальное соединение или сессию), или без соединений (на основе UDP-датаграмм).

WPF. Windows Presentation Foundation (WPF) ― это система следующего поколения для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем. С помощью WPF можно создавать широкий спектр как автономных, так и размещенных в браузере приложений. На следующем рисунке показан пример одного из таких приложений

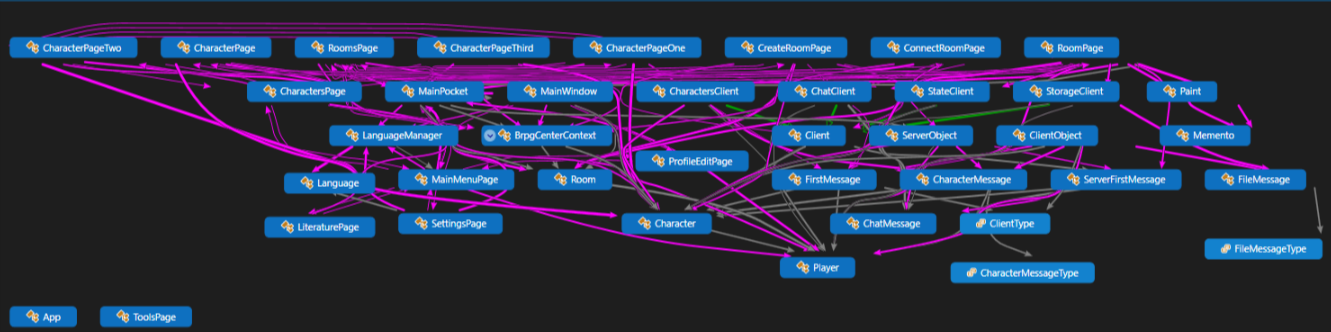


В основе WPF лежит векторная система визуализации, не зависящая от разрешения и созданная с расчетом на возможности современного графического оборудования. WPF расширяет базовую систему полным набором функций разработки приложений, в том числе Extensible Application Markup Language (XAML), элементами управления, привязкой данных, макетом, 2-D- и 3-D-графикой, анимацией, стилями, шаблонами, документами, мультимедиа, текстом и оформлением. WPF входит в состав Microsoft .NET Framework и позволяет создавать приложения, включающие другие элементы библиотеки классов .NET Framework.

ADO.NET. **ADO.NET** (ActiveX Data Object для .NET) — технология, предоставляющая доступ и управление данными, хранящимся в базе данных или других источниках (Microsoft SQL Server, Microsoft Access, Microsoft Excel, Microsoft Outlook, Microsoft Exchange, Oracle, OLE DB, ODBC, XML, текстовые файлы), основанных на платформе .NET Framework и входящая в состав .NET Framework 2.0, представляет собой набор библиотек. В отличие от технологии ADO, которая была в основном предназначена для тесно связанных клиент-серверных систем, ADO.NET больше нацелена на автономную работу с помощью объектов DataSet. Объекты DataSet представляют локальные копии взаимосвязанных таблиц данных, каждая из которых содержит набор строк и столбцов. Объекты DataSet позволяют вызывающей сборке (наподобие веб-страницы или программы, выполняющейся на настольном компьютере) работать с содержимым DataSet, изменять его, не требуя подключения к источнику данных, и отправлять обратно блоки измененных данных для обработки с помощью соответствующего адаптера данных. Но, пожалуй, самое фундаментальное различие между классической ADO и ADO.NET состоит в том, что ADO.NET является управляемой кодовой библиотекой, и, значит, подчиняется тем же правилам, что и любая управляемая библиотека. Типы, составляющие ADO.NET, используют протокол управления памятью CLR, принадлежат к той же системе типов (классы, интерфейсы, перечисления, структуры и делегаты), и доступ к ним возможен с помощью любого языка .NET. Классы ADO.NET находятся в сборке System.Data.dll.

Структура системы

Большая часть компонентов так или иначе связана между собой в стороне остаются лишь небольшая часть от всех классов.



Алгоритмы системы

На момент запуска приложение пользователю предлагают либо зарегистрироваться, либо автоматически подгружает его данные если он уже проделывал эту операцию.

Как только пользователь создает комнату, система автоматически создает и открывает сервер делая его доступным для всех кто знает IP и порт данной комнаты. Затем пользователя(хоста) подключают в качестве клиента к своему серверу.

На этом моменте на сервере запущены и функционируют следующие алгоритмы:

-Процесс прослушивания подключений;

-Процесс прослушивания chatClient и входящих сообщений с него и автоматическая отправка остальным участникам комнаты;

-Процесс прослушивания charactersClient и изменений персонажей, находящихся в комнате и автоматическое оповещение об изменении других пользователей комнаты;

-Процесс прослушивания fileClient и поступающих оттуда команд.

