**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 Захаров.К.П. |
|  |  |  |

Владимир 2023

# АННОТАЦИЯ

В данном документе приведено описание разработки многопользовательской онлайн игры на BacterIO. Игра реализована на языке Python с использованием библиотек pygame. Суть игры состоит в реализации многопользовательской IO игры, чтобы человек из любой любой точки мира мог подключиться к игровой сессии и поучаствовать в массовом сражении за самую большую и доминирующую бактерию.

Разработка прототипа многопользовательской игры состоит из следующих этапов:

* Проработка тематики, механики и архитектуры игры
* Разработка моделей игроков и объектов
* Разработка сервера
* Разработка протокола обмена данными, клиент-сервер
* Разработка механики и классов объектов в игре
* Тестирование и отладка

ОГЛАВЛЕНИЕ

[АННОТАЦИЯ……………………………………………………………………………………………..2](#_Toc1)

1 РАЗРАБОТКА ПРОТОТИПА МНОГОПОЛЬЗОВАТЕЛЬСКОЙ ИГРЫ…....4

1.1 Основные требования…………………………………………………………4

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

1.3 Разработка моделей игроков и объектов…………………………………….7

1.4 Разработка сервера……………………………………………………………9

1.5 Разработка протокола обмена данными, клиент-сервер…………………..10

1.6 Разработка клиента…………………………………………………………..11

1.7 Разработка механики и классов объектов в игре………………………..…12

1.8 Тестирование и отладка………………………………………………………

2 ПРОВЕРКА ВЫПОЛНЕНИЯ ТРЕБОВАНИЙ К ИГРЕ……………………..

