Муниципальное автономное общеобразовательное учреждение

«ХИМИКО-ТЕХНОЛОГИЧЕСКАЯ ШКОЛА «СИНТЕЗ»

**Индивидуальный исследовательский проект**

**СОЗДАНИЕ ПРОСТЕЙШЕЙ 2D ИГРЫ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON**

«Допущен к защите» Руководитель Шрамко Екатерина Сергеевна \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(Подпись)*

«\_\_» \_\_\_\_\_\_\_ 20\_\_ г.

Работу выполнил Ученик 10 класса Поспелов Артём Андреевич \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(Подпись)*

Руководитель: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(Подпись)*

ПЕРМЬ

2022 г.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc95926100)

[ГЛАВА 1. ИСТОРИЯ ЯЗЫКА PYTHON И СТРУКТУРА КОМПЬЮТЕРНЫХ ИГР 6](#_Toc95926101)

[Подглава 1.1 История языка программирования Python 6](#_Toc95926102)

[Подглава 1.2 Компьютерные игры. Их классификация и особенности 14](#_Toc95926103)

[ГЛАВА 2. СОЗДАНИЕ СВОЕЙ 2D ИГРЫ 18](#_Toc95926104)

[Подглава 2.1 Ход проделанной работы 18](#_Toc95926105)

[ЗАКЛЮЧЕНИЕ 20](#_Toc95926106)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 21](#_Toc95926107)

[ПРИЛОЖЕНИЯ 22](#_Toc95926108)

# **ВВЕДЕНИЕ**

На сегодняшний день огромное количество людей проводят свое свободное время за компьютером, будь то играя в игры, работая и зарабатывая деньги или же просто читая новости. К таким людям можно отнести самых обычных детей и взрослых, программистов и предпринимателей, инженеров и художников. Из всех перечисленных мы бы хотели поговорить о программистах, потому что именно они связаны с нашим проектом. Итак, программист - это специалист, который занимается разработкой исходного кода, алгоритма и компьютерной программы на основе специальных математических моделей. Такой программой может быть операционная система компьютера, видеоигра, web-сайт, мобильное приложение, текстовый редактор и даже алгоритм работы кофемашины. Программный код пишется на специальном языке программирования. Он состоит из обычных слов и некоторых специальных символов, присвоенных своему языку. В наше время насчитывается более 100 языков программирования, но самые распространенные из них – Java, Python, PHP, C#, JavaScript, C, С++, Objective-C, Swift, ABC Pascal и другие. Так как создание нашей игры будет происходить на языке программирования Python, расскажем о нём поподробнее. Python — это высокоуровневый язык программирования, который используется в различных сферах IT, таких как машинное обучение, разработка приложений, web и другие. Python очень популярен среди новичков и любителей программирования, потому что он очень прост и удобен в использовании. В 2019 году **Python** стал самым популярным языком программирования, обогнав Java. От выбора языка, программист зависит от конкретной задачи и собственных знаний того или иного языка. Его выбор полностью зависит от его намерений, например, создать игру, мобильное приложение или же собственный сервер. Квалифицированный программист может уверенно использовать 2-4 языка. У программистов широкая сфера приложения профессиональных знаний, к такой сфере можно отнести IT-сферу. Сокращение IT означает Information Technology, что в переводе с английского – информационные технологии. IT-сфера – одна из самых перспективных отраслей в мировой экономике, это деятельность по изучению и разработке процессов сбора, хранения, поиска, обработки и передачи информации, а также изучение и разработка процессов управления информацией с помощью компьютеров и других устройств. Сегодня IT-специалистами называют относительно большую группу людей профессионалов, занимающихся одним и тем же видом деятельности и имеющих общие интересы и задачи, чья сфера деятельности так или иначе связана с информационными технологиями в целом и разработкой программного обеспечения. Сфера IT и программирования с каждым днем набирает обороты и все больше и больше развивается в новых технологиях по всему миру вперед, упрощая жизнь обычных людей.

Наш проект направлен на создание 2D игры на языке программирования Python, в последствии чего, мы будем знакомиться с языком на более высоком уровне, изучая его функции, классы, структуры, библиотеки, циклы, операторы и многое другое, а также найдем новую информацию о строении, структуре и особенностях компьютерных 2D игр. Целью нашей работы является создание своей игры 2D на языке программирования Python. В самом начале нашей работы нужно определиться с редактором кода, библиотеками и классами, которые будут использоваться нами при написании игры. Мы выбирали среди огромного количества библиотек и классов, имеющихся в Python, которые в последствии увеличат возможности в моей игре. С выбором редактора кода у нас не было никаких проблем, ведь за основу всех проектов, созданных нами, мы использовали Python Charm. В момент создания нашей игры мы планируем еще больше углубиться в изучение языка программирования Python, выйдя на более высокий уровень знаний, а также обратить наше внимание на технологии создания игры, конечно, не без помощи интернета. После того, как игра будет готова я хочу показать ее своим друзьям и окружающим меня людям, послушать их мнение об этой игре и поделиться с ними своими впечатлениями во время разработки и написания кода для этой игры.

# **ГЛАВА 1. ИСТОРИЯ ЯЗЫКА PYTHON И СТРУКТУРА КОМПЬЮТЕРНЫХ ИГР**

**Подглава 1.1 История языка программирования Python**

Происхождение названия и история создания

Python смог захватить малую часть рынка веб-разработки, иногда используется для написания десктопных приложений и, конечно, тотально доминирует в сфере машинного обучения. Дескоптные приложения - это программы, обрабатываемые на стороне клиента и запускаемые в виде обыкновенного исполняемого файла на устройстве пользователя. Кроме того, на нём создаётся много прототипов, которые позволяют быстро набросать структуру, функционал и внешний вид будущего проекта.

Создатель языка Python — нидерландский программист Гвидо ван Россум. Он был участником проекта по написанию ABC Pascal, языка для обучения программированию. В конце 1989 года Гвидо приступил к разработке своего нового языка и задумал его как потомка ABC, способного к обработке исключений и взаимодействию с операционной системой Amoeba. На тот момент он работал в центре математики и информатике в Нидерландах.

Он **назвал его в честь британского комедийного шоу «Monty Python», что в переводе означает** «Летающий цирк Монти Пайтона», которое было популярно в начале 1970-х годов. Это телешоу позволяло автору расслабиться и отвлечься от разработки языка. Однако, несмотря на настоящее происхождение названия, для людей более очевидно связывать Python со словом «змея». Этому также способствует логотип, на котором изображена рептилия. И хотя создатель языка не раз говорил, что название никак не связано со змеями, повлиять на мнение общества так и не удалось. Так и получился Python.

**Развитие Python**

Python свободно распространялся через интернет и со временем у него появились последователи — люди, заинтересованные в развитии этого языка программирования. Первая публикация Python состоялась в феврале 1991 года — это была версия 0.9.0. В 1994 году Гвидо опубликовал Python 1.0, а потом одна за другой вышли и другие версии: до 2.0 язык обновился в октябре 2000, до 3.0 — в декабре 2008. В октябре 2021 мир увидела самая свежая версия — Python 3.10.0.

Python — простой язык

Синтаксис Питона всегда выделял его на фоне других языков программирования. Он не страдает избыточностью, схожесть синтаксиса с обычным английским позволяет понять код даже обычному пользователю, кроме того, программист пишет меньше строк кода, потому что нет необходимости использовать символы: «;», «{», «}». Вложенность обозначается отступами, что повышает читаемость кода и приучает новичков к правильному оформлению.

Простота отчасти обусловлена тем, что Питон написан на основе языка ABC, который использовался для обучения программированию и повседневной работы людей, не являющихся программистами.