Со стороны клиента запущены и функционируют следующие алгоритмы:

-Процесс прослушивания входящих сообщений, перенаправленных с сервера;

-Процесс обновления списка персонажей, на основе отправленных данных с сервера;

-Процесс обновления списка файлов внутри комнаты, на основе отправленных данных с сервера.

Базы данных

Для создания баз данных я использовал EntityFramework на основе СУБД MySQL.

В БД созданы следующие таблицы:

-Character

int Id

string FullName

string WorldName //Название мира в котором находится персонаж

int TL //Технический уровень

string Status //статус персонажа в мире

int Age //возраст

string Birthday //дата рождения

string Eyes //описание глаз

string Hair //описание волос

string SkinColor //описание цвета кожи

string MainHand //основная рука

int Growth //рост

int Weight //вес

string Gender //пол

string Race //раса

string Religion //религия

int SM //модификатор размера

string MainNote //заметки

int ST //сила

int DX //ловкость

int IQ //интелект

int HT //здоровье

int HP //жизни

int Move //движение

int Speed //скорость

int Will //воля

int Per //восприятие

int FP //усталость

string Wounds //раны

string Fatigue //усталость

string AdvantagesDisadvantages//преимущества/недостатки

string Skills //способности

string Equip //экипировка

-Room

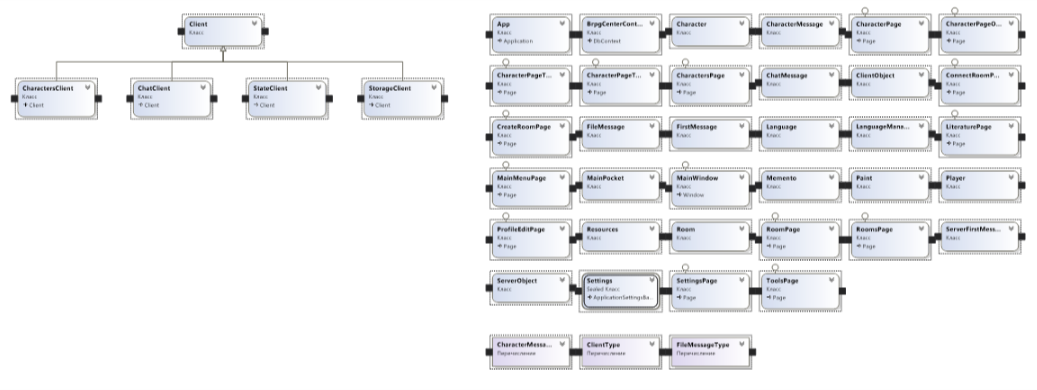
int Id

string Name

string Ip

int Port

Структура проекта



Разработка интерфейса

Интерфейс был выполнен в разных цветах с общей фоновой картиной, это было сделано с целью передать уникальность каждой страницы приложения и при этом не дать пользователю забыть, что это все еще одно целое.

Я пытался сделать интерфейс таким что бы он привлекал внимание на важные детали. Кнопки были выполнены в виде полупрозрачного белого блока с обрезанными нижними углами для того чтобы кнопка как бы сливалась с контентом содержащемся наверху, но и при этом выделялась.

Все блоки в приложении будь это ListBox или просто Rectangle заполнены белой заливкой с прозрачностью, заданной на 50%. Это было выполнено с целью создать визуальное сочетание всех блоков с друг-другом и фоном, так как все фоны в приложении разных цветов то подобрать один цвет для всех блоков было бы не приемлемо, но и задавать каждому блоку разные цвета не имеет смысла так как потеряется нить между окнами приложения. И для решения этих проблем был использован полупрозрачный белый цвет для того что бы акцентировать внимание на контенте внутри, но и при этом иметь оттенки фона что позволяет всем блокам в приложении сочетаться друг с другом.

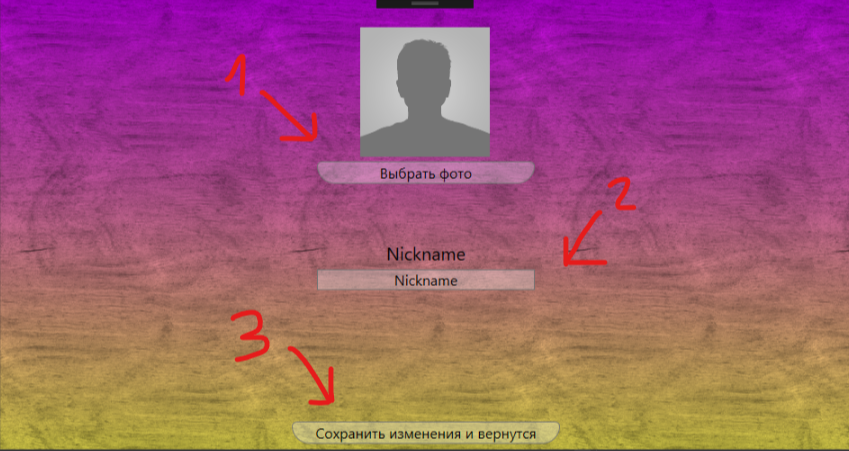
Макеты отсутствуют в данном приложении так как были уже утеряны в связи выполнениям их на листе бумаги.

