**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Владимирский государственный университет

имени Александра Григорьевича и Николая Григорьевича Столетовых» (ВлГУ)

**РАЗРАБОТКА ПРОТОТИПА МНОГОПОЛЬЗОВАТЕЛЬСКОЙ ИГРЫ**

Пояснительная записка

На // листах

|  |  |  |
| --- | --- | --- |
| Руководитель |  | к.т.н. доцент кафедры ИЗИ Ю.М. Монахов |
| Исполнитель |  | студент гр. ИСБ-122 Юрастов Н.А. |
|  |  |  |

Владимир 2023

**Аннотация:**

**Название проекта:**

Клон браузерной игры Agar.IO.

**Функциональность:**

* Локальный или многопользовательский режим игры
* Смена имени

**Описание:**

Данный технический документ описывает проект клона браузерной игры Agar.IO. Игра реализована на языке python и выполняется в python.

Игровой процесс предоставляет пользователям возможность участвовать в сетевых соревнованиях, а серверная часть отвечает за координацию и синхронизацию игровой сессии. Коммуникация между игроками осуществляется через TCP-сокеты, которые гарантируют доставку сообщений в том порядке, в котором они были отправлены.

Техническая архитектура включает клиентскую часть, реализованную на языке python, а также серверную часть, также реализованную на языке python.

ОГЛАВЛЕНИЕ

[АННОТАЦИЯ……………………………………………………………………..2](file:///C:\Users\1\Downloads\Telegram%20Desktop\Documentation.docx#_Toc1)

Основные требования…………………………………………………………….4

Проработка тематики, механики и архитектуры игры…...…………………….5

Разработка моделей игроков и объектов..………………………………………6

Разработка сервера………………………………………………………………..8

UML – диаграмма последовательности общения между клиентами и сервером………………………………………………………………………….12

