**Название:** Pacman\_JavaFX

**Ссылка на репозиторий:** <https://github.com/kopch02/Pacman_JavaFX>

**Авторы проекта:** Беломытцев Владислав Романович (GosuUHC), Искандиров Марат Ринатович (kopch02)

**Описание идеи:** реализация на языке Java всемирно известной аркадной игры Pac-Man своими силами. Задача игрока — управляя Пакманом, съесть все точки в лабиринте, избегая встречи с призраками, которые гоняются за героем.

**Описание реализации:**

**GamePlayController** – основной игровой класс, при инициализации которого (запуск игры с кнопки в меню) создаются экземпляры классов:

* **GameMap** - класс карты игры, реализующий создание стен, очков(точек), а также алгоритмы вычисления коллизий игрока с призраками и точками
* **Renderer** – класс игры, реализующий отрисовку объектов
* **Player** – класс игрока, отвечающий за обновление состояния игрока(перемещение)
* **KeyPolling** – вспомогательный класс игры, отвечающий за обработку нажимаемых пользователем клавиш
* **RedGhost**, **PinkGhost**, **BlueGhost** – классы призраков, реализующие алгоритмы поведения призраков в лабиринте
* **GameLoopTimer** – класс игры, расширяющий javafx.AnimationTimer. Экземпляр этого класса, созданный в отдельном потоке, при каждом кадре вызывает метод tick(), отвечающий за обновление глобального состояния игры для алгоритмов отрисовки, просчитывания коллизий, обновления счета игрока.

**MenuController** – класс меню игры, который позволяет пользователю запустить игру или посмотреть таблицу с результатами

**DeathController** – класс меню игры, который инициализируется после смерти игрока (столкновение с призраком) и в новом потоке отправляет результаты игрока на сервер. Позволяет вернуться в главное меню или посмотреть результаты

**ScoresController** – класс, реализующий отображение результатов игроков в javafx.TableView. При его инициализации, в отдельном потоке происходит запрос к серверу для получения данных о результатах игроков из БД

**Net** – вспомогательный класс, отвечающий за прием и передачу данных между сервером и клиентом

**Server** – класс сервера игры. При подключении к серверу создается новый поток, в котором происходит запрос к БД. Отправляет или сохраняет данные из/в БД

**DBUtil**, **PlayerDAO**, **PlayerBD** – классы для взаимодействия с БД. Использовался паттерн Data Access Object

**Описание технологий:** OpenJDK, JavaFX, Oracle Database

**Ссылка на видео с демонстрацией работы:** <https://www.youtube.com/watch?v=yma2jtikWB4>

**Вклад конкретного участника:**

Беломытцев В.Р – классовое разбиение проекта; реализации классов: GamePlayController, Renderer, Player, KeyPolling, GameLoopTimer, MenuController, DeathController, ScoresController, Net, Server, DBUtil, PlayerDAO, PlayerBD (рендер игровых объектов, многопоточность, сетевое взаимодействие, работа с базой данных); реализация мультиплеера

Искандиров М.Р. – Реализация классов: GameMap, Entity, Ghost, RedGhost, PinkGhost, BlueGhost, Point (Алгоритмы поведения призраков, взаимодействие игровых элементов (столкновение призраков с игроком, поедание точек), механизм работы “Энерджайзера” – большая точка, при съедании которой, игрок может съедать призраков)