 Евразийский национальный университет им. Л.Н. Гумилёва

Факультет информационных технологий

Кафедра “Информационные системы”

ОТЧЕТ

**Учебная практика**

Студент: Батталова Мадина

Группа: ИС-12

\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководящий: PhD, Жукабаева Тамара Кокеновна

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Астана, 2023

**Содержание**

[**Введение** 3](#_Toc136832209)

[1.1. Ознакомления с правилами и ресурсами, предоставленными преподавателем 4](#_Toc136832210)

[1.2. Создание аккаунтов в GitHub и в Jira 4](#_Toc136832211)

[1.3. Написание плана в Jira 5](#_Toc136832212)

[**2.** **Техническое задание** 6](#_Toc136832213)

[2.1. Знакомство с Pong. История. Концепт и цель игры 6](#_Toc136832214)

[2.2. Требования к игре 6](#_Toc136832215)

[2.3. Аналоги игры 7](#_Toc136832216)

[**3.** **Описание игровой механики** 9](#_Toc136832217)

[3.1. Основные правила в игре 9](#_Toc136832218)

[3.2. Описание управления и взаимодействия игрока с игрой 9](#_Toc136832219)

[3.3. Обзор игровых элементов 10](#_Toc136832220)

[3.4. Фишки в игре. Отличие нашего проекта от других игр 10](#_Toc136832221)

[**4.** **Реализация** 12](#_Toc136832222)

[4.1. Язык программирования Python 12](#_Toc136832223)

[4.2. Описание основных этапов игры (блок схема , Mind map и UML) 15](#_Toc136832224)

[4.3. Обзор использованных библиотек 16](#_Toc136832225)

[4.4. Примеры кода, демонстрирующие ключевые аспекты реализации игровой механики и функциональности 17](#_Toc136832226)

[4.5. Объяснение основных компонентов игры, таких как игровой цикл, обработка событий, отображение графики и звука 18](#_Toc136832227)

[**5.** **Тестирование и отладка** 21](#_Toc136832228)

[5.1. Описание подхода к тестированию игры 21](#_Toc136832229)

[5.2. Обзор выявленных проблем и ошибок в процессе разработки 22](#_Toc136832230)

[5.3. Описание примененных методов отладки и исправления ошибок 22](#_Toc136832231)

[**6.** **Результаты и выводы** 24](#_Toc136832232)

[6.1. Обсуждение достигнутых результатов и функциональности игры 24](#_Toc136832233)

[6.2. Оценка качества реализации и соответствия исходной концепции 24](#_Toc136832234)

[6.3. Рефлексия о преимуществах и ограничениях выбранного подхода разработки игры 24](#_Toc136832235)

[**7.** **Возможности для дальнейшего развития** 25](#_Toc136832236)

[7.1. Идеи и предложения по улучшению игры или добавлению новых функций 25](#_Toc136832237)

[7.2. Рекомендации по дальнейшей оптимизации производительности или улучшению пользовательского опыта 25](#_Toc136832238)

[**Заключение** 27](#_Toc136832239)

[**Список использованных источников** 28](#_Toc136832240)

# **Введение**

В современном мире игровой индустрии существует множество проблем, требующих незамедлительного решения для улучшения нашего же игрового опыта. Как студент IT факультета, хорошо осознав тонкости и проблемы хочется внести свой вклад в улучшение этой сферы. В рамках учебной практики, проходящей с 22 мая по 10 июля, мы решили принять вызов и попробовать себя в роли разработчиков игрового приложения, используя язык программирования Python. Целью проекта является создание простой игры, позволяющей нам получить первоначальный опыт в разработке игровых приложений. Мы осознали, что создание игр является сложным и захватывающим процессом, и эта возможность помогла глубже погрузиться в мир программирования на Python. Однако, несмотря на наши стремления, игровая индустрия все еще сталкивается с рядом проблем, которые заслуживают особого внимания. Некоторые из них включают в себя недостаток разнообразия в игровых жанрах, проблемы с балансом между доступностью и сложностью игр, а также вопросы, связанные с безопасностью и защитой геймеров по всему миру.

Надеюсь, что драгоценный опыт, который я сейчас получаю, поможет мне в будущем внести свой вклад в решение этих проблем и улучшение игровой индустрии в целом.

В данном отчете мы представим наши идеи, с которыми мы столкнулись в процессе разработки, а также поделимся опытом и расскажем о трудностях, с которыми столкнулись в процессе решения этих проблем. Мы подробно рассмотрим методы и инструменты, которые использовали, чтобы достичь поставленных целей. Также, обратим внимание на основные результаты нашей работы и поделимся нашими выводами и рекомендациями.

**Требования к работе**

## Ознакомления с правилами и ресурсами, предоставленными преподавателем

Этот отчет и первая учебная практика имеют особое значение, поскольку это первый опыт в этой области. Преподаватель, не оставив без поддержки сразу же предоставил важные и полезные инструкции. Он поделился требованиями, необходимыми для написания качественного отчета, а также рекомендовал использовать определенные приложения, которые окажутся полезными процессе.

Данные советы и наставления, дали ценный набор инструментов для успешного создания нашего проекта. Они стали непреходящим руководством в путешествии по разработке игрового приложения на языке программирования Python. Получение этих советов от опытного преподавателя стало источником вдохновения и помогло преодолеть различные трудности на пути к реализации наших идей.

Благодаря преподавателю, под его руководством, экспертные советы стали фундаментом для проекта. Они подарили уверенность и позволили построить качественное и инновационное игровое приложение на языке программирования Python.

## Создание аккаунтов в GitHub и в Jira

Одной из ценных рекомендаций преподавателя было использование Jira и GitHub при выполнении проекта. Зарегистрировавшись на платформах началось изучение всех преимуществ. В то же время, активно обсуждалось как будет организовываться работа с помощью этих мощных инструментов.

Jira, инновационное решение для управления проектами, и GitHub Рис.1, популярная платформа для совместной разработки программного обеспечения, оказались идеальным сочетанием для наших потребностей. Они предоставили эффективные средства планирования, отслеживания и управления проектом. Полученная информация помогла решать задачи, определять приоритеты, устанавливать сроки и назначать ответственных людей в Jira. В то же время, GitHub позволил контролировать версии кода, совместно работать над разработкой и обмениваться изменениями в проекте.

Полностью осознав ценность платформ планируется продолжать использовать Jira и GitHub в дальнейшем. Они являются неотъемлемой частью индустрии разработки программного обеспечения и широко востребованы в профессиональном сообществе. Jira и GitHub предоставляют надежные и удобные инструменты для эффективного управления проектами, обеспечивая синхронизацию работы команды и облегчая процесс совместной разработки.

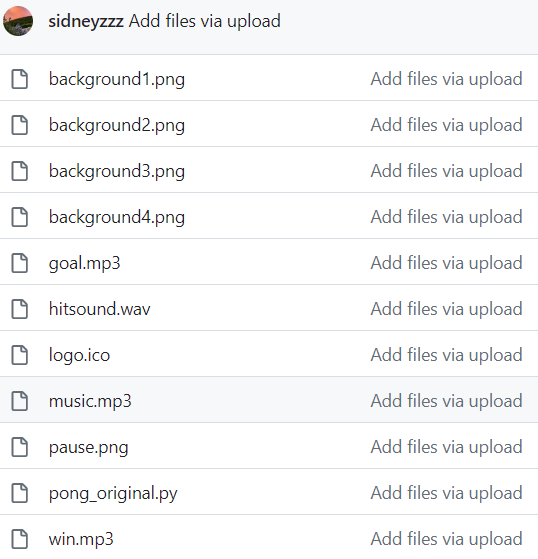


Рисунок 1.1. GitHub файлы

## Написание плана в Jira

Однажды Джон Максвелл, выдающийся мыслитель и автор заявил: "Потерянное время – это ценность, не подлежащая восстановлению. Хорошо спланированное время – самое бесценное сокровище, которое может открыться перед нами." Проведя мозговой штурм, команда принялась за выполнение работы. Составив план на грандиозную задачу, со множеством мельчайших компонентов, учитывались индивидуальные особенности каждого. В этом непростом деле нам превосходно помогла Jira – мощный инструмент управления проектами, разработанный компанией Atlassian. Jira, широко используемая для координации работ, отслеживания ошибок и управления задачами в команде разработчиков. Она позволяет создавать и наблюдать за задачами, устанавливать приоритеты, определять сроки и назначать ответственных исполнителей. Это придает эффективности в планировании, организации и мониторинге хода работ. Визуальное воплощение планов в Jira помогло наглядно подмечать все тонкости.