Разработка протокола обмена данными, клиент-сервер………………………13

Инструкция для пользователя…………………………....……………………..14

Инструкция для разработчика…………………………………………………..15

Ссылка на репозиторий………………………………………………………….18

Список литературы………………………………………………………………19

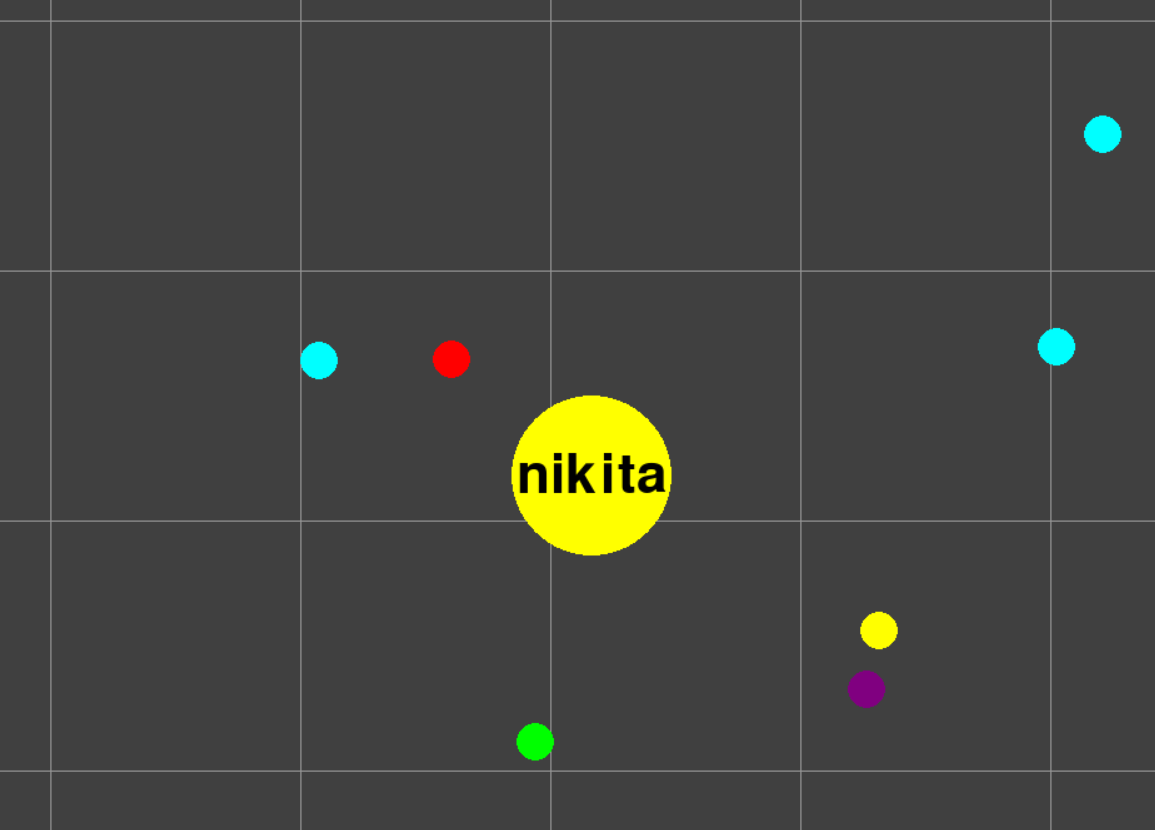
**Основные требования**

Разработка будет проводиться в соответствии со следующими требованиями:

1. Должны присутствовать клиентская и серверная часть.
2. Игра должна быть многопользовательской.
3. Должна присутствовать возможность играть как по локальной сети, так и при помощи удалённого сервера.
4. Должен присутствовать обмен данными между клиентом и сервером.
5. Должны быть реализованы функции из оригинальной игры Agar.IO.

**Проработка тематики, механики и архитектуры игры:**

1.Тематика

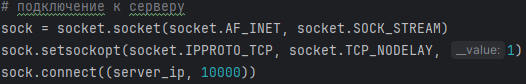
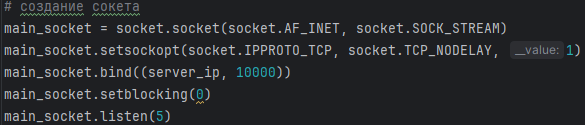


Тематикой игры является браузерная игра Agar.IO, но с некоторыми изменениями. В созданной игре нельзя победить, так как поле является квадратным, а игрок окружностью, и он не может занять всё пространство поля. В случае смерти, игрок будет вынужден перезапускать клиент или же exe файл. Отсутствует подсчёт очков, нужный для победы.

2. Механика

Есть игровое поле, на котором находятся сами игроки, корм и боты. Игроки и боты могут свободно перемещаться по осям. Игроки и боты могут поглощать корм для набора массы, если у бота или игрока масса будет больше, чем у другого бота или игрока, то они смогут поглотить и его. Боты подражают поведению игрока.

3. Клиент-серверная архитектура

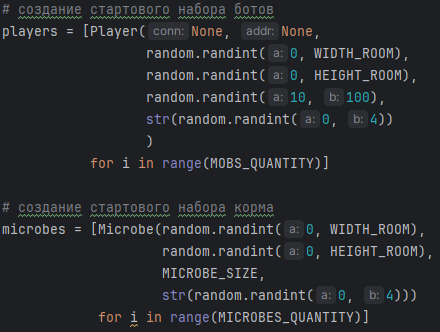


Сокет – это переменная отвечающая за работу порта нашего компьютера.

