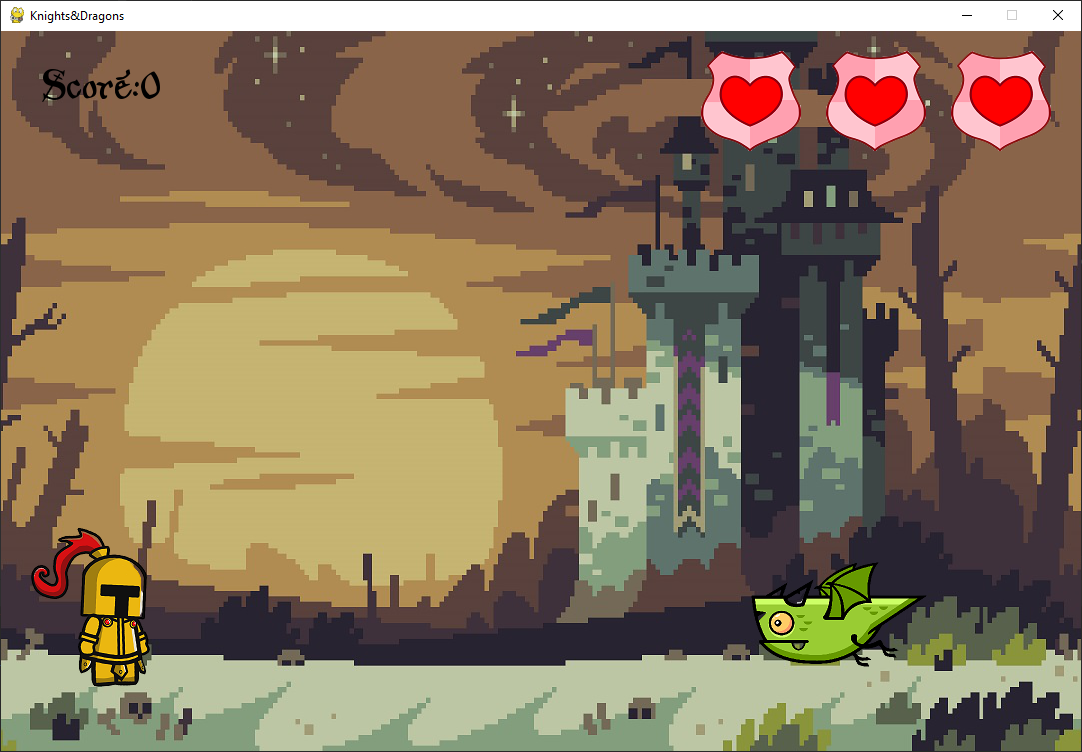
KnightRunner

Проект реализовал студент 3 курса группы БСБО-07-18 Малышев Евгений Алексеевич.

Игра своего рода тайм-киллер. Смысл игры в преодолении надвигающихся противников путём перепрыгивания их, тем самым зарабатываются очки. Нужно набрать как можно больше очков и достичь новых, больших рекордов.



Основные классы Player, Mob, Button и Save, в которых описаны игрок, противник, кнопки и сохранение соответственно.

Класс игрока имеет функции, описывающие его отрисовку спрайтов из списка, логику прыжка, проверку по позиции Y и возврат местоположения на исходную. Класс противника имеет функцию увеличения скорости, и так же, как и класс игрока, имеет функции отрисовки спрайтов, проверку позиции по Y и возвращение на исходную. Класс кнопки имеет функцию по её отрисовке и выбору уровня игры для дальнейшего использования. Класс сохранения даёт возможность сохранять и получать результаты максимального рекорда, чтобы после выхода из игры рекорд был сохранен и была возможность продолжить с уже поставленным рекордом.

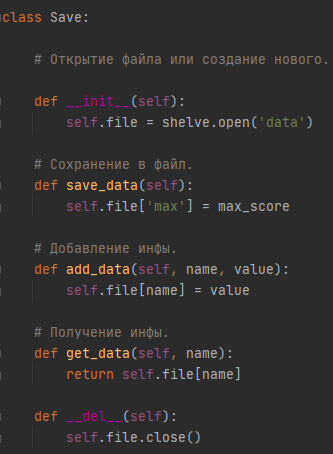
Что реализовано в игре:

* Интерфейс пользователя (главное меню, в котором присутствуют кнопки, старта, выбора одного из двух уровней игры и выход игры(функция menu(); draw\_button();)).
* Сохранение и получение максимального рекорда для продолжения игры с уже поставленным рекордом.(класс Save)
* Выбор уровней, в которых изменены параметры противника, его спрайты и бэкграунд уровня.(функция draw\_button();)
* Звуковое сопровождение игры и звуковые эффекты кликов по кнопкам, нажатие паузы.
* Паузы во время игры, возможность снять паузу и продолжить игру.(функции is\_paused(); is\_unpaused(); run();)
* Повторить попытку заново после смерти.(функция restart();)
* На каждой попытке игрок имеет 3 жизни, при столкновении с противником, отнимается 1 жизнь.(функции draw\_health(); check\_hp();)
* Возможность восполнить потерянные жизни, путём подбора прыгающей розочки. (Класс Regen, функции check\_rect\_hp(); check\_hp();)
* Набор очков по ходу игры.(функции speed\_inc(); scoring();)
* Реализованы функции проверки ректанглов на столкновения игрока, противника и розочки, добавляющей хп. (функции check\_rect(); check\_rect\_hp();)
* Получение нового образа игрока при достижении 50 очков рекорда.(функция draw\_button();)
* Прочитать информацию об управлении. (функции menu(); draw\_button();)

Очень интересна реализация сохранения рекорда. Модуль shelve позволяет сохраняет объекты в файл с определенным ключом. Затем по этому ключу может извлечь ранее сохраненный объект из файла. Благодаря этому модуля была реализована возможность сохранения максимального рекорда.

Подключение:

import shelve

Класс:  


Сохранение рекорда:

save.save\_data()

Получение сохраненного рекорда из файла:

max\_score = save.get\_data('max')