Текстовод Уникальность https://textovod.com/unique

Отчет проверки уникальности текста

Дата проверки: 2023-05-08 19:57:43

Уникальность 72%

Хорошо. Подойдет для большинства текстов.

Текст

Введение

Компьютерные игры появились более полувека назад. С тех пор игры сильно изменились. Вместе с играми развивалось и аппаратное обеспечение, от которого зависели игры. Игры, которые создаются в наше время, не могли быть запущены на компьютерах того времени, по крайней мере, из-за недостатка мощности.

Эволюционировало не только аппаратное обеспечение, но и инструменты разработки для этих самых игр. Есть даже инструменты, ориентированные на людей, не знакомых с программированием, в прежние времена это было невозможно. Создание новых и усовершенствование существующих инструментов происходит за счет сокращения времени и усилий, затрачиваемых на разработку игрового приложения.

На данный момент существует множество платформ, для которых разрабатываются игровые приложения, включая персональные компьютеры, смартфоны, планшеты, игровые приставки и другие. Платформы развиваются, как и установленные на них операционные системы. Если имеет смысл разрабатывать игру для нескольких платформ или операционных систем, то кроссплатформенная разработка имеет множество преимуществ, которые позволяют сократить время и усилия, затрачиваемые на разработку сразу для нескольких платформ и операционных систем.

Pygame - это бесплатная кроссплатформенная библиотека с открытым исходным кодом для разработки мультимедийных приложений, таких как видеоигры, с

использованием Python. Он использует простую библиотеку DirectMedia Layer и несколько других популярных библиотек для абстрагирования наиболее распространенных функций, что делает написание этих программ более интуитивно понятной задачей.

PyQT – библиотека, в которой разработан набор кроссплатформенных библиотек C++, который позволяет реализовать высокоуровневые API. Эти библиотеки предоставляют доступ к различным сервисам, таким как службы определения местоположения и позиционирования, мультимедиа, NFC и Bluetooth. Более того, в рамках этой библиотеки был реализован веб-браузер на основе Chromium. Кроме того, данная библиотека предоставляет возможность для традиционной разработки пользовательского интерфейса. Таким образом, использование предложенного набора библиотек C++ может значительно ускорить процесс создания приложений на различных платформах и устройствах.

1. Обзор жанра RogueLike

Жанром игрового приложения был выбран RogueLike

RogueLike – один из жанров компьютерных игр. Характерными чертами такого жанра являются генерируемые случайным образом уровни, пошаговость и необратимость смерти персонажа – в таком случае игрок должен заново проходить все уровни, нельзя загрузить игру в определенном месте.

Обычно игры такого жанра представлены в пиксельной стилистике (рис. 1, рис. 2)

Рисунок 1 – игра The Binding of Isaac

Рисунок 2 – игра Space Robinson: Hardcore Roguelike Action

Игровой персонаж — объект, перемещающийся в комнате под воздействием игрока и атакующий злодеев.

2. Технические предложения

Для разрабатываемого игрового приложения можно выделить следующие требования:

- игра в жанре RogueLike;
- поддержка операционной системы Windows;
- вид сверху;
- случайная генерация комнаты;
- плавное анимированное перемещение игрового персонажа по комнате;
- генераций разных количеств мобов;
- присутствие поощрительных комнат, где игрок может улучшить свои навыки.

3. Разработка сцены игрового приложения

В сцене игрового процесса есть два корневых элемента – комната с мобами, которая содержит коробки, которые мешают, как игроку, так и мобам, и мобов, которые являются главными противниками нашего героя

Игрок может перемещаться одним способом: для перемещения в прямоугольном лабиринте используются следующие кнопки:

- «D» для перемещения вправо;
- «W» для перемещения вверх;
- «A» для перемещения влево;
- «S» для перемещения вниз.

Камера статично стоит на месте.

4. Разработка сцены игрового приложения

Приложение было разработано в соответствии с методологией объектно-ориентированного программирования, с учетом таких принципов, как:

- абстракция — использование тех характеристик объекта, которые достаточны для его представления в системе;

- инкапсуляция — ограничение доступа одних компонентов системы к другим, а также подключение данных методами, предназначенными для их обработки;

- наследование — концепция повторного использования свойств одних объектов в других;

- полиморфизм — способность объектов и методов обрабатывать различные типы данных.

В языке программирования Python объекты понимаются как классы и интерфейсы. В разрабатываемом приложении есть много классов. В дальнейшем они будут рассмотрены.

