

1 Сценарий

Главная цель игры передвигаться по полю, поедая корм и других игроков, которые имеют меньший размер, чем он. После того, как шарик съел некоторое количество, он начинает расти в диаметре. Основная задача выжить и попасть в турнирную таблицу.

При заходе в игру пользователь попадает в меню, где может:

- Зарегистрировать и начать игру;
- Просмотреть игровое поле;
- Подключить бота.

Серверная часть должна отвечать за обработку данных о координатах всех пользователей и рассылать всем клиентам обновленные результаты. Как только сервер будет запущен, он должен ожидать клиентов. Количество клиентов на первоначальном этапе не будет превышать 8 пользователей, включая ботов.

После старта клиента игроку нужно управлять шариком с помощью клавиатуры (↓, →, ↑, ←, s, d, w, a). Поле, по которому движется шарик, ограничено. В режиме демонстрации поля передвижение по карте будет происходить аналогично с помощью стрелок и выше представленных букв. Вместо человека в игре могут соревноваться боты. Для этого бот должен обладать простейшим искусственным интеллектом. Клиенты - боты и люди, могут связываться с сервером (для обмена информации между собой) через API.

2 Архитектура

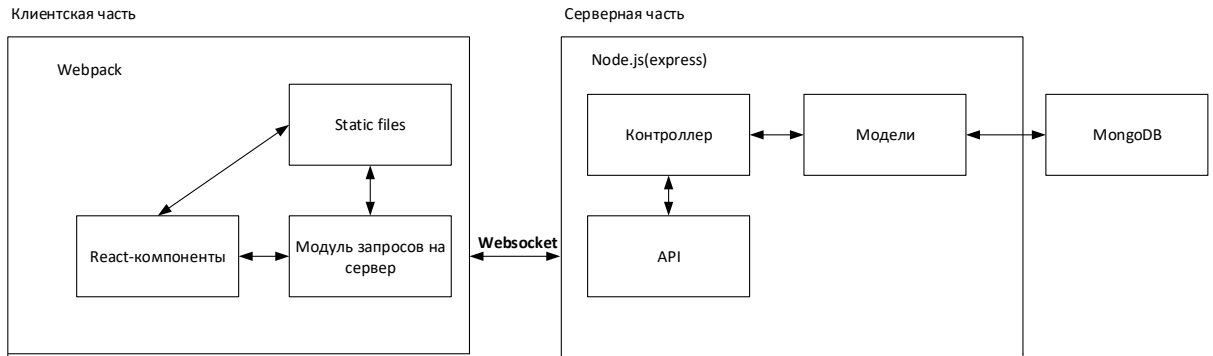


Рисунок 1 — Архитектура приложения

3 Словарь

При разработке игры «Agar.io» используются следующие понятия:

- Игрок;
- Бот;
- Клавиатура;
- Дисплей;
- Игра;
- Рейтинговая таблица;
- Баллы.

3.1 Игрок

Взаимодействует с программой, обеспечивая тем самым выполнение своих целевых функций. Взаимодействие осуществляется через внешние устройства компьютера: клавиатуру и дисплей.

Основные действия игрока можно разделить на:

- Управление ходом игры;
- Регистрация (ввод имени);
- Просмотр игрового поля.

Прямого взаимодействие игрока с программой не происходит. Между ними существует посредник в виде внешних устройств компьютера: клавиатуры, монитора, которые обеспечивают преобразования физических воздействий человека в программные события посредством использования

клавиатуры. Обратная связь осуществляется за счет визуализации изменения состояния программы на экране дисплея.

3.2 Бот

Взаимодействует с сервером через API. Главным отличием от игрока является заложение искусственного интеллекта, в котором будут использованы простейшие алгоритмы передвижения и взаимодействия с другими игроками.

3.3 Игра

Основной программный модуль, решающий целевую задачу. Взаимодействует с клавиатурой, реагируя изменением внутреннего состояния на посылаемые воздействия игрока. Игра является достаточно сложным понятием, которое можно рассматривать как композицию следующих дополнительных понятий:

- модель игры;
- вид игры;
- контроллер игры.

Подобное видение определяется одним из наиболее распространенных в настоящее время подходом к реализации интерактивных приложений на основе концепции модель-вид-контроллер.

3.4 Рейтинговая таблица

Рейтинговая таблица будет содержать в себе лучших 5 игроков, которые имеют большее количество баллов (очков). Игроки в таблице нужно расположить в порядке убывания их полученных очков, которые можно набрать, поедая на игровом поле пищу либо других игроков.