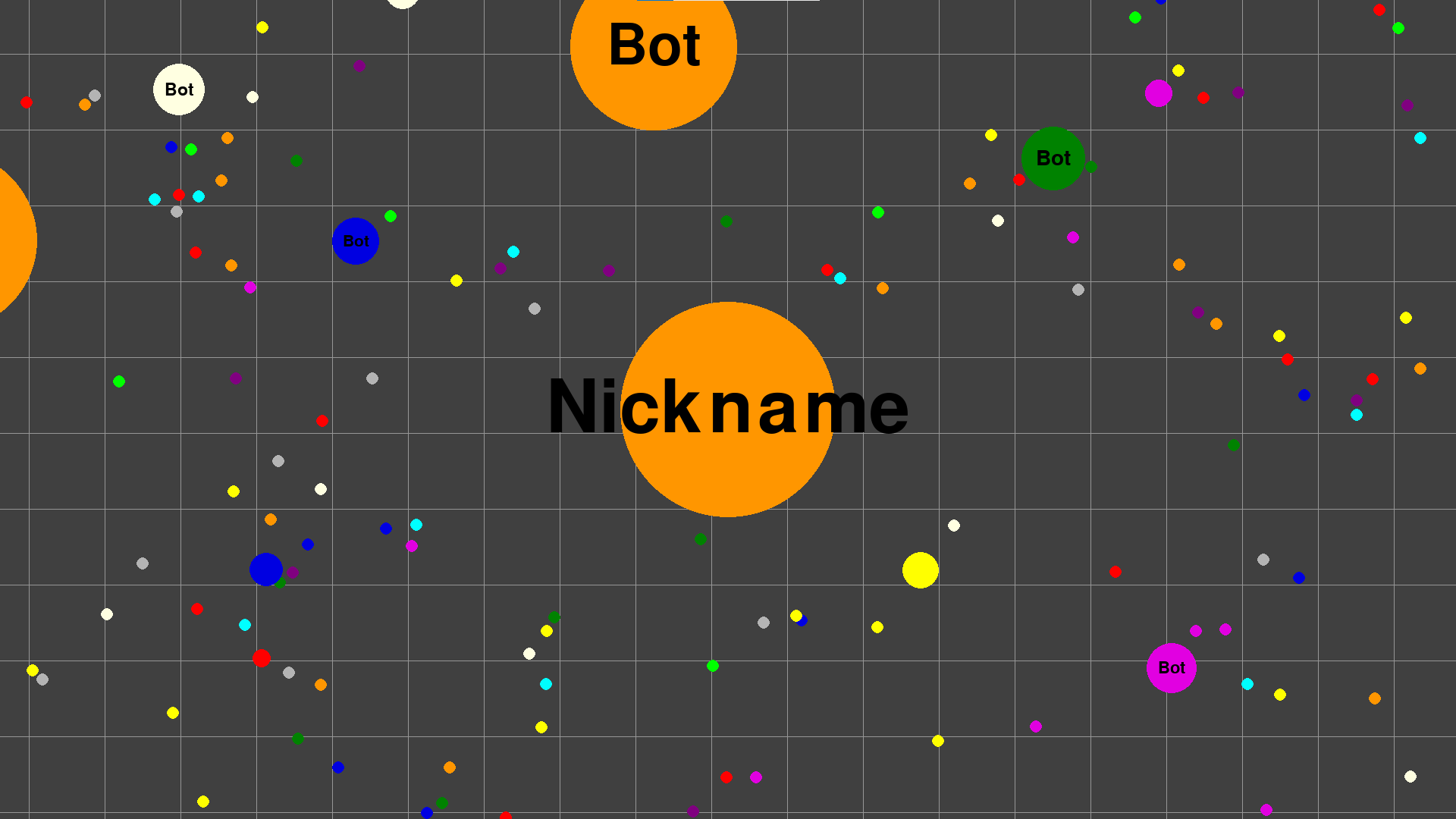
**1** РАЗРАБОТКА ПРОТОТИПА МНОГОПОЛЬЗОВАТЕЛЬСКОЙ ИГРЫ

## **Основные требования**

Разработка будет производиться в соответствии со следующими требованиями:

1. Должны присутствовать клиент и сервер;
2. Игра обязана быть многопользовательской, т.е. чтобы любой пользователей мог присоединиться к определенному IP и начать игру;
3. Должен присутствовать обмен данными между клиентом и сервером;
4. Должны быть реализованы функции соответствующие AgarIO.
   1. **Проработка тематики, механики и архитектуры игры**

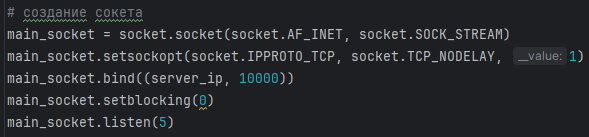
1) Тематика

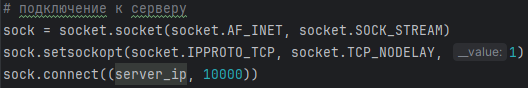


Тематика игры классическое AgarIO, в которой нельзя победить, игроки будут всегда заходить сражаться и в случае смерти пере заходить на сервер, нет никакого подсчета очков, для победы.  
  
 2) Механика

У игроков есть поле в котором они могут свободно передвигаться по всем осям и уничтожать противников с меньшей массой. Каждый игрок может свободно поглощать других игроков, если его масса больше. Игроки так же могут уничтожать микробов и ботов, которые подражают поведению игрока.