**Python упрощает написание кода и делает разработку быстрой, всё потому что он обладает следующими особенностями:**

* **Динамическая типизация**. Программисту не нужно указывать тип переменных, язык присвоит его сам. Константы, переменные и числовые значения разных типов, участвующие в одной операции, автоматически приводится к нужному по определённым правилам.
* **Удобный возврат нескольких значений функцией**. Их можно перечислить через запятую, и они автоматически преобразуются в список, то есть не нужно выделять память и передавать указатели в функцию.
* **Автоматическое выделение памяти**. Программисту не нужно самостоятельно выделять память под что-либо. С одной стороны, это уменьшает контроль программиста над программой, а с другой, разработка значительно ускоряется.
* **Сборщик мусора**. Если объект становится бесполезным или на него перестаёт что-либо ссылаться, он автоматически удаляется сборщиком мусора. Сборщик мусора позволяет оптимизировано использовать память и не удалять бесполезные объекты вручную.
* **a, b = b, a**. Эта строка меняет местами значения переменных, теперь то, что было в a, находится в b и наоборот. Такое возможно, потому что Питон сначала рассматривает переменные справа от знака «=» и помещает их в список, то же он делает с элементами слева от «=», затем он связывает каждый элемент правого списка с левым. Таким способом можно обменивать значения не только двух переменных, но и трёх, пяти и так далее.
* **Привязка типа данных**. Тип данных привязан к значению, а не к переменной. То есть значение - это какой-то объект с атрибутами, которые определяют его тип и другие характеристики, а переменная - просто ссылка на этот объект. Такой подход позволил обойтись без явного определения типов и значительно упростил повторное присваивание значения переменной.
* **Интерпретируемый язык**. Написанный код не нужно компилировать, достаточно запустить его и получить результат. Более того, можно работать в интерактивном режиме и получать результат буквально после каждой операции.

Python сочетает в себе и простоту и мощный инструментарий. Его можно использовать для создания прототипа практически любой программы.

Чтобы ускорить разработку, часть программы, обычно не сильно влияющую на скорость работы пишут на Питоне.

Именно благодаря простоте этот язык программирования смог занять доминирующее место в сфере машинного обучения. Люди, так или иначе связанные с наукой, предпочитают не тратить много времени на такие вещи, как написание кода, поэтому Python отлично подошёл для реализации поставленных перед ними задач.

Популярность

Несмотря на то что языку уже более 31 года, он популярен среди программистов всего мира. Python используется почти в каждом среднем или крупном проекте, если не как основной инструмент разработки, то как инструмент для создания прототипа или написания какой-то его части.

Он собрал вокруг себя огромное сообщество разработчиков, предпочитающих именно этот язык программирования

Преимущества, чем хорош Python

Специалисты в этой сфере выделяют огромное множество преимуществ у Python. К ним относятся:

* **Простота и минималистичность**. Чтение программы на Python очень напоминает текст английского языка, хотя и достаточно строгого. Такая конструкция Python является одной из его самых сильных сторон.
* **Простота синтаксиса**. Низкий порог0вхождения (при изучении нового языка программирования можно быстро и без больших усилий добиться значительных результатов). Код языка чистый и понятный, без лишних символов и выражений.
* **Расширяемость и гибкость**. Python можно легко расширить для взаимодействия с другими программными системами и библиотеками или встроить в другие программы в качестве компонента. Например, если вам нужно, чтобы какая-то часть программы работала очень быстро или вы вынуждены скрыть часть алгоритма, вы можете написать эту часть программы на C или C++, а затем вызывать её из программы на Python.
* **Интерпретируемость.**Интерпретатор Python есть для всех популярных платформ и по умолчанию входит в большинство дистрибутивов Linux. Программа просто выполняется из исходного текста. Python сам преобразует этот исходный текст в некоторую промежуточную форму, называемую байт-кодом. Байт-код - это код виртуального процессора, который обычно интерпретируется в машинный. Всё это заметно облегчает использование Python, поскольку нет необходимости заботиться о компиляции программы, подключении и загрузке нужных библиотек и т.д.
* **Кроссплатформенность**. Python имеет возможность запускать программы не только для программных компьютеров, но и для телефонов, приставках и другое. Все ваши программы смогут запускаться на любой из этих платформ без каких-либо изменений, если только вы не использовали системно-зависимые функции.
* **Стандартизированность.** У Python есть единый стандарт для написания кода — Python Enhancement Proposal или PEP, что в переводе означает Предложения по развитию Python, благодаря чему язык остаётся читабельным даже при переходе от одного программиста к другому.
* **Open Source.** Open Source – это обширный и постоянно растущий ресурс для разработчиков, изучая который, они прокачивают свои навыки, начинают понимать больше и становятся лучше как программисты. У интерпретатора Python открытый код, то есть любой, кто заинтересован в развитии языка, может поучаствовать в его разработке и улучшении.
* **Большое сообщество.** Вокруг Python образовалось дружественное сообщество, которое поможет новичку или уже опытному разработчику и разобраться в его проблеме. Во всём мире проходит много мероприятий, где можно познакомиться с коллегами и узнать много нового.
* **Востребованность на рынке труда и поддержка гигантами IT-сферы.** Python-разработчики0востребованы во многих проектах и им несложно найти работу. Разработку на Python ведут в Google, Facebook, Spotify, Netflix, Microsoft Intel, а в России — «Яндекс», «ВКонтакте» и «Сбербанк». Это серьёзно влияет на статус языка.

Недостатки Python

У Python, как и у любого другого языка, есть не только плюсы, но и минусы, а среди разработчиков - не только фанаты, но и ненавистники. К недостаткам Python можно отнести следующее:

* **Низкая производительность.** Python требует высоких вычислительных мощностей серверов и компьютеров. Это делает его не таким быстрым, он отстаёт от других языков по производительности. В условиях сильного развития мощности это не так заметно, как раньше, но всё равно даёт о себе знать. Чтобы нивелировать эту проблему, разработчики обращаются к языку программирования C для реализации проблемного участка кода.
* **Динамическая типизация.** Python относится к языкам с динамической типизацией, что делает его0невероятно гибким при разработке. Но несмотря на это, он потребляет0много ресурсов и имеет низкую скорость выполнения программы.
* **Глобальная0блокировка интерпретатора (GIL).** Это способ синхронизации потоков,0который используется в некоторых интерпретируемых языках программирования. Хотя GIL является самым простым способом избежать конфликтов при одновременном обращении разных потоков к одним и тем же участкам памяти, у такого подхода есть недостаток — ограничение параллельности вычислений. Также он не позволяет достигать высокой эффективности вычислений при работе на многоядерных системах.

Что можно написать на Python

Python используют во многих областях программирования, поэтому на нём можно создать что душе угодно:

* **Интерфейс и внутренний функционал web-сайта**. Программист легко может работать со связями URL адресов, обращениями к базам данных и созданием HTML файлов, которые пользователь видит в браузере.
* **Blockchain**. Блокчейн — это последовательная цепочка блоков, где каждый блок содержит информацию и всегда связан с предыдущим. Технология особенно популярна в финансовой сфере и криптовалюте. Блокчейн совмещает в себе защищенность и открытость информации, он позволяет получить доступ к данным из любой точки мира, но в то же время его практически невозможно взломать, данные хранятся на каком-то главном компьютере, а взламывать каждый блок очень затратно и долго.
* **Бот**. Бот - Это программа, автоматически0выполняющая какие-либо действия в заданное время или в ответ на поступивший сигнал. Боты могут примитивно0симулировать поведение человека, поэтому они часто используются для работы в технической поддержке и поиска информации в интернете
* **База данных**. База данных — это информация, систематизированная по общим0признакам и специальным правилам. В любом большом проекте используются базы данных, в них хранится информацию о пользователях, изменениях в программе и т. д.
* **VR-объекты**. VR дополняет физический мир с помощью виртуальных технологий. То есть виртуальные объекты проецируются на реальное окружение, и имитируют признаки и поведение обычных физических объектов.
* **Игра**. В Python в основном создаются самые простые игры, начиная от змейки закачивая 2D платформером. Создавать 3D игры – очень и очень трудно. Python используется либо для разработки прототипа, либо для реализации какой-то части игры. Для написания простенькой игры можно воспользоваться библиотекой Pygame или Arcade, которые дают все необходимые инструменты и возможности для создания небольшой 2D игры.
* Язык программирования. Python достаточно высокоуровневый язык, поэтому создавать на нём ещё один язык программирования будет глупо, но возможно. Полезнее будет разработать интерпретатор или библиотеку для Python или другого языка программирования.

