Лицей Академии Яндекса

Детский технопарк «Кванториум»

Проект по направлению Pygame по теме:

**«Компьютерная игра «Гонки против течения»**

Проект разработал:

Кирпа Юрий Павлович

Руководитель проекта:

Гринчишин Михаил Александрович,  
преподаватель   
Лицей Академии Яндекса

Великий Новгород

2025

Оглавление

[Введение 3](#_Toc189817192)

[Описание реализации 4](#_Toc189817193)

[Техническое задание 4](#_Toc189817194)

[Схема проекта 8](#_Toc189817195)

[Используемые библиотеки 10](#_Toc189817196)

[Реализуемые технологии 10](#_Toc189817197)

[Реализация проекта 10](#_Toc189817198)

[Скриншоты выполненного проекта 11](#_Toc189817199)

[Тестирование приложения 16](#_Toc189817200)

[Заключение 16](#_Toc189817201)

# Введение

Компьютерные игры — это вид программного обеспечения, предназначенный для развлечения пользователей и развития их навыков.

В современном мире компьютерные игры стали неотъемлемой частью жизни многих людей и оказывают значительное влияние на общество. Они предоставляют возможность расслабиться, отдохнуть после трудного дня, а также развить свои навыки и знания.

Поэтому мне пришла идея написать на Python с использованием pygame игру «Гонки против течения», которая сочетала бы в себе элементы различных жанров, развивала ращличные навыки и была направлена на широкую аудиторию.

**Цель:**

Создать компьютерную игру «Гонки против течения» с использованием технологии pygame.

**Задача:**

1. Составить техническое задание по теме
2. Изучить технологии pygame и необходимые инструменты
3. Протестировать работу игры

# 

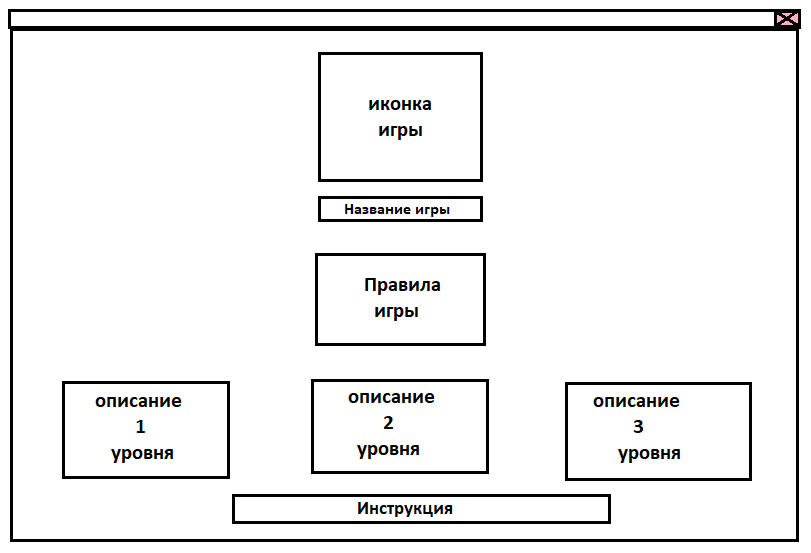
# Описание реализации

## Техническое задание

Игра должна содержать:

* Стартовое окно, содержащее в себе иконку, название и правила игры, описание её уровней
* Окно ввода имени пользователя с полем ввода и инструкцией
* 1 и 2 уровни, включающие в себя фон, машину, звезды, встречные машины, имя и картинку пользователя, количество собранных звезд, количество взрывов, инструкцию и мотивацию
* 3 уровень с теми же объектами 1 и 2 уровней за исключением фона, общее расположение всех объектов отличается от 1 и 2 уровней
* Окно между уровнями, содержащее смайлик (для мотивации) и инструкцию, что делать дальше
* Окно проигрыша со смайликом и указаниями по дальнейшим действиям
* Окно победы, включающая в себя картинку победного кубка, мотивационный текст, инструкцию, количество собранных звезд, процент пользователей, которых обошел текущий пользователь