Ниже предоставляется план, подробно расписанный в Jira

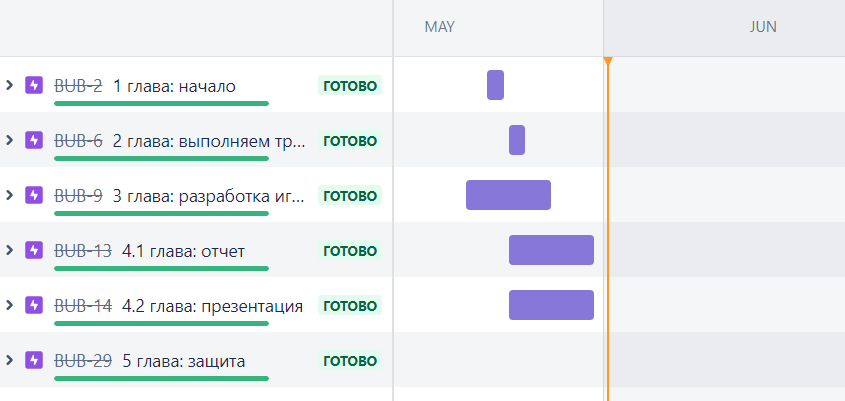


Рисунок 1.2. План проекта в Jira

# **Техническое задание**

## Знакомство с Pong. История. Концепт и цель игры

## 

Как упомянуто выше, основной целью стало побыть в роли разработчиков игр. Поэтому в качестве проекта была отобрана игра Pong. Pong считается одной из самых первых видеоигр в истории, она заложила основу для будущего развития игровой индустрии благодаря своему колоссальному успеху. Игра была разработана Нолан Бушнелл и Аллен Элкон и выпущена в 1972 году. Создатели Pong, впоследствии основали компанию Atari, которая стала одной из самых влиятельных и успешных компаний в области видеоигр. Информацию можно дополнить тем, что Atari впоследствии создала: «Breakou»t и «Asteroids» и многие другие культовые игры. Как сказал доктор Пауль Йозеф Геббельс: "Всё гениальное просто", наверно, поэтому в свое время Pong произвел настоящую революцию в мире игр. Разработка стала настолько популярной, что во многих аркадных залах появились специальные "Pong-новые" автоматы. Pong относится к жанру спортивных видеоигр. Этот жанр включает игры, которые симулируют различные виды спорта или спортивные активности.

Концепт игры «Pong» довольно прост: два игрока управляют ракетками, отбивая мяч между собой на экране. Целью игры является набрать больше очков, чем противник, путем отбивания мяча так, чтобы он пролетел мимо ракетки соперника и затронул его сторону поля.

Игра «Pong» была разработана в эпоху зарождения видеоигр и имела простой и легко понятный геймплей. Ее целью было создание веселой и занимательной соревновательной игры, которая могла быть доступной для широкой аудитории. Целью команды стало попробовать создать Pong на языке программирования Python

## Требования к игре

Перед разработкой игры Pong необходимо было учесть и сформировать все требования, чтобы создать качественную игровую платформу, которая соответствует ожиданиям пользователей.

Во-первых, игровой концепт игры Pong должен был быть ясно определен. Игра представляет собой классическую аркадную игру, где два игрока управляют платформами, отбивая мяч и пытаясь забить гол в ворота противника.

Цель игры - набрать максимальное количество очков. Определение основной идеи игры и ее жанра помогло в определении механик игрового процесса и визуального стиля игры. Вторым важным требованием было определение способов управления в игре. Для игры «Pong» было решено использовать клавиатуру для управления платформами игроков. Каждый игрок имел возможность управлять своей платформой с помощью соответствующих клавиш, перемещая ее по вертикальной оси. Это требование обеспечивало простоту и интуитивность управления для игроков. Третьим требованием было обеспечение сохранения прогресса игрока. Для этого был реализован механизм сохранения данных, который позволял игрокам сохранять свой текущий прогресс, достижения и другую информацию. Это позволяло игрокам продолжать игру с последнего сохранения и не терять свои достижения.

Одним из ключевых аспектов разработки игры Pong было проведение тестирования и отладки. Тестирование игры было необходимо для обнаружения и исправления возможных ошибок, а также для проверки функциональности и стабильности игры. Были использованы различные сценарии игры, чтобы проверить различные аспекты игрового процесса и выявить проблемные моменты. Отладка кода также была важной частью процесса разработки, позволяя идентифицировать и исправлять ошибки в программе.

Документация также являлась неотъемлемой частью проекта. Была создан отчёт, который включает в себя описание игры, инструкции по установке и использованию, а также руководство для игроков и разработчиков. Документация обеспечивала понимание игрового процесса и предоставляла необходимую информацию для игроков и разработчиков.

Наконец, производительность игры стала одним из наиболее важных требований. Было стремление к тому, чтобы игра работала плавно и без задержек на различных платформах. Оптимизация кода и ресурсов, использование эффективных алгоритмов и методов рендеринга позволили достичь хорошей производительности игры и удовлетворить потребности пользователей.

## Аналоги игры

«Pong» была одной из первых и наиболее известных аркадных игр, и ее успех вдохновил самих разработчиков и других на создание подобных улучшенных игр или клонов. Представлены некоторые из аналогов Pong и ранних случаев клонирования игры:

Существует множество аналогов игры Pong, которые доступны для игры на телефонах, компьютерах и ноутбуках. Эти аналоги предлагают различные вариации и модификации игры Pong, а также добавляют новые элементы и функции. Некоторые из популярных аналогов Pong включают:

«Curveball» рис. 2.1. Аналог Pong, в котором поле игры имеет кривую форму, что делает игру более сложной и интересной. Curveball, как и другие аналоги Pong, также доступен на различных платформах. Вы можете найти его в магазинах приложений для смартфонов (iOS и Android) или найти онлайн-версии игры, которые можно запустить на вашем ноутбуке или компьютере через веб-браузер.

Изображение игры:



Рисунок 2.1. Curveball

«Pong Game» рис. 2.2. Простой аналог Pong с обновленной графикой и некоторыми дополнительными функциями. Игры с простым названием "Pong Game" доступны на множестве платформ, включая смартфоны (iOS и Android), а также ноутбуки/компьютеры (Windows, macOS и Linux). Вы можете найти их в магазинах приложений или на игровых платформах.

Изображение игры:



Рисунок 2.2. Pong Game

«Neon Pong» рис. 2.3. Версия Pong с яркой неоновой графикой и различными вариациями игры. Аналоги Pong с яркой неоновой графикой также могут быть доступны на различных платформах. Вы можете искать их в магазинах приложений для смартфонов или на игровых платформах, а также найти онлайн-версии, которые можно играть в веб-браузере на ноутбуке или компьютере.

Изображение игры:



Рисунок 2.3. Neon Pong

«Pong Solo» рис. 2.4. Аналог Pong, который предлагает одиночную игру против компьютерного противника с различными уровнями сложности. Вы можете искать их в магазинах приложений для смартфонов или на игровых платформах, а также найти онлайн-версии для игры на ноутбуке или компьютере.

Изображение игры:



Рисунок 2.4. Pong Solo

# **Описание игровой механики**

## Основные правила в игре

Игра «Pong» является классической аркадной игрой, которая существует уже на протяжении не одного десятилетия. Она проста в своей концепции, но при этом предлагает захватывающий геймплей и заряжает атмосферой соревнования. В этом пункте рассматриваются основные правила, которые делают игру «Pong» столь увлекательной.

Первое правило: геймплей игроков. В игре «Pong» участвуют два игрока, каждый из которых управляет своей платформой. Они размещены на противоположных концах игрового поля. Один игрок управляет левой платформой, другой – правой платформой. Задача игроков состоит в том, чтобы отбить мяч и предотвратить его попадание в свои ворота.

Второе правило: отбивание мяча. Цель игры «Pong» заключается в том, чтобы отбивать мяч платформой так, чтобы противник не смог его поймать. Когда мяч приближается к платформе, игрок должен использовать свою реакцию и навыки, чтобы переместить свою платформу вовремя и в нужное место, чтобы отразить мяч обратно в сторону противника и не дать ему возможности отбить мяч.

Третье правило: попадание мяча. Игрок успешно отбивает мяч и направляет его в сторону противника, противник должен поймать мяч своей платформой, чтобы продолжить игру. Если противник не сумеет поймать мяч и пропустит его, сопернику будет засчитано очко. Игра продолжается до тех пор, пока один из игроков не достигнет заданного количества очков для победы. В разработанном случае это 15 игровых очков. Однако, когда обоим игрокам остается финальное очко, победное увеличивается, чтобы придать игре соревновательный жанр. То есть при достижении обоими игроками 14 игровых очков, забив еще один гол игрок не выиграет, ему нужно оторваться от оппонента на 2 очка для победы.

Четвертое правило: переигрывание и победные очки. Существует не только счетчик игровых очков, но и счетчик победных очков. Когда один из игроков набирает нужное количество игровых очков, тем самым побеждая игру, ему присваивается очко в победном счетчике. Отличие их в том, что у победного счетчика нет ограничений, соревнование может происходить и до 400 и до 1000 очков.

Пятое правило: дополнительные элементы. В некоторых версиях игры Pong могут быть добавлены дополнительные элементы, которые влияют на геймплей. Например, это могут быть преграды, которые меняют направление мяча, или бонусы, которые даруют игрокам временные преимущества.