4 Диаграмма прецедентов

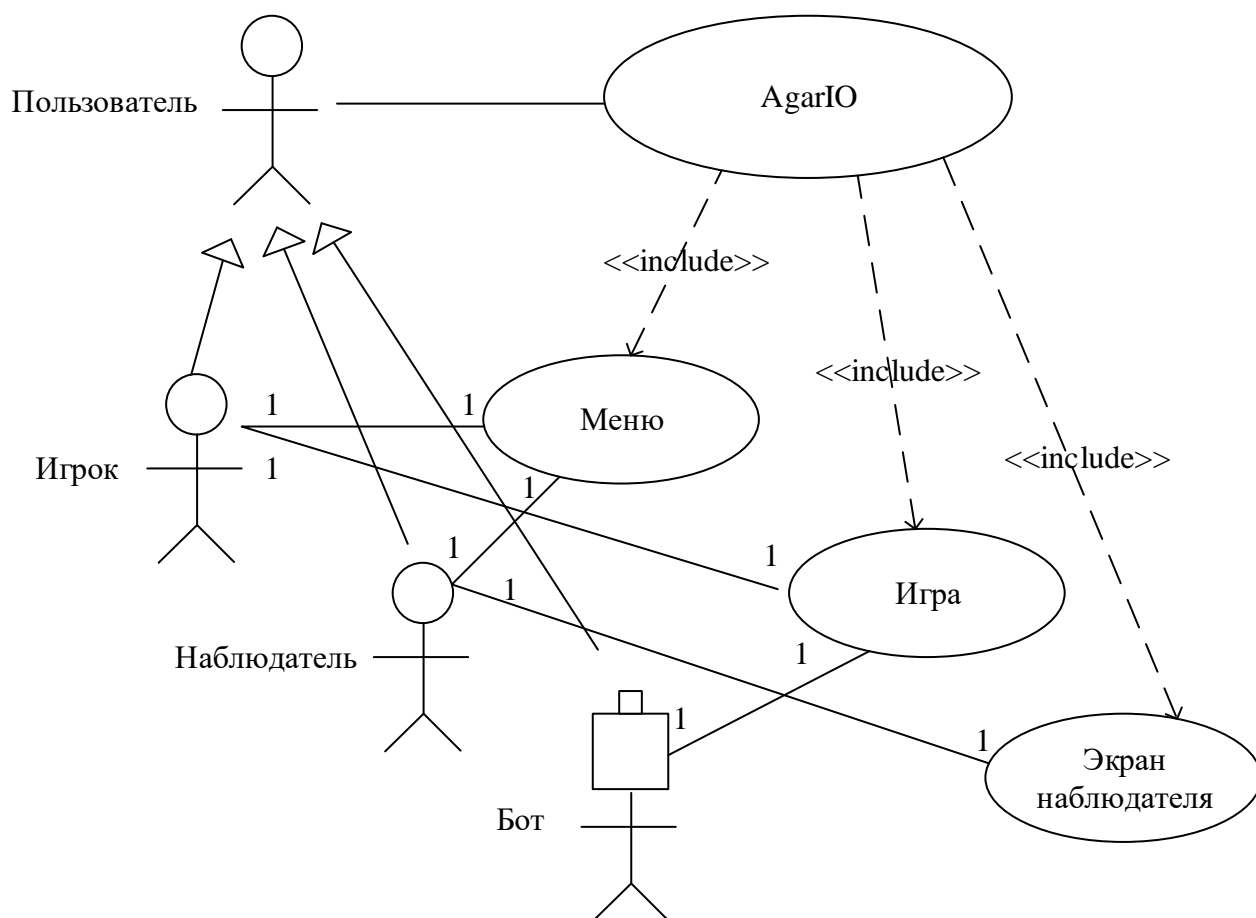


Рисунок 2 — Диаграмма прецедентов

5 Алгоритм работы программы

Описание состояний продемонстрировано на рисунке 3.

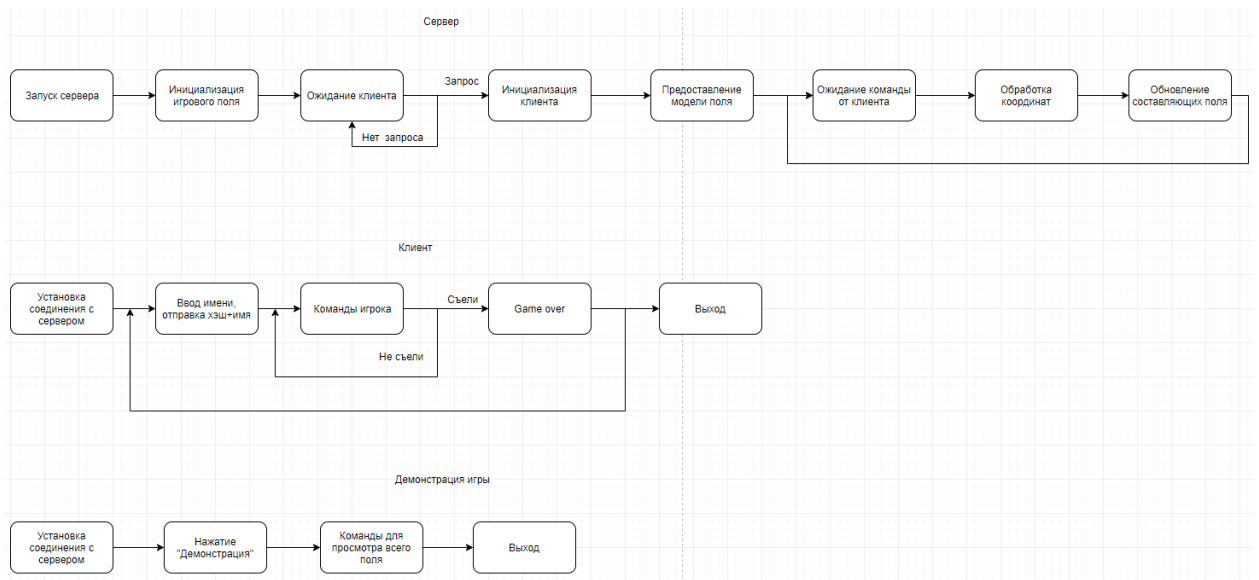


Рисунок 3 — Описание состояний

6 Структура программы

6.1 Игрок и бот

Отличительными чертами являются функции, изменяющие координаты игрока и бота. Игрок — по нажатию клавиш, бот — заложенный алгоритм.

```

var player = {
    var id,
    x, y, //координаты
    screenWidth, screenHeight //ширина и высота экрана
    name, //имя (отправляется один раз)
    hue, //оттенок шаров
    mass //масса
}
  
```

6.2 Еда

```

var food = {
    var hue, //оттенок еды
    x, y, //координаты
    radius //радиус шара
}
  
```

7 Модель

