|  |
| --- |
| Никита Поглазов и Павел Савин представляют: |
| Проект PyGame.  2D платформер “Paulum Ambulare”. |
| Пояснительная записка |

|  |
| --- |
| 18.01.2021 |

# Цель проекта

Цель нашего проекта – создание игры с использованием возможностей языка программирования Python и движка PyGame основанного на мультимедийной библиотеке SDL.

**Структура** **программы**

Для удобного взаимодействия с отображением экрана был создан класс Display. Он содержит в себе метод update обновляющий все составляющие картинки на экране. Один из них – класс Camera, который просчитывает смещение персонажа относительно центра экрана и перемещает все объекты уровня на это смещение, грубо говоря, удерживая персонажа в центре экрана.

Самый большой по объему и значимости класс – Character. Он содержит в себе набор инструкций, которым следует персонаж в определенных игровых ситуациях: от перемещения собственного спрайта (метод move) до запуска драматичной анимации смерти, при столкновении с врагом или от падения в бездну (метод set\_death). Помимо гоблинов (или как принято в нашей команде – гоблов), предусмотрены столкновения с препятствиями, монетами и финишными флагами. Персонаж также имеет набор анимаций: покой, бег, прыжок, падение, смерть. Смена анимаций происходит с помощью метода set\_animation.

Во всех играх должны быть противники и наша – не исключение. Главный герой должен избегать столкновения с гоблинами, рыщущими по всему уровню, которые задаются классом Enemy. Но гоблины не такие ловкие, как персонаж и все что они умеют – бегать из стороны в сторону (метод move) со скоростью в 5 раз меньше персонажа, отталкиваясь либо от обычных блоков, либо от невидимых блоков – ограничителей которые не позволяют им выйти за пределы платформы. Структура этого класса схожа с Character. Противники также имеют анимацию бега и покоя.

В нашей игре присутствуют несколько экранов: главное меню, пауза, смерть и победа. Для их отображения созданы функции main\_menu, pause\_menu, game\_over и finish. Все они отображают пункты меню, между которыми можно переключаться на стрелочки и выбирать на Enter. Для превращения сырого текста в красивый создан метод text\_format, который переписывает исходный текст, но с использованием заданных параметров шрифта, цвета и размера.

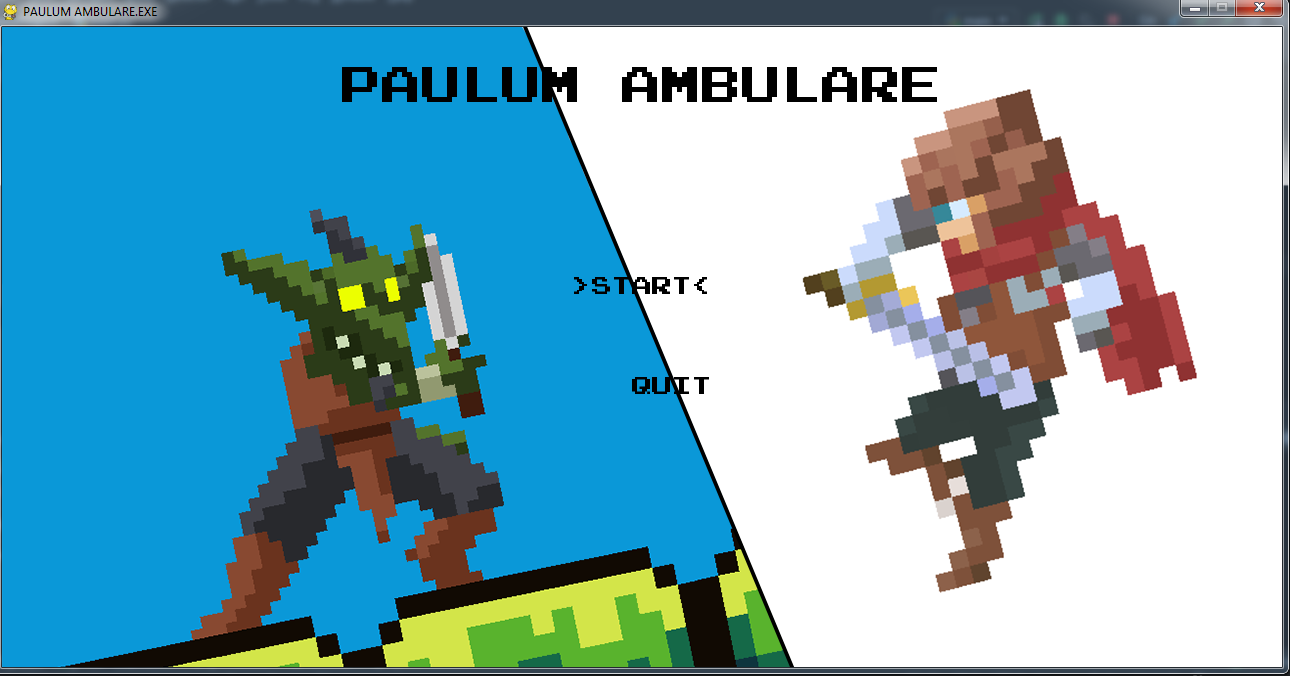
Для загрузки уровня созданного в TiledMapEditor разработан класс TiledMap, который загружает данные из tmx файла и рисует с их помощью уровень, используя классы Tile и его “наследники”: Block, Finish и Deco - а также класс Coin, который, в отличии от Tile, имеет особую анимацию.

