

Министерство образования и науки Пензенской области

Управление образования города Пензы

Муниципальное бюджетное общеобразовательное учреждение гимназия № 42

НПК школьников «Я исследую мир» 2021/2022, Пенза

**«Создание компьютерных игр»**

Выполнил:

Кочегаров Данила Сергеевич,  
ученик 10а класса

Руководитель:

Щелчкова Елена Валентиновна

2021 год

# Содержание

<b>I.</b>	<b>Введение.....</b>	<b>3</b>
<b>II.</b>	<b>Основное содержание. ....</b>	<b>6</b>
<b>2.1.</b>	<b>Теоретические основы программирования на языке Python.....</b>	<b>6</b>
2.1.1.	Сущность языка Python. ....	6
2.1.2.	История и философия языка Python. ....	8
<b>III.</b>	<b>Практическое применение языка Python для создания игр. ....</b>	<b>10</b>
<b>3.1.</b>	<b>Игра «IRRUPTION».....</b>	<b>10</b>
3.1.1.	Философия игры. ....	10
3.1.2.	Архитектура кода.....	11
3.1.3.	Поэтапное создание игры.....	12
<b>3.2.</b>	<b>Игра «SHIELDS».....</b>	<b>13</b>
3.2.1.	История создания игры и её смысл. ....	13
3.2.2.	Архитектура кода.....	14
3.2.3.	Факты об игре и идеи для развития.....	15
<b>IV.</b>	<b>Заключение.....</b>	<b>16</b>
<b>V.</b>	<b>Источники и литература. ....</b>	<b>17</b>
<b>VI.</b>	<b>Приложение.....</b>	<b>18</b>

# I. Введение.

«Кто сам программирует свои компьютерные игры, тот наслаждается дважды».

(Ж. Арсак)

Я хотел бы рассказать и показать, какую роль занимают игры в наше время. Игры присутствуют в нашей жизни с самого рождения. Люди на протяжении всего своего развития каждый раз придумывали новые способы развлечения. Некоторые из вас даже не догадываются, что игры, в частности, настольные, появились аж за 5000 лет до нашей эры, ещё до появления письменности!

Первой настольной игрой принято считать «Кости», которыми мы привыкли играть и по сегодняшний день. Набор из 49 вырезанных и покрашенных камешков был найден во время раскопок 5000-летнего кургана Башур Хаюк на юго-востоке Турции. Они являются самыми древними игровыми компонентами, найденными археологами.

Значение слова *игра* постоянно меняется.

Многие очень умные люди пытались дать точное определение слову *игра*. Например:

- В 1978 году, в своей книге «The Grasshopper», Бернард Сьютс (бывший в ту пору профессором философии в университете Ватерлоо) заявил, что «игра — это добровольная попытка преодоления ненужных препятствий»<sup>1</sup>.
- Легенда игровой индустрии Сид Мейер сказал, что «игра — это серия интересных решений».
- В книге «Game Design Workshop» Трейси Фуллертон определила игру как «закрытую формальную систему, которая вовлекает игроков в организованный конфликт и разрешает его неопределенность в неравноценности результатов»<sup>2</sup>.
- В книге «Game Design Theory» Кит Бургун представил намного более узкое определение игры: «Система правил, руководствуясь которыми агенты конкурируют друг с другом, делая неоднозначные, эндогенно значимые решения», иначе говоря, следуя интуиции<sup>3</sup>.

---

<sup>1</sup> Bernard Suits, The Grasshopper (Toronto: Toronto University Press, 1978), 56.

<sup>2</sup> Tracy Fullerton, Christopher Swain и Steven Hoffman. Game Design Workshop: A Playcentric Approach to Creating Innovative Games, 2<sup>nd</sup> ed. (Boca Raton, FL: Elsevier Morgan Kaufmann, 2008), 43.

<sup>3</sup> Keith Burgun. Game Design Theory: A New Philosophy for Understanding Games (Boca Raton, FL: A K Peters/CRC Press, 2013), 10, 19.

Теперь, когда у вас есть примерное представление о понятии игры, мы можем продолжить разговор о них.

Жанров игр на данный момент существует бесчисленное множество, одними из самых популярных являются:

## **RPG<sup>4</sup>**

Где игрок отыгрывает выбранную им роль.