Исходный код приложения приведен в приложении 1.Класс направление (Player) – это класс, представляющий направление в двухмерном пространстве персонажа, а также атаку. Он имеет следующие поля:

- __init__ – главный метод, который описывает, движение, атаку, начальное положение самого персонажа, анимации (рис.4, рис.5, рис.6, рис.7).

- player_pos – метод положения самого персонажа, который изменяет положение и сохраняет его.

- possibility_of_movement – ограничение движения игрока, когда он пытается пройти сквозь ящики, стены.

- Update_moving– метод, который обновляет координаты персонажа в зависимости от направления.

- update – метод, который обновляет координаты пули и описывает движение, когда стреляет персонаж.

Класс направление (Enemy) – это класс, представляющий направление в двухмерном пространстве противника, а также атаку. Он имеет следующие поля:

- __init__ – главный метод, который описывает, движение, атаку, начальное положение самого противника, анимации (рис.3).

- Moving_away – метод, который описывает поведение противника и его движениею

- Shooting – метод, который описывает атаку противника.

- update – метод, который обновляет координаты пули и описывает движение, когда стреляет противника.

Класс направление (Player_bullet) – это класс, представляющий направление в двухмерном пространстве пули персонажа. Он имеет следующие поля:

- __init__ – главный метод, который описывает поведение пули, анимацию.

- update – метод, который обновляет положение пули.

- draw_bullet – метод, который связывает анимацию и поведение пули.

Класс направление (Enemy_bullet) – это класс, представляющий направление в двухмерном пространстве пули противника. Он имеет следующие поля:

- __init__ – главный метод, который описывает поведение пули, анимацию.

- update – метод, который обновляет положение пули.

- draw_bullet – метод, который связывает анимацию и поведение пули.

Класс (Door) – это класс, представляющий поведение двери

- __init__ – главный метод, который описывает поведение двери, анимацию.

- Output – метод, который связывает анимацию и поведение двери.

- Update - метод, который описывает дверь (меняется картинка, в зависимости от того, есть ли противники в комнате или нет).

Класс (Box) – это класс, представляющий поведение коробки

- __init__ – главный метод, который описывает поведение коробки, анимацию.

Класс (Interface) – это класс, представляющий весь интерфейс игры

 __init__ – главный метод, который показывает количество жизней, патронов, а также меняет локацию в зависимости от того, есть ли противники на карте или нет.

 Output – метод, который связывает анимацию и поведение интерфейса.

Класс (HP) – это класс, представляющий бонус HP игрока

 __init__ – главный метод, который описывает поведение HP, анимацию.

 Update - метод, который обновляет количество жизней.

Класс (FastMovePlayer) – это класс, представляющий бонус к движению игрока

 __init__ – главный метод, который описывает движение игрока, анимацию

 Update - метод, который обновляет движение игрока.

Класс (FastMoveBulletsPlayer) – это класс, представляющий бонус к движению пули

 __init__ – главный метод, который описывает движение пули, анимацию

 Update - метод, который обновляет движение пули.

Класс (MoreStrongerBulletsPlayer) – это класс, представляющий бонус к силе пули

 __init__ – главный метод, который описывает силу пули, анимацию

 Update - метод, который обновляет силу пули.

Класс (Map) – это класс, который описывает саму карту, стенки и генерацию

 __init__ – главный метод, который описывает саму карту

 New_level – метод, который описывает загрузку нового уровня.

 Generate_room – метод, который генерирует новую команту.

 Next_room – метод, который проверяет на возможность перемещения в следующую локацию

 Pos_player_get – метод, который начальное положение игрока.

 Fon_get – метод, который передает фон для отрисовки на экране

 Door_state_get – метод, который проверяет наличие врагов для того, чтобы открыть дверь.

Рисунок 3 – анимация противника

Рисунок 4 – анимация игрока при движении налево

Рисунок 5 – анимация игрока при движении направо

Рисунок 6 – анимация игрока при атаке, когда он находится слева

Рисунок 7 – анимация игрока при атаке, когда персонаж находится справа

5. Заключение

В рамках учебно-практической работы были рассмотрены платформы, операционные системы, жанр RogueLike, аналоги жанра labyrinth, внесены технические предложения, даны технологии для разработки и выбрана наиболее подходящая технология, разработаны требования к игровому приложению, разработана архитектура и код, проведено тестирование завершена подготовка к развертыванию на выбранных платформах, собраны файлы, разработана документация для конечного пользователя для доставки конечному пользователю.

Источники

- https://revolution.allbest.ru/programming/01181659_0.html (28%)