Министерство науки и высшего образования РФ

ФГБОУ ВПО «Пермский государственный национальный исследовательский университет»

Кафедра прикладной математики и информатики

**Индивидуальное задание №2**

***по дисциплине «Алгоритмизация и программирование»***

|  |  |
| --- | --- |
|  | **Выполнил:** |
|  | Студент 1-го курса  Института компьютерных наук и технологий  \_\_\_\_\_\_\_\_\_\_\_ Арланова А.А.,  группа ПМИ-1,2-2022 |
|  | **Проверил:** |
|  | Старший преподаватель кафедры прикладной математики и информатики ПГНИУ,  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Шеина Т. Ю. |
|  | «\_\_\_»\_\_\_\_\_\_\_\_\_ 2023 г. |

Пермь 2023

## **Оглавление**

[1 Постановка задачи 3](#_Toc131437059)

[2 Определение идеи алгоритма, выбор методов решения 4](#_Toc131437060)

[3 Тестирование программы 5](#_Toc131437061)

[4 Текст программы 6](#_Toc131437062)

# Постановка задачи

По заданию необходимо написать программу-игру «Война с инопланетянами», в которой с верхней части экрана будут падать мобы (инопланетяне), а внизу экрана будет находиться игрок, который может сбивать врагов выстрелами (с помощью мыши). Вести подсчет количества сбитых врагов. Игрок должен иметь 3 жизни, каждую из которых он теряет при столкновении с врагом. Добавить в игру звуковое сопровождение (фоновую музыку, звуки выстрелов и звуки взрывов при попадании). Общее количество мобов - не более 30. Падать они должны неравномерно и с разной скоростью.

Так же я хочу реализовать возможность игры вдвоем на одном устройстве, следовательно, и выбор управления, так как два игрока не должны иметь один и тот же тип управления. Добавить, чтобы существовало два типа врагов, за которых добавляло разное количество очков. А чтобы игроки бесконечно не стреляли во врагов, решила, что необходимо ограничить количество пуль, которые они могут использовать. Но чтобы не было так сложно, необходимо добавить различные бонусы, такие как, увеличение количество жизней, увеличение запаса пуль и на время бесконечные пули. Реализовать два игровых режима. Первый – сюжетный, где вначале проигрывается катсцена (текст с картинками), после запускается сама игра, и при наборе определенного очков, чтобы выходил босс инопланетян (враг с очень большим запасом здоровья и разными атаками). После победы над ним проигрывается завершающая катсцена. Второй режим – бесконечный, в нем будут отсутствовать какие-либо катсцены и босс, но зато есть возможность ставить рекорды, которые сохраняются в файле. Так же есть желание создать главное меню с кнопками, которые позволяют не только выбрать какой-либо режим игры, но и выключить/включить музыку, сменить полноэкранный режим на оконный. И последняя моя задача - ставить игру на паузу при нажатии клавиши.

# Определение идеи алгоритма, выбор методов решения

Для создания этой игры я использовала IDE PyCharm Community Edition и модуль Pygame.

В ходе написания данной программы, я пыталась придерживаться методологии ООП, где основные элементы программы должны представлять собой объекты, у которых есть свойства и методы. Рассмотрим классы, которые я сформировала:

1. **Player**

Описание: Player является объектом, с которым себя идентифицирует игрок. Пользователь способен при помощи клавиш или мышки его перемещать и создавать объекты – пули.

Свойства: изображение, размер, скорость, координаты, тип управления (мышка, стрелки ←→ + ↑ или клавиши ad + w), список с объектами Bullet, бесконечные ли пули (логическая переменная), время до окончания бесконечных пуль, прямоугольная область (объект класса Rect, который в дальнейшем позволит отслеживать столкновение других объектов), количество жизней, пуль и очков

Методы:

1. Движение по оси x

Если управление через клавиши, то проверяем, какие клавиши нажаты. Если совпадает тип управления и необходимые клавиши для него, то прибавляем (или отнимаем) к координатам по x скорость игрока.

Если же управление через мышку, то просто приравниваем координаты x значению координат мышки по x

1. Стрельба

Если количество пуль в игре не превосходит максимальное количество пуль и при этом игрок жив (то есть у него жизней больше чем 0), тогда мы проверяем, какая клавиша была отжата, и если она совпала с типом управления, то добавляем в список пуль новый объект Bullet, проигрывая звук выстрела.

1. Отрисовка

Метод, который отображает игрока и его пули на заданной поверхности.

1. **Bullet**

Описание: объект, цель которого перемещаться по оси y, и если встречает какой-то объект, то самоуничтожаться при этом, давая очки игроку и уничтожая врага, (или отнимая жизнь у игрока, если пуля была выпущена боссом)

Свойства: изображение, размер, скорость, координаты, прямоугольная область

Методы:

1. Движение по оси y

Если объект находится в видимом диапазоне, то меняется координата y в зависимости от скорости, в ином случае удаляется из списка

1. Столкновение

Если объект был создан игроком, то проверяется, есть ли столкновение пули с врагом. В зависимости от того в кого она попала, выполняются разные действия. Если в босса, то у него отнимаются жизни. Если в моба, то он удаляется из списка врагов, и даются очки игроку, проигрывая звук смерти врага. Так же с некоторым шансом генерируется бонус для игрока.

Если объект был создан боссом, то проверяется было ли столкновение с игроком. В случае если это произошло, то пуля удаляется из списка и у игрока вычитается жизнь, проигрывая звук ранения.

1. Отрисовка

Отображает объект на заданной поверхности

1. **Enemy**

Описание:

Свойства: изображение, размер, скорость по x, скорость по y, координаты, тип врага, прямоугольная область

Методы:

1. Движение
2. Столкновение
3. Отрисовка
4. **FallingBuff**

Описание:

Свойства: изображение, размер, скорость, координаты, тип бонуса, прямоугольная область

Методы:

1. Движение по y
2. Столкновение
3. Отрисовка
4. **BossEvent**

Описание:

Свойства: изображение, размер, координаты начальные, координаты конечные, здоровье, объект Laser, список с объектами Bullet

Методы:

1. Появление
2. Стрельба
3. Атака лазером
4. Отрисовка
5. **Laser**

Описание:

Свойства: изображение, размер текущий, размер максимальный, координаты, прямоугольная область

Методы:

1. Распространение
2. Столкновение
3. Отрисовка
4. **Button**

Описание:

Свойства: изображение текущее, изображение для не наведённого состояния, изображения для наведенного состояния, размер, координаты, текст, прямоугольная область

Методы:

1. Обновление состояния
2. Нажатие
3. Отрисовка
4. **Star**

Описание:

Свойства: координаты, непрозрачность

Методы:

1. Сияние
2. Отрисовка

# Тестирование программы

# ЛИСТИНг программы