Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра программного обеспечения информационных технологий

Дисциплина: Компьютерные системы и сети(КСИС)

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

на тему

Игровое приложение

БГУИР КП I–40 01 01 038 ПЗ

Выполнил

студент: гр. 151004 Ермолович Д.О.

Проверил: Красковский П.Н.

Минск 2022

содержание

[Введение 3](#_Toc134132251)

[1 Аналитический обзор литературы 4](#_Toc134132252)

[1.1 Анализ существующих аналогов 4](#_Toc134132253)

[1.1.1 BGC 4](#_Toc134132254)

[1.2 Перечень функциональных требований 5](#_Toc134132255)

[1.3 Состав и параметры технических и программных средств 5](#_Toc134132256)

[2 Моделирование предметной области 6](#_Toc134132257)

[2.1 Архитектура «клиент-сервер» 6](#_Toc134132258)

[2.2 Протоколы TCP и UDP 7](#_Toc134132259)

[2.3 Отрисовка 8](#_Toc134132260)

Введение

Клониум – это игра, в которой поле состоит из квадратных ячеек. Игровое поле может быть различной формы (круг, квадрат, крест). Стандартное квадратное поле состоит из 36 ячеек (6\*6).

В начале игра у игроков есть по одной фишке. На фишках есть точки и у каждого игрока она своего цвета. Их может быть от 1 до 4.

В начале на фишке 3 точки. Когда на фишке становится 4 точки она «взрывается» и от ячейки где была эта фишка соседним фишкам по вертикали и горизонтали прибавляется одна точка. Если соседняя ячейка была пустая, то на ней появляется фишка с одной точкой. Также если на соседней ячейке была фишка с тремя точками к ней прибавится одна точка и она тоже «взорвется». Так могу вызываться цепные реакции. Если на соседней ячейке была вражеская фишка она становится союзной (меняет цвет) с тем же количеством точек плюс один.

Каждый игрок ходит по очереди. Время на ход неограниченно. Игра заканчивается когда на поле все фишки одного цвета. Тот игрок, чьи фишки остались на поле становится победителем.

# Аналитический обзор литературы

## Анализ существующих аналогов

Аналог данной курсовой работы мобильная игра BGC.

BGC это каталог игр, одна из которых клониум

### BGC

Рисунок 1.1 – Интерфейс BGC

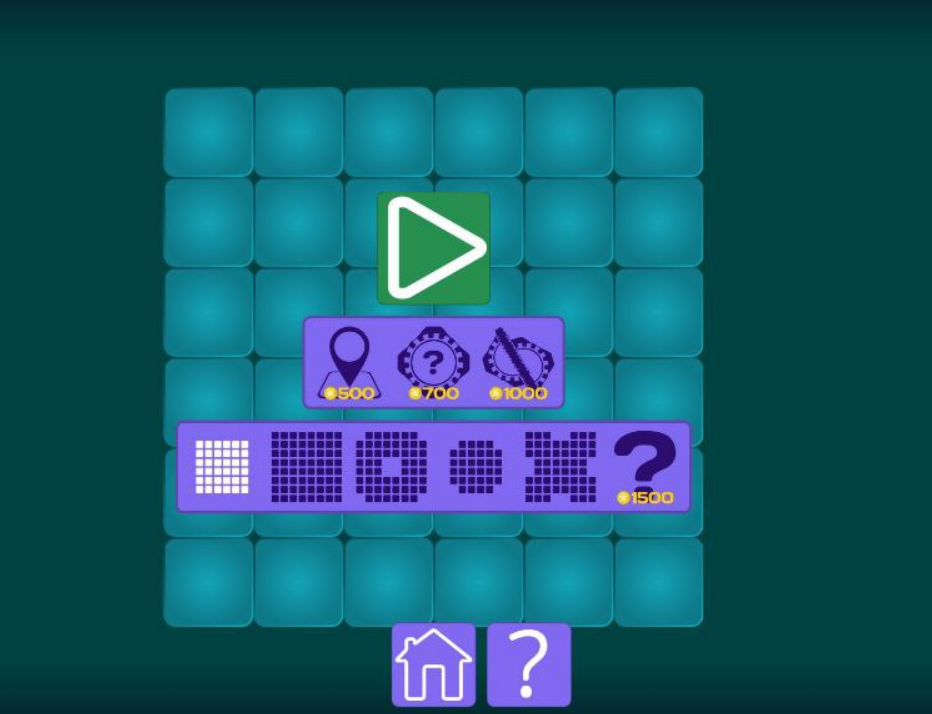


Рисунок 1.2 – Интерфейс игры клониум

Плюсы:

* Удобный интерфейс;
* Низкие требования к устройству;
* Захватывающий геймплей;
* Игра развивает стратегической мышление и учит думать наперед.

