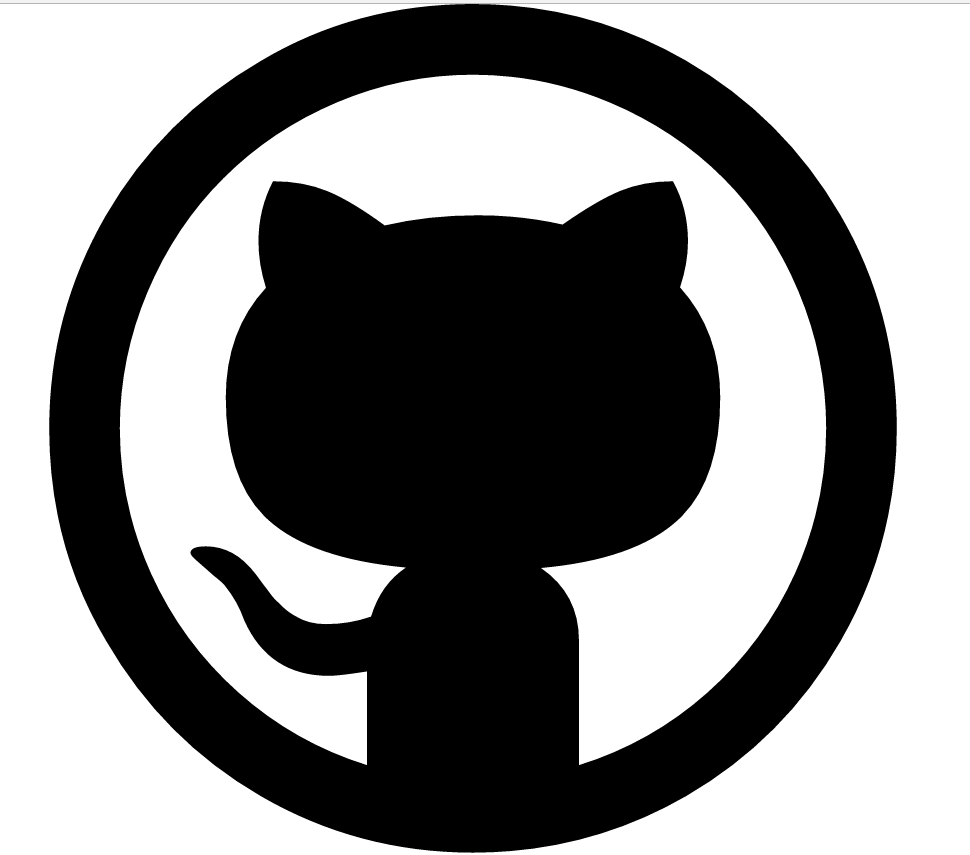
Christmas Adventures

Этот проект представляет собой игру в жанре платформер, сделанная по примеру [Trap Adventure 2](https://en.wikipedia.org/wiki/Trap_Adventure_2) и стилизованная под тему Нового Года.

Ссылка[](https://github.com/Ulbandus/ChristmasAdventures/):

**Установка:**

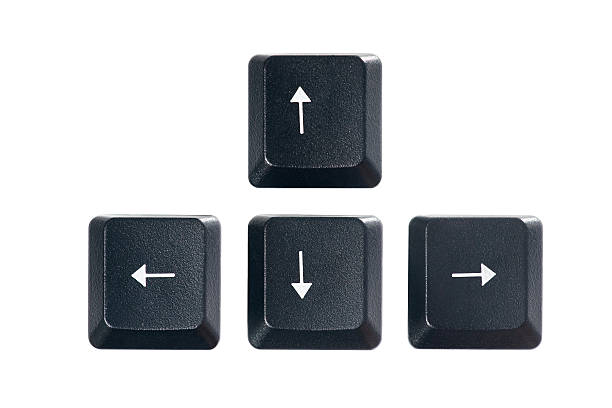
 **Linux:**

apt install git python3

git clone https://github.com/Ulbandus/ChristmasAdventures/

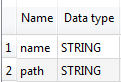
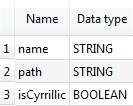
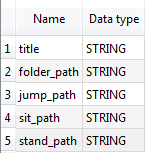
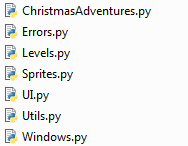
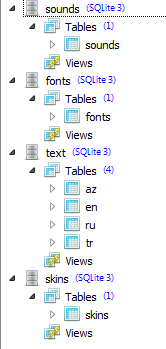
cd ChristmasAdventures

pip install -r requirements.txt

python3 ChristmasAdventures

**Управление:**

* **Up – Прыжок**
* **Down – Уклониться**
* **Left, Right - Движение на лево и направо**

