Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Пермский национальный исследовательский**

**политехнический университет»**

**(ПНИПУ)**

Электротехнический факультет

кафедра «Информационные технологии и автоматизированные системы»

направление подготовки: 09.03.04 Разработка программно-информационных систем

**О Т Ч Е Т**

**по учебной практике**

Выполнила студентка

гр.РИС-23-3б

Шуракова А.А.

###### \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись)

###### Проверил:

Доцент кафедры ИТАС Петренко А.А.

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

(оценка) (подпись)

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

(дата)

**Пермь 2024**

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc170210273)

[**1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ** 4](#_Toc170210274)

[**Вывод** 6](#_Toc170210275)

[**2 ТЕХНОЛОГИИ РЕАЛИЗАЦИИ** 7](#_Toc170210276)

[**2.1 Создание и реализация действий игрового поля** 7](#_Toc170210277)

[**2.2 Визуализация игрового поля и использование библиотеки визуализации** 9](#_Toc170210278)

[**2.3 Руководство пользователя** 11](#_Toc170210279)

[**Вывод** 12](#_Toc170210280)

[**ЗАКЛЮЧЕНИЕ** 13](#_Toc170210281)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 14](#_Toc170210282)

[**ПРИЛОЖЕНИЕ А** 15](#_Toc170210283)

[**ПРИЛОЖЕНИЕ Б** 20](#_Toc170210284)

# **ВВЕДЕНИЕ**

В современном мире информационных технологий человек ежедневно сталкивается с работой на компьютере. Постоянное решение сложных задач утомляет людей. Ежедневное решение сложных задач и умственная нагрузка приводят к утомлению и усталости людей. Игры являются популярным средством развлечения, которое привлекают людей разных возрастов и интересов. Одной из самых известных игр является “Лабиринт”.

Игра "Лабиринт" представляет собой двухмерную аркадную игру, разработанную с использованием библиотеки Pygame на языке Python. Игра позволяет игроку управлять персонажем, который должен пройти через лабиринт, избегая столкновения с врагами и достигая финального объекта. В отчете представлены результаты разработки и реализации игры "Лабиринт".

Целью работы является создание увлекательного игрового опыта, который предоставит игрокам возможность проверить реакцию, ловкость и стратегическое мышление.

Отчет включает в себя описание основных компонентов игры, таких как игровое поле, главный герой, враги, стены, а также особенности их взаимодействия. Рассмотрены алгоритмы и логика, лежащие в основе игрового процесса.

**1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

Игра "Лабиринт" принадлежит к жанру аркадных игр, где игрок управляет персонажем для преодоления различных препятствий и достижения поставленных целей. Такие игры характеризуются простыми двухмерными графическими элементами и интуитивно понятным управлением.

В игре "Лабиринт" игрок управляет персонажем, который должен пройти через лабиринт, состоящий из различных стен. В лабиринте присутствуют два вражеских объекта, которые перемещаются по определенным траекториям и представляют опасность для игрока. Игроку необходимо избежать столкновения с ними, чтобы не проиграть. Конечная цель игры - достичь финального игрового объекта, который обозначает победу.

Цель игры: Заключается в том, чтобы игрок, управляя персонажем, смог пройти через лабиринт, состоящий из различных стен, избежав столкновения с врагами и достигнув финального игрового объекта, который обозначает победу в игре.

Игровое поле: Игровое поле в игре "Лабиринт" представляет собой двухмерное игровое пространство, на котором расположены следующие элементы:

1. Фоновое изображение: Для создания атмосферы игры используется фоновое изображение, которое занимает все игровое окно.

2. Стены лабиринта: Лабиринт состоит из различных стен, которые игрок должен обходить. Стены имеют определенные размеры и расположены в различных частях игрового поля.

3. Игровые объекты:

- Персонаж, управляемый игроком: Главный игровой объект, которым управляет игрок, перемещаясь по лабиринту.

- Вражеские объекты: Два вражеских объекта, которые движутся по заданным траекториям и представляют опасность для игрока.

- Финальный игровой объект: Объект, обозначающий конечную цель игры, которую игрок должен достичь.

Размеры игрового поля задаются с помощью переменных win\_width и win\_height, определяющих ширину и высоту игрового окна соответственно.

Ход игры:

1. Игрок управляет персонажем, используя клавиши клавиатуры для перемещения влево, вправо, вверх и вниз.

2. Персонаж должен пройти через лабиринт, состоящий из различных стен.

3. В лабиринте присутствуют два вражеских объекта, которые перемещаются по заданным траекториям.

4. Игроку необходимо избегать столкновения с врагами, чтобы не проиграть.

5. Конечная цель игры - достичь финального игрового объекта, который обозначает победу.

6. Если игрок сталкивается с врагами, игра заканчивается поражением, и отображается соответствующий экран.

7. Если игрок достигает финального объекта, отображается экран победы.