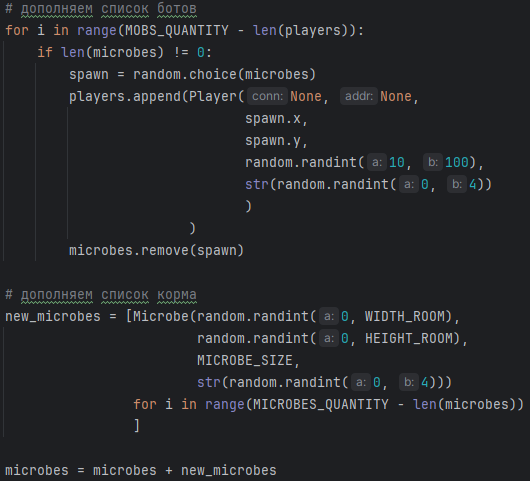
Сокет отключает алгоритм Нейгла, это нужно для того чтобы сокет не упаковывал данные в большие пакеты и отсылал их в нужное время. На картинках показана только основная часть создания клиент-серверного соединения и настройка сокетов, чтобы было возможно подключение и обмен информацией.

**Разработка моделей игроков и объектов:**

Всего в игре существует 2 вида игровых моделей: игроки (наш игрок, боты и другие игроки), а также корм.

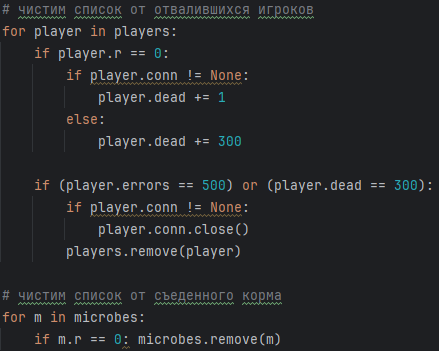


При запуске сервера создаётся базовый набор корма и ботов. При желании их количество можно увеличить или сократить.

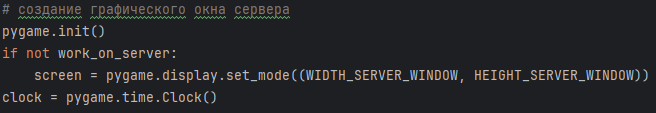


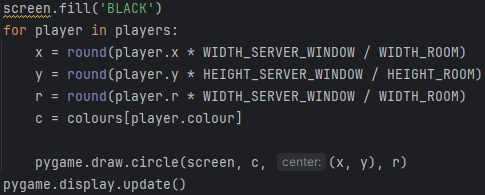
Для того чтобы поле всегда было заполнено ботами и кормом мы дополняем их список каждую часть цикла.

Дополнительно мы отчищаем список от отключившихся или погибших игроков, а также список съеденного корма.

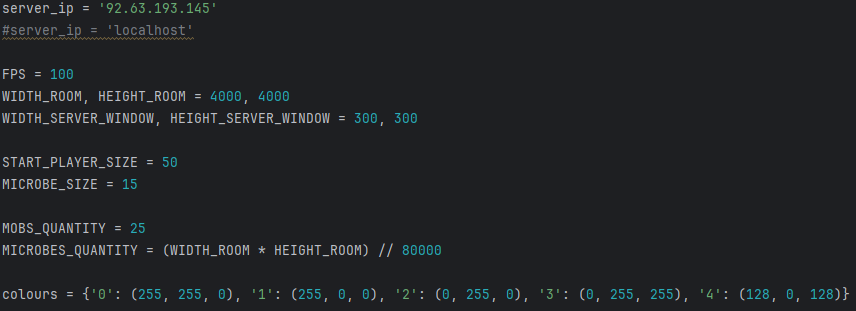


**Разработка сервера:**





При помощи библиотеки pygame создается графическое окно севера, на котором будут отображены все объекты и игроки.