****

**Структура баз данных и основные файлы проекта**

**Классы:**

**ChristmasAdventures.py:**

***GameManager*** – Лаунчер игры:

* start — Иницилизация окна
* next\_level — Запуск следущего уровня
* run\_game — Запуск окна

**Errors.py:**

***SpriteError –*** Ошибка отсутствия определённого спрайта

**Levels.py:**

***FirstLevel, SecondLevel, ThirdLevel, FourthLevel, FifthLevel*** *-*

классы уровней с мктодами get\_sprites в котором находится

расположение всех элементов

**Sprites.py:**

***Sprite*** *–* базовый класс спрайта

* setup — Объявление таких переменных как skin, height, width
* set\_group — Добавляет скин в группу скинов
* to\_spawn — Телепортирует игрока на спавн
* teleport — Телепорт игрока
* full\_file\_path — Возвращает полный путь спрайта
* set\_skin — Устанавливает картинку
* change\_skin — Изменяет картинку
* is\_in\_window — Проверяет находится ли игрок в пределах окна

***Player*** *–* класс игрока (унаследован от Sprite)

* change\_player\_skin — Смена костюма игрока
* die — Отнимает одну жизнь у игрока и телепортирует его на спавн
* get\_collide\_object- Возвращает объект, который находится в коллизии сигроком
* get\_skins — Возвращает словарь со скинами
* is\_collide — Проверка находится ли игрок в коллизии с чем-либо
* move — Движение игрока
* update — Обновление состояния
* flip – Отражаерт спрайт
* jump\_skin – Выбирает спрайт для прыжка
* move – Двигает персонажа
* sit – перевод персонажа в сидячее положение
* stand – перевод персонажа в стоячее положение
* update – Обновляет состоянее перонажа

**Particles** — класс партиклов

* get\_particles – Получение частиц
* update– Обновление состояния

**UI.py:**

***Button*** – базовый класс кнопки

* load\_icon — Загрузка иконки и её кроп
* set\_icon — Установка иконки на кнопку
* get\_rect — Получение pygame.Rect по заданным параметрам
* move — Перемещение кнопки
* is\_pressed — Проверка нажата ли кнопка

**Label** – базовый класс надписей

* connect\_to\_button — Привязка к кнопке
* move — Перемещение надписи

**Message** – класс для отображения уведомлений

* is\_alive – Проверка открыто ли окно

**Text** – класс для отображения текста на экране

* get\_font – Получение специального шрифта
* get\_text – Получение отренеренного текста
* draw – Отрисовка текста

**Utils.py:**

**Sounds** – класс для получения звуков и работы с ними

* get\_sounds — Возвращает словарь со звуками
* play — Проигрывает звук
* stop — Останавливает звук
* stop\_all — Останавливает все звуки

**Settings** – класс настроек, парсщий их из settings.ini, методы:

* all\_settings – Получение всех настроек из файла
* save — Сохранение настроек

**DataBase** – работа с базой данных sqlite3

* execute — Выполнение запроса к базе данных
* get\_sounds — Получение списка со звуками
* get\_font — Получение шрифта по его имени
* get\_text — Возвращает словарь с текстом для программы

**Windows.py:**

**EndWindow, LoseWindow, ShopWindow, MainWindow** – Дочерние классы Window

**Level** – класс уровней

* game\_cycle -Запускает уровень
* event\_handler — Обрабатывает события
* draw — Показывает и обновляет объекты в окне

**Window** – базовый класс окна

* get\_screen — получает рабочее окно

Библиотеки:

* Pygame – основная библиотека, в которой будет организовано графическое представление игры и работа со спрайтами
* Sqlite3 – библиотека для работы с базами данных
* Pygame\_Gui - Библиотека отвечающая за пользовательский интерфейс
* ConfigParser - запись и чтение настроек
* Pyganim - работа с анимацией

Автор:

[Гусейнов Эльдар](https://github.com/Ulbandus/) – GitHub