


Christmas Adventures



Этот проект представляет собой игру в жанре платформер, сделанная по примеру Trap Adventure 2 и стилизованная под тему Нового Года.

Ссылка: 

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



Name	Data type	Name	Data type
1 play	STRING	1 name	STRING
2 shop	STRING	2 path	STRING
3 faq	STRING	3 isCyrillic	BOOLEAN
4 settings	STRING		
5 faq_text	STRING		
6 skin_blocked	STRING		
7 ordinary_level_tip	STRING		
8 build_level_tip	STRING		

Структура баз данных и основные файлы проекта

Классы:

ChristmasAdventures.py:

GameManager — Лаунчер игры:

- start — Инициализация окна
- next_level — Запуск следующего уровня
- run_game — Запуск окна

Errors.py:

SpriteError — Ошибка отсутствия определённого спрайта

Levels.py:

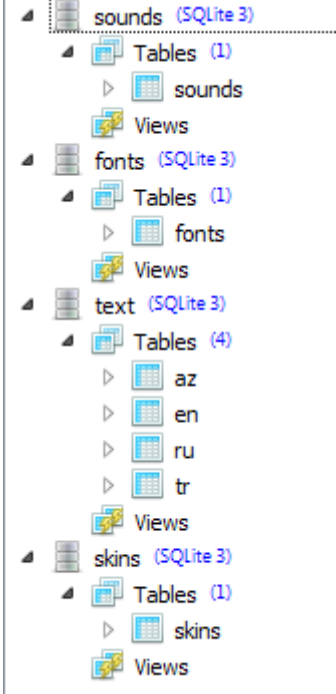
FirstLevel, SecondLevel, ThirdLevel, FourthLevel, FifthLevel -

классы уровней с методами `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 с игроком
- `get_skins` — Возвращает словарь со скинами
- `is_collide` — Проверка находится ли игрок в коллизии с чем-либо
- `move` — Движение игрока
- `update` — Обновление состояния
- `flip` — Отражает спрайт
- `jump_skin` — Выбирает спрайт для прыжка
- `move` — Двигает персонажа
- `sit` — перевод персонажа в сидячее положение
- `stand` — перевод персонажа в стоячее положение
- `update` — Обновляет состояние персонажа

Particles — класс частиц

- `get_particles` — Получение частиц
- `update` — Обновление состояния

UI.py:

Button – базовый класс кнопки

- `load_icon` — Загрузка иконки и её крор
- `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 — [GitHub](#)