Разработка инсталлятора

Инсталлятор был создан по инструкции в приложении “InstallShield 11”, в инсталлятор были помещены все файлы необходимой для полной работоспособности разрабатываемого приложения. На данном этапе был выбран и отредактирован значок ярлыка для приложения.

Инструкция пользования

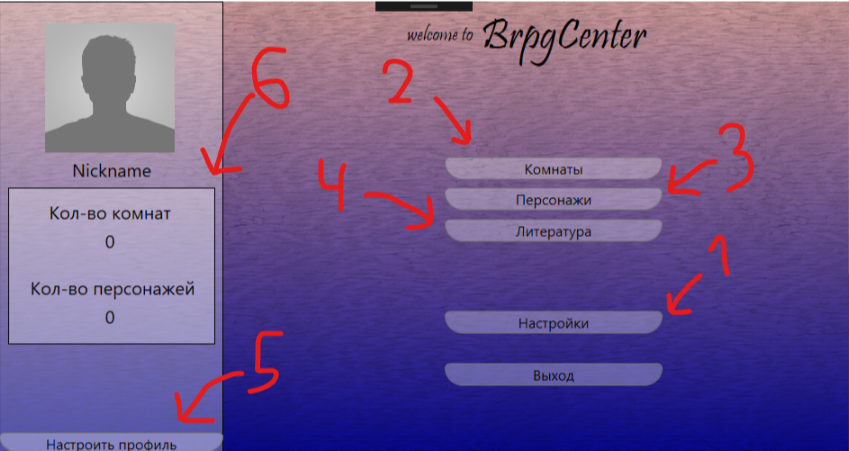
Объяснение будет проходить следующим образом, сперва идет скриншот с стрелками и нумерациями затем под скриншотом следует объяснение.



1-выбираете фото профиля.

2-вводите желаемый Nickname.

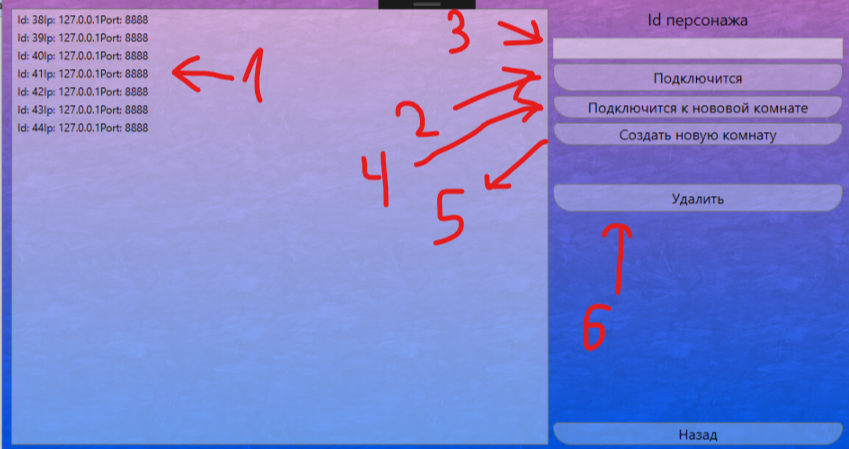
3-сохраняете изменения.



1-настройки помогут вам настроить приложение под себя.

2-раздел комнаты предназначен для подключения к друзьям или создания сервера.

3-раздел персонажи отвечает за создание и хранение созданных вами персонажей.



1-список комнат, сюда сохраняются все комнаты, к которым вы подключались.