Ход работы:

1. Создать графические элементы для игрового поля, стен, персонажа и вражеских объектов.

2. Реализовать логику управления персонажем с помощью клавиш.

3. Создать вражеские объекты, которые будут перемещаться по заданным траекториям.

4. Обеспечить обнаружение коллизий между персонажем и врагами, а также с финальным объектом.

5. Реализовать отображение конечных экранов для победы и поражения.

**Вывод**

Проведение анализ предметной области игры "Лабиринт" позволил получить четкое представление о ее ключевых особенностях и игровом процессе. Были определены основные элементы игрового поля, включающие фоновое изображение, стены лабиринта, управляемого игроком персонажа, вражеские объекты, а также финальный игровой объект, обозначающий победу. А также были представлены цель игры, ход игры, ход работы.

# **2 ТЕХНОЛОГИИ РЕАЛИЗАЦИИ**

## **2.1 Создание и реализация действий игрового поля**

Для реализации игры "Лабиринт" используется язык программирования Python и библиотека Pygame. Pygame – это библиотека модулей для языка Python, созданная для разработки 2D игр. Также Pygame могут называть фреймворком. [1] Игровые объекты представлены в виде классов, наследующих от родительского класса Parents. Каждый объект имеет свои атрибуты, такие как изображение, координаты, скорость и т.д. Управление персонажем осуществляется с помощью обработки нажатий клавиш клавиатуры. Вражеские объекты перемещаются по заданным траекториям, а обнаружение коллизий реализуется с использованием функций Pygame. Pygame представляет собой оболочку Python для библиотеки SDL (Simple DirectMedia Layer). SDL обеспечивает межплатформенный доступ к базовым мультимедийным аппаратным компонентам вашей системы, таким как звук, видео, мышь, клавиатура и джойстик. [2]

Создание игрового поля: На рисунке №1 изображено, как загружается фоновое изображение "fon.png" и масштабируется до размеров игрового окна (win\_width, win\_height). Это создает фон для игрового поля.

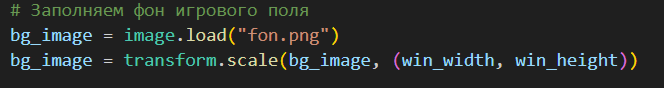


Рис.1 – Заполнение фона игрового поля

Создание стен лабиринта: На рисунке №2 изображено, как создается класс Wall, который представляет собой стены лабиринта. Конструктор класса принимает параметры image, x, y, width, и height, которые определяют изображение, положение и размеры стены. Внутри конструктора масштабируется изображение и создается прямоугольный объект self.rect, соответствующий размерам стены. Метод draw\_wall() отрисовывает стену на игровом поле.

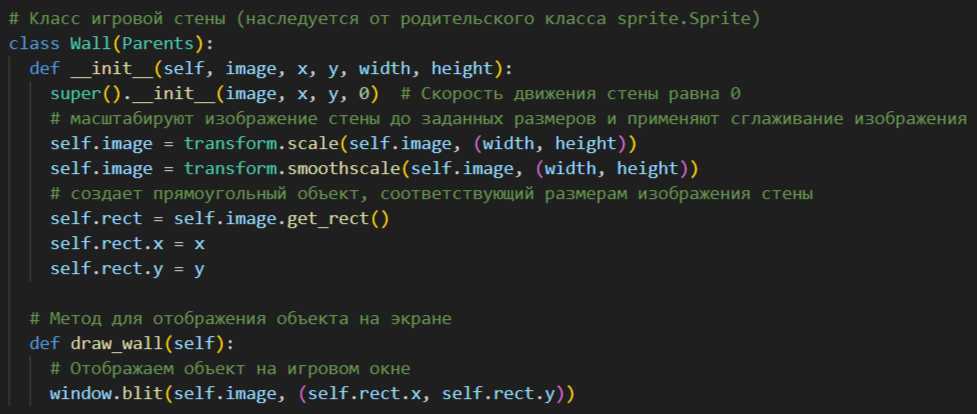


Рис.2 – класс игровой стены

Реализация движения игрока: На рисунке №3 изображено, как обрабатывается нажатие клавиш для перемещения игрока. Если нажата клавиша влево, игрок движется влево, если нажата клавиша вправо, игрок движется вправо, и аналогично для клавиш вверх и вниз. Также проверяется, чтобы игрок не выходил за границы игрового поля.