Минусов не было выявлено.

## Перечень функциональных требований

После анализа аналогов был составлен следующий список выполняемых функций:

1. Удобный интерфейс
2. Возможность играть сети

Для разработки программного средства будет использоваться язык программирования C# и среда разработки visual studio.

Для UI будет использоваться WindowForms как простой и целесообразный способ создать окно с кнопками.

## Состав и параметры технических и программных средств

Приложение должно функционировать на персональных компьютерах со следующими характеристиками:

* процессор Intel Celeron N4020 или лучше;
* оперативная память 128 MB или лучше;
* свободное место на диске в размере 3 мегабайта или больше.

Приложение должно функционировать в окружении операционной системы Windows XP или новее.

В данном разделе указаны минимальные технические требования для запуска программного средства. Для эксплуатации в реальных могут потребоваться более мощные технические средства. Программное средство должно корректно функционировать и на более мощном оборудовании.

# Моделирование предметной области

## Архитектура «клиент-сервер»

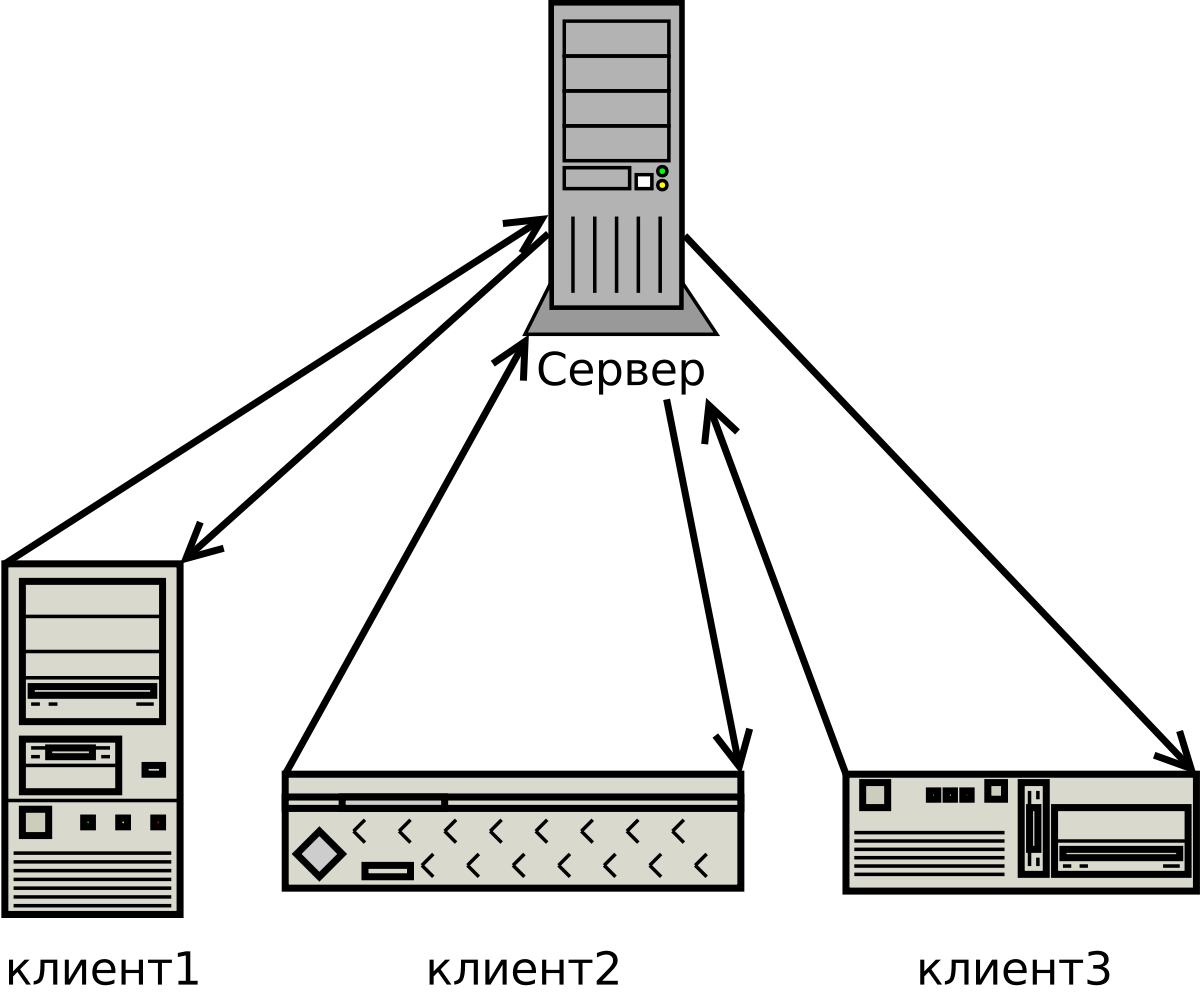


Рисунок 2.1 – Архитектура «клиент-сервер»

Архитектура «клиент-сервер» – это вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами. Серверы предоставляют различные функции или сервисы клиентам, которые инициируют запросы на такие сервисы посредством сетевых протоколов. Клиенты могут быть веб-браузерами, мобильными приложениями или другими программами, которые используют ресурсы или данные, предоставляемые серверами. Например, когда вы заходите на веб-сайт, ваш браузер отправляет запрос на веб-сервер, который отвечает на этот запрос, отправляя вам данные. Архитектура «клиент-сервер» имеет ряд преимуществ:

* серверы могут обслуживать множество клиентов одновременно;
* серверы могут хранить и обрабатывать данные в одном месте, что упрощает управление и обеспечение безопасности данных;
* сервер может обеспечивать взаимосвязь между клиентами.