Тут показаны все глобальный переменные, которые задействует сервер: server\_ip – уникальный числовой идентификатор устройства в компьютерной сети;

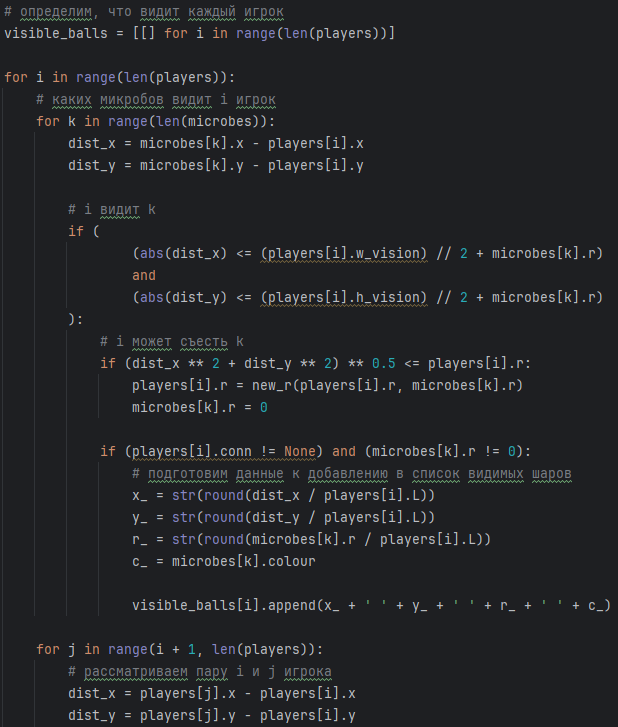
WIDTH\_ROOM, HEIGHT\_ROOM – ширина и высота игрового поля; WIDTH\_SERVER\_WINDOW, HEIGHT\_ SERVER\_WINDOW – ширина и высота серверного окна;

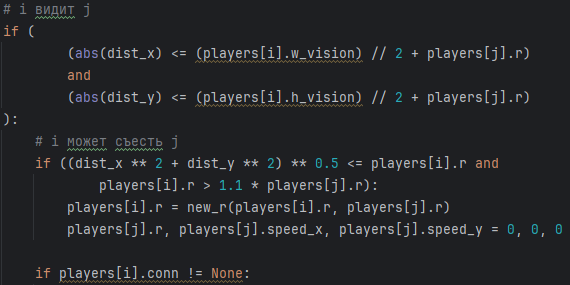
START\_PLAYER\_SIZE – стартовый размер игрока;

MICROBE\_SIZE – размер корма;

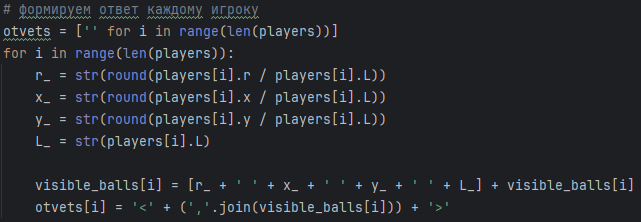
MOBS\_QUANTITY и MICROBES\_QUANTITY – количество ботов и корма на поле;

colours – цвета, которые может получить каждый объект.



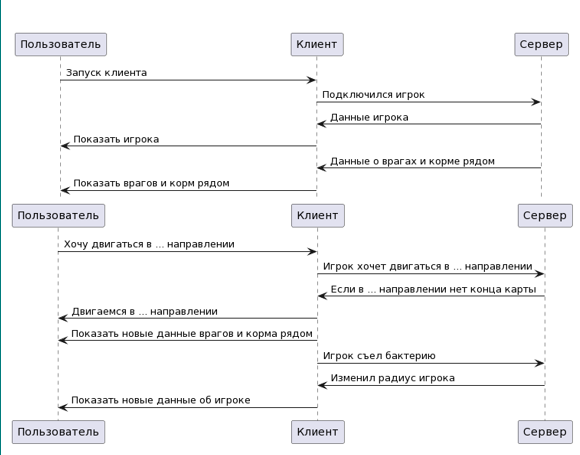


Данные модули ответственный за определение объектов, видимых игроку, а также возможности их уничтожения.



Данный участок кода формирует пакет данных и расположение видимых шаров.