Конечно, это не весь перечень вещей, которые поддаются созданию на Python. Создать можно что угодно, начиная с обычного калькулятора, заканчивая масштабными работающими серверами, необходимо только время и знания.

**Подглава 1.2 Компьютерные игры. Их классификация и особенности**

Компьютерные игры – это разработанные программы, предназначенные для организации игрового процесса, которые использует человек в качестве обучения или развлечения. К компьютерным играм также относят видеоигры и мобильные игры. Изначально существовали определенные отличия между видеоиграми и компьютерными играми. Видеоигры ассоциировались с отдельными устройствами, компьютеры же представляли собой программные разработки, требующие реализации на стационарных компьютерах. Уже с начала 1980-х годов эта грань стала стремительно стираться, и термины «видеоигра» и «компьютерная игра» превратились в синонимы.

На сегодняшний день почти каждый ребенок проводит определенное количество времени за компьютерными играми. Даже взрослы и состоятельным мужчина любит потрепать свои нервишки в видеоиграх. Компьютерные игры рассчитаны на разные возрастные категории, начиная с детей, заканчивая взрослыми людьми. Пик популярности компьютерных игр начался в момент технологического прогресса середины XX века. Компьютерные игры появились раньше игровых компьютерных программ. Это связано с особенностями постановления и развития компьютерных технологий. Электронные устройства были компактнее, нежели первые компьютеры, которые едва ли умещались в отдельную комнату.

История создания

История появления первых компьютерных игр начинается с 1940 года. Долгое время игры были непопулярны из-за того, что люди не имели компьютеров. Уже с 1970 годов начинают появляться аркадные автоматы, игровые консоли и домашние ПК – Персональные компьютеры.

Первый прототип компьютерной игры был создан в США. Эта игра получила название Space War, что в переводе означает Космическая война и была рассчитана на двух игроков. Суть игры была уничтожить своего противника с помощью ракет. К сожалению, в те времена игра не получила должного внимания и популярности из-за дороговизны ПК, и нужного оборудования.

В 1984 году в России появилась всеми известная электронная игра «Ну, погоди». Этот был небольшой планшет с экраном был произведён под торговой маркой «Электроника ИМ-2», помимо игровой функции, данное устройство умело показывать время и работать в качестве будильника. Данная игрушка является полной копией игры Nintendo EG-26 Egg. Заграничный волк был заменён на нашего волка из советского мультика, а петушок на зайца. Думаю, все знают в чем заключается задача этой игры, поэтому рассказывать об этом нет смысла

Классификация жанров компьютерных игр.

* **Приключенческие.** Эти компьютерные игры называются «бродилки», которые опираются на исторические сюжеты. Они оформлены как мультфильм с возможностью управлять главным персонажем. Такие игры тренируют сообразительность и логику. Главный герой на своём пути встречает разные предметы, которые помогают ему увеличить свои способности или, наоборот, могут нанести вред, а также попадает в различные квесты.
* **Стратегии.** В жанре этой игре необходимо уметь планировать, рассчитывать свои силы, анализировать ситуацию и принимать верные решения. Это достаточно сложные игры, но именно они формируют усидчивость и умение планировать свои действия заранее.
* **Аркады.** Это многоуровневые игрушки. Задачей игры является прохождение от уровня к уровню, которую можно выполнить, проделывая конкретные задания и получив специальную награду. Для такой игры характерна система баллов, которые получает игрок за прохождение каждого уровня. Такая игра хорошо развивает глазомер и внимание.
* **Обучающие игры**. В их основе лежит необходимость совершения тех или иных действий, направленных на тренировку своих определенных навыков;
* **Симуляторы**. Это виртуальные игры, в которых ты можешь почувствовать себя учителем, таксистом, стримером и другими.
* **Экшены**. Это игры, с захватывающим сюжетом, предполагающим активные и непрекращающиеся действия со стороны игрока, основанные на постоянных нажатиях клавиатуры и использования игровой мышки
* **Головоломки и логически игры.** К ним можно отнести все настольные игры, например, пасьянс, шахматы, монополия.
* **Смешанные**. Эти игры имеют в себе не один, а несколько жанров из перечисленных выше.

Основным жанром в игровой индустрии последних лет является смешанный жанр, реализации качественной 3D графики, наполненный игровым сюжетом, историей и динамикой.

Особенности компьютерных игр.

Компьютерные игры всегда поднимают настроение людям. Это происходит из-за того, что они имеют определенные характерные особенности, с помощью которых не дают заскучать и дарят игроку положительные эмоции. К таким особенностям относятся следующие:

1. Разработанный, занимательный сюжет, благодаря которому можно занимательно и увлекательно провести всё свое свободное время.
2. Если игра не нуждается в донате – это отличная игра, потому что нет необходимости платить деньги в игру, чтобы прокачать свой аккаунт.
3. Так же внимание стоит уделить на разнообразие видеоигр, так как разработчики постоянно создают новые обновления и предложения, тем самым стараются подчеркнуть их основные рабочие качества.
4. Постоянно выпускаются разные модификации и дополнения, поэтому игровой процесс становится более увлекательным, захватывающим и привлекательным.
5. Игра имеет качественную производительность, графику и звук. Благодаря этому можно спокойно добиться своей задачи и получить удовольствие от игры

Компьютерные игры все разные для новичков и профессиональных игроков существует огромное множество, поэтому стоит предварительно ознакомиться с описанием, деталями и правилами, чтобы не столкнуться с проблемами. А также важно помнить о том, что каждая игра имеет системные требования, к которым не нужно пренебрегать, поскольку именно так можно добиться практичности и удобства применения скачанной компьютерной игры в последующем.

# **ГЛАВА 2. СОЗДАНИЕ СВОЕЙ 2D ИГРЫ**

## **Подглава 2.1 Ход проделанной работы**

Итак, приступаем к нашей работе. Первыми шагами к написанию нашей работы было ознакомление со всей нужной нам информацией, а именно с внутренней составляющей компьютерных игр и языком программирования Python. После чего мы начали прописывать определенные части своей работы, чтобы в последствии упростило нам написание нашего кода. Когда мы были готовы приступать к нашей работе, мы открыли свой редактор кода, который используем для написания программ на Python. Им является PyCharm, потому что он очень прост в чтении языка, имеет приятное и понятное для глаз оформление и удобен в устранении своих ошибок при написании кода программы. Далее мы создали файл своей работы, чтобы производить дальнейшие сохранения по ходу написания кода.

Написание нашей игры начиналось с выбора библиотек, классов и модулей. У Python существует 2 библиотеки, которые подходят для создания игр: Pygame и Arcade. Прочитав про каждую из них, мы решили остановиться на Pygame, потому что наша игра одноуровневая и не требует множества кнопок для управления, как, например, на платформере. Pygame — это «игровая библиотека», набор инструментов, помогающих программистам создавать игры. К ним относятся: графика, анимация, звук, управление. Конечно, создавать какую-то новейшую игру, которую никто никогда не делал мы не собираемся, ведь превыше всего для нас является лишь ознакомление с организацией процесса написания игры, а также усовершенствование этой игры на более яркую и интересную, с нашим кодом и нашими созданными моделями. Мы создаю игру, чем-то схожую на очень старую классическую аркадную игру Space Invaders, созданную еще в 1978 году для аркадных автоматов. Моя же игра называется Космические воины.

Написание кода, автоматизации процессов и проверка на работоспособность нашей игры у нас отняли около 1 месяца моей работы. Мы думаем, так как не многие разбираются в IT-сфере и программировании, расписывать наши подробные написания и действия мы не станем, потому что многим они будут не понятны и будут только занимать место на листе. Именно поэтому мы опишем это вкратце и на более простом языке.