3) Клиент-серверной архитектуры

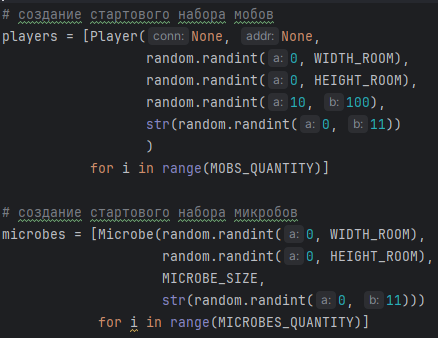




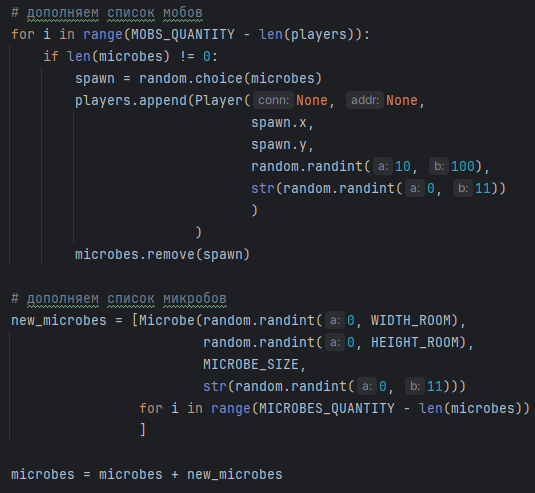
Сокет – это переменная отвечающая за работу порта нашего компьютера. При помощи библиотеки socket создается объект, который работает по протоколу TCP. Сокет отключает алгоритм Нейгла, чтобы сокет не упаковывал данные в большие пакеты и отсылал их в нужное время. На картинках показана только основная часть создания клиент серверного соединения и настройка сокетов, чтобы можно было подключаться и обмениваться информацией.

* 1. **Разработка моделей игроков и объектов**

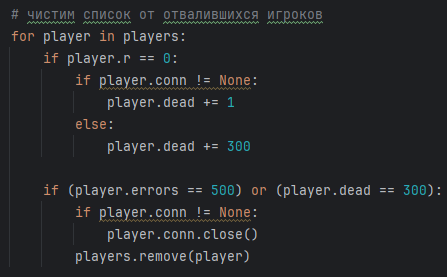
Всего в игре существует 4 вида сущностей: наш игрок; другие игроки; боты, выполняющие роль замены обычных игроков, копирующие их поведение; микробы, выполняющие роль корма для набора массы.

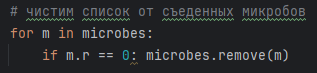


При запуске сервера создается базовый набор мобов и микробов. При желании их количество можно увеличить или сократить.



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

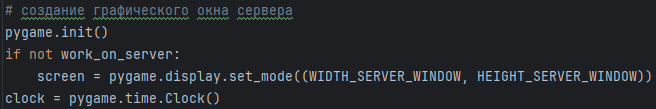


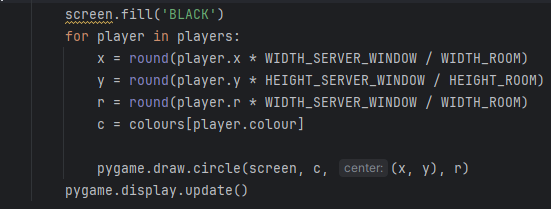


Дополнительно мы отчищаем список от отключившихся или погибших игроков, а так же список съеденных микробов.

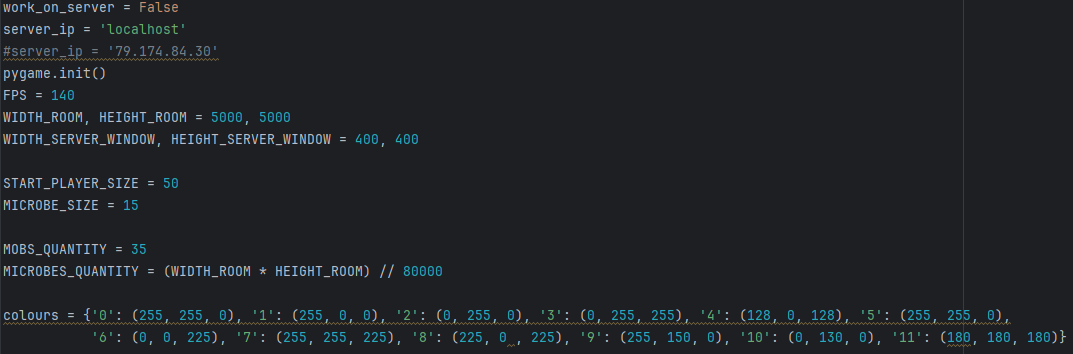
Подробнее о классовой модели будет рассказано ниже.