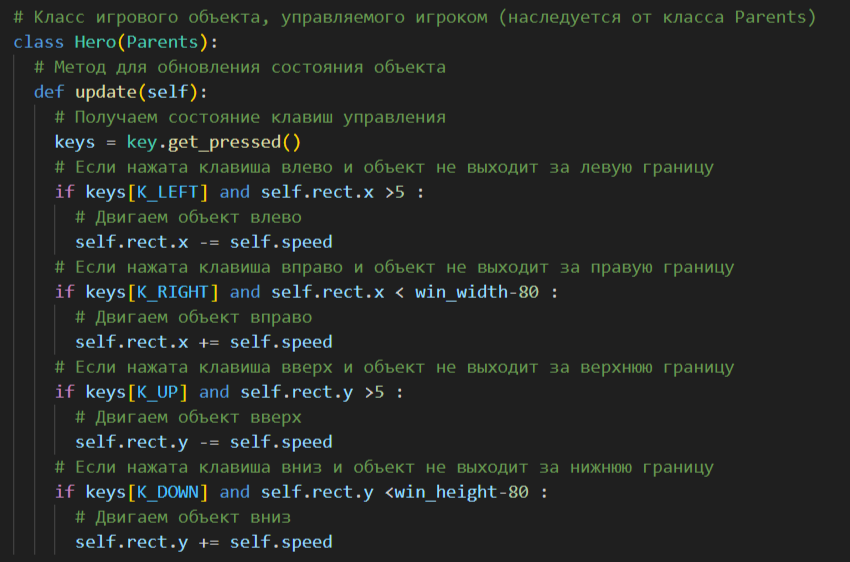


Рис.3 – класс игрового объекта

## **2.2 Визуализация игрового поля и использование библиотеки визуализации**

Для визуализации игрового поля в представленном коде используется библиотека Pygame.

Создание игрового окна: На рисунке №4 изображено, как задаются размеры игрового окна с помощью переменных win\_width и win\_height, устанавливается заголовок окна с помощью display.set\_caption() и создается само окно с помощью display.set\_mode().

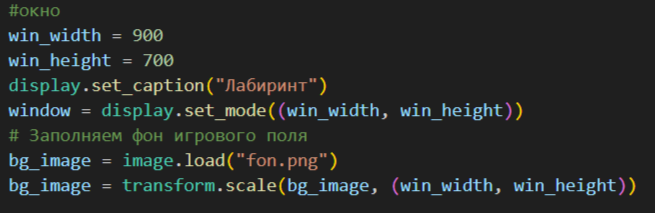


Рис.4 – Создание игрового окна

Отображение игровых объектов: На рисунках 5, 6, 7 изображено отображение игровых объектов, с помощью дополнительных методов.



Рис.5 – Отображение фонового изображения

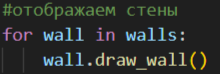


Рис.6 – Отображение стен

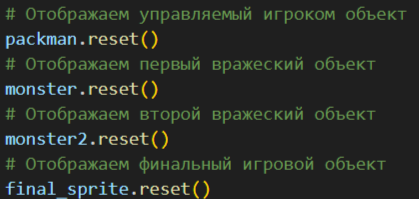


Рис.7 – Отображение игровых объектов

Эти операции выполняются в основном игровом цикле, позволяя динамически обновлять игровое поле и перемещать игрока. Библиотека Pygame предоставляет удобные функции и методы для работы с графикой, обработки ввода и управления игровым процессом.

## **2.3 Руководство пользователя**

В этой игре задача пользователя - управлять главным героем (зайцем). Цель пользователя - пройти лабиринт от начала до конца, избегая столкновений с вражескими объектами и достигая финального объекта.

**Управление:**

* Движение игрока осуществляется с помощью клавиш со стрелками:
* Стрелка влево: Движение игрока влево.
* Стрелка вправо: Движение игрока вправо.
* Стрелка вверх: Движение игрока вверх.
* Стрелка вниз: Движение игрока вниз.

**Правила игры:**

* Игрок начинает движение с определенных координат и перемещается по полю.
* Цель игры - пройти весь лабиринт от начала до конца.
* Игрок должен избегать столкновений с врагами.
* Игра заканчивается, когда игрок достигает финального объекта.

**Важно для успешной игры:**

* Не сталкиваться с вражескими объектами.
* Игрок должен сохранять ориентацию в лабиринте, чтобы найти кратчайший путь к выходу.

Образец игры представлен на рисунке 8.



Рис. 8 – Основное окно игры

## **Вывод**

Анализ технологий, используемых для реализации игры "Лабиринт", показал, что ключевым инструментом для ее разработки является библиотека Pygame, которая предоставляет богатый функционал для создания двухмерных игр на языке Python. Важным аспектом реализации является обработка ввода от пользователя, осуществляемая через отслеживание нажатий клавиш клавиатуры. Это позволяет игроку управлять перемещением персонажа по игровому полю. Для обеспечения динамичности игрового процесса применяются алгоритмы управления движением вражеских объектов по заданным траекториям, а также механизмы обнаружения коллизий между игровыми объектами. Эти функциональные возможности предоставляются библиотекой Pygame.

# **ЗАКЛЮЧЕНИЕ**

В ходе разработки игры "Лабиринт" были исследованы различные аспекты предметной области, такие как описание игры и постановка задачи.