Мы установили библиотеку Pygame, импортировали ее и начали писать код. Помимо этой библиотеки мы так же использовали модуль sys, которыйпредоставляет доступ к переменным, используемым или поддерживаемым интерпретатором, а также к функциям, которые тесно взаимодействуют с интерпретаторами, а также Спрайт — это элемент компьютерной графики, представляющий объект на экране, который может двигаться. В двухмерной игре все, что вы видите на экране, является спрайтами. Изначально мы распределили все части своей игры, которые нужно будет прописать в отдельных файлах. Их количество составило 7: Главный файл, который содержал в себе все данный остальных файлов (см. Приложение 2); файл пули (см. Приложение 4); файл своего космического корабля, то есть пушки (см. Приложение 3); файл инопланетян (см. Приложение 5); файл счёта и рекорда игры (см. Приложение 6); файл статистики, сохраняющей твоей рекорд (см. Приложение 7); файл со всеми управляющими функциями и данными, которые задействованы в остальных файлах (см. Приложение 8).

Начали мы с создания окна, цвета, текста и фона. Затем приступили к написанию кода работы пушки (см. Приложение 2 и Приложение 3) и создании модельки которую мы нарисовали при помощи сайта Pixilart (см. Приложение 10). После чего нашей задачей стало создание пули, вылетающей из моей пушки. Никаких проблем при ее создании не возникло, потому что в библиотеке, которую мы использовали есть уже готовые мини-модели пуль (см. Приложение 4). Далее мы приступили к самому сложному, созданию пришельца и его движения к моей пушке. Его модельку мы так же сделали на сайте Pixilart (см. Приложение 9). Именно написание кода пришельца заняло у меня очень много дней (см. Приложение 5). И вот, уже конец, осталось только добавить счёт, который накапливается по ходу устранений пришельцев, оставшиеся жизни и собственный личный рекорд (см. Приложение 6). Игра готова! (см. Приложение 1)

# **ЗАКЛЮЧЕНИЕ**

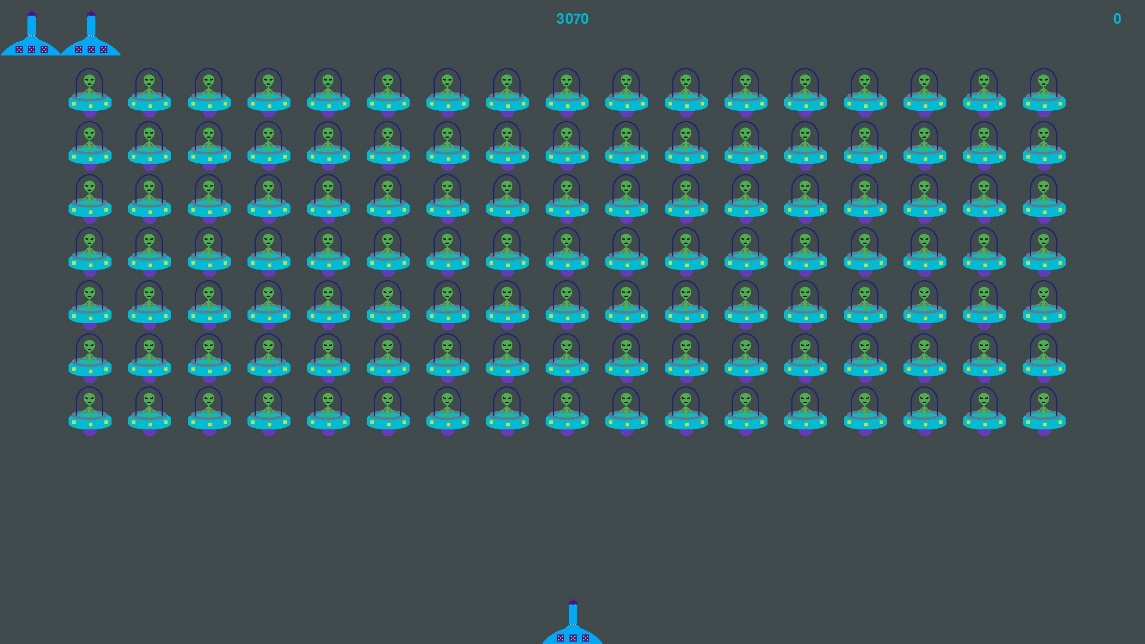
В ходе моей проделанной исследовательской работы нам удалось добиться желаемого результата. У нас получилось создать игру на языке программирования Python, конечно, не без каких-либо проблем и недочётов. К такому недочёту можно отнести отсутствие звуков, что в дальнейшем мы постараемся исправить. Мы ожидали, что создание игры не будет какой-то тяжелой работой, но оказалось все совсем иначе. В отличие от создания 2D моделей, написание кода и автоматизация работы функций в моих кодах было очень трудоемким занятием, ведь если мы сделаем хоть один не правильный шаг, то будто бы попадаем в зыбучие пески, из которых очень тяжело выбраться. Но все же мы справились с нашими задачами и поставленной целью и в дальнейшем постараемся доработать свою игру и выйти на еще более высокий уровень. Я попросил своих друзей оценить мою игру и что удивительно, игра, по их мнению, оказалась очень даже хороша для первого раза и понравилась всем.