## **Аркады**

Если говорить на простом языке и не вмешиваться в подробности, то аркада — это мини-игра, которая характеризуется простотой игрового процесса, слабой сюжетной линией (не всегда) и наличием игрового счёта, являющегося показателем мастерства игрока.

## **Sandbox<sup>5</sup>**

Жанр, где игроку дан бескрайний мир, с которым он может делать всё, что угодно, в рамках механики игры конечно же.

## **Симуляторы**

Как ясно из названия, симуляторы — это жанр игр, характеризующийся полной или практически полной копией процессов, профессий, тематик настоящего мира, симуляция которого происходит в игре под управлением игрока. Например, это может быть управление транспортом (самолётом, автомобилем, кораблём и т.д.), или тематическая занятость (фермерство, охота, рыбалка и т.д.).

## **Adventure<sup>6</sup>**

Игра-повествование, в которой управляемый игроком герой продвигается по сюжету и взаимодействует с игровым миром посредством применения предметов, общения с другими персонажами и решения логических задач.

Я с самого детства заинтересовался компьютерными играми и тем, как они работают. Нажимая на ту или иную кнопку, я думал, что же это за чудо и как это всё работает. После долгих лет внедрения в IT-сферу, изучая основы программирования в целом и углубляясь в практическое применение изученного,

---

<sup>4</sup> (расш. Role Play Game) (англ. Ролевая игра)

<sup>5</sup> (англ. Песочница)

<sup>6</sup> (англ. Приключение)

мне удалось создать две собственных игры, одну жанра Приключения, другую - жанра аркада.

Хотелось отметить, что одна из разработанных мной игр имеет **экологическую направленность**, отражающую на сегодняшний день одну из глобальных проблем **«Защиту окружающей среды и предупреждение экологических катастроф»**.

Исходя из вышеизложенного, была определена **тема исследования: «Создание компьютерных игр на языке программирования Python»**

**Цель исследования:** разработка игр на языке программирования Python

**Объект:** язык программирования Python

**Предмет:** компьютерные игры «IRRUPTION»<sup>7</sup>, «SHIELDS»<sup>8</sup>.

**Гипотеза:** создавать игры на языке программирования Python несложно, если

- изучить особенности этого языка программирования;
- узнать основные алгоритмы, необходимые для правильной работы данной игры.

**- захотеть этого;**

На основании цели и гипотезы были поставлены следующие **задачи исследования:**

- раскрыть сущность языка программирования Python
- изучить историю данного языка
- понять философию языка Python
- изучить среду разработки PyCharm
- изучить работу программ для работы со звуком и видео
- разработать игры на языке программирования Python

**Методы исследования:** анализ печатных и электронных источников информации по данной теме; анализ алгоритмов Python; синтез выявленных алгоритмов в единую программу.

---

<sup>7</sup> Irruption - (англ.) Вторжение.

<sup>8</sup> Shields - (англ.) Щиты.

## **II. Основное содержание.**

### **2.1. Теоретические основы программирования на языке Python.**

#### **2.1.1. Сущность языка Python.**

Python (в русском языке распространено название питон) — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций. Язык обладает чётким и последовательным синтаксисом, продуманной модульностью и масштабируемостью, благодаря чему исходный код написанных на Python программ легко читаем.

Python — активно развивающийся язык программирования, новые версии с добавлением и изменением языковых свойств выходят примерно раз в два с половиной года. Он находит применение во множестве сфер человеческой деятельности.

Python – не самый «молодой» язык программирования, но и не слишком старый. К моменту его создания уже существовали такие языки как «Паскаль» или «Си». А потому при создании «питона» авторы старались взять лучшее из различных платформ для разработчиков. Фактически Python представляет собой своеобразный «джем» удачных решений более чем из 8 различных языков.

Питон поддерживает практически все распространенные операционные системы. Он может прекрасно работать на карманных компьютерах, так и на больших серверах. В случае если платформа значительно устареет, она исключается из поддержки ядра. К примеру, версии языка, начиная от 2.6, уже не работают с платформами Windows 95, 98 и ME. В случае необходимости можно воспользоваться более старыми версиями, отказавшись от применения современных инструментов языка. И тогда приложение будет работать, в том числе с этими ОС.

Python относится к наиболее востребованным и популярным языкам программирования, о чем свидетельствуют многочисленные рейтинги и анализ предложений на рынке разработки программных продуктов.



