**Пояснительная записка.**

Проект «Rodeo went wrong».

Автор: Дорохов Евгений Александрович.

**Идея.**

В ходе родео скакун падает с лошади и теперь вынужден от него убегать, продержаться как можно дольше.

В приложении должны быть использованы все пройденные темы.

**Реализация.**

После подключения необходимых модулей, инициализации pygame, создания окна начинается разработка функций и классов.

Классы:

*Tile* – «плитки» поля.

*Character* – персонаж. В игре их двое: парень и лошадь.

*Camera –* камера, меняющая картинку относительно спрайта игрока (им является парень, убегающий от лошади).

Функции:

*generate\_map()* – отвечает за рандомную генерацию карты, представляющую из себя текстовый файл.

*load\_level()* – возвращает содержимое файла “map.txt” в виде списка строк.

*load\_image()* – возвращает конвертированное изображение.

*start\_screen() –* начальное окно программы.

*final\_screen() –* конечное окно программы.

*generate\_level()* – создание экземпляров класса Tile и класса Character.

*terminate()* – для завершения работы программы.

*main\_generation()* – генерация игрового поля при помощи выше перечисленных функций.

*writing\_round() – экран смены раунда.*

*main()* – игровой цикл.

Что происходит в игровом цикле?

Сначала генерируется мир, затем функция стартового окна.

Бесконечный цикл, обрывающийся только при закрытии окна.

Кроме этого события (закрытие окна) главным образом обрабатываются ещё четыре – это стрелки на клавиатуре.

По умолчанию персонаж движется вверх. При нажатии на стрелку он начинает двигаться в соответствующем направлении – вычитание единицы из его координат каждую итерацию цикла с частотой кадров в секунду – 150. Если же игрок врезается в препятствие частота кадров в секунду падает до 75, и игрок движется медленнее в два раза в “удобном” ему направлении (“удобное” направление определяется по предыдущему). Лошадь не изменяет скорости своего движения ни в одном случае.

Лошадь движется путём добавления в список её действий команд, изменяющих её координаты. Это делается для того, чтобы она могла повторить траекторию движения игрока и в случае поймать его на ошибке.

Проверка на столкновение персонажей делается при помощи метода *colliderect.*

По проходе 15 секунд начинается новый раунд и карта генерируется по новой.

Введётся счёт раундов, рекордные показатели хранятся в файле record.txt и в случае корректируются.

**Использованные технологии.**

Язык программирования - *Python.*

Модули, необходимые при разработке проекта:

pygame, sys (exit), os (path), random (sample, choice), time (perf\_counter).

**\***

О более мелких подробностях в комментариях кода.