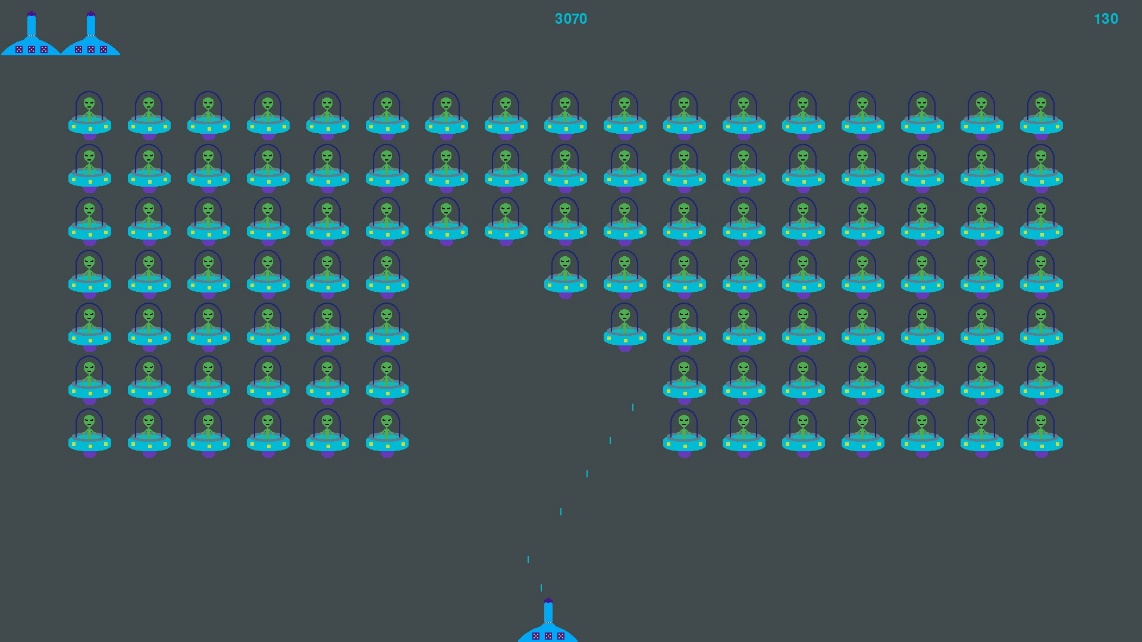
# **БИБЛИОГРАФИЧЕСКИЙ СПИСОК**

1. Как устроены компьютерные игры: 2D игры // zen.yandex.ru URL: https://zen.yandex.ru/media/id/5dc19d782fda8600af6cbc4d/kak-ustroeny-kompiuternye-igry-2d-igry-5e21f73a92414d00b16b53f5 (дата обращения: 08.02.2022).
2. Библиотека Pygame / Часть 1. Введение // Хабр URL: https://habr.com/ru/post/588605/ (дата обращения: 03.02.2022).
3. Компьютерные игры: классификация и особенности // Playground URL: https://www.playground.ru/misc/news/kompyuternye\_igry\_klassifikatsiya\_i\_osobennosti-323484 (дата обращения: 15.02.2022).
4. Создайте 2D-игру в Unity // Unity URL: https://unity.com/ru/how-to/beginner-2D-game-resources (дата обращения: 08.02.2022).
5. Самоучитель Python // Python 3 для начинающих URL: https://pythonworld.ru/samouchitel-python (дата обращения: 03.02.2022).
6. Как программировать игры на Python с помощью библиотеки Pygame // wikiHow URL: https://ru.wikihow.com/программировать-игры-на-Python-с-помощью-библиотеки-Pygame (дата обращения: 10.02.2022).
7. Создание игр на Python 3 и Pygame: Часть 1 // envatotuts+ URL: https://code.tutsplus.com/ru/tutorials/building-games-with-python-3-and-pygame-part-1--cms-30081 (дата обращения: 28.01.2022).
8. Создание 2D платформера на Python // itProger URL: https://itproger.com/news/sozdanie-2d-platformera-na-python (дата обращения: 28.01.2022).
9. 2D. Что такое 2d? // TOPAuthor URL: https://www.topauthor.ru/2d\_\_chto\_takoe\_2d\_f664.html (дата обращения: 03.02.2022)
10. Что такое Python: чем он хорош, где пригодится и как его выучить // Медиа Технологии URL: https://netology.ru/blog/python (дата обращения: 15.02.2022).
11. Что такое Python и для чего он используется // Все о Python URL: https://all-python.ru/osnovy/yazyk-programmirovaniya.html (дата обращения: 15.02.2022).
12. Уроки начинающим // PythonRu URL: https://pythonru.com/ (дата обращения: 07.02.2022).
13. ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И СВЯЗЬ. МОЖНО ЛИ СДЕЛАТЬ КАРЬЕРУ В IT? // ПРОФориентатор URL: https://proforientator.ru/publications/articles/informatsionnye-tekhnologii-i-svyaz-mozhno-li-sdelat-kareru-v-it.html (дата обращения: 05.02.2022).
14. CREATE AND SHARE ART // PixilArt URL: https://www.pixilart.com/ (дата обращения: 27.01.2022).
15. Что такое видеоигра: жанры, история, зависимость // SPRINTinvest.RU URL: https://sprintinvest.ru/chto-takoe-videoigra-zhanry-istoriya-zavisimost (дата обращения: 10.02.2022).
16. Ю.Д. Романова Информатика и информационные технологии - 3-е изд. - Новое Экономическое Образование, 2009. - 688 с.
17. Брякалов Г. А. и др. Информатика и информационные технологии. - 1-е изд. - СПб.: Корона принт, 2005. - 672 с.
18. Марк Лутц Изучаем Python. - 4-е изд. – Пер. с англ. - СПб: Символ-Плюс, 2011. - 1280 с.
19. Прохоренок Н., Дронов В. Python 3 и PyQt 5. Разработка приложений. - 2-е изд. - СПб.: Профессиональное Программирование, 2016. - 832 с.
20. Эл Свейгарт Учим Python, делая крутые игры. - 4-е изд. – Пер. с англ. - М.: Бомбора, 2018

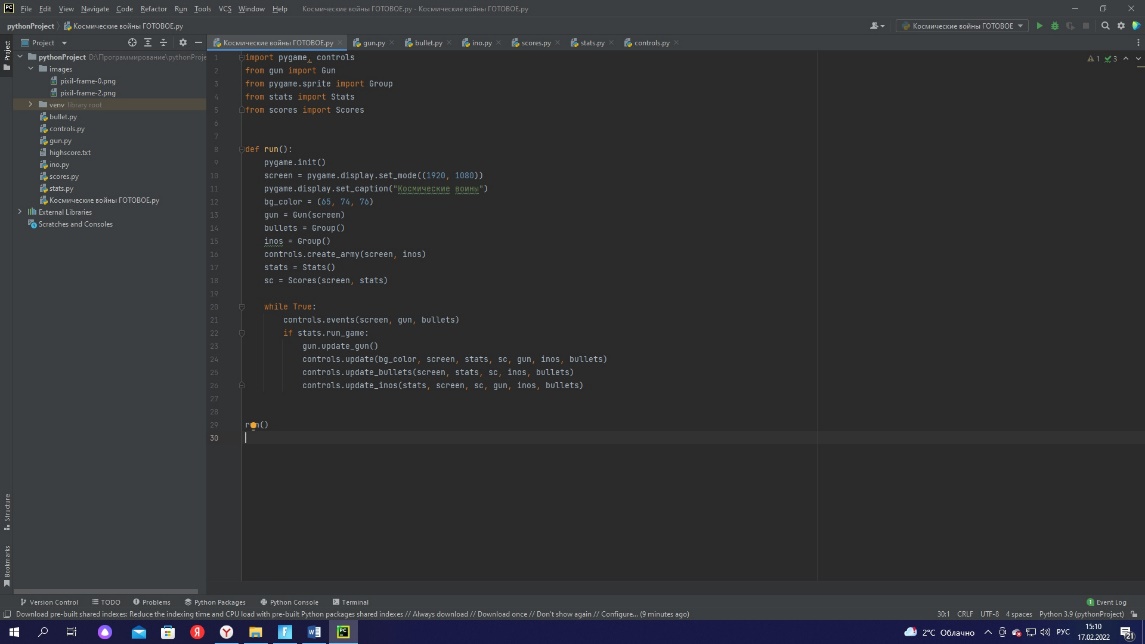
# **ПРИЛОЖЕНИЯ**

Приложение 1

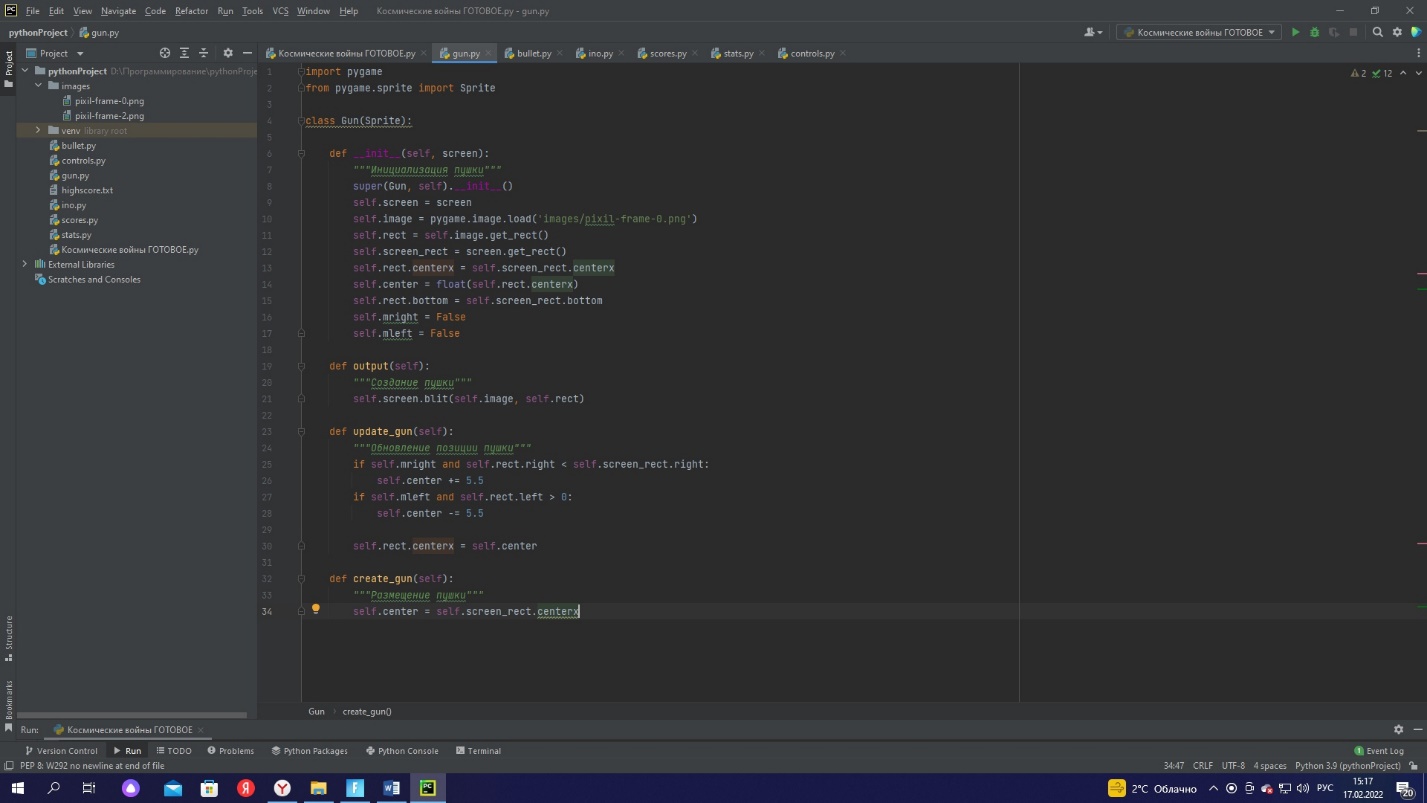
[](#_Подглава_2.1_Ход)