При разработке игрового процесса была создана и реализована логика построения игрового поля, движения и расположения объектов на поле, а также их взаимодействие.

Для визуализации игрового поля была использована библиотека paygame. Она была выбрана из-за удобства использования спрайтов, так как Pygame даёт все необходимые инструменты для создания небольшой 2D игры. [3]

Были разработаны методы, отвечающие за визуализацию игрового поля, а также за обработку событий клавиатуры для управления игрой.

В целом, разработка игры "Лабиринт" позволила углубиться в предметную область и изучить основные аспекты этой игры. Реализация игрового поля и его визуализация с использованием библиотеки paygame позволили создать интерактивную и увлекательную игру, которая может быть использована для развлечения и развития реакции. Дальнейшие улучшения и расширения проекта могут включать добавление звуковых эффектов, уровней сложности, улучшенной графики и других игровых механик. Этот проект может служить хорошим отправным пунктом для дальнейшего исследования разработки игр и расширения своих навыков программирования с использованием paygame.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Pygame и разработка игр. [Электронный ресурс] URL: https://younglinux.info/pygame/pygame
2. «PyGame: учебник по программированию игр на Python» [Электронный ресурс] URL: <https://uproger.com/pygame-uchebnik-po-programmirovaniyu-igr-na-python/>
3. Язык программирования Python: что такое и где используется – Логотип и история [Электронный ресурс] URL: https://all-python.ru/osnovy/yazyk-programmirovaniya.html