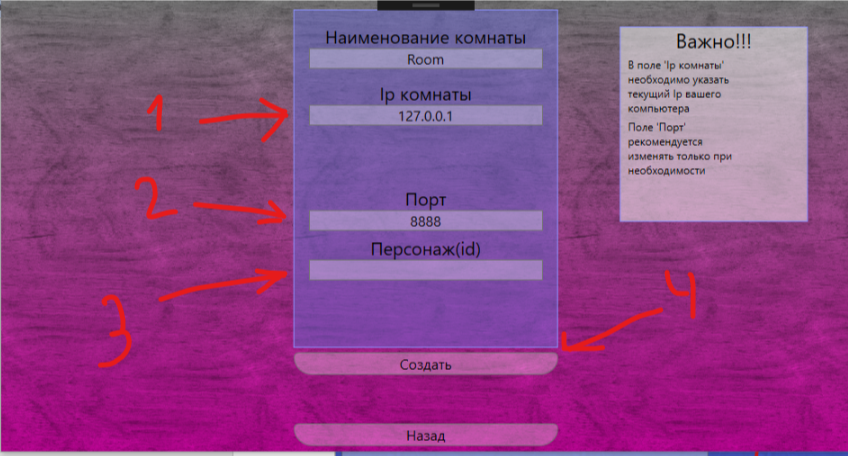
2-вы можете подключится к выбранной из списка комнате, но для этого необходимо сперва заполнить поле “Id персонажа”.

3-данное поле отвечает за то какой персонаж будет направлен в комнату от вашего лица.

4-подключение к новой комнате, перенаправляет вас на новую страницу где нужно заполнить поля.

5-перенаправляет вас на страницу создания сервера.

6-удаляет комнату из списка.

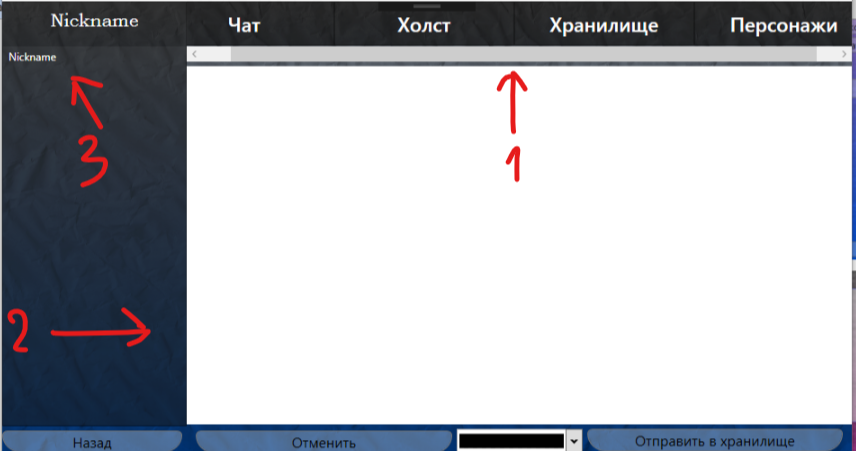


1-“Ip комнаты” вам необходимо указать текущий Ip вашего ПК.

2-указывается порт по которому будет проходить соединение.

3-персонаж под которым вы будете играть.

4-как только вы заполните все поля, можно смело нажимать.



1-вкладки, с помощью вкладок можно удобно переключатся между всеми функциями комнаты.

2-контент вкладки, сюда помещается функционал текущей вкладки будь это чат или хранилище.

3-список игроков, находящихся в комнате.

Инструкция пользования завершена.

Вывод

И вот документация подходит к концу и пришло время вывода, какую выгоду я извлек из данного проекта и какие потери понес, будет описано именно в данном разделе.

По мере работы над данных проектом я сталкивался с разными проблемами и ошибками разные задачи предо мной стояли, и вот он тот самый, ожидаемый финиш, мог ли я сделать функционал лучше или просто сделать его больше, определенно мог, но под конец забега сил уже не то что бы было. Определенно были трудности с тем что я просто устал от этого проекта где-то в середине работы, мне стало просто лень его писать из-за того, что я столкнулся с проблемой написания клиент-серверной части. Я просто не мог сообразить, как правильно можно и нужно отслеживать поставку данных от клиента к серверу, и в итоге спустя примерно неделю негодований было решено снести всю клиент-серверную часть и переписать с нуля, честно говоря в тот момент я думал, что никогда ее не напишу, я был просто обескуражен. Но клиент-серверная часть все-таки была написана и всего за дня два, новая архитектура классов давала легче понимать написанный код. Я писал все составляющие клиента и сервера по отдельности и сразу же их проверяя жутко радовался тому что не все потеряно, именно так я и осилил клиент-серверную часть, пожалуй, все-таки самую сложную часть для меня. Больше особых трудностей не возникало, если и были ошибки, то исправлялись они максимум за час.

Литература

Сетевое программирование - <https://lecturesnet.readthedocs.io/net/intro.html>

ADO.NET - <https://ru.wikipedia.org/wiki/ADO.NET>

WPF - <https://ru.wikipedia.org/wiki/Windows_Presentation_Foundation>

Введение - <https://nauchniestati.ru/blog/vvedenie-v-referate/>