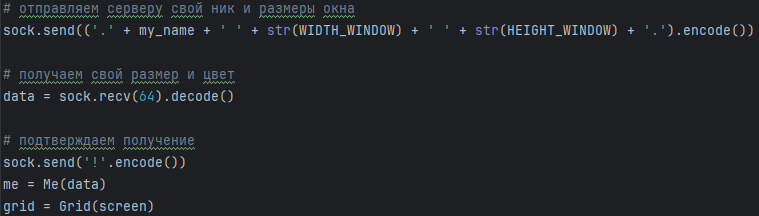
**UML – диаграмма последовательности общения между клиентами и сервером**



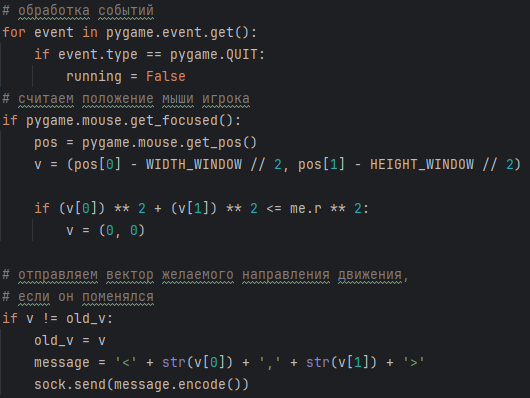
**Разработка протокола обмена данными, клиент-сервер:**



Данный модуль отвечает за считывание команд, отправленных игроком.



Эти строки ответственны за отправку клиентских данных на сервер.

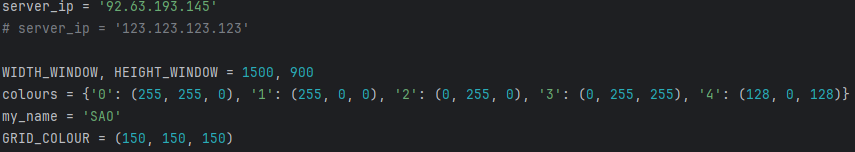


Данный модуль представляет собой обработчик игровых событий. Он считывает положение мышки игрока относительно поля и отправляет на сервер вектор желаемого игроком перемещения, если игрок меняет своё положение относительно поля. Если целевой игрок не осуществляет никаких действий с мышкой, следовательно положение его бактерии остается статичным относительно поля и на сервер не приходит сообщение с вектором желаемого перемещения.

**Инструкция для пользователя:**

Для запуска игры необходимо открыть клиент с помощью Python IDLE или приложения pycharm.

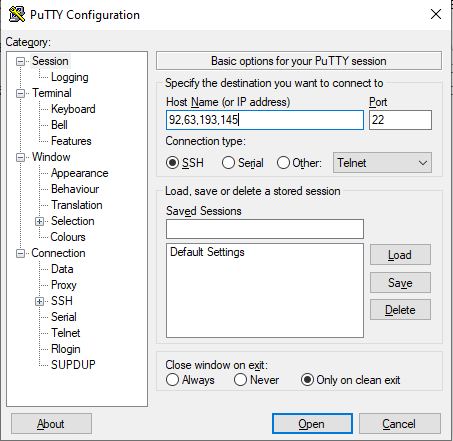
В открытом окне можно изменить имя пользователя (my name), выбрать тип подключения (сервер или локальная сеть), изменить цветовую палитру сущностей (colours), изменить цвет сетки поля (grid colour), изменить разрешение экрана (width and height windows):



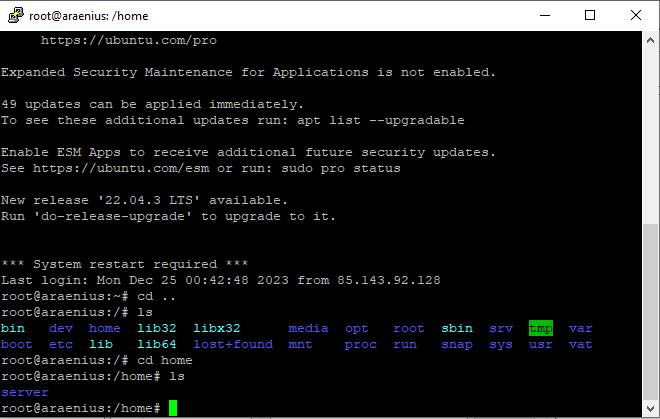
После настройки клиента, пользователь может запустить игру с помощью кнопки F5 или же в случае использования pycharm сочетание клавиш shift+F10.