# **ПРИЛОЖЕНИЕ А**

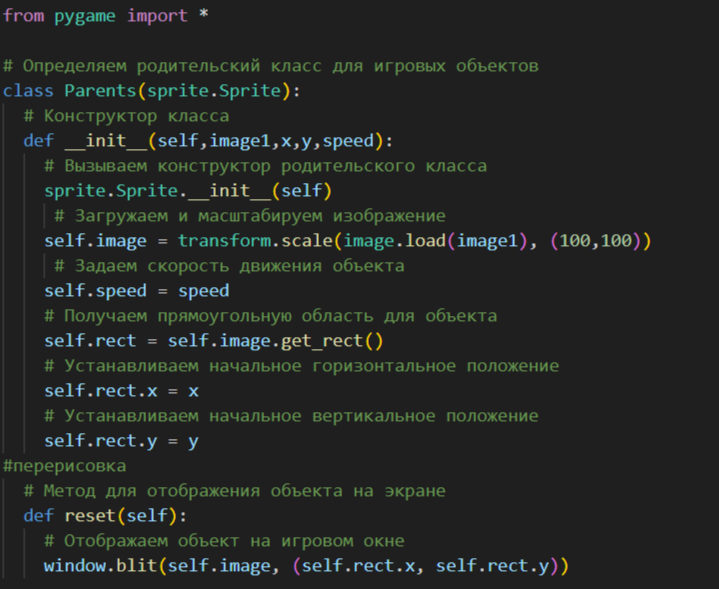
****

Рисунок А.1 – Пример фрагмента кода

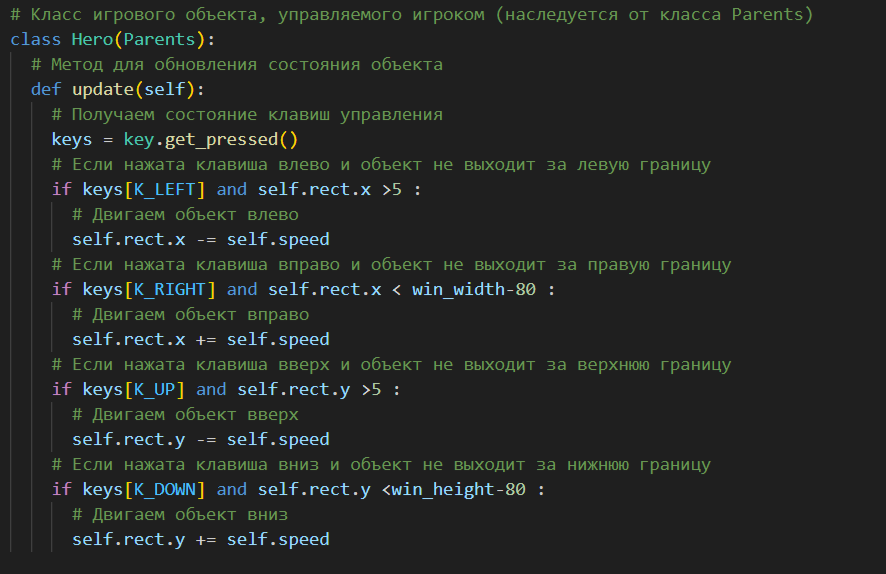


Рисунок А.2 – Пример фрагмента кода

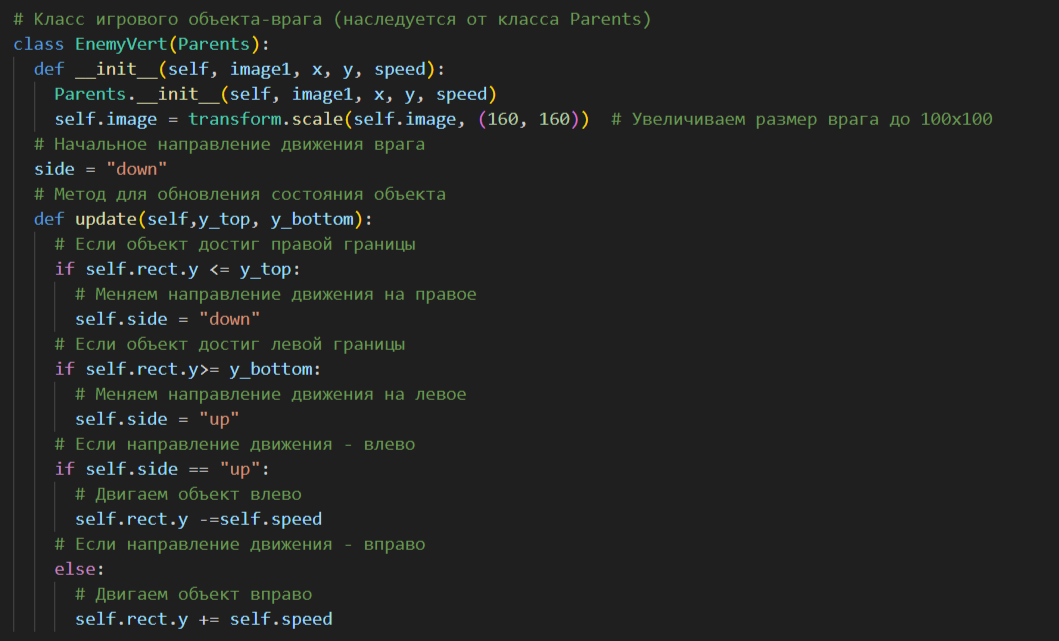


Рисунок А.3 – Пример фрагмента кода

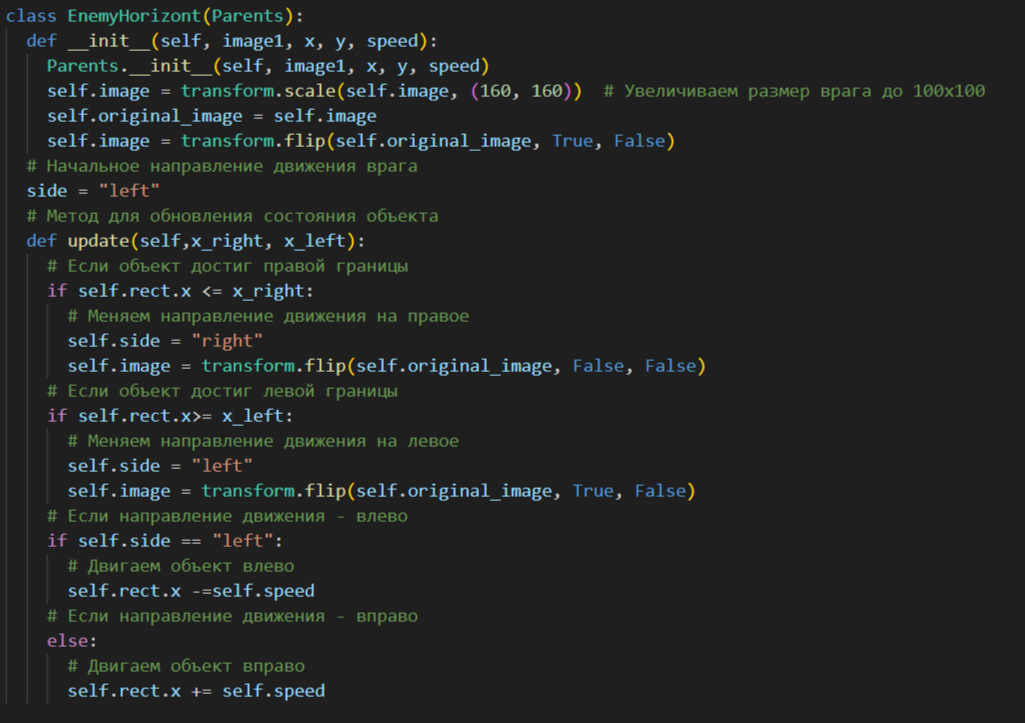