В заключение, игра Pong предлагает простые, но захватывающие правила, которые подразумевают соревновательный дух и требуют от игроков быстрой реакции и координации движений. Она является одной из наиболее узнаваемых и популярных в истории видеоигр, и ее простота привлекают игроков всех возрастов. Правила Pong легко понять, но стать настоящим мастером требует много тренировок и опыта.

## Описание управления и взаимодействия игрока с игрой

Основной элемент управления в игре «Pong» — это платформа, которую игрок контролирует на своей стороне поля. Платформа может двигаться только вверх и вниз по вертикальной оси. Это позволяет игроку настраивать свою позицию и подстраиваться под полет мяча. Для управления платформой обычно используются клавиши со стрелками вверх и вниз на клавиатуре или другие соответствующие клавиши или контроллеры.

Цель игрока в «Pong» - отбить мяч и предотвратить его попадание на свою сторону поля. В то время, когда мяч приближается к платформе, игрок должен быстро и точно реагировать, перемещая свою платформу в нужное место и нанося удар мячу в нужном направлении.

Взаимодействие между игроками происходит во время отбивания мяча. Если игрок успешно отбивает мяч и направляет его в сторону противника, противник должен поймать мяч своей платформой, чтобы игра продолжалась. Противник использует аналогичное управление своей платформой для перехвата мяча и его возвращения обратно на сторону первоначального игрока. Этот процесс создает напряженную динамику, где игроки постоянно соревнуются друг с другом.

Счет игры отслеживается с помощью специальных отметок на стенах поля. Каждый раз, когда мяч проходит мимо платформы и достигает стены противника, игроку, забившему мяч, засчитывается очко. Счет служит показателем успеха игроков и стимулирует их к достижению новых рекордов.

Управление и взаимодействие игрока с игрой «Pong» являются ключевыми элементами, которые делают игру захватывающей и увлекательной. Они требуют от игроков быстроты реакции, точности и стратегического мышления. Простота игры Pong исключает любые излишества и делает управление и взаимодействие игрока с игрой простыми и интуитивно понятными, позволяя каждому наслаждаться игровым процессом и соревноваться с друзьями или компьютером.

## Обзор игровых элементов

Игра «Pong» содержит несколько основных игровых элементов, каждый из которых играет важную роль в игровом процессе. Важно знать об этих элементах, чтобы понять, как они взаимодействуют между собой и как влияют на игровую динамику.

Мячик является главным объектом в игре «Pong». Он представляет собой небольшой шар, который движется по игровому полю, отскакивая от различных объектов. Мячик начинает движение с определенной скоростью и направлением и отражается от стенок и платформ, меняя свое направление. Главная цель игроков - отбивать мячик с помощью платформы и не допустить его проникновения в свою зону.

Присутствуют два игрока, каждый управляет своей платформой. Платформа представляет собой горизонтальную полосу, которая может двигаться по вертикальной и горизонтальной полосе в ограниченном пространстве. Игроки используют платформу для отбивания мячика и его направления в сторону противника. Управление платформой осуществляется с помощью клавиатуры.

Каждый игрок имеет свою зону, которые находится на противоположных сторонах игрового поля. Ворота представляют собой края поля, и если мяч пересекает край, игровое очко достается забившему игроку. Цель каждого игрока - не допустить проникновение мячика в свою воротную зону и отразить его обратно в направлении противника. Если мячик проникает в ворота, противник получает очко, а мячик перезапускается для нового раунда.

В игре присутствует счетчик, который отслеживает количество очков, набранных каждым игроком. Он отображается вверху экрана и обновляется после каждого успешного попадания мячика в ворота противника. Счетчик позволяет игрокам следить за текущим положением в игре и определить победителя в конце матча.

## Фишки в игре. Отличие нашего проекта от других игр

Фишка, которая предлагает смену фоновой картинки в главном меню каждый раз, добавляет разнообразие в пользовательский опыт игры. Вместо того, чтобы использовать одну и ту же фоновую картинку каждый раз, игра может случайным образом выбирать изображение из заданного списка или директории и отображать его как фоновую картинку в главном меню.

Также можно добавлять свои фотографии, которые необходимо поместить в папку с игрой, она будет рабочей. В следствии, случайным образом станет фоновым изображением игры.

Эта фишка обладает несколькими преимуществами. Во-первых, она позволяет игре предложить игрокам новый и свежий внешний вид каждый раз, при заходе в главное меню. Это создает ощущение уникальности и разнообразия, делая игровой процесс более интересным. Во-вторых, смена фоновой картинки помогает избежать монотонности и утомительности, которые могут возникнуть, когда игроки видят одну и ту же картинку снова и снова.

# **Реализация**

## Язык программирования Python

У Python много готовых библиотек для решения задач. Библиотеками в программировании называют инструменты для решения конкретных типов задач. Вот несколько примеров популярных библиотек для Python:

* Pygame. Библиотека для создания небольших игр и мультимедийных приложений.
* NumPy. Библиотека для работы с искусственным интеллектом и машинным обучением. Используется для сложных математических вычислений.
* Pandas. Библиотека для работы с большими данными.
* SQLAlchemy. Библиотека для работы с базами данных.
* Django, Flask. Библиотеки для разработки серверной части приложений.

Наличие библиотек значит, что под каждую задачу есть свой инструмент. Придумывать что-то сложное с нуля не придется. Python используют компании-гиганты Многие известные нам компании и организации используют Python:

* + Spotify и Amazon используют Python для анализа данных и создания рекомендаций.
  + Walt Disney использует Python как скриптовый язык для анимации.
  + YouTube и Instagram целиком написаны на Python.
  + Если этого недостаточно, есть ещё NASA: их система автоматизации процессов WAS тоже создавалась средствами Python.

В чем заключаются преимущества языка Python?

Язык Python имеет следующие преимущества:

* + Разработчики могут легко читать и понимать программы на Python, поскольку язык имеет базовый синтаксис, похожий на синтаксис английского.
  + Python помогает разработчикам быть более продуктивными, поскольку они могут писать программы на Python, используя меньше строк кода, чем в других языках.
  + Python имеет большую стандартную библиотеку, содержащую многократно используемые коды практически для любой задачи. В результате разработчикам не требуется писать код с нуля.
  + Разработчики могут легко сочетать Python с другими популярными языками программирования: Java, C и C++.
  + Активное сообщество Python состоит из миллионов поддерживающих разработчиков со всего мира. При возникновении проблем сообщество поможет в их решении.
  + Кроме того, в Интернете доступно множество полезных ресурсов для изучения Python. Например, можно легко найти видеоролики, учебные пособия, документацию и руководства для разработчиков.
  + Python можно переносить на различные операционные системы: Windows, macOS, Linux и Unix.

Где применяется Python?

Язык Python имеет несколько стандартных примеров использования при разработке приложений, в числе которых:

Веб-разработка на стороне сервера.

Веб-разработка на стороне сервера включает в себя сложные серверные функции, с помощью которых веб-сайты отображают информацию для пользователя. Например, веб-сайты должны взаимодействовать с базами данных и другими веб-сайтами, а также защищать данные при их отправке по сети.

Python полезен при написании серверного кода, поскольку он предлагает множество библиотек, состоящих из предварительно написанного кода для сложных серверных функций. Также разработчики используют широкий спектр платформ Python, которые предоставляют все необходимые инструменты для более быстрого и простого создания интернет-приложений. Например, разработчики могут создать «скелет» интернет-приложения за считанные секунды, потому что им не нужно писать код с нуля. Затем его можно протестировать с помощью инструментов тестирования платформы независимо от внешних инструментов тестирования.

Автоматизация с помощью скриптов Python:

Язык скриптов — это язык программирования, который автоматизирует задачи, обычно выполняемые людьми. Программисты широко используют скрипты Python для автоматизации многих повседневных задач, среди которых:

• Одновременное переименование большого количества файлов.

• Преобразование файла в другой тип файла.

• Удаление повторяющихся слов в текстовом файле.

• Выполнение базовых математических операций.

• Отправка сообщений электронной почты.

• Загрузка контента.

• Выполнение базового анализа журналов.

• Поиск ошибок в нескольких файлах.

Наука о данных и машинное обучение:

Наука о данных извлекает ценную информацию из данных, а машинное обучение (ML) позволяет компьютерам автоматически учиться на данных и делать точные прогнозы. Специалисты по работе с данными используют Python для решения следующих задач:

• Исправление и удаление неверных данных (очистка данных).

• Извлечение и выбор различных характеристик данных

• Разметка данных добавляет данным значимые имена

• Поиск статистической информации в данных

• Визуализация данных с помощью диаграмм и графиков: линейных диаграмм, столбчатых диаграмм, гистограмм и круговых диаграмм

Специалисты по работе с данными используют библиотеки Python ML для моделей машинного обучения и создания классификаторов, которые точно классифицируют данные. Классификаторы на основе Python используются в различных областях и применяются для выполнения таких задач, как классификация изображений, текста и сетевого трафика, распознавание речи и распознавание лиц. Специалисты по работе с данными также используют Python для глубокого обучения — передовой техники машинного обучения.