Rank	Language	Type	Score
1	Python	🌐 🖥️ ⚙️	100.0
2	Java	🌐 📱 🖥️	95.4
3	C	📱 🖥️ ⚙️	94.7
4	C++	📱 🖥️ ⚙️	92.4
5	JavaScript	🌐	88.1
6	C#	🌐 📱 🖥️ ⚙️	82.4
7	R	🖥️	81.7
8	Go	🌐 🖥️	77.7
9	HTML	🌐	75.4
10	Swift	📱 🖥️	70.4

Таблица 1. Рейтинг языков программирования 2021 года по версии IEEE Spectrum.

Он достаточно прост, а потому изучение языка не займет слишком много времени. При запуске Python появляется окно интерпретатора Python Shell. Оно переводит понятный человеку код в машинный язык, то есть в код, который может выполнить процессор устройства. Здесь находятся вкладки «Файл», «Редактировать», «Отладка», «Опции», «Окно», «Помощь». Для создания программ необходимо зайти во вкладку «Файл» и создать новый файл. Перед нами откроется окно, в котором мы будем писать код.

Для создания программ часто необходимы дополнительные функции. Для этого существуют специальные библиотеки. Библиотеки могут использоваться для создания оконных приложений с кнопками, картинками и так далее. Существуют специальные библиотеки для создания игр. Некоторые из них встроены в Python, некоторые нужно скачивать отдельно.

На основании вышеизложенного можно сделать вывод, что Python - активно развивающийся и самый перспективный язык программирования, который поддерживает почти все операционные системы. Он простой и удобный в использовании.

## 2.1.2. История и философия языка Python.

История языка программирования Python началась в конце 1980-х. Гвидо ван Россум задумал Python в 1980-х годах, а приступил к его созданию в декабре 1989 года в центре математики и информатики в Нидерландах. Язык Python был задуман как потомок языка программирования ABC, способный к обработке исключений и взаимодействию с операционной системой Амёба. Ван Россум является основным автором Python и продолжал выполнять центральную роль в принятии решений относительно развития языка вплоть до 12 июля 2018 года.

Название языка произошло вовсе не от названия семейства пресмыкающихся. Автор назвал язык в честь популярного британского комедийного телешоу 1970-х «Летающий цирк Монти Пайтона». Впрочем, всё равно название языка чаще связывают именно со змеей, нежели с передачей – даже эмблема на сайте python.org (до выхода версии 2.5) изображают змеиные головы. Важная цель разработчиков Python — создавать его забавным для использования. Это отражено в его названии, которое пришло из Монти Пайтона. Также это отражено в иногда игривом подходе к обучающим программам и справочным материалам.

Разработчики языка Python придерживаются определённой философии программирования, называемой «The Zen of Python»<sup>9</sup>. Её текст выдаётся интерпретатором Python по команде `import this` (работает один раз за сессию). Автором этой философии считается Тим Петерс (Tim Peters). По сути, это набор основных принципов, которые разработчики языка считают очень важными.

---

<sup>9</sup> «Дзен Питона», или «Дзен Пайтона»



### Текст философии:

- Красивое лучше, чем уродливое.
- Явное лучше, чем неявное.
- Простое лучше, чем сложное.
- Сложное лучше, чем запутанное.
- Плоское лучше, чем вложенное.
- Разреженное лучше, чем плотное.
- Читаемость имеет значение.
- Особые случаи не настолько особые, чтобы нарушать правила.
- При этом практичность важнее безупречности.
- Ошибки никогда не должны замалчиваться.
- Если не замалчиваются явно.
- Встретив двусмысленность, отбрось искушение угадать.
- Должен существовать один – и, желательно, только один – очевидный способ сделать это.
- Хотя он поначалу может быть и не очевиден, если вы не голландец.
- Сейчас лучше, чем никогда.
- Хотя никогда зачастую лучше, чем прямо сейчас.
- Если реализацию сложно объяснить – идея плоха.
- Если реализацию легко объяснить – идея, возможно, хороша.
- Пространства имён – отличная вещь! Давайте будем делать их больше!

Можно сделать вывод, что язык Python появился относительно недавно. Разработчики «питона» подходят к его созданию с юмором для облегчения работы с языком. Изучив философию Python можно ещё раз убедиться в этом. По мнению его разработчиков, важно как можно больше упростить процесс программирования, сделать его доступным и понятным для каждого.