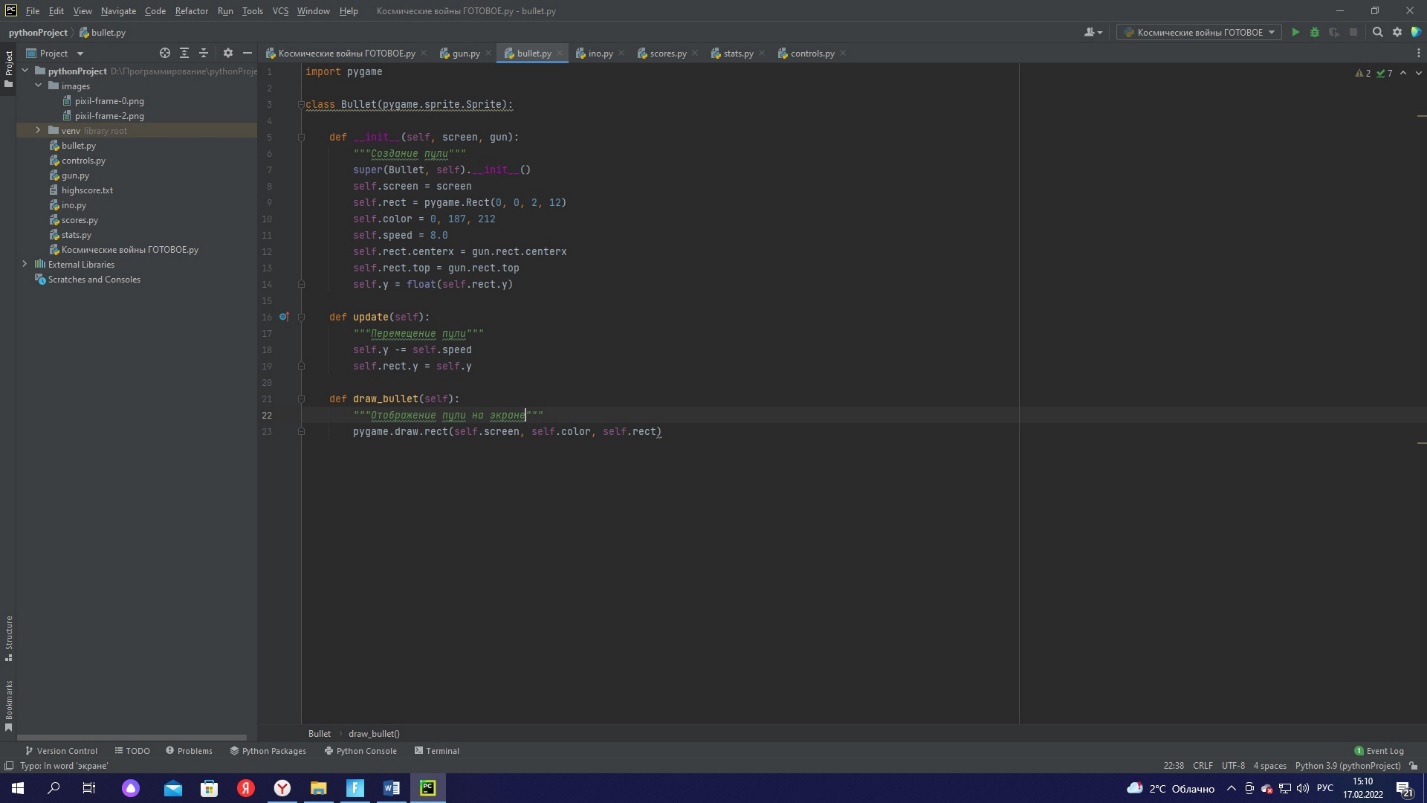
Приложение 2



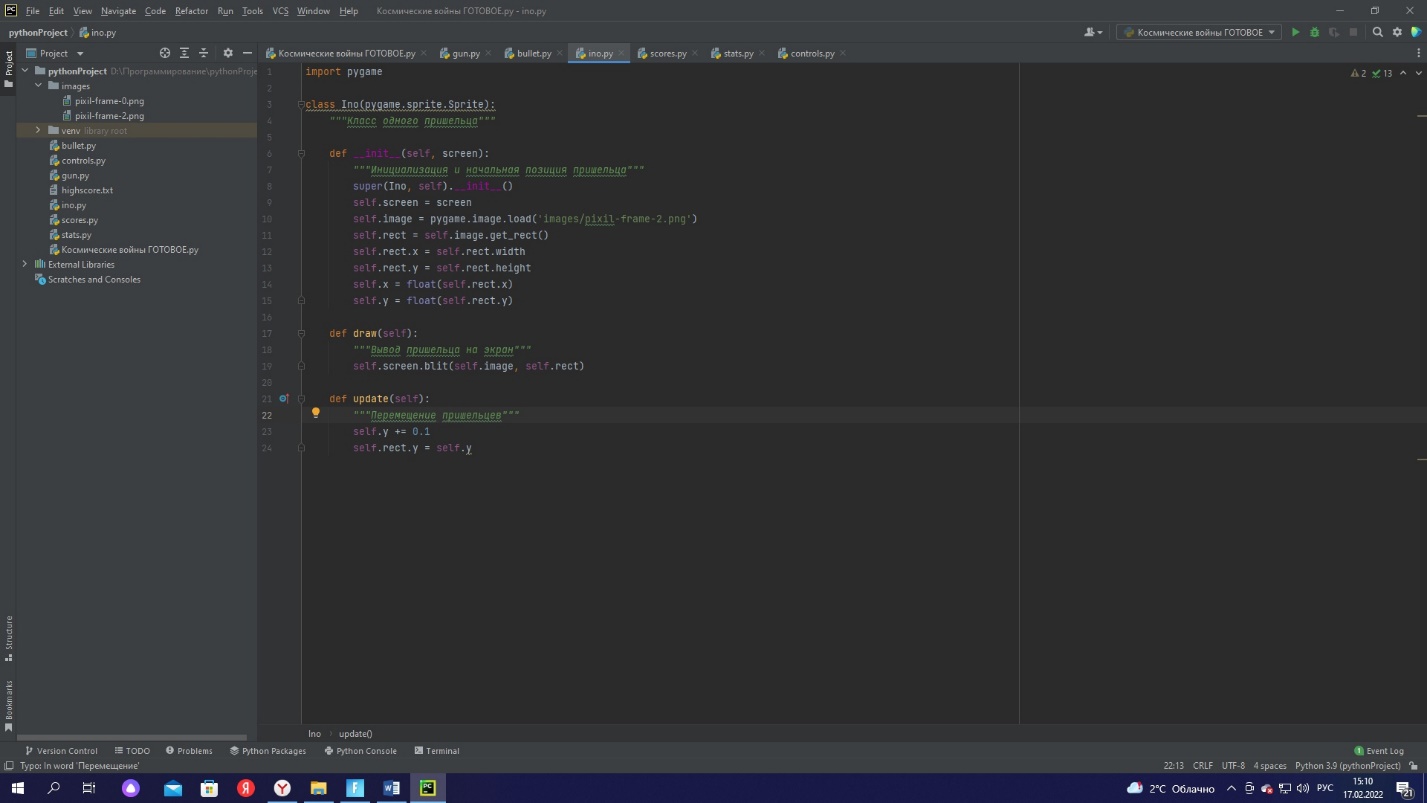
Приложение 3