Разработчики программного обеспечения часто используют Python для различных задач разработки и программных приложений, среди которых:

• Отслеживание ошибок в программном коде.

• Автоматическая сборка программного обеспечения.

• Управление программными проектами.

• Разработка прототипов программного обеспечения.

• Разработка настольных приложений с использованием библиотек графического пользовательского интерфейса (ГПИ).

• Разработка игр: от простых текстовых игр до сложных видеоигр.

Автоматизация тестирования программного обеспечения:

Тестирование программного обеспечения — это процесс проверки соответствия фактических результатов программного обеспечения ожидаемым результатам, который позволяет убедиться, что программное обеспечение не содержит ошибок.

• Разработчики используют среды модульного тестирования Python (Unittest, Robot и PyUnit) для тестирования написанных функций.

• Тестировщики программного обеспечения используют Python для написания тестовых примеров для различных сценариев. Например, язык применяется для тестирования пользовательского интерфейса интернет-приложения, нескольких программных компонентов и новых функций.

Разработчики могут использовать несколько инструментов для автоматического запуска тестовых скриптов. Эти инструменты известны как инструменты непрерывной интеграции / непрерывного развертывания (CI/CD). Тестировщики и разработчики программного обеспечения используют инструменты CI/CD (Travis CI и Jenkins) для автоматизации процесса тестирования. Инструмент CI/CD автоматически запускает тестовые скрипты Python и сообщает о результатах тестирования всякий раз, когда разработчики вносят новые изменения в код.

Python разработан Гвидо Ван Россумом (Guido Van Rossum), программистом из Нидерландов. Он начал работу над языком в 1989 году в центре Centrum Wiskunde & Informatica (CWI). Изначально язык был полностью любительским проектом: Ван Россум просто хотел чем-то занять себя на рождественских каникулах. Название языка было взято из телешоу BBC «Летающий цирк Монти Пайтона», большим поклонником которого являлся программист.

Язык Python уникален благодаря следующим особенностям:

1. Интерпретируемый язык.

Python является интерпретируемым языком, то есть он выполняет код построчно. Если в коде программы присутствуют ошибки, она перестает работать. Это позволяет программистам быстро найти ошибки в коде.

1. Простой в использовании язык.

Python использует слова, подобные словам английского языка. В отличие от других языков программирования, в Python не используются фигурные скобки. Вместо них применяется отступ.

1. Язык с динамической типизацией.

Программистам не нужно объявлять типы переменных при написании кода, потому что Python определяет их во время выполнения. Эта функция позволяет писать программы на Python значительно быстрее.

1. Язык высокого уровня.

Python ближе к естественным языкам, чем ряд других языков программирования. Благодаря этому программистам не нужно беспокоиться о его базовой функциональности, например об архитектуре и управлении памятью.

1. Объектно-ориентированный язык

Python рассматривает все элементы как объекты, но также поддерживает другие типы программирования (например, структурное и функциональное программирование).

Изначальная версия игры Pong была написана на языке программирования Assembly. Assembly — это низкоуровневый язык программирования, который позволяет разработчикам напрямую взаимодействовать с аппаратурой компьютера. Такой выбор языка был обусловлен техническими особенностями и доступностью на тот момент.

Нолан Бушнелл и Аллен Элкон, создатели Pong, разработали игру для аппаратного обеспечения аркадного автомата на основе процессора MOS 6502. Они использовали язык программирования Assembly, чтобы написать код, который контролировал отображение графики на экране, обработку ввода от игроков и логику игры.

В дальнейшем Pong была портирована на другие платформы, включая домашние консоли и персональные компьютеры, где использовались различные языки программирования, такие как BASIC, С и другие, чтобы адаптировать игру к спецификациям и возможностям каждой платформы.

## Описание основных этапов игры (блок схема , Mind map и UML)

Mind-Map или Ментальная карта игры:

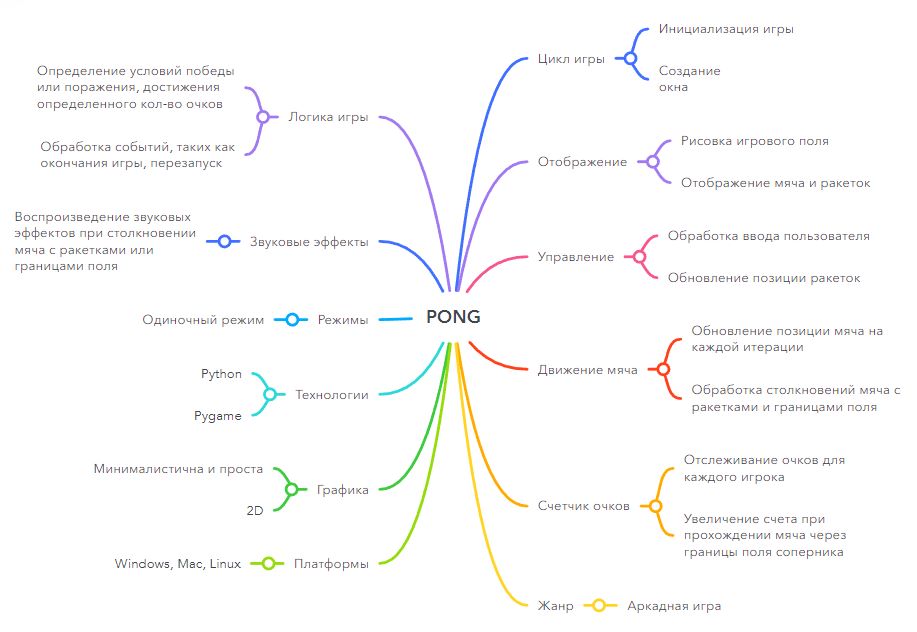


Рисунок 4.1. Mind map

UML проекта приведена ниже:

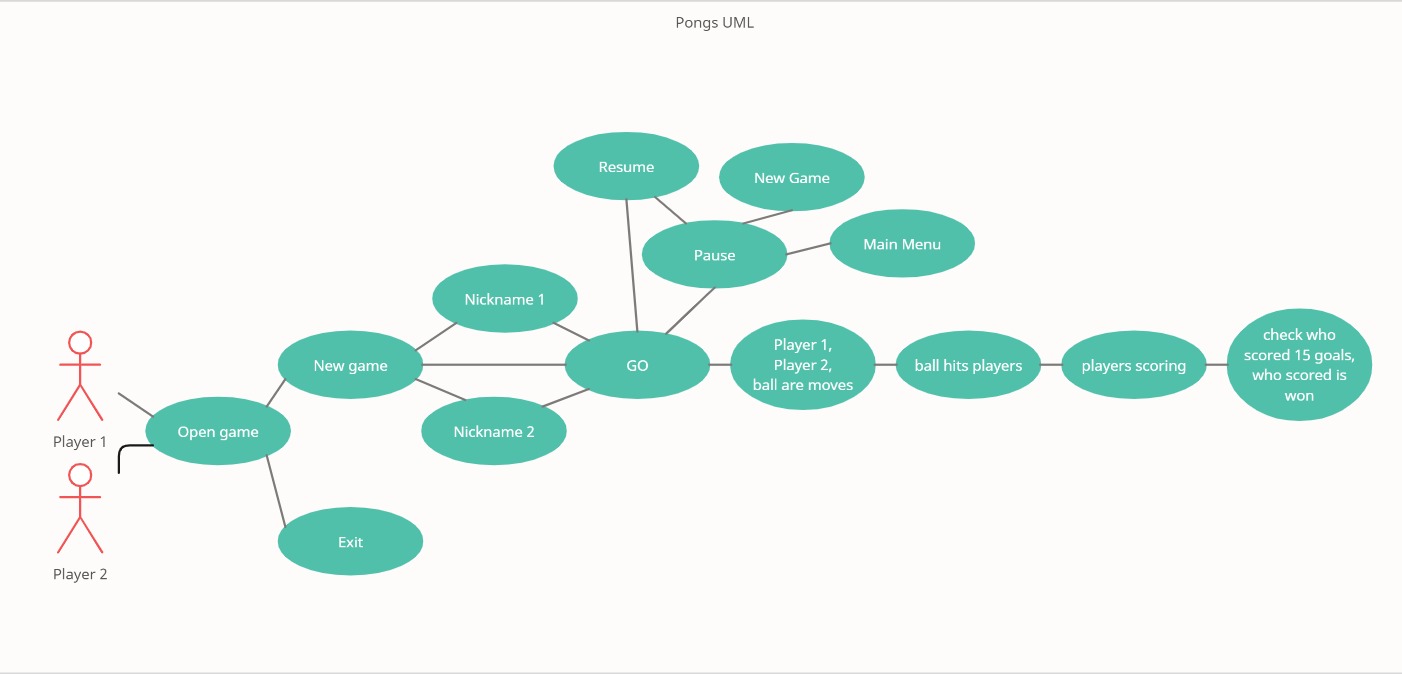


Рисунок 4.2. UML игры «Pong»