Рисунок А.4 – Пример фрагмента кода

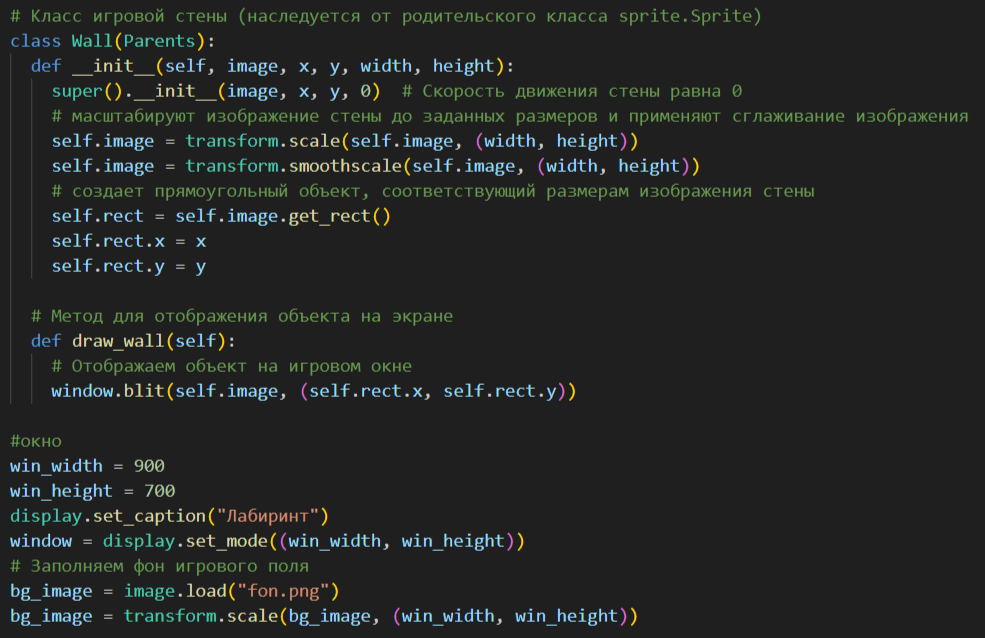


Рисунок А.5 – Пример фрагмента кода

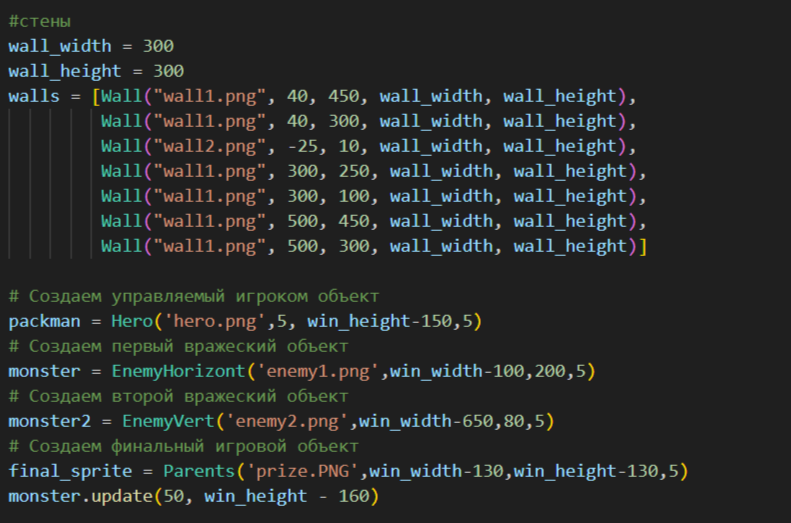


Рисунок А.6 – Пример фрагмента кода

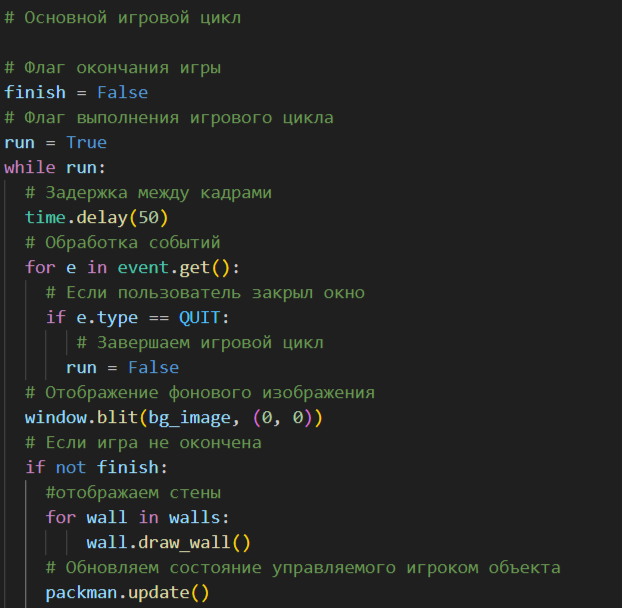


Рисунок А.7 – Пример фрагмента кода

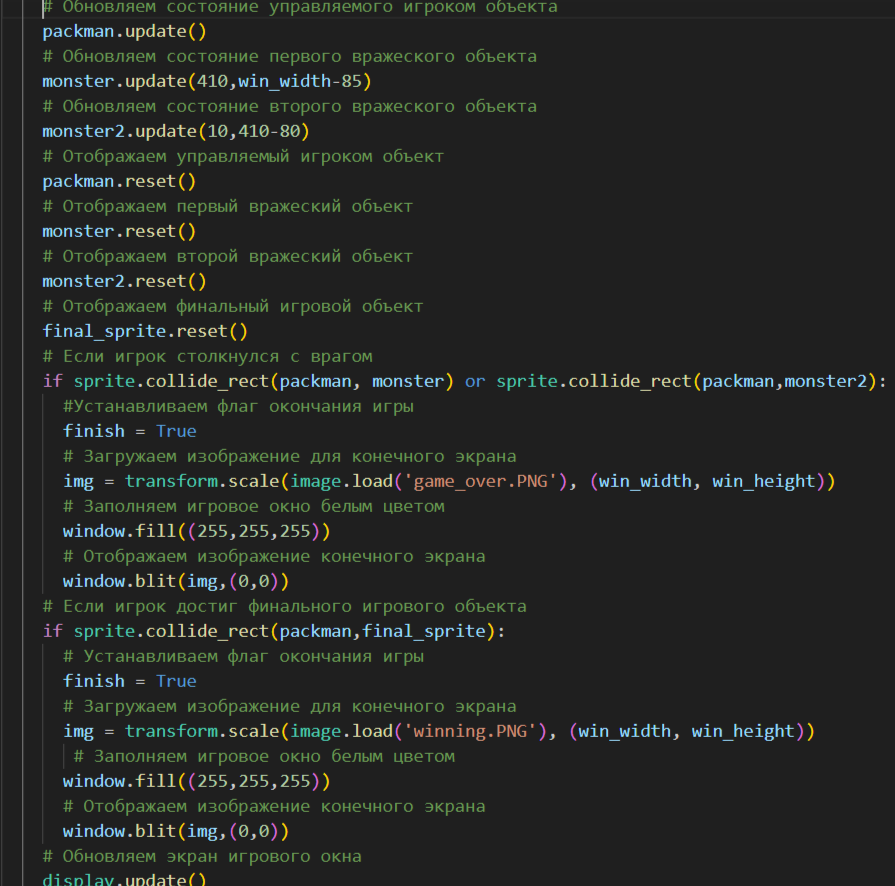