**Инструкция для разработчика:**

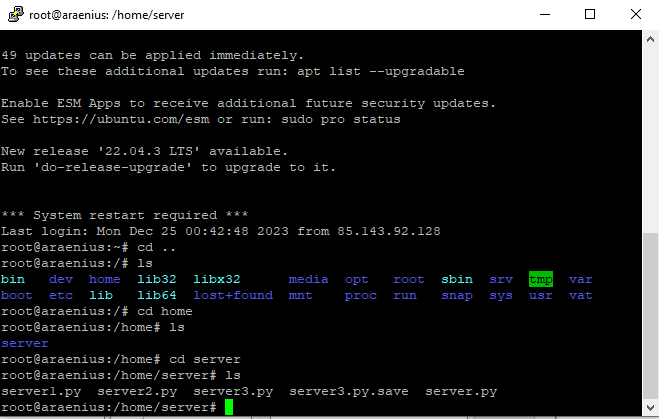
В проекте также реализовано подключение к удалённому серверу. Оно нужно для того, чтобы можно было подключить клиенты с разных компьютеров к одному серверу, который загружен на хост. Для этого я использовал VDS сервер. VDS (Virtual Dedicated Server, виртуальный выделенный сервер) – это услуга хостинга, при которой клиенту выделяется виртуальный сервер целиком с полными административными правами, которые дают возможность установить на сервер любое программное обеспечение. При помощи программы PuTTY я установил подключение к удалённому серверу.



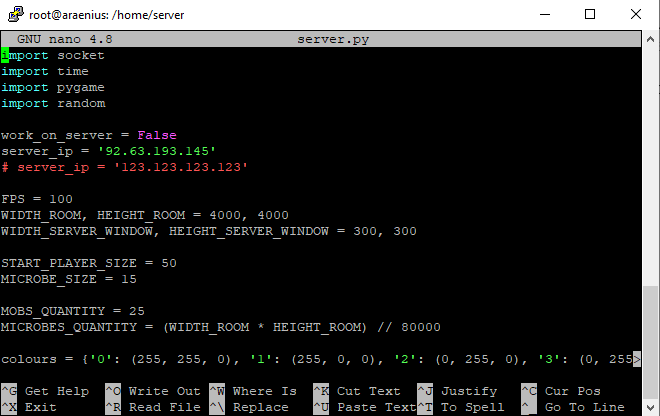
Затем в папке home, я создал папку server.



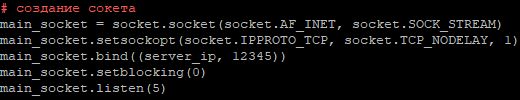
В папке сервер я создал файл server.py.



В файле server.py я скопировал код серверной части.

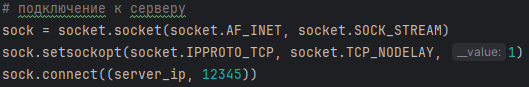


В строке server\_ip я указываю ip своего VDS сервера, а в строке main\_socket.bind значение порта указываю 12345

.

Это все настройки VDS сервера. В клиентской части в строке server\_ip указываю ip своего VDS сервера, а в строке sock.connect значение порта указываю 12345.





Для запуска файла server.py на VDS сервере надо указать команду python3 server.py. А для подключения к серверу надо запустить client.py при помощи pycharm или python IDLE.

**Ссылка на репозиторий**

(https://github.com/omax137/AgarIO)

**Список литературы**

# [Библиотека](https://habr.com/ru/articles/588605/) Pygame