Блок-схема игры приведена ниже:

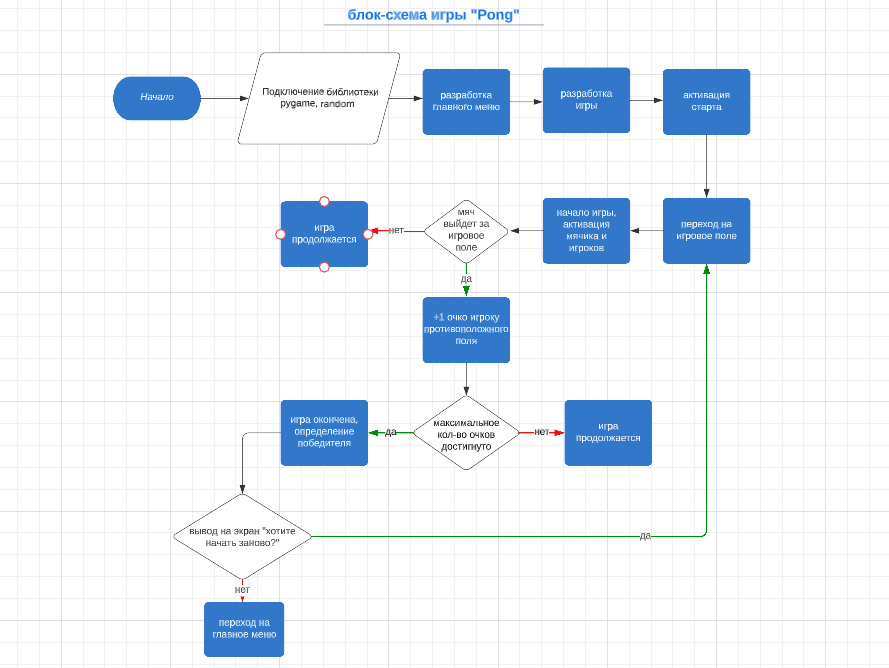


Рисунок 4.3. Блок-схема игры «Pong»

## Обзор использованных библиотек

Для создания игры «Pong» использовалась библиотека Pygame. Она представляет широкий набор функций и возможностей, которые делают ее привлекательным выбором в реализации целей проекта. Самое главное преимущество данной библиотеки – ее простота и доступность. Pygame предоставляет интуитивно понятный интерфейс и простую синтаксическую структуру, что делает ее подходящей для разработчиков всех уровней. Даже те, кто только начинает знакомиться с разработкой игр, могут быстро освоить основы Pygame и начать создавать свои собственные проекты. Учитывая это, данная библиотека дает возможность быстро и легко изучить необходимые инструменты для создания собственной игры, а не копаться в деталях и разбираться в сути всех функций.

Из возможностей Pygame предоставляет широкий спектр инструментов для работы с графикой, звуком, анимацией и управлением пользовательским вводом. Это позволяет создавать игры с разнообразными возможностями и эффектами.

Еще одно преимущество Pygame - переносимость. Библиотека поддерживает различные операционные системы, включая Windows, macOS и Linux, что позволяет разрабатывать игры, которые будут доступны для широкой аудитории пользователей. Это важно, потому что игры на данной библиотеке возможно распространять на различных платформах или делать их доступными для игроков с разными устройствами.

И наконец, выбор Pygame был обусловлен его популярностью и широкой поддержкой в сообществе разработчиков игр. Благодаря этому, легко найти множество ресурсов, туториалов и обучающих материалов, которые помогут в разработке игр и решении возможных проблем

Для создания фишки использовалась библиотека Random. Библиотека Random является одной из стандартных библиотек языка программирования Python и предоставляет функции для работы с случайными числами и случайными выборками. Она является важным инструментом для создания случайности в программах и играх, а также для моделирования случайных событий.

Создается список с путями до разных изображений и их индексами, после чего для генерации случайных целых чисел в заданном диапазоне можно использовать функцию randint(a, b). Она возвращает случайное целое число включительно в диапазоне от a до b. Например, random.randint(1, 10) будет генерировать случайное целое число от 1 до 10.

## Примеры кода, демонстрирующие ключевые аспекты реализации игровой механики и функциональности

Одним из ключевых аспектов реализации игровой механики во многих играх является движение игровых объектов. В частности, движение платформы по вертикальной оси является важным элементом во многих аркадных и спортивных играх, включая игру Pong. Движение платформы по вертикальной оси позволяет игроку контролировать положение платформы на игровом поле и взаимодействовать с другими игровыми объектами, такими как мячик. Это создает динамичность и интерактивность в игре, позволяя игроку адаптироваться к изменяющимся условиям и стратегически выбирать положение платформы для достижения цели.

Ниже представлен код на языке Python, реализующий движение платформ в игре:



Рисунок 4.4.

Код, реализующий движение платформ по вертикальной оси

Важным шагом является использование функции «pygame.key.get\_pressed()», которая позволяет определить состояние клавиш на клавиатуре в текущий момент времени. Эта функция возвращает список, где каждый элемент соответствует одной клавише, и значение элемента равно 1, если клавиша нажата, и 0, если клавиша не нажата.

Далее следует условное выражение, которое проверяет состояние определенных клавиш. Если состояние клавиши "W" истинно (равно 1), то происходит уменьшение скорости игрока по вертикальной оси на 8 пикселей, что позволяет игроку двигаться вверх. Если же нажата клавиша со стрелкой вниз, то скорость игрока увеличивается на 8 пикселей, что позволяет изменить его положение в соответствии с желанием игрока.

Дополнительно, в коде предусмотрены проверки, которые ограничивают движение платформ по вертикальной оси, чтобы они не выходили за пределы игрового поля и оставались видимыми для игроков.

После обработки нажатия клавиш, следуют условные выражения, которые проверяют положение платформы игрока. Если платформа игрока достигает верхней границы поля, то ее положение по вертикальной оси остается неизменным, чтобы платформа не вышла за пределы поля. Аналогично, если платформа игрока достигает нижней границы поля, ее положение также остается неизменным, предотвращая выход за пределы игрового поля.

## Объяснение основных компонентов игры, таких как игровой цикл, обработка событий, отображение графики и звука

Объяснение основных компонентов игры, таких как игровой цикл, обработка событий, отображение графики и звука, является важной частью создания игрового приложения.

Один из ключевых компонентов игры - игровой цикл. Игровой цикл представляет собой бесконечный цикл, который обрабатывает и обновляет состояние игры. В игре Pong игровой цикл отвечает за движение объектов, обработку коллизий и обновление экрана. Pygame содержит в своем арсенале функцию pygame.time.Clock() для контроля скорости обновления кадров и функцию clock.tick(FPS) для определения задержки между кадрами. Это позволяет игре работать с постоянной скоростью обновления и поддерживать плавную анимацию.

Обработка событий является еще одним важным компонентом игры. События включают в себя действия игрока, такие как нажатие клавиш или движение мыши. В игре Pong, события обрабатываются для управления движением платформ и взаимодействия с мячом. Библиотека Pygame предоставляет функцию pygame.event.get(), которая возвращает список всех событий, произошедших за один кадр. Затем можно проверить каждое событие и принять соответствующие действия, например, изменить координаты положения платформы при нажатии определенной клавиши.

Отображение графики является неотъемлемым компонентом игры Pong. Pygame предоставляет мощные инструменты для отрисовки спрайтов, фигур и текста на экране. В игре Pong, отображение графики включает в себя рисование платформ, мяча, игрового поля и других элементов интерфейса. Функция pygame.draw предоставляет широкий спектр методов для создания графических объектов, таких как pygame.draw.rect(), pygame.draw.circle() и pygame.draw.line(). Кроме того, можно использовать изображения и спрайты для создания более сложных графических эффектов.

Звук является продолжением, и важным компонентом игры Pong. Pygame предоставляет возможность воспроизведения и управления звуком в игре. С помощью функций pygame.mixer.Sound() и pygame.mixer.music можно загружать и воспроизводить аудиофайлы различных форматов. В игре Pong, звук может использоваться для воспроизведения звука удара мяча о платформу или звука гола. Это помогает создать более реалистичное и вовлекающее игровое окружение.

Использование библиотеки Pygame облегчает реализацию всех этих компонентов игры. Она предоставляет простой и лаконичный интерфейс для работы с игровой механикой, обработкой событий, отображением графики и звуком. Богатый функционал Pygame позволяет разработчикам создавать качественные игровые приложения. Она является отличным выбором для разработки игры Pong и многих других проектов, требующих графики, звука и взаимодействия с пользователем.