# III. Практическое применение языка Python для создания игр.

## 3.1. Игра «IRRUPTION».

### 3.1.1. Философия игры.

*Игра — последовательность интересных выборов.*

*Сид Мейер*

Проблема экологии является ключевой для человечества. Ее решение возможно если человек осмыслит взаимоотношение человека и природы. Бережное отношение к окружающей среде позволит избежать экологических катастроф. Чтобы предупредить или исключить экологические катастрофы, прежде необходимо понять, насколько важно сохранять природную гармонию.

Сюжет моей игры основывается на персонажах: **Волчи**, **Рыська** и **Ирбис**.

Люди настолько загрязнили окружающую среду разными химикатами, что животные начали мутировать: у них развился мозг, прямохождение, разум и они стали похожи на людей.

Животным не очень нравится их новый облик, ведь так они не будут уживаться с другой расой - самими людьми. По-хорошему просить не получалось, поэтому отряды животных, один из которых возглавляет главнокомандующий Ирбис, начали замышлять как бы им разобраться с людьми

**Выход один – война.**

На нашей планете разразился хаос и вследствие этого Земли больше не стало.



Рисунок 1. Фрагмент из игры - взрыв Земли.

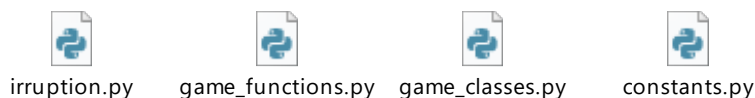
### 3.1.2. Архитектура кода.

Прежде чем перейти к написанию игры, или же программы, необходимо тщательно продумать, из каких частей она будет состоять. Иначе говоря, разработать её архитектуру.

Архитектура программы – это описание компонентов программы.

Структура моего программного кода состоит из классов функций и различных переменных.

Иерархия файлов кода такова:



- Файл `irruption.py`

Этот файл отвечает за работоспособность кода – последовательную и правильную работу функций.

- Файл `game_functions.py`

Данный файл хранит в себе весь функционал моей игры, от отрисовки персонажа, до работы с Базами Данных, для получения результатов игроков.

- Файл `game_classes.py`

В файле расположены классы - описания структур объектов и методов работы с ними.

- Файл `constants.py`

Тут находятся переменные, которые не изменяются на протяжении игры, например:

- загружаемые изображения (задний фон, спрайт-лист анимации игрока<sup>10</sup>);

- количество отрисовываемых кадров в секунду<sup>11</sup>;

- диалоги между персонажами.

Весь код для игры занял 1104 строчки кода и были использованы такие библиотеки<sup>12</sup>, как: `pygame`, `sqlite3`, `os`, `sys`, `time`, `random`, `moviepy`.

<sup>10</sup> Спрайт-лист – это изображение на котором расположены все кадры анимации.

<sup>11</sup> FPS – расш. Frames Per Second.

<sup>12</sup> Библиотека в языке Python – это коллекция модулей, которая упрощает написание кода.

### 3.1.3. Поэтапное создание игры.

По большей части игры редко создаются одним человеком, потому что это очень трудоёмкий процесс. Для проектов уровня трипл-А, так называют высокофинансированные игры, собираются целые команды разработчиков, дизайнеров, продюсеров и людей различных профессий. Но всё же есть и люди, которые могут организовать свои возможности так, что смогут создать полноценную игру в одиночку. Чтобы создать хоть какую-то *свою* игру, нужно уметь программировать, рисовать, работать со звуком и быть знакомым с огромным спектром программного обеспечения.

Жанр этой игры, не совсем приключение, это – платформер. Такими называют игры, в которых основной задачей игрока является прыгать по платформам. С графикой для игры мне помог друг – художник, который нарисовал всё, что можно увидеть в ней, от кнопок до кат-сцен<sup>13</sup>.

Создание началось задолго до начала реализации проекта. Для начала нужна была идея. Идея пришла в голову моему другу, и я скоро развил её до итогового варианта. С пришедшей идеей я стал развивать этот проект и думать, как представить сюжет в хорошем виде. Вскоре пришла идея создания игры платформера с диалогами. Сами же диалоги

Начал я с реализации движения игрока: я сделал так, чтобы игрок мог бегать и прыгать.

Затем решил создать начальный экран на котором разместил кнопки настроек, старта, выхода, и кнопку, нажав на которую можно узнать на каком уровне остановился игрок с определенным никнеймом<sup>14</sup>.

