# Структура проекта

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

Изображение выглядит как текст, снимок экрана, программное обеспечение

Контент, сгенерированный ИИ, может содержать ошибки.

# Описание ключевых классов

1. Класс Player (KKKickers.GameObjects)

Я написал класс для персонажа в 2D-игре: прыжки, скольжение, движение, смерть, отрисовка — всё моё.

* Описание:
  + Поля: Position, MoveSpeed, Gravity, JumpForce, спрайты (\_slideSprite, \_jumpSprite, \_fallSprite, \_deadSprite).
  + Флаги: IsDead, IsSliding, IsFacingRight, JumpsRemaining (до 2 прыжков).
  + Методы:
    - Jump(): Прыжок, второй — с разворотом.
    - Slide(): Скольжение, возвращает прыжки.
    - Die(): Смерть.
    - ApplyGravity(): Падение/подъём, меняет спрайты.
    - Draw():Отрисовка персонажа с поворотом и направлением.
    - Reset(): Сброс на старт.
    - HandleMovement(): Бег вправо/влево.
* Закодил всю логику, физику, анимации и нарисовал спрайты, протестировал геймплей.

1. Класс Wall (KKKickers.GameObjects.Walls)

Я разработал абстрактный класс Wall для представления стен в 2D-игре KKKickers, обеспечивающий базовую функциональность для препятствий.

* Назначение: определяет общую структуру для всех типов стен, включая их позицию, размеры, отображение и взаимодействие с игроком.
* Основные компоненты:
  + Свойства:
    - Bounds (RectangleF): определяет позицию и размер стены.
    - Sprite (Image, абстрактное): Спрайт стены, реализуемый в производных классах.
    - IsActive (bool): указывает, активна ли стена (по умолчанию true).
  + Конструктор:
    - Принимает position (PointF) и size (Size) для инициализации Bounds.
  + Методы:
    - Draw(Graphics g): отрисовывает спрайт стены, если она активна и спрайт не null.
    - HandleCollision(Player player): обрабатывает столкновение с игроком, используя вспомогательный класс WallCollisionHelper. Виртуальный, может переопределяться.
    - Update(): Виртуальный метод для обновления состояния стены (по умолчанию пустой).
* Разработал архитектуру класса, включая абстрактную основу для различных типов стен. Реализовал логику отрисовки и обработки столкновений. Создал базовую структуру, позволяющую легко расширять функциональность через наследование.

1. Класс GameManager (KKKickers)

Я разработал класс GameManager для управления основной логикой игрока, стен, фона и игрового процесса.

* Назначение: управляет состоянием игры, включая игрока, препятствия, фон, очки и подсказки.
* Основные компоненты:
  + Свойства:
    - Player: Экземпляр класса Player для управления персонажем.
    - CurrentWalls, NextWalls: Списки текущих и следующих стен (препятствий).
    - BG: Фоновое изображение.
    - bgBounds: Массив RectangleF для позиций фона.
    - Score, HighScore: Текущий и рекордный счёт.
    - Tips: Массив подсказок, отображаемых при достижении WinWall.
    - OutOfBounds: Флаг выхода игрока за границы формы.
    - FormRectangle: Границы игрового окна.
  + Конструктор:
    - Инициализирует Player, списки стен, фон и его позиции (bgBounds).
  + Ключевые методы:
    - Reset(): сбрасывает игру: очищает стены, сбрасывает игрока, счёт и фон.
    - InitializeBackground(): задаёт начальные позиции фона.
    - InitializeFirstStructure(), CreateStructure0/1/2/3(): создают начальную и случайные структуры стен (обычные, шипы, пилы, разрушаемые, отбойные, победные).
    - GenerateNextStructure(): генерирует следующую структуру стен случайным образом.
    - Update(): обновляет состояние игры: гравитацию игрока, движение фона, столкновения, обновление пил.
    - CheckCollisions(): обрабатывает столкновения игрока со стенами, обновляет счёт при достижении WinWall.
    - MoveBackground(): двигает фон и стены при движении игрока вверх.
    - ToggleSpikes(), AddSpikes(), RemoveSpikes(): управляют активацией/деактивацией шипов на стенах.
* Разработал всю логику управления игрой: координацию игрока, препятствий, фона и счёта. Реализовал генерацию структур стен и динамическое перемещение фона. Создал систему подсказок и обработку столкновений. Настроил механику шипов и обновление игровых объектов.

Класс GameManager — ядро игры, обеспечивающее плавный геймплей и взаимодействие всех компонентов.

# Логика работы

Главное меню – нажатие «Играть».

GameManager создает начальную и следущую за ней случайную структуры.

Игрок прыгает, избегая препятствий.

При выходе наивысшей точки текущей структуры за границы поля зрения, она текущая становится следующей, а следующая случайно выбирается (без повторов предыдущей).

При падении игрока за границы экрана – экран с кнопкой «Еще раз», счетом и советом.