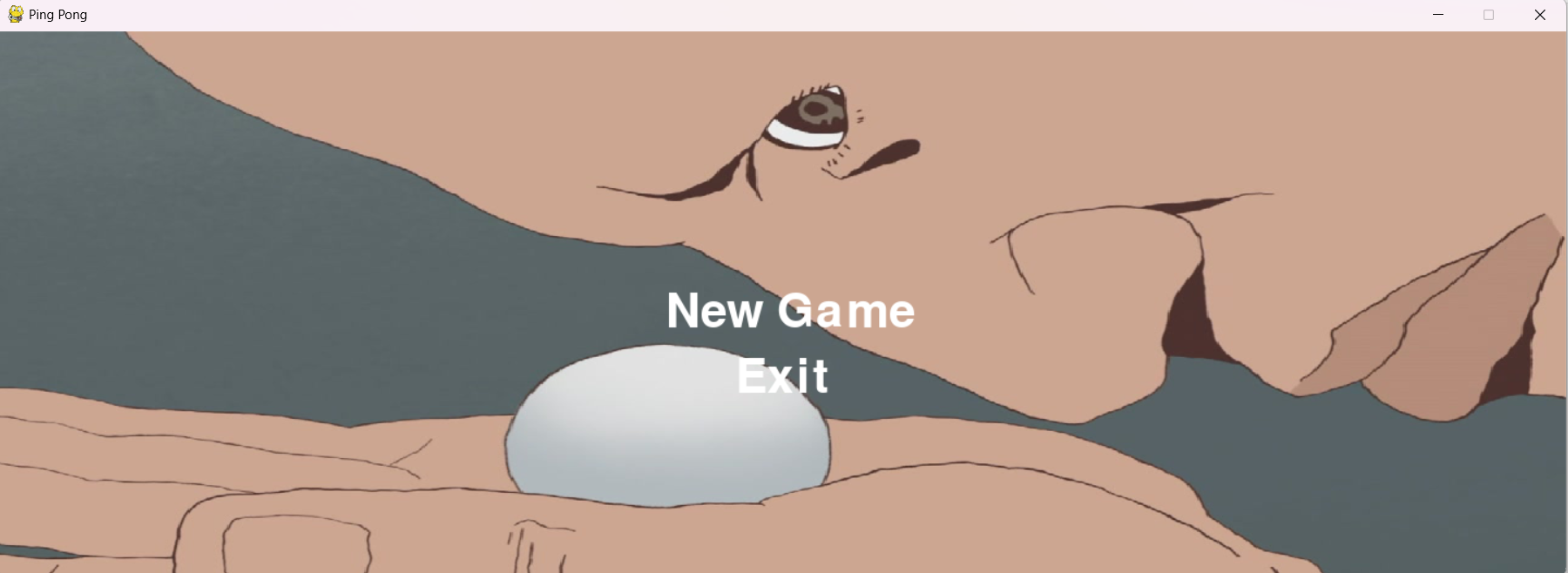


Рисунок 4.5.

Скриншот главного меню, с фоновым изображением

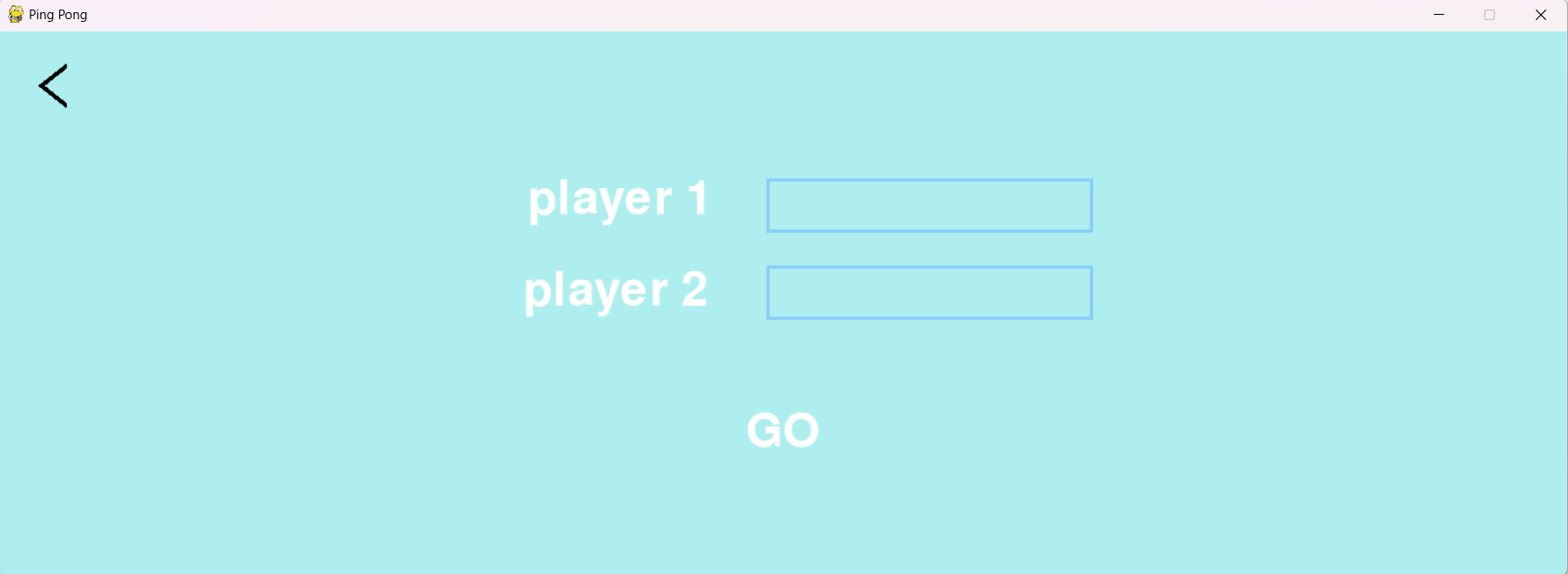


Рисунок 4.6.

Начало игры, ввод никнеймов

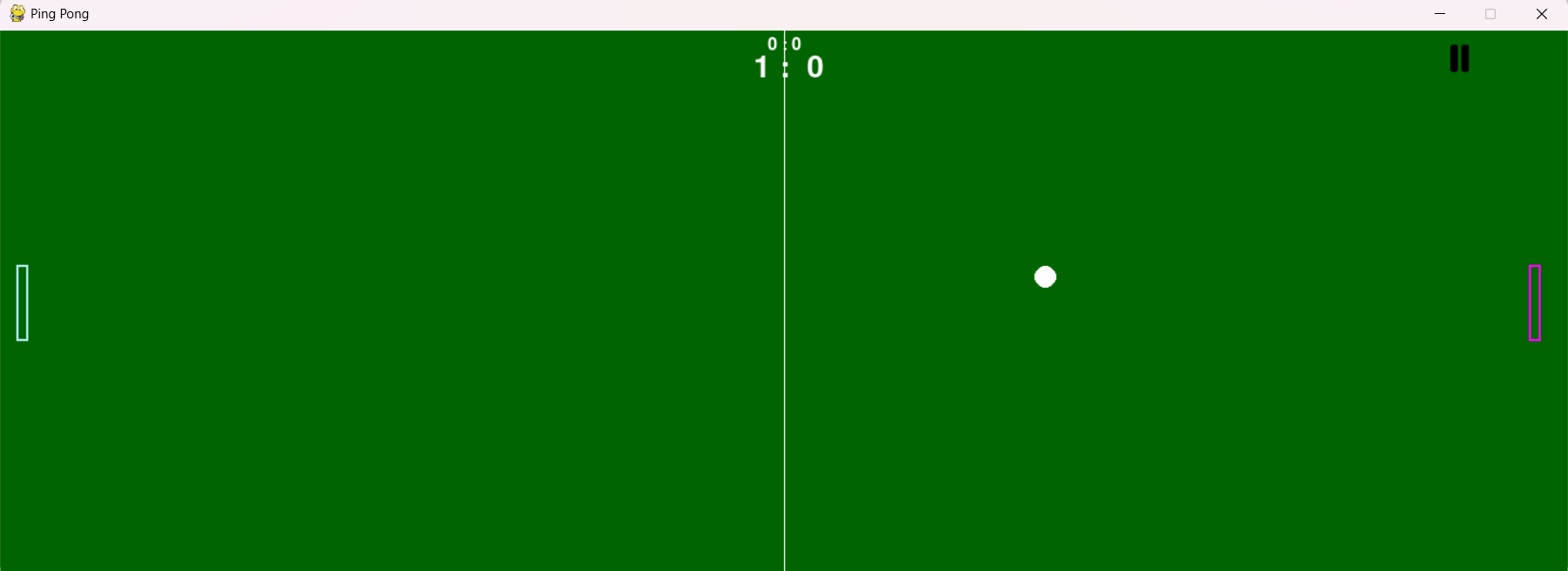


Рисунок 4.7.