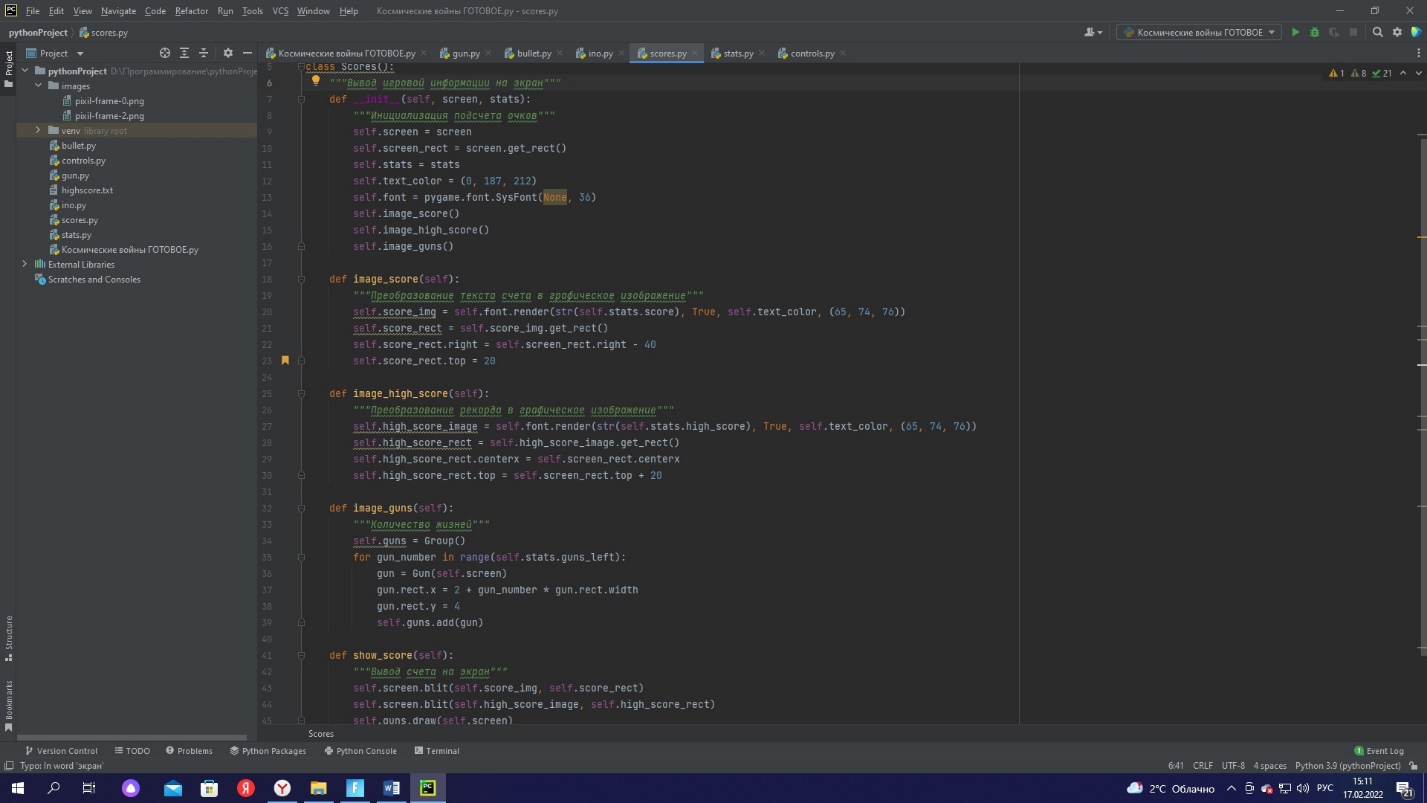
Приложение 4



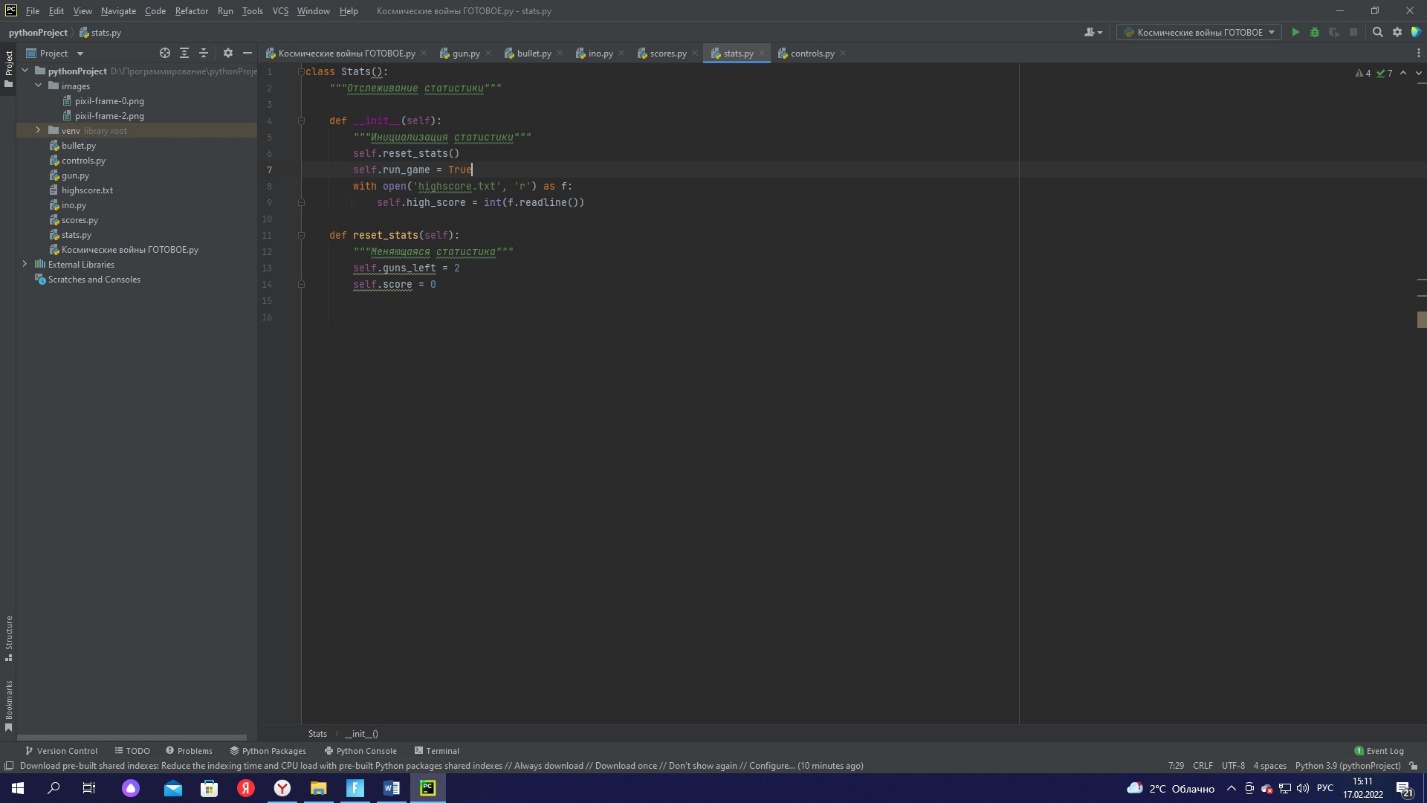
Приложение 5



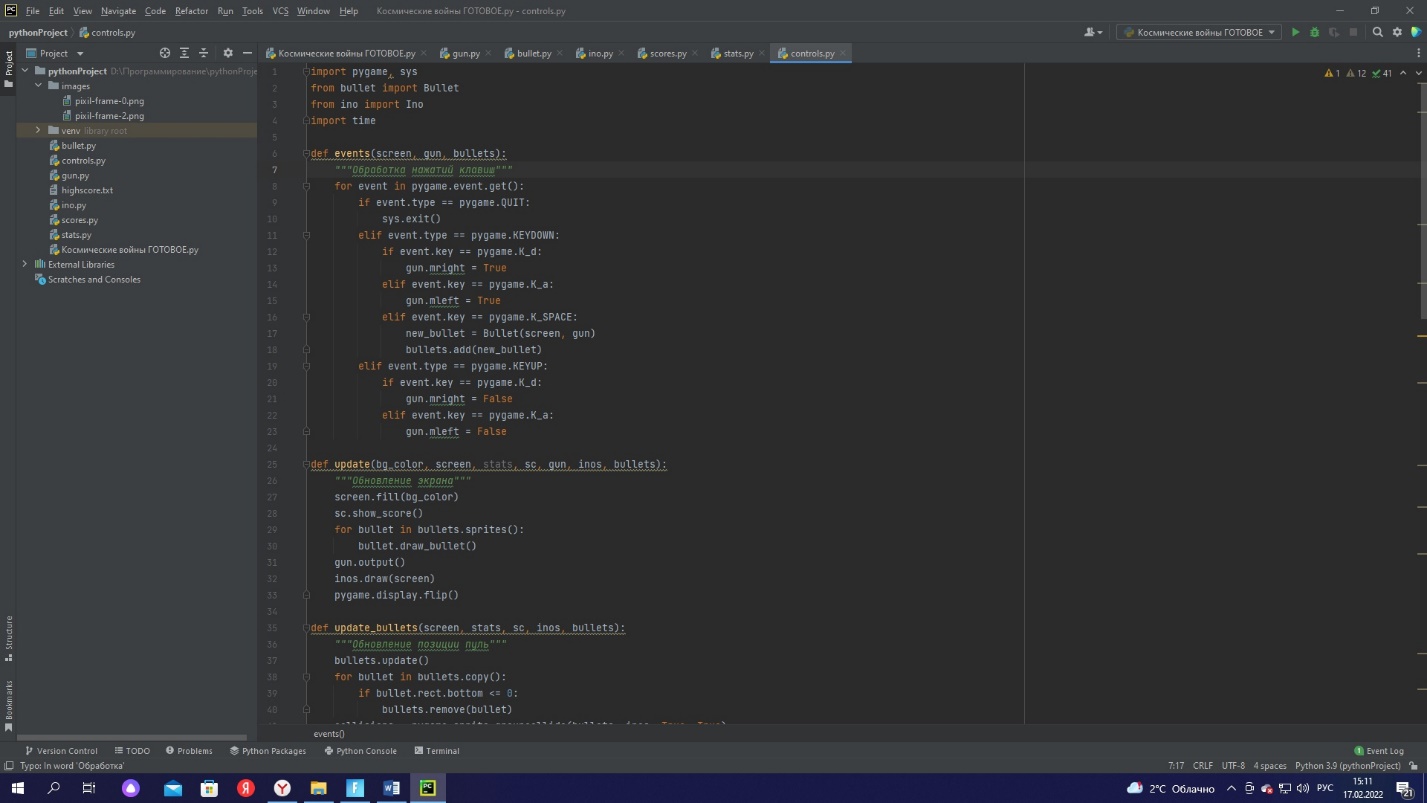
