

На создание игры super mario меня натолкнуло несколько вещей:

Относительная простота выполнения

Личный интерес к данной игре

Практика в создании собственной игры

Основные классы в коде:

Класс Animation отвечает за анимацию объектов. В конструкторе класса указываются изображения, используемые для анимации, а также другие параметры, такие как текущий индекс изображения и задержка между сменой изображений. Методы класса позволяют обновлять анимацию и изменять текущее изображение.

Класс Camera используется для управления камерой в приложении. В конструкторе класса задается начальная позиция камеры и объект, за которым следует камера. Метод move() перемещает камеру в зависимости от положения объекта.

Класс Collider предназначен для обработки столкновений объектов в приложении. В конструкторе класса указываются объект и уровень, на котором происходят столкновения. Методы класса позволяют проверять столкновения по осям X и Y.

Класс Dashboard отвечает за отображение данных на экране, таких как счет, количество монет и время. Конструктор класса принимает путь к файлу со шрифтом, его размер и экран. Метод update() обновляет отображаемые данные.

Класс CreatureCollider используется для обработки столкновений между объектами. В конструкторе класса указывается объект, для которого выполняется проверка столкновений. Метод check() проверяет столкновение с целевым объектом.

Класс CollisionState представляет состояние столкновения и содержит информацию о том, произошло ли столкновение и было ли оно сверху.

Класс "Font" наследуется от класса "Spriteheet" и отвечает за загрузку шрифтов. Он инициализирует массив символов, определяет функцию "loadFont", которая загружает изображения символов шрифта, используя метод "image\_at" из класса "Spriteheet". Загруженные шрифты хранятся в словаре "font".

Класс "Blur" отвечает за применение гауссового размытия к поверхности. Он инициализирует размер ядра размытия и имеет метод "filter", который принимает поверхность, позицию, ширину и высоту и возвращает новую размытую поверхность.

Класс "Input" отвечает за обработку ввода с клавиатуры и мыши. Он инициализирует координаты мыши и объект сущности, проверяет ввод с клавиатуры (нажатие клавиш WASD, пробела и клавиш SHIFT), обрабатывает ввод с мыши (нажатие левой и правой кнопки), а также обрабатывает события завершения приложения и перезапуска.

Класс "Level" отвечает за загрузку уровня и его отрисовку. Он инициализирует объекты спрайтов, звук, экран и уровень, а также список сущностей. Он имеет методы для загрузки уровня из файла JSON, загрузки сущностей, слоев и объектов из данных уровня, а также обновления сущностей и отрисовки уровня.

Класс Vec2D представляет двумерный вектор с координатами x и y. Используются следующие библиотеки: json, sys, os, pygame.

Класс Menu отвечает за отображение и взаимодействие с главным меню игры.

Класс Pause отображает паузу в игре.

Класс Sound отвечает за управление звуковыми эффектами и музыкой в приложении. Он инициализирует звуковые каналы и загружает звуковые файлы.

Класс Sprite представляет спрайт и его свойства, такие как изображение, коллизии, анимация и необходимость перерисовки фона.

Класс Sprites отвечает за загрузку спрайтов из файлов JSON. Он использует классы Spritesheet и Sprite для создания объектов спрайтов из изображений.

Класс Spriteheet представляет лист спрайтов, загружает изображение и предоставляет методы для извлечения отдельных спрайтов из листа.

Класс Block представляет тайл и его свойства, такие как спрайт и прямоугольник.

Класс Coin описывает объект монеты (Coin) в игре. У него есть атрибуты screen, spriteCollection, animation, type. Метод update обновляет анимацию объекта и отображает его на экране.

Класс CoinBox описывает коробку с монетой (CoinBox) в игре. У него есть атрибуты screen, spriteCollection, animation, type и др. Метод update обновляет анимацию коробки, а также отображает на экране пустую коробку и спавнит монеты из коробки.

Класс CoinBrick описывает кирпичную коробку с монетой (CoinBrick) в игре. У него есть атрибуты screen, spriteCollection, image, type и др. Метод update обновляет изображение кирпичной коробки, а также отображает на экране пустую коробку и спавнит монеты из коробки.

Класс CreatureBase является базовым классом для всех игровых объектов и содержит общую логику для всех объектов. Он содержит атрибуты vel, gest, gravity и другие, а также методы для применения гравитации, обновления особых черт объекта и т.д.

Класс Goomba описывает объект гумбы (Goomba) в игре. У него есть атрибуты screen, spriteColl, animation, type и др. Метод update обновляет позицию и анимацию гумбы, проверяет столкновения с другими объектами и выполняет соответствующие действия при столкновениях.

Класе "Item" реализует создание и отображение монетки, которая движется вверх и вниз и исчезает после определенного времени.

Класс "Коора" реализует движение и анимация творения "Коора", а также обработка его столкновения с другими объектами.

Класс "Mario" реализует управление персонажем "Mario", его движение, столкновение с другими объектами и взаимодействие с ними.

Класс "RedMushroom" реализует движение и отображение объекта "RedMushroom", а также обработка его столкновений.

Класс "RandomBox" отображает объект "RandomBox", его анимацию и обработку запуска рандомного предмета при взаимодействии.

Класс movements реализует функциональность отскока объекта и обновление его состояния, реализует движение объекта влево и вправо с анимацией. Включает различные параметры для скорости и ускорения, реализует функциональность прыжка объекта с учетом гравитации и высоты прыжка, реализует движение объекта влево и вправо по заданному уровню. Использует класс Collider для обнаружения столкновений.

В целом, приложению необходимы доработки, потому что оно еще очень далеко находится от оригинала. Но очень важно отметить, если рассматривать, что я еще в этом деле новичок, то для моего уровня, я считаю, это довольно хороший результат