Уже впоследствии я создал окно выбора персонажа, систему диалогов с NPC<sup>15</sup>, генерацию уровней с определённой сложностью и воспроизведение кат-сцен. Кроме того, поработал над звуковой составляющей – добавил музыку на задний план и добавил эффектности и яркости кат-сценам.

Таким образом, я создал игру с сюжетом, изучил все этапы её создания и понял как тяжело организовывать свою работу так, чтобы успеть всё в ограниченные сроки.

---

<sup>13</sup> Кат-сцена — это эпизод в компьютерной игре, в котором игрок слабо или вообще никак не может влиять на происходящие события, обычно с прерыванием геймплея. Кат-сцены используются для развития сюжета, визуального представления игрового прогресса и заполнения пауз в геймплее. Сцены могут быть анимированы средствами самой игры, либо использовать отснятый видеоматериал с живым действием.

<sup>14</sup> Никнейм – имя игрока, зачастую выдуманное.

<sup>15</sup> NPC - расш. Non-Player Character, англ. Неигровой персонаж

## 3.2. Игра «SHIELDS».

### 3.2.1. История создания игры и её смысл.

Получив репутацию создателя игр, один из моих друзей предложил мне идею создания стратегической игры-аркады, которую он придумал. Недолго думая, я согласился и в течение месяца реализовал проект. Идея игры проста – это способ посоревноваться с друзьями в игре, похожей на «Морской Бой». В ней игроки поочередно расставляют определённое количество щитов на поле 6x6, тем самым защищаясь от будущего нападения друг на друга. После они нападают на поля друг друга, причём поле разделено на кольца из полей, попав на которые игрок получает определённое количество очков.



Рисунок 2. Фрагмент из игры. Правила, управление в игре.

Главным преимуществом данной игры является то, что – это standalone-приложение<sup>16</sup>.

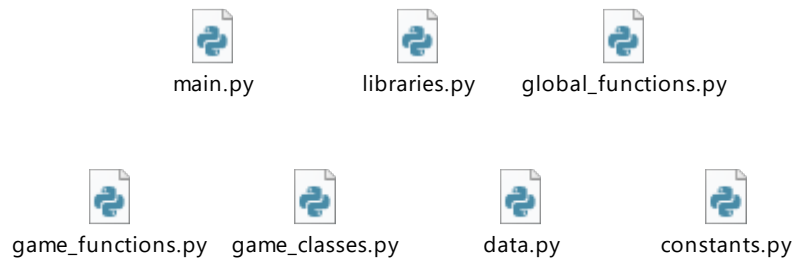
<sup>16</sup> Standalone-приложение – это приложение, которое может запускаться с любого компьютера, простым нажатием на ярлык программы.

### 3.2.2. Архитектура кода.

Весь код написан на языке Python и занимает 1042 строчки кода.

Структура моего программного кода состоит из классов функций и различных переменных.

Иерархия файлов кода такова:



- main.py

Файл, благодаря которому игра обладает работоспособностью.

- libraries.py

Файл, где импортируются все библиотеки для работы.

- global\_functions.py

Функции, которые могут быть вызваны в любом фрагменте кода.

- game\_functions.py

Функции, благодаря которым игра откликается на действия игрока и отрисовывает ему соответствующие кадры.

- game\_classes.py

Файл с классами объектов игры.

- data.py

Файл, в котором записаны переменные, ссылающиеся на графические данные игры.

- constants.py

Файл с неизменяющимися переменными.

### 3.2.3. Факты об игре и идеи для развития.

*Выигрышной стратегией, что я выяснил в ходе многочисленных тестовых игр, является нападение на внутреннее кольцо из полей, попадание в которое даёт наибольшее количество очков.*



Рисунок 3. Идеи для развития игры "Shields"

Создание и продумывание этих игр занимает очень много времени и сил. В заключение, я хочу сказать, что, конечно, тот уровень, на котором созданы эти игры очень мал, но даже в них принимали участие несколько человек: гейм-дизайнер и художник. Было интересно заниматься не только программированием, но и монтажом, работой со звуком и реализацией проекта в целом.

Я буду стараться развиваться в этой сфере, и я уверен, что в ближайшем будущем добьюсь намного больших успехов.

## IV. Заключение.

Python – активно развивающийся и самый перспективный язык программирования, который поддерживает почти все операционные системы. Он простой и удобный в использовании.