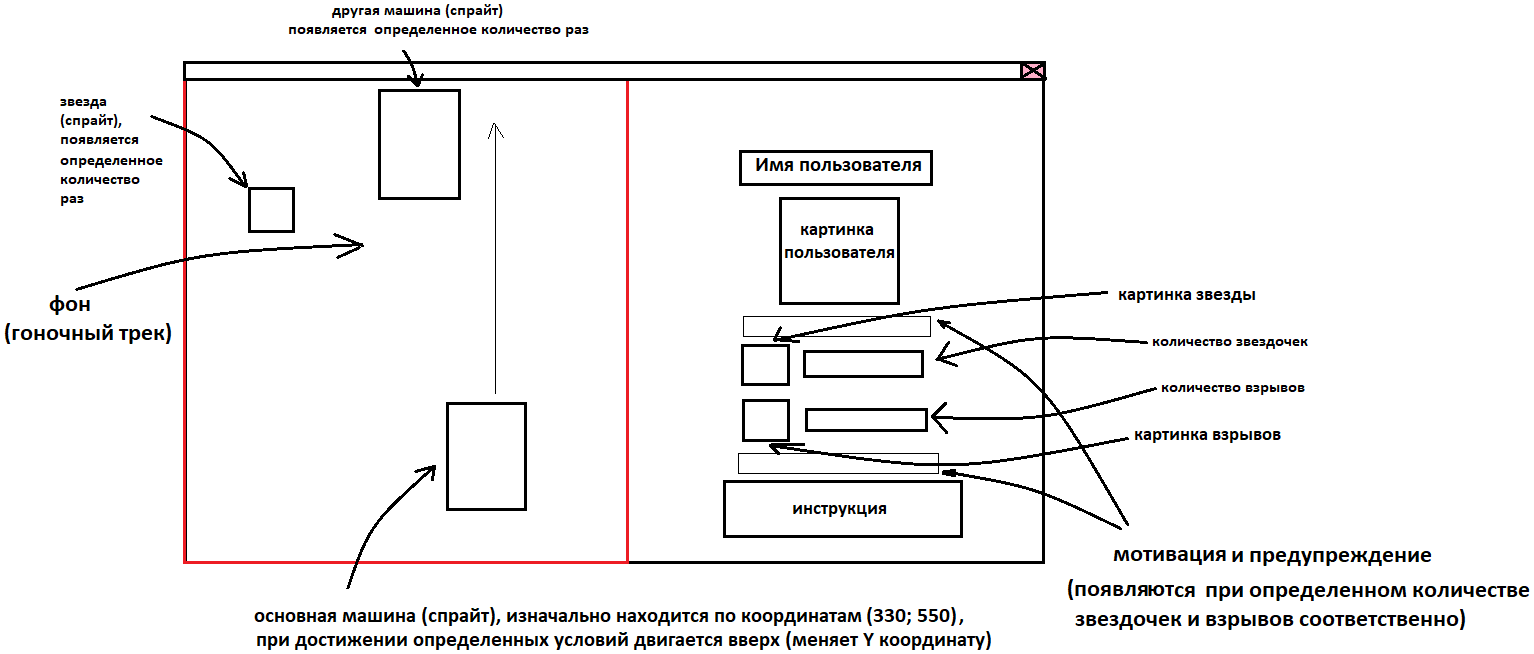
**Стартовое окно**



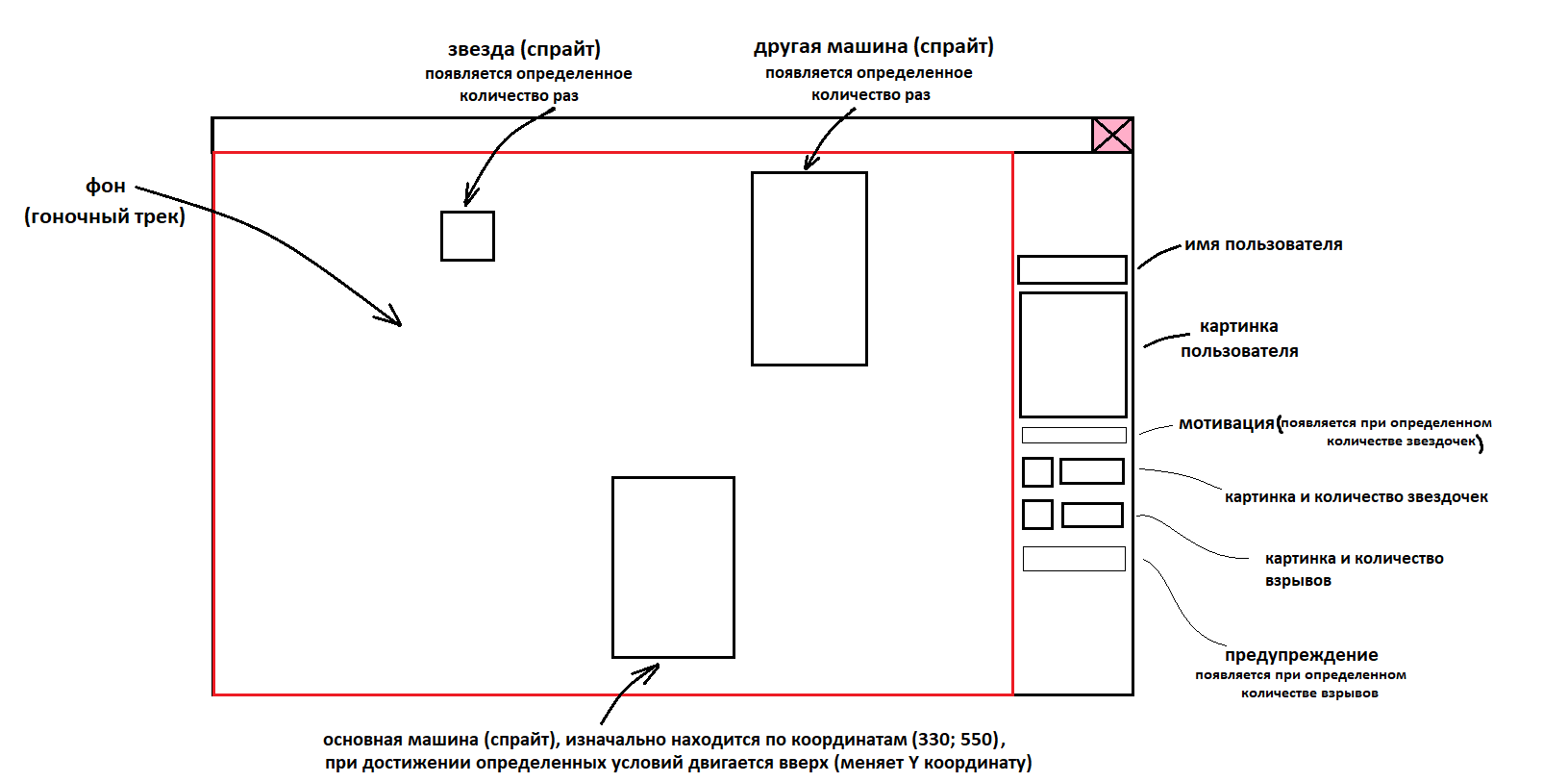
**Окно ввода имени пользователя**

****

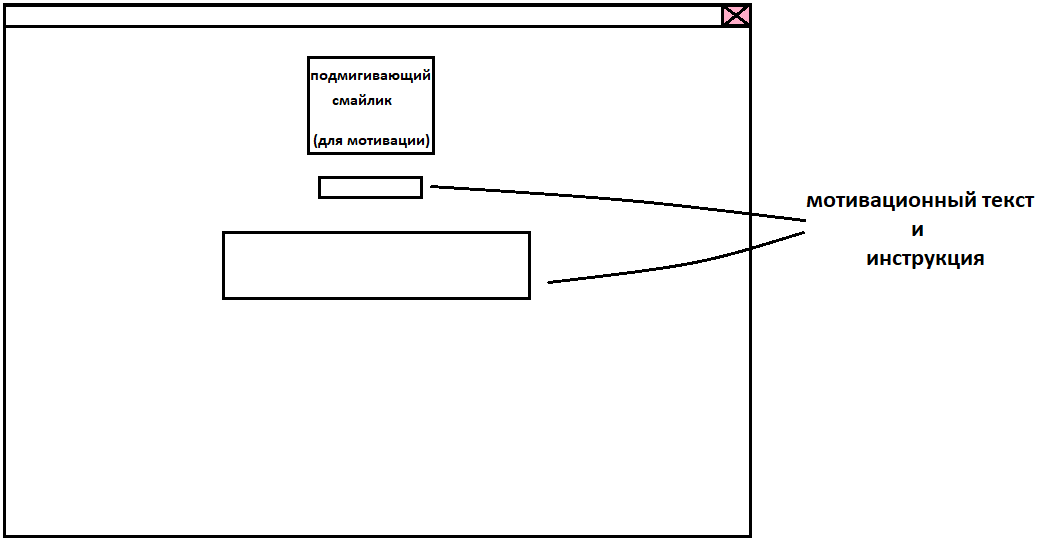
**1–2 уровни**

****

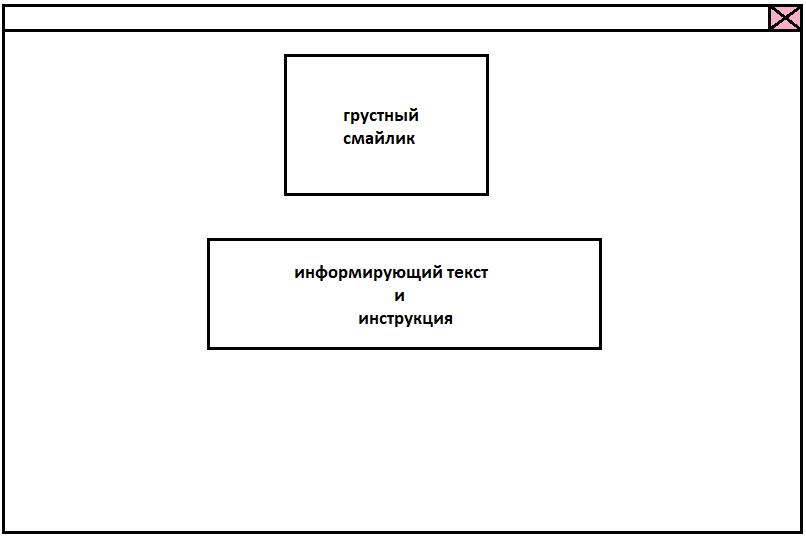
**3 уровень**

****

**Окно между уровнями**

****

**Окно проигрыша**



**Окно выигрыша**

**Изображение выглядит как текст, диаграмма, снимок экрана, линия

Автоматически созданное описание**

## Схема проекта

Изображение выглядит как текст, диаграмма, План, карта

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана, меню

Автоматически созданное описание

**Изображение выглядит как текст, диаграмма, снимок экрана, План

Контент, сгенерированный ИИ, может содержать ошибки.**

Изображение выглядит как текст, снимок экрана, диаграмма, Прямоугольник

Контент, сгенерированный ИИ, может содержать ошибки.

## Используемые библиотеки

pygame==2.6.1

## Реализуемые технологии

В процессе реализации своего проекта я использовал такие возможности библиотеки pygame, как функцию blit, которая позволяет прикрепить изображения и тексты, изменение шрифта с помощью font, функцию fill, закрашивающая весь экран определенным цветом, тем самым позволяя изменить дизайн окна, pygame.image.load(), которые помогает загружать изображения, различные функции, позволяющие создать спрайты и спрайты с анимацией, определять их маски, осуществлять поиск их пересечений, создание и добавление спрайтов в группы, уничтожать какие-то спрайты, работать с их методами.