* 1. **Разработка сервера**

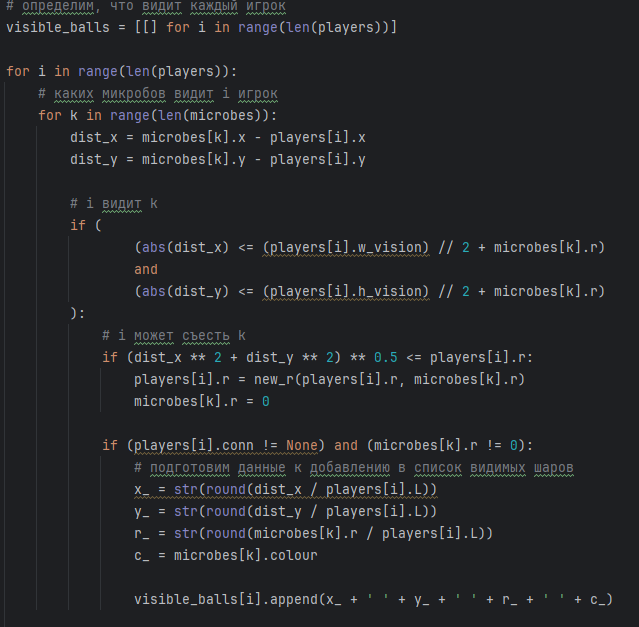
****

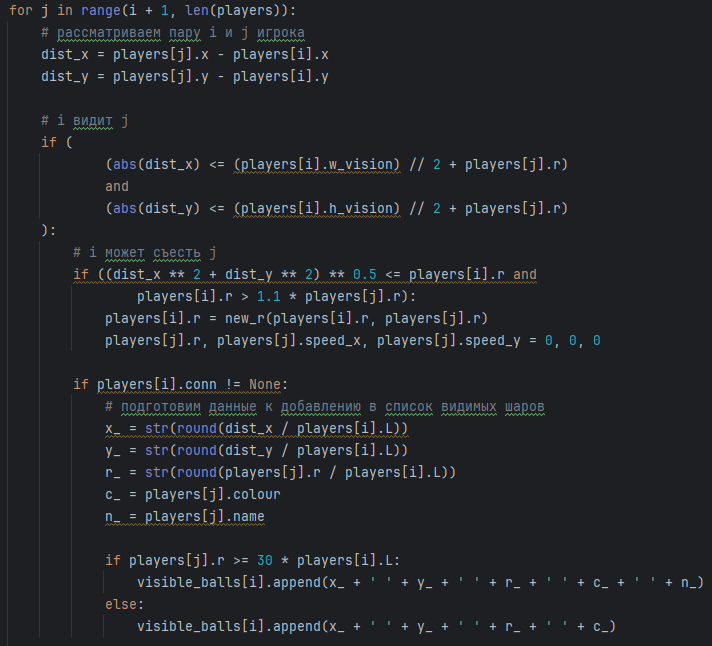
****

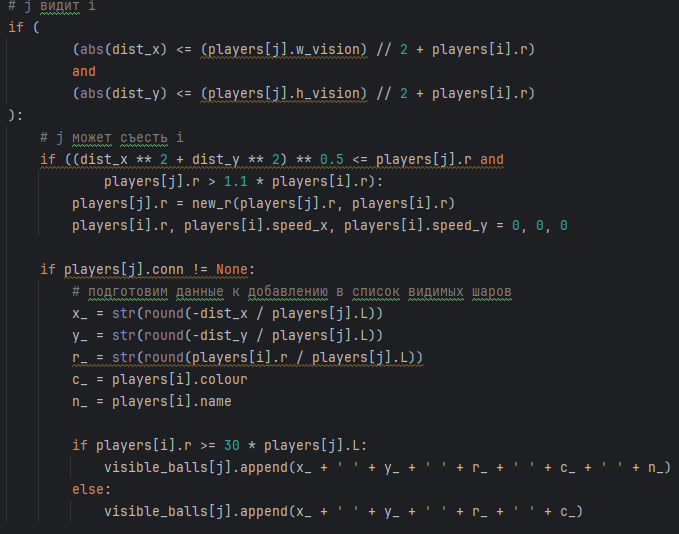
С помощью библиотеки pygame создается графическое окно севера, на котором будут отображены все объекты и игроки.