Для удобного управления музыкой был создан класс Music, с помощью которого можно переключать треки, а также ряд функций: play, pause и unpause, название которых говорит само за себя.

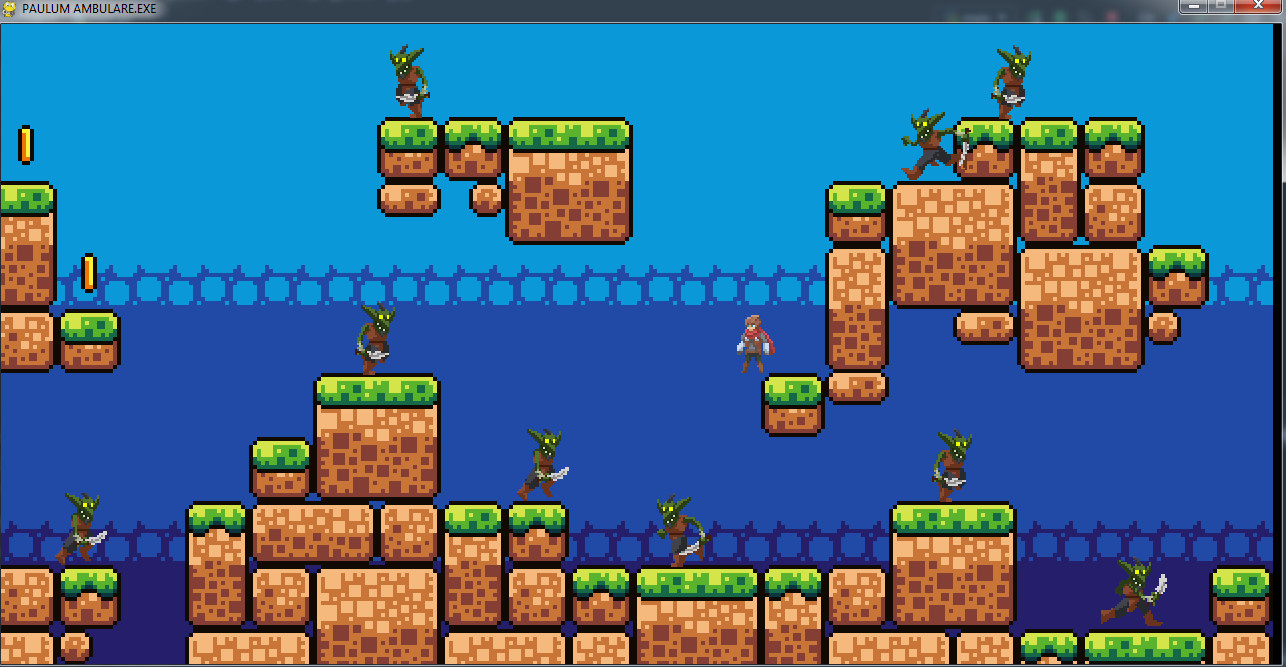
Чтобы мотивировать игрока играть усерднее, был написан класс Stats, который отслеживает количество собранных монет и попыток со временем, затраченных на прохождение уровня.