Само игровое поле, по бокам платформы игроков и мяч

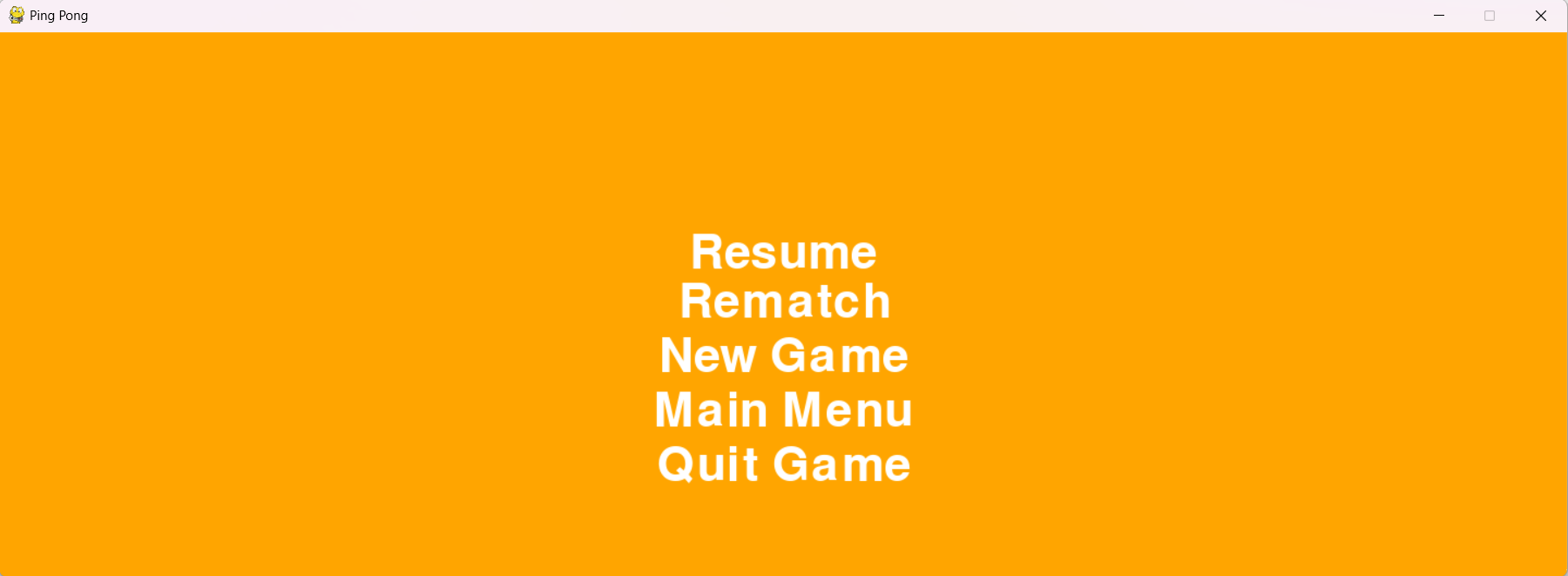


Рисунок 4.8.

Пауза игры, возможность выхода, реванша, новой игры



Рисунок 4.9.

Выход в главное меню, смена изображения – фишка нашей игры

# **Тестирование и отладка**

## Описание подхода к тестированию игры

Тестирование – важный этап в разработке, помогающий выявить ошибки и баги, проверить функциональность и производительность игры. Тестирование проходит поэтапно, как и разработка игры. То есть первым в разработке являются основные элементы: платформы игроков, мяч, бортики игрового поля. Проверяется функциональность платформ, их движение по вертикальной оси и способность контактировать с мячом. Тестирование физики мяча позволяет выявить проблемы с отскакиванием мяча от платформ игроков и границ игрового поля.

Unit-тестирование является важной практикой в разработке программного обеспечения. Оно позволяет автоматизировать процесс проверки отдельных компонентов кода и убедиться в их правильности и надежности.

Для проведения unit-тестирования в Python можно использовать фреймворк unittest. Этот фреймворк предоставляет набор инструментов для определения тестовых случаев, выполнения тестов и проверки ожидаемых результатов.

Вначале создается класс, наследуемый от unittest.TestCase. В этом классе определяются различные методы, каждый из которых представляет собой отдельный тестовый случай. Методы должны начинаться с префикса "test\_", чтобы фреймворк unittest мог автоматически его обнаружить и выполнить рис.5.



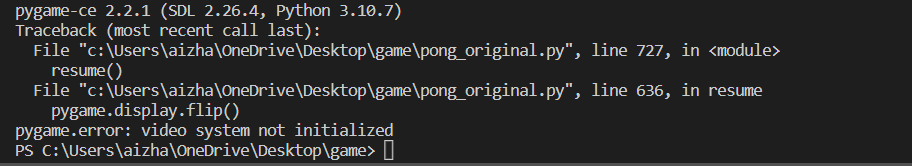
Внутри каждого метода тестового случая мы вызываем функции или методы, которые необходимо протестировать, и используем специальные утверждения (assertions), чтобы проверить ожидаемый результат. Например, мы можем использовать метод assertequals для сравнения двух значений и убедиться, что они равны.

Когда мы запускаем тесты, фреймворк unit test выполняет каждый метод тестового случая и отображает результаты выполнения. Если все тесты проходят успешно, мы получаем уведомление о прохождении тестов. В противном случае, если тест не проходит, мы получаем информацию об ошибке и можем анализировать причину неудачи.

## Обзор выявленных проблем и ошибок в процессе разработки

По ходу создания проекта, выявлялось много ошибок и, из-за неопытности, было выделено на них колоссальное количество времени. Как говорилось ранее, первом этапом осуществляется разработка основных элементов – игровое поле, игроки, мяч. Создание данных элементов не вызвало затруднений, сложнее всего было реализовать физику. Изначально возникли проблемы с реализацией движения платформ по нажатию определенных клавиш на клавиатуре. По итогу не приходила идея как возможно воплотить задуманное. Однако это не потребовало много времени, потому что возникла еще одна преграда: при постоянном движении платформа выходила за рамки игрового поля. Первоначально было принято так и оставить, но спустя некоторое время нашелся вариант решения проблемы.

Единственный баг за все время разработки заключался в отключении игры и ошибки кода при быстром нажатии паузы и выхода в главное меню. Однако, позже нашелся момент, из-за которого случалась данная ошибка. Оказалось время значение изменения кадров, было высоким, из-за чего игра не успевала быстро обрабатывать события на экране и выключалась. Баг казался неисправим, но в будущем, случайным образом, получилось избавиться от него, при помощи создание дополнительного окна при выходе в главное меню (тем самым, выключение игры избегалось, и появилась возможность оставить высокое значение ФПС).



«pygame.display.flip()» - обновление экрана, то есть оно не успевало обновляться.

## Описание примененных методов отладки и исправления ошибок

Важным компонентом разработки игры является одновременное тестирование. Были проведены тестирования каждого модуля и компонента отдельно, чтобы убедиться, что они работают правильно. Следующий этап – проведение тестирование игры. Для этого использовали разные сценарии игры, проверяли различные варианты взаимодействия игрока с игровыми объектами и анализировал результаты. Это помогло выявить и исправить ошибки, а также проверить работу игры в различных ситуациях.

Один из методов отладки, которым часто пользовались, — это вывод отладочной информации. Добавляется код, который выводил значения переменных, состояние объектов и другую полезную информацию на консоль или всплывающие окна. Это позволяет отслеживать текущее состояние игры и идентифицировать возможные проблемные места.

Также для улучшения производительности игры была проведена оптимизацию кода. Имелись проблемные места, где происходили лишние вычисления или ненужные операции, были устранены. Также использовались алгоритмы и структуры данных, которые обеспечивали более эффективную работу программы. Это помогало ускорить процесс обработки событий, отображения графики и других компонентов игры. Применение различных методов отладки и исправления ошибок было существенным в процессе разработки игры Pong. Они позволили выявить и устранить проблемы, повысить производительность и улучшить пользовательский опыт.

# **Результаты и выводы**

## Обсуждение достигнутых результатов и функциональности игры

После множества усилий, было достигнуто значительные результаты. Разработанная игра успешно функционирует и работает. В ходе проекта были усвоены более глубокие знания в изучение языка программирования Python. Основная функциональность игры включает в себя игроков, их платформы, мяч, игровое поле, графический интерфейс и звуковые эффекты. Важно отметить, что игра основана на использовании библиотеки «Pygame», которая обеспечивает ее функционирование и стабильность. Благодаря этой библиотеке получилось реализовать все необходимые компоненты и создать увлекательное игровое взаимодействие.

## Оценка качества реализации и соответствия исходной концепции

Оценка качества реализации и исходной концепции нашей игры подразумевала создание полноценной электронной версии настольного тенниса. Однако, в ходе разработки, было решено изменить изначально задуманную концепцию, чтобы в результате получить классическую аркадную игру, которая уже известна и стала исторической. Это решение позволило применить элементы и стилистику, знакомые многим игрокам, и создать ностальгическую атмосферу. Также было уделено особое внимание игровому процессу и плавности, чтобы игроки получили максимальное удовольствие от нашей игры.

## Рефлексия о преимуществах и ограничениях выбранного подхода разработки игры

Для разработки игры, был выбран конкретный подход, основанный на языке программирования Python и использовании библиотеки Pygame. Известно, что Python не является основным языком программирования для создания игр, и из-за этого существует ограниченный набор возможностей для реализации полноценной графической игры.

