# **Пояснительная записка к проекту "Яблочная ферма"**

## **1. Введение**

Компьютерные игры стали неотъемлемой частью нашей культуры, предлагая не только развлечение, но и ценные образовательные возможности. Они помогают развивать внимание, скорость реакции, логическое мышление и даже социальные навыки.

Проект "Яблочная ферма" представляет собой динамичную 2D-игру, в которой игрок управляет фермером, стремясь поймать падающие яблоки. Игра разработана с использованием библиотеки Pygame, что позволяет легко создавать графику, анимацию и обрабатывать пользовательский ввод. Для хранения данных об игровых объектах применяется база данных SQLite, что обеспечивает эффективный контроль над информацией о различных типах яблок, их характеристиках и поведении в игре.

Цель данного проекта заключается в изучении основ разработки игр, включая работу с графикой, обработку событий, анимацию и звуковое сопровождение. Этот проект также демонстрирует, как можно интегрировать базы данных в игровую механику.

## **2. Цель и задачи проекта**

**Цель проекта:** Создание простой, но увлекательной аркадной игры, которая демонстрирует принципы работы с библиотекой Pygame, анимацией, базами данных и обработкой событий на языке Python.

Задачи проекта:

1. Разработать графический интерфейс игры с удобным управлением.
2. Реализовать механику падения яблок с различными характеристиками (скорость, очки).
3. Внедрить систему подсчета очков и ограничения на пропущенные яблоки.
4. Обеспечить управление персонажем с помощью клавиш влево (←) и вправо (→).
5. Добавить возможность выбора уровня сложности (Easy, Hard).
6. Реализовать хранение данных об игровых объектах в базе данных SQLite.
7. Интегрировать музыкальное сопровождение и звуковые эффекты для создания атмосферы.

## **3. Функциональные возможности игры**

### **Управление персонажем**

Игрок управляет фермером, перемещая его по экрану с помощью клавиш ← и →. Персонаж анимирован, что делает его движение более реалистичным и приятным для глаз.

### **Различные типы яблок**

В игре представлено несколько видов яблок, отличающихся:

* Скоростью падения: медленные, средние и быстрые яблоки.
* Количеством начисляемых очков: редкие яблоки приносят больше баллов.
* Звуковыми эффектами: каждое яблоко имеет уникальный звук при сборе.

### **Система уровней сложности**

***Easy режим:*** Яблоки падают медленно.  
***Hard режим:*** Яблоки падают быстрее, что делает игру более сложной.

### **Подсчет очков**

За каждое пойманное яблоко игрок получает баллы.  
Если игрок пропускает 5 яблок, игра заканчивается.  
После завершения игры отображается итоговый счет и предложение сыграть снова.

### **Аудиосопровождение**

***Фоновая музыка:*** создает атмосферу деревенской фермы.  
***Звуковые эффекты:*** звуки падения и сбора яблок добавляют реализма.

### **Дружественный интерфейс**

***Главное меню включает:***

* Выбор уровня сложности (Easy, Hard).
* Кнопку информации с правилами игры.
* Возможность выхода из игры.

## **4. Техническая реализация**

### **Язык программирования**

Проект написан на языке Python, который известен своей простотой и мощностью в области разработки игр и приложений.

### **Используемые библиотеки**

***Pygame:***

* Создание и обработка графики, анимации, пользовательского ввода.
* Управление игровым циклом и событиями.

***SQLite3:***

* Хранение данных о яблоках, их характеристиках и поведении.
* Оптимизация загрузки данных через запросы к базе.

***Random:***

* Генерация случайных позиций и характеристик яблок.

***OS:***

* Управление ресурсами (изображения, звуки).

### **Графика**

Персонаж (фермер):

* Анимированный спрайт, загружаемый из файла farmer2.png.
* Движение реализовано через последовательную смену кадров.

Яблоки:

* Несколько видов яблок, загружаемых случайным образом.
* Размер каждого яблока масштабируется до 60x60 пикселей.

Фон:

* Статическое изображение фермы (farm\_bg.png).

### **Звуковое сопровождение**

Фоновая музыка: файл music\_1.ogg, воспроизводимый в бесконечном цикле.  
Эффекты падения и сбора яблок: уникальные звуки для каждого типа яблока.

### **База данных SQLite**

База данных содержит таблицу apples, где хранятся следующие данные:

* apple\_id: уникальный идентификатор яблока.
* image\_path: путь к изображению яблока.
* points: количество очков за сбор яблока.
* fall\_speed: скорость падения (медленная, средняя, быстрая).
* spawn\_rate: вероятность появления яблока.
* music\_path: путь к звуковому эффекту.

Пример структуры таблицы:  
*CREATE TABLE apples (*

*apple\_id INTEGER PRIMARY KEY,*

*image\_path TEXT,*

*points INTEGER,*

*fall\_speed TEXT,*

*spawn\_rate REAL,*

*music\_path TEXT*

*);*

### **Логика работы игры**

При запуске игры отображается главное меню.  
Игрок выбирает уровень сложности или просматривает правила.  
Начинается игровой процесс:

* Яблоки появляются случайным образом и падают сверху вниз.
* Игрок управляет фермером, стараясь поймать яблоки.
* За каждое пойманное яблоко начисляются очки.
* Если яблоко падает мимо, увеличивается счетчик пропущенных яблок.
* Игра заканчивается, когда счетчик пропущенных яблок достигает лимита (5).
* После завершения игры отображается итоговый счет и предложение сыграть снова.

## **5. Возможные улучшения проекта**

### **Дополнительные уровни и режимы**

**Испытания с ограниченным временем:** игрок должен собрать как можно больше яблок за определенное время.

### **Система достижений и рекордов**

Таблица лидеров: сохранение лучших результатов игроков.  
Открытие новых скинов: игроки могут разблокировать новые внешние виды персонажа.