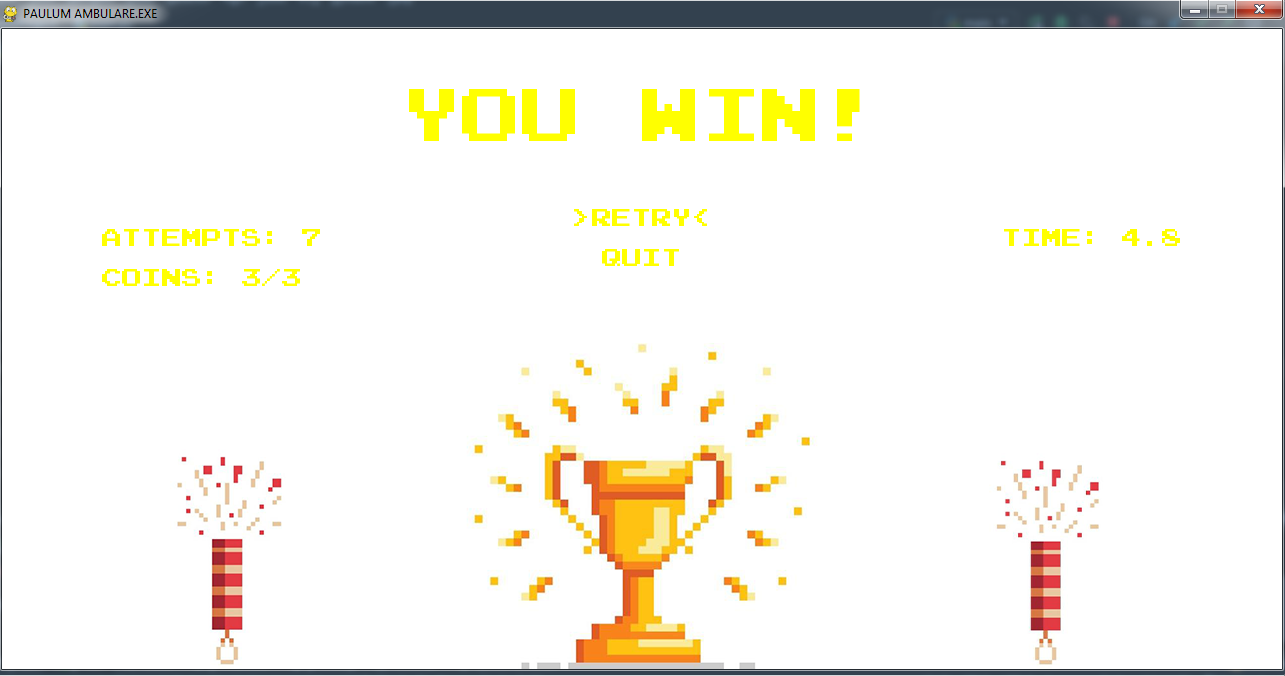
Для работы необходимо иметь установленный Git и аккаунт на GitHub. По выданной ссылке склонировать репозиторий себе и установить библиотеки pygame и pytmx.

Главное меню



Запущенный уровень



Экран победы со статистикой

Экран смерти