Рисунок А.8 – Пример фрагмента кода

# **ПРИЛОЖЕНИЕ Б**

****

Рисунок Б.1 – Управляемый игроком объект

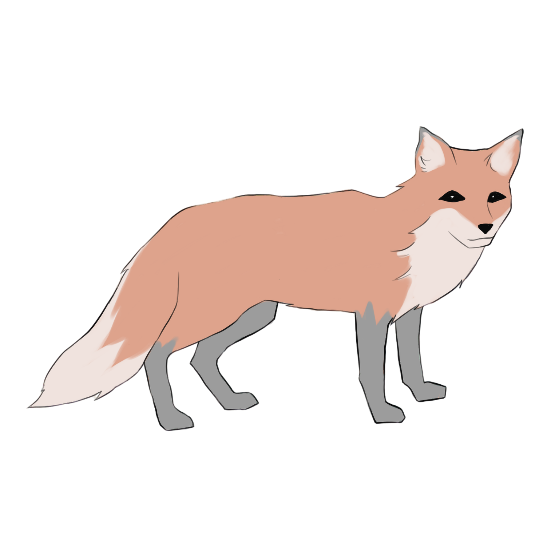


Рисунок Б.2 – Первый вражеский объект



Рисунок Б.3 – Второй вражеский объект

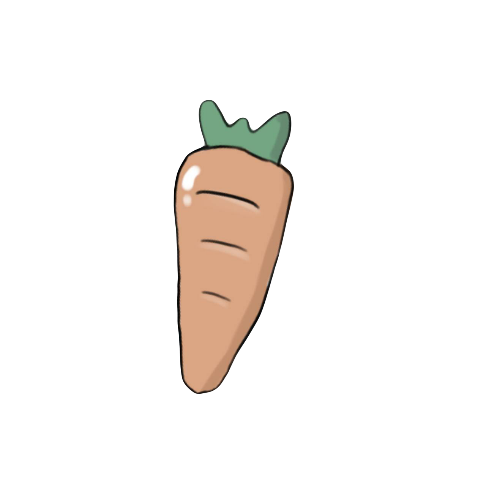