Однако архитектура «клиент-сервер» также имеет некоторые недостатки:

* клиенты не могут работать без подключения к серверам и зависят от их доступности и производительности;
* серверы могут быть перегружены запросами от большого числа клиентов и стать узким местом в системе;
* серверы могут быть атакованы злоумышленниками, которые пытаются получить доступ к данным или нарушить работу системы.

## Протоколы TCP и UDP

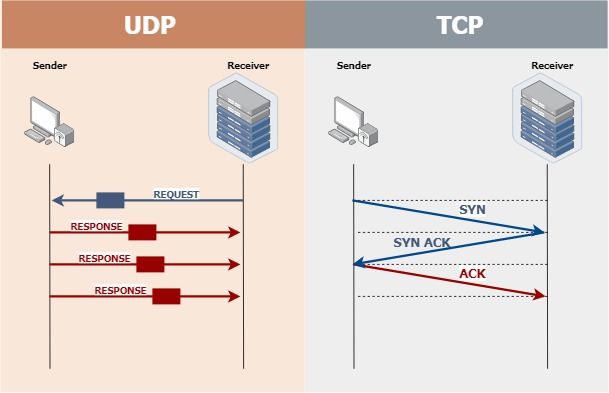


Рисунок 2.2 – Протоколы TCP и UDP

В играх чаще применяется UDP, так как он обеспечивает более высокую скорость и меньшую задержку передачи данных, чем TCP. UDP не гарантирует доставку и порядок пакетов, но это не критично для реального времени, где важнее актуальность информации. TCP же может приводить к замедлению и сбоям в игре из-за повторной отправки потерянных пакетов и автоматического контроля пропускной способности. Однако UDP требует дополнительной работы по реализации некоторых функций TCP, таких как надежность, упорядоченность и шифрование данных. Но поскольку шахматы – пошаговая игра, не требующая высокой актуальности данных, то протокол TCP будет более целесообразен.

В случае с TCP порт клиента по умолчанию – любой свободный (не менее 2000, не более 65535), работает на транспортном уровне модели OSI, что обеспечивает надежность и целостность передаваемой информации.

Протокол TCP предусматривает работу в соединительном режиме, т.е. в режиме “установка соединения – передача данных – разрыв соединения”. Является протоколом точка-точка, что означает, что взаимодействие происходит между двумя конечными устройствами.

Данный протокол хорошо подходит для передачи информации, доставка которой требует подтверждения получения и контроля ошибок.

## Отрисовка

Поле и фишки являются картинками и выводятся напрямую на форму.

Для этого используется класс Graphics из стандартной библиотеки c# System.Drawing. Сами картинки отрисовываются в событии формы под названием Paint. Для вывода самих картинок используется функция DrawImage. Для обновления формы используется функция Refresh.