Приложение 6

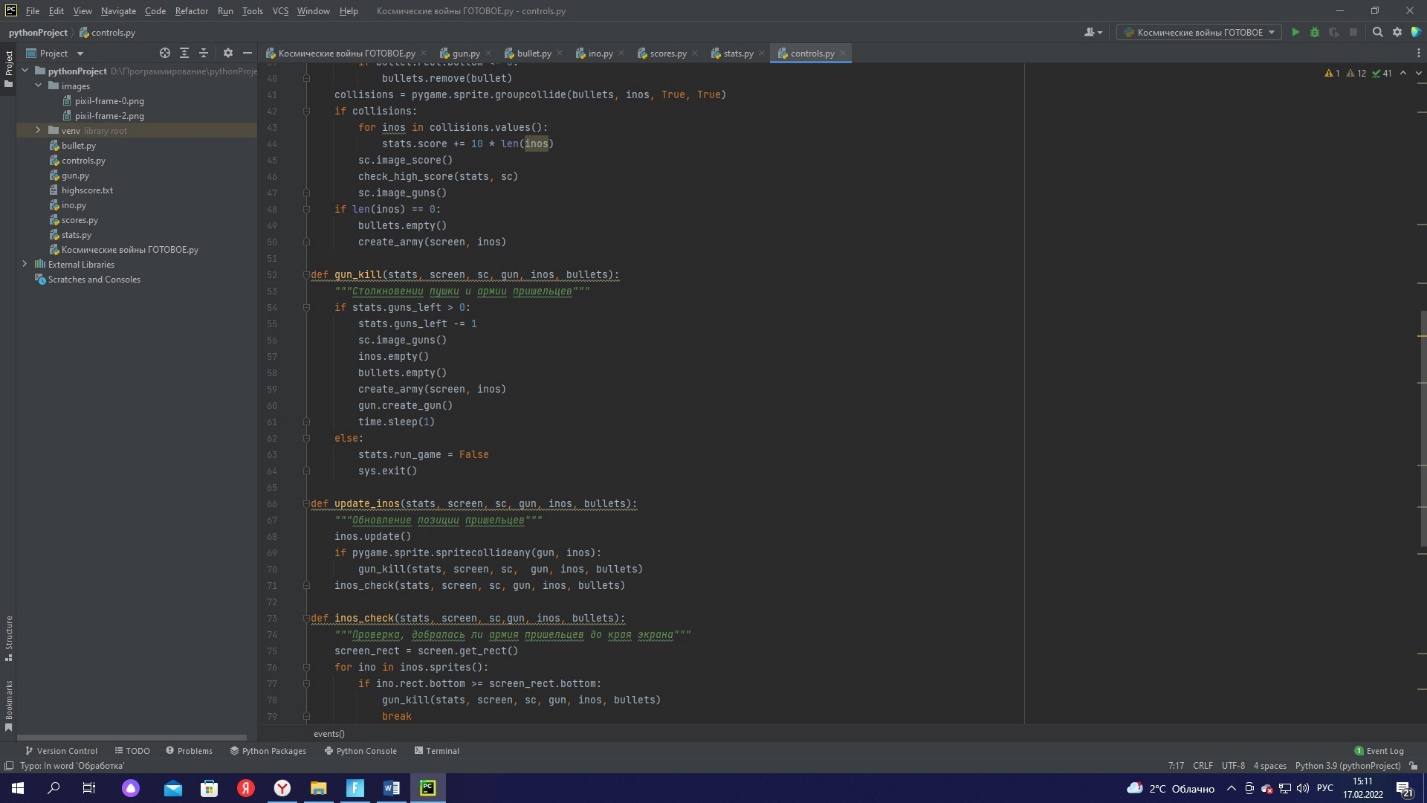


Приложение 7



Приложение 8







Приложение 9



Приложение 10