Появился Python относительно недавно. Изучив философию Python, можно ещё раз убедиться в том, что этот язык доступен каждому, кто хочет создать свой программный продукт. По мнению его разработчиков, важно как можно больше упростить процесс программирования, сделать его доступным и понятным для каждого.

Изучив Python и его принципы, мы разработали игры «**IRRUPTION**» и «**SHIELDS**».

Продумав архитектуру игры и обозначив необходимые для её работы компоненты, мы написали алгоритмы наших игр.

В играх использовалась авторская графика. Все персонажи, элементы интерфейса и основные сюжетные картины прорисованы в редакторе MediBang Paint PRO. А подогнаны под хорошее расположение на экране в Photoshop. Совокупность этих факторов делает эти игры творческими и уникальными.

Таким образом, мы изучили язык программирования Python и смогли написать игру на этом языке. Было очень интересно и увлекательно изучать историю и философию Python и создавать собственные игры, которыми ты можешь поделиться со всеми.



## V. Источники и литература.

Википедия: Python. – Режим доступа: <https://ru.wikipedia.org>

Tech: Python – краткий обзор языка и его назначения. – Режим доступа:  
<https://techrocks.ru>

Яндекс Дзен: Дзен питона. – Режим доступа: <https://zen.yandex.ru>

<https://zen.yandex.ru/media/topgames/samyepopuliarnyeigrovyejanry-5c5ee1031174ac00ad9fccad>

<https://tesera.ru/article/778208/>

Жанры игр: самые популярные жанры игр. Режим доступа:

<https://stopgame.ru/blogs/topic/92843>

[https://en.wikipedia.org/wiki/Homo\\_Ludens](https://en.wikipedia.org/wiki/Homo_Ludens)

[https://ru.wikipedia.org/wiki/Игры\\_и\\_люди](https://ru.wikipedia.org/wiki/Игры_и_люди)

Арсак Ж. Программирование игр и головоломок/ Ж. Арсак. – Наука, 1990. – 76с.

Трофимов В. Программирования игр, создание с нуля / В. Трофимов. – 2020.

Джереми Бонд «Unity и C#. Геймдев от идеи до реализации. 2-е изд.» 2019.

Bernard Suits, The Grasshopper (Toronto: Toronto University Press, 1978), 56.

Tracy Fullerton, Christopher Swain и Steven Hoffman. Game Design Workshop: A Playcentric Approach to Creating Innovative Games, 2nd ed. (Boca Raton, FL: Elsevier Morgan Kaufmann, 2008), 43.

Keith Burgun. Game Design Theory: A New Philosophy for Understanding Games (Boca Raton, FL: A K Peters/CRC Press, 2013), 10, 19.

## **VI. Приложение.**

**(Видео)**

**Ссылка: <https://youtu.be/q4mNMlkJzP0>**

РЕЦЕНЗИЯ  
на работу обучающегося 10 класса  
МБОУ гимназии № 42 г Пензы  
Кочегарова Данилы Сергеевича

по теме: **«Создание компьютерной игры экологической направленности на языке программирования Python»**

Работа Кочегарова Данилы посвящена созданию игры затрагивающей проблему экологии, «Защиту окружающей среды и предупреждение экологических катастроф», написанная на языке программирования.

Автором была поставлена цель – создать игру используя современный язык программирования Python.

Следует отметить, что автором была определена философия игры.

На этапе концептирования придумана идея игры и проработка дизайна. Авторское прототипирование персонажей, которые были созданы средствами графического редактора.

Четко определена архитектура программы, описание компонентов программы.

При написании программного кода были применены основные конструкции языка Python.

Данная работа имеет и личную значимость для ее автора, так как она свидетельствует о способности самостоятельно ставить проблему и находить пути ее решения.


В приложении представлен ход работы над проектом. А также работа над достижением поставленной цели.

Работа оформлена с учетом всех необходимых требований, содержит приложение со скриншотами созданного программного продукта.

Рецензируемая работа представляет собой серьезную и интересную разработку. Автор подробно описал ход работы над проектом.

Материал в проекте Кочегарова Данилы изложен последовательно и чётко. Выводы и заключение сделаны правильно. Считаю, что проект может быть представлен на конкурс «Высший пилотаж»

РЕЦЕНЗЕНТ



(подпись)

Щелчкова Е.В.

(ФИО)

«10 » января 2022 г