Рисунок Б.4 – Финальный игровой объект

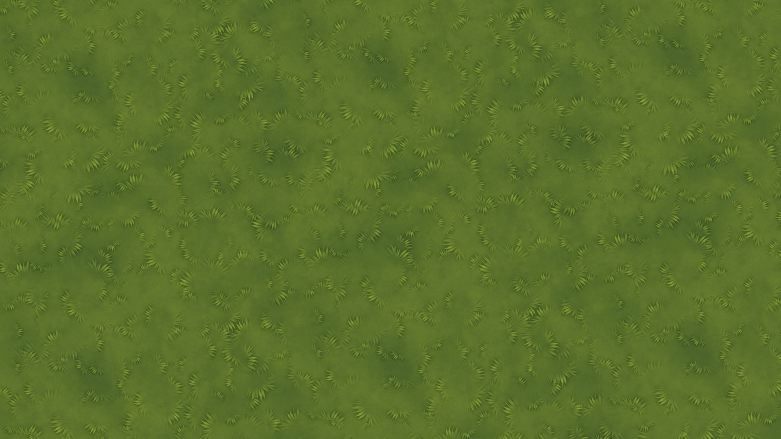


Рисунок Б.5 – Фон

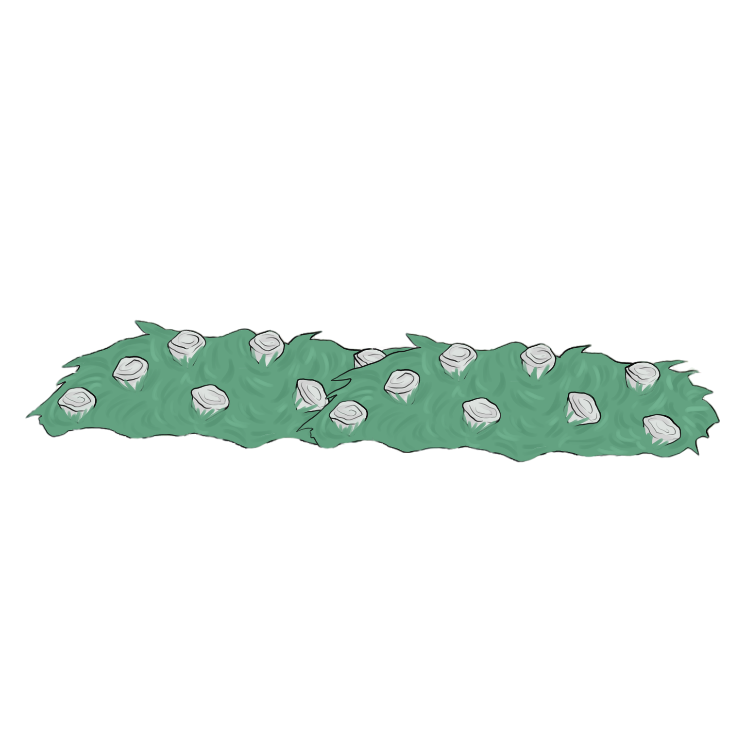


Рисунок Б.6 – Стена

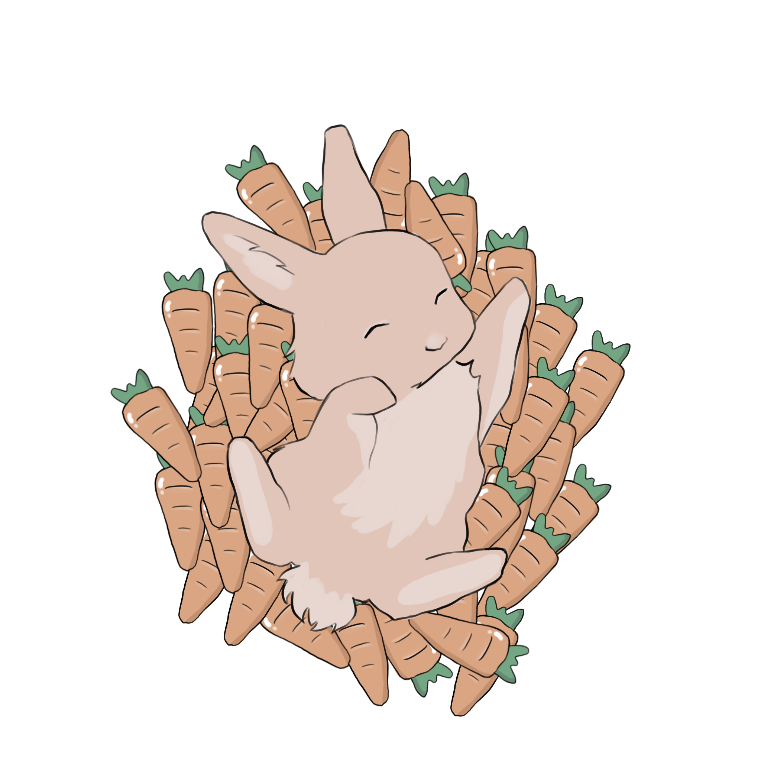


Рисунок Б.7 – Изображение для конечного экрана (выигрыш)



Рисунок Б.8 – Изображение для конечного экрана (проигрыш)