Для того, чтобы сохранять и работать с информацией о пользователях, мной был создан пустой текстовый файл «имя пользователей.txt», взаимодействие с которым осуществляется с помощью встроенных функций языка Python.

## Реализация проекта

После разработки зависимостей и технического задания я приступил к созданию проекта.

Сначала я нашел в Интернете все графические элементы, которые необходимы для игры, а также самостоятельно создал фон для уровней в программе Paint.

Далее, я перешел в среду разработки и начал реализовывать функцию draw(), которая отвечает за отрисовку 1 и 2 уровней. В дальнейшем, я создал такие классы, которые наследуется от pygame.sprite.Sprite, как Car(), который отвечает за основную машину, OtherCars(), который отвечает за встречную машину, Star(), который отвечает за звездочку и AnimatedBoom(), который отвечает за анимированный взрыв. Затем, я создал экземпляр класса Car(), объект clock, отслеживающий время, счетчик, словарь d, хранящий в себе некоторую информацию для уровней (например, частоту появления встречных машин, звездочек, скорость передвижения фона и т.д.). После этого я стал реализовывать игровой цикл, в котором производится прорисовка окна, движения фона, основной машины (если необходимо), встречных машин, звездочек, взрывов, отрисовка кадра, увеличение счетчика.

Впоследствии я создал такие функции как draw\_3(), отвечающую за отрисовку 3 уровня, start\_screen(), отвечающую за отрисовку стартового окна, screen\_between\_levels(), отвечающую за отрисовку окна между уровнями, victory\_screen(), отвечающую за отрисовку победного окна, loss\_screen(), отвечающую за отрисовку окна проигрыша, переменную ans, хранящую текущий уровень, а также внес некоторые дополнения в словарь, касающиеся 3 уровня, и в игровой цикл, связанные с переключением между уровнями, взаимодействия с новыми окнами.

В конце проектирования я сделал пустой текстовый документ «имя пользователей.txt» и функцию screen\_entry\_name(), отвечающую за отрисовку экрана ввода имени пользователя и внес некоторые изменения, касающиеся изменений выше.

## Скриншоты выполненного проекта

Изображение выглядит как текст, снимок экрана, машина

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Шрифт, диаграмма

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, мультфильм, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, эмотикон, смайлик, мультфильм

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, Мобильный телефон, дизайн

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, эмотикон, смайлик, желтый

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, трофей, кружка

Контент, сгенерированный ИИ, может содержать ошибки.

## Тестирование приложения

Игру тестировал несколько дней. За это время была найдена проблема в движении встречных машинок и обнаружена недоработка, связанная со звездочками. Все недостатки были устранены.

В остальном программа работает корректно.

Тестирование проводилось еще 3 людьми. Приложение работало стабильно.

**Не понравилось:**

* Простые уровни
* Отсутствие возможности поменять пользователя

**Понравилось:**

* Игра работает
* Легкое и интуитивно понятное
* Дизайн
* Интересная идея

# Заключение

В результате выполнения проекта на языке Python были успешно решены поставленные задачи и достигнуты основные цели. Разработанное программное обеспечение обладает высокой функциональностью и производительностью.

В процессе работы были изучены история создания игр, основные сведения о библиотеке pygame, её возможности, классификация компьютерных игр и этапы создания игры.

Графический интерфейс интуитивно понятен и прост в использовании, что делает его доступным для широкой аудитории.

В процессе тестирования появились некоторые идеи улучшения моего проекта. Например, добавить несколько видов основной машины и возможность выбирать среди них понравившуюся модель.

В процессе разработки были применены технологии Python и pygame, что обеспечило гибкость и эффективность создания моего проекта. Использование этих инструментов позволило быстро настраивать внешний вид моей игры, а также обрабатывать данные, полученные от пользователей.

В целом, проект является успешным примером применения современных технологий для создания удобной и функциональной компьютерной игры.