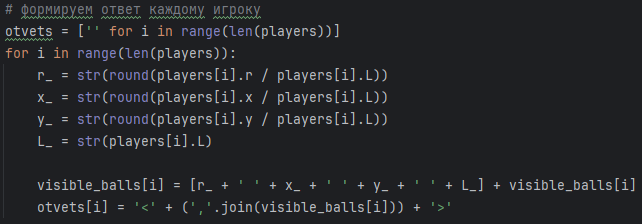
Тут показаны все глобальный переменные, которые задействует сервер: server\_ip – уникальный числовой идентификатор устройства в компьютерной сети; FPS – количество кадров в секунду; WIDTH\_ROOM, HEIGHT\_ROOM – ширина и высота игрового поля; WIDTH\_SERVER\_WINDOW, HEIGHT\_ SERVER\_WINDOW – ширина и высота серверного окна; START\_PLAYER\_SIZE – стартовый размер игрока; MICROBE\_SIZE – размер микроба; MOBS\_QUANTITY и MICROBES\_QUANTITY – количество ботов и микробов на поле; colours – цвета, которые может получить каждый объект.







Данные модули ответственный за определение объектов видимых игроку а так же возможности их уничтожения.

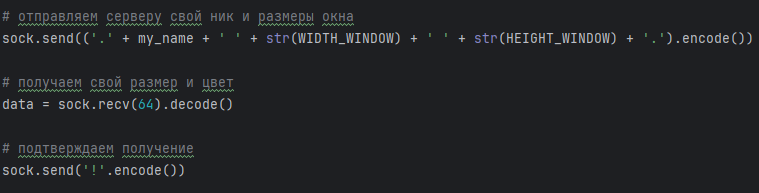
****

Данный участок кода формирует пакет в виде [r x y L] и расположение видимых шаров.

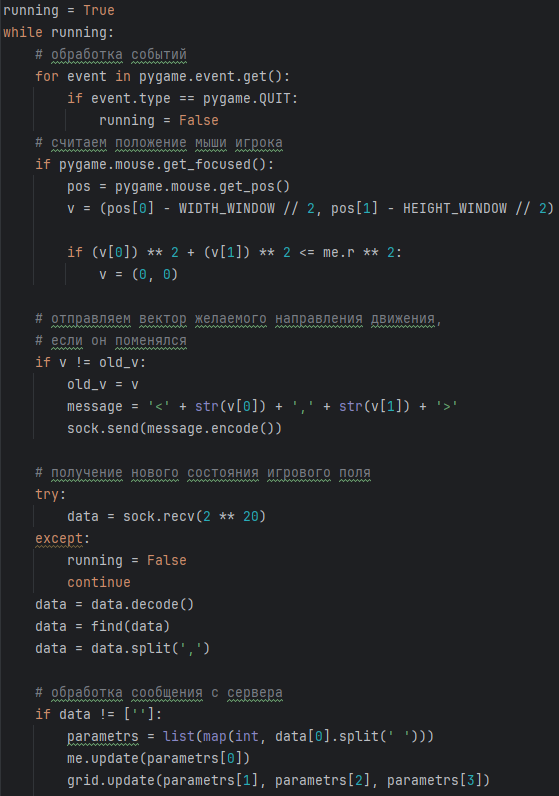
* 1. **Разработка протокола обмена данными, клиент-сервер**



Данный модуль отвечает за считывание команд, отправленных игроком, точнее получает имя и размеры окна игрока и вектор желаемого направления перемещения, выбранный игроком.