Интересно отметить, что исходная версия игры Pong была написана на языке Assembly, который изначально был предназначен для низкоуровневого программирования. Была предпринята попытка реализовать игру на языке Python, который обладает более высоким уровнем абстракции и предоставляет более простой и понятный синтаксис.

Оба языка - Assembly и Python - имеют свои преимущества и недостатки. Язык Assembly обеспечивает максимальное управление аппаратными ресурсами компьютера и позволяет достичь высокой производительности, однако его использование требует глубоких знаний и опыта в низкоуровневом программировании.

С другой стороны, Python обладает более простым и читаемым синтаксисом, что делает его более доступным для начинающих разработчиков. Он также предоставляет обширную библиотеку инструментов, включая Pygame, которая облегчает разработку игр. Однако, из-за более высокого уровня абстракции, Python может иметь ограничения в производительности и доступе к аппаратным ресурсам.

Хоть и были проблемы с ограничениями Python при создании графической игры, в результате, они были успешно преодолены, благодаря библиотеке Pygame. Этот опыт также позволил лучше понять и оценить преимущества и недостатки каждого из языков программирования.

# **Возможности для дальнейшего развития**

## Идеи и предложения по улучшению игры или добавлению новых функций

В ходе нашей работы, у нас получилось создать игру Pong с нуля. При написании основного кода, нам хотелось, чтобы наша игра отличалась от других схожих аналогов. Перед нами стоял большой выбор, как можно было дополнить, украсить игру.

* Различные режимы игры: добавить разнообразие в игру, включив различные режимы игры, такие как режим с препятствиями, где на поле будут появляться препятствия, через которые нужно будет отбивать мяч. Можно также добавить режим с разными уровнями гравитации или изменяющейся скоростью мяча.
* Многопользовательский режим: реализовать возможность играть в Pong с друзьями или другими игроками онлайн. Добавьте функцию многопользовательской игры через Интернет или локальную сеть.
* Улучшенная физика и анимация: добавить более реалистичную физику мяча, отскоки и движение ракеток. Улучшить анимацию игры, чтобы она выглядела более плавной и привлекательной.
* Разнообразные power-ups: Добавьте различные power-ups, которые могут влиять на игровой процесс. Например, power-up, увеличивающий размер ракетки, power-up, ускоряющий мяч или power-up, превращающий мяч в огненный шар, который разрушает ракетки при соприкосновении.
* Режим карьеры или турниры: создать режим карьеры, в котором
* Интеграция социальных сетей: разрешить игрокам делиться своими достижениями и рекордами на социальных сетях, таких как Facebook или Twitter. Добавьте возможность вызывать своих друзей на игру и соревноваться с ними.
* Режим игры против компьютерного противника: добавить возможность играть против компьютерного противника разной сложности. Настроить и улучшите искусственный интеллект компьютера, чтобы он представлял более интересного и вызывного противника.
* Кастомизация: предоставить игрокам возможность настраивать внешний вид и настройки игры. Позвольте им изменять цвета ракеток и мяча, выбор

## Рекомендации по дальнейшей оптимизации производительности или улучшению пользовательского опыта

В игре Pong существует несколько способов оптимизации производительности и улучшения пользовательского опыта. Можно рассмотреть несколько рекомендаций по дальнейшей оптимизации и улучшению игры Pong.

Одним из первых аспектов, на которые следует обратить внимание, является производительность игрового цикла. Игровой цикл отвечает за обновление и отображение игрового состояния. Для обеспечения плавной игры рекомендуется оптимизировать выполнение игрового цикла, минимизировать нагрузку на процессор и оптимизировать вычисления. Это можно достичь путем использования эффективных алгоритмов и структур данных, избегая излишних вычислений и операций.

Другой важный аспект - оптимизация отрисовки графики. Рекомендуется использовать аппаратное ускорение, предоставляемое библиотекой Pygame, чтобы обеспечить более быструю отрисовку спрайтов и фонов. Также стоит избегать излишнего использования прозрачности и сложных графических эффектов, которые могут негативно сказаться на производительности.

Оптимизация обработки событий является еще одной важной частью. Рекомендуется обрабатывать только необходимые события, минимизируя избыточные проверки и обновления. Кроме того, следует учесть возможность асинхронной обработки событий, чтобы игра реагировала быстро на пользовательский ввод.

Для улучшения пользовательского опыта рекомендуется добавить звуковые эффекты и музыку. Звуковое сопровождение может значительно повысить атмосферу игры и делать ее более привлекательной для игрока. Используйте соответствующие библиотеки или инструменты, чтобы воспроизводить звуковые файлы с минимальной задержкой.

Кроме того, можно добавить дополнительные функции и возможности, которые сделают игру Pong более интересной и разнообразной. Например, реализуйте различные уровни сложности, добавьте режим для игры с компьютерным противником или внедрите многопользовательский режим через сеть.

В заключение, оптимизация производительности и улучшение пользовательского опыта в игре Pong являются важными задачами для достижения успеха. Следуя рекомендациям по оптимизации игрового цикла, отрисовки графики, обработке событий и добавлению звукового сопровождения, можно создать более плавную и увлекательную игру Pong, которая будет приносить удовольствие игрокам.

# **Заключение**

Создание игры на языке программирования Python стало для нас уникальным опытом, который позволил расширить уже имеющиеся знания и освоить новые навыки в этой области. Мы поняли, что важно делать первые шаги и выходить за пределы зоны комфорта, чтобы преодолеть преграды и достичь новых высот.

В процессе разработки игры мы столкнулись с различными трудностями, от непредвиденных ошибок в коде до сложностей в реализации конкретных игровых механик. Однако, мы не позволили этим трудностям остановить нас. Мы искали решения, изучали документацию, обменивались знаниями и сотрудничали в команде, чтобы преодолеть каждое препятствие.

Важно отметить, что процесс разработки игры на Python потребовал от нас выхода из зоны комфорта. Мы столкнулись с новыми концепциями, архитектурой игрового движка, а также пришлось изучить и применить библиотеку Pygame для создания графического интерфейса и обработки игровых событий. Эти вызовы помогли нам развиться как разработчикам и обрести уверенность в своих способностях.

В итоге, мы гордимся результатом нашей работы. Мы получили удовольствие от процесса создания и открыли для себя новую сферу разработки. Быть в роли разработчиков дало нам возможность проявить креативность, применить наши знания и умения, и увидеть, как наш труд и усилия превращаются в рабочую игру.

В целом, опыт создания игры на языке программирования Python стал для нас ценным уроком. Мы поняли, что важно не бояться бросать себе вызовы, идти в ногу с технологическими изменениями и стремиться к постоянному развитию.

# **Список использованных источников**

1. Sweigart A. Making Games with Python & Pygame. – 2012.

2. Белолюбский М. М., Сорочинский М. А. Изучение модуля «PYGAME» для создания игр на языке программирования PYTHON //Современное образование: традиции и инновации. – 2020. – №. 2. – С. 220-222.

3. Сараев А. К. и др. Игра виртуальной реальности ping pong vr. – 2018.

4. Kelly S., Kelly S. Basic introduction to pygame //Python, PyGame and Raspberry Pi Game Development. – 2016. – С. 59-65.

5. Широбокова С. Н., Хашиева Л. Н. Разработка информационных моделей экономических систем с использованием унифицированного языка моделирования UML. – 2002.

6. Faulkner A., Contributor. Lucidchart for easy workflow mapping //Serials Review. – 2018. – Т. 44. – №. 2. – С. 157-162.

7. Ortu M. et al. The emotional side of software developers in JIRA //Proceedings of the 13th international conference on mining software repositories. – 2016. – С. 480-483.

8. Cosentino V., Luis J., Cabot J. Findings from GitHub: methods, datasets and limitations //Proceedings of the 13th International Conference on Mining Software Repositories. – 2016. – С. 137-141.

9. Андерсон М. И., Адаменко К. А. Влияние игр аркадного направления на развитие индустрии компьютерных игр //Молодой исследователь Дона. – 2019. – №. 2 (17). – С. 85-90.

10. Свейгарт Э. Учим Python, делая крутые игры. – Litres, 2018.

11. https://elib.sfu-kras.ru/bitstream/handle/2311/141582/vkr\_ravodin.pdf?sequence=1

12. Чеботарев Д. А., Харламов Д. А. ПРИМЕР РЕАЛИЗАЦИИ РЕШЕНИЯ ИГРЫ" СУДОКУ" В PYTHON //Молодежь и научно-технический прогресс. – 2022. – С. 200-203.

13. https://habr.com/ru/companies/otus/articles/433358/

14. Leitner A. et al. Efficient unit test case minimization //Proceedings of the twenty-second IEEE/ACM international conference on Automated software engineering. – 2007. – С. 417-420.

15. Okken B. Python Testing with pytest. – Pragmatic Bookshelf, 2022.

16. This game - https://github.com/madina-battalova/bublikkk