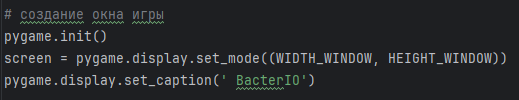
****

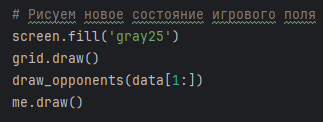
Эти строки ответственны за отправку клиентских данных на сервер и их обработку.



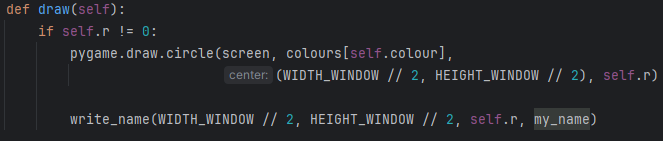
Данный модуль представляет собой обработчик игровых событий, так называемых event –ов. Он считывает положение мышки игрока относительно поля и отправляет на сервер вектор желаемого игроком перемещения, если игрок меняет своё положение относительно поля. Если целевой игрок не осуществляет никаких действий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

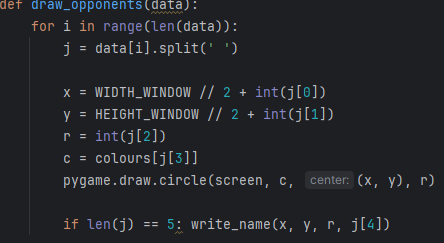
**1.6 Разработка клиента**





С помощью библиотеки pygame создаем клиентское игры, а также отрисовка игрового поля.

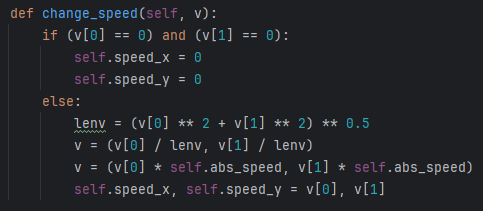


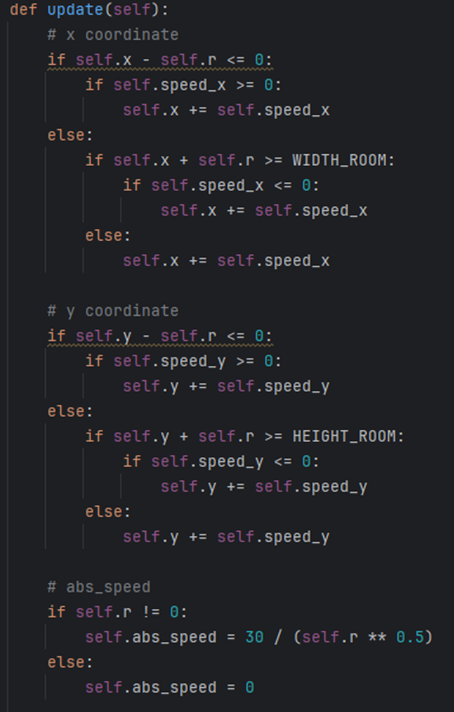


Все отрисовки происходят в основном цикле игры. На скриншотах только функции, которые всегда обновляются и при необходимости запускаются.

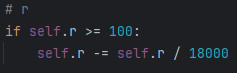
**1.7 Разработка механики и классов объектов в игре**

1. Механика



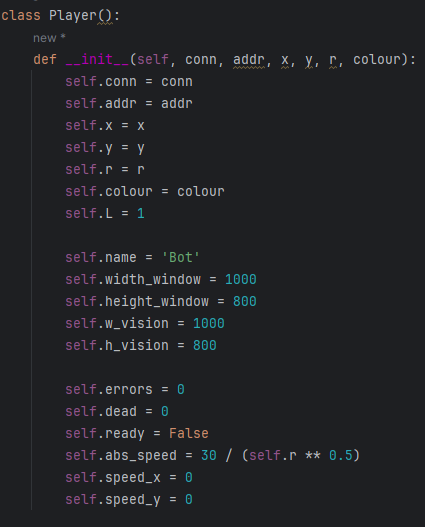


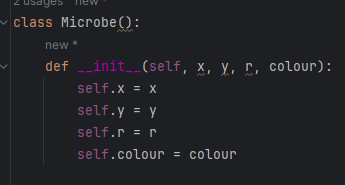
Обработкой движения занимается функция change\_speed и update. Они принимают пакеты, которые им присылает клиент и обрабатывают. Эти функции так же обрабатывают изменение скорости и ограничивает игрока в рамках поля.



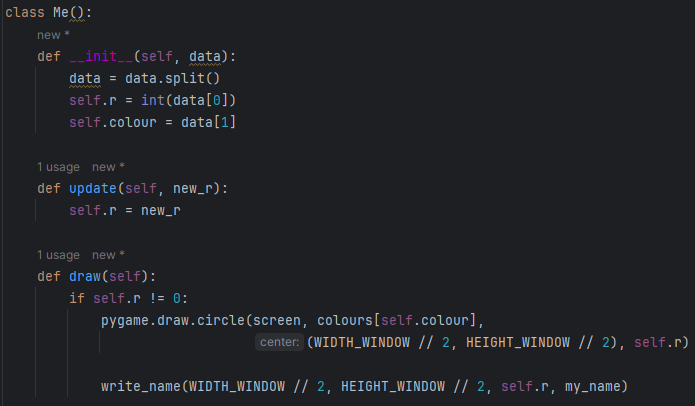
Для внутри игрового баланса была создана функция, которая не позволяет бактерии бесконечно увеличивать свой размер.

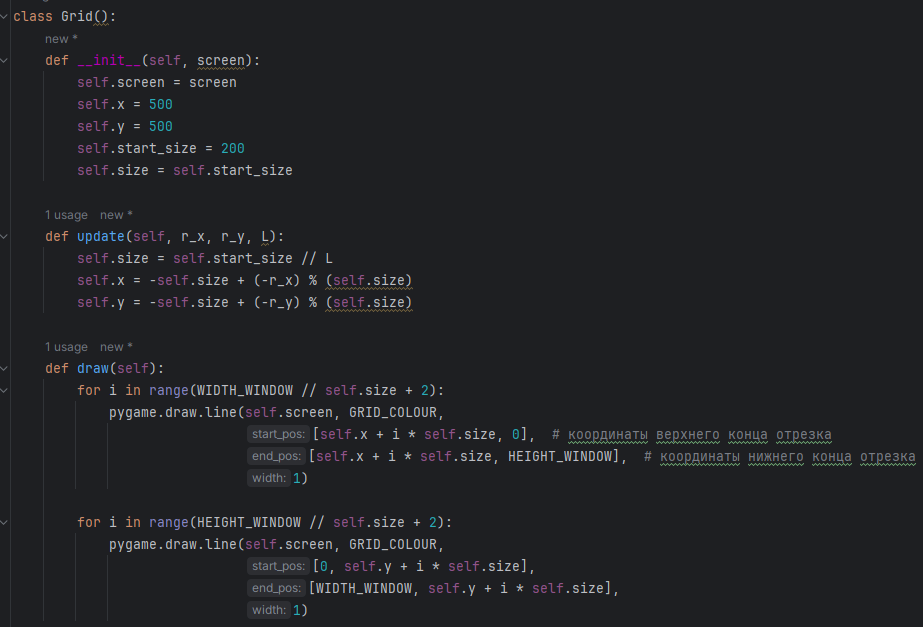
1. Классы





В игре существует три основных класса, сам игрок (player), бот(Bot, подкласс игрока) и микроб(Microbe). Данные модули позволяют создать объекты каждого из классов и задать им параметры такие как: x,y координаты на поле; r – размер объекта; colour – цвет; abs\_speed – абсолютна скорость и тд.





В клиенте сделаны классы Me(), для отрисовки игрока в клиенте,  
класс Grid(), для создания сетки на поле.