*Скворцов Алексей, Провоторин Лев*

Проект «Tanks.ReLoad». Пояснительная записка.

Описание с точки зрения пользователя

Данный проект представляет собой игру-аркаду. Игроку (или двум игрокам, если выбран соответствующий режим) дается по танку на игровом поле, задача – набрать как можно больший счет. Для набора очков нужно побеждать врагов (танки под управлением программы).

При запуске программы предлагается UI с вариантами: начать новую игру, продолжить существующую игру, изменить настройки, выйти. В изменении настроек можно изменить управление танками игроков (выбрать один из двух вариантов раскладки, причем два танка в режиме на двух игроков могут управляться одним набором клавиш) и громкость звука в игре.

В течение игры пользователи зарождаются (далее в тексте *спаунятся*) на определенном месте, заданном в описании уровня. Танки врагов спаунятся через некоторый интервал времени в случайной из нескольких точек спауна врагов, так же заданных в описании уровня. Если у врага кончается уровень HP (health-point, очки здоровья, в коде обозначено как свойство класса .strength – прочность), то он «погибает» и исчезает с уровня. Если HP кончается у игрока, то он возрождается на своей точке спауна через определенное время.

И игроки, и враги обладают системой прокачки. Есть три уровня. При возрождении игроки получают первый уровень (даже после смерти танка игрока в течение игры его уровень сбрасывается до стартового). Танки врагов получают уровень случайным образом. Чем больше уровень, тем сильнее урон корпусом (действует только на объекты карты, но не на врагов) и урон выстрелом (действует на все объекты, которым можно нанести урон).

В коде постоянно проверяется, остались ли еще танки на уровне (либо на карте, либо еще не заспауненные). Если танков не осталось, то игроки переходят на следующий уровень. Перед каждым уровнем игра уведомляет об этом игроков и предлагает начать путем нажатия любой клавиши. После завершения игры игроки видят свой счет к концу игры.

Счет начисляется только за убитых врагов, но пропорционально их уровню (чем сильнее враг, тем больше очков он дает). За смерть игрока очки вычитаются (возможно даже уйти в минус). При игре вдвоем счет общий.

В игре доступно меню паузы. Если оно активировано при помощи клавиши Esc, весь игровой процесс останавливается. Оттуда можно вернуться в игру, перейти в раздел настроек, выйти в меню или выйти из игры совсем.

У объектов карты присутствует разрушаемость, некоторые объекты могут гореть. Это добавляет игре реализма.

Описание с точки зрения реализации

В коде реализованы и используются много классов и функций.

В классе Tile определены все кусочки поля. Они выступают исключительно как фон и никак не влияют на игровой процесс.

Класс Object – базовый класс для разрушаемых объектов. В нем определена возможность нанесения урона объекту и его уничтожения, если параметр прочности .strength не больше нуля.

При помощи классов Boarding (забор), Train (поезд), Tree (дерево) определены основные объекты сцены после игроков и противников. Это разрушаемые объекты, объекты класса Tree также могут гореть (если выстрелить по дереву, то оно загорается, горит с анимацией и через некоторое время удаляется с поля, когда сгорело).

В классе Building изначально были определены здания. Позднее в процессе разработки здания были удалены, и теперь класс Building используется для создания неразрушаемых объектов (такими являются, например, камни, которыми ограничена карта уровня). Основное отличие от разрушаемых объектов (помимо разных групп спрайтов) - параметр .strength установлен как float(‘inf’), т.е. их прочность бесконечна. Поэтому они не могут быть уничтожены.

Класс Player создает спрайт игрока, класс Enemy – спрайт врага. Они также унаследованы от класса Object, т.к. обладают ограниченной прочностью.

Класс AnimatedSprite – базовый класс для анимированных спрайтов, *не обладающих разрушаемостью*. Он используется для анимации выстрела. Реализован с поддержкой «на будущее» пока не используемых функций: например, поддерживается любая длительность жизни анимации (параметр .lifetime; анимация показывается ровно столько раз, после чего объект самоуничтожается)

Для выстрела используются 3 класса. Класс Shot производит создание невидимого спрайта выстрела, который движется и проверяет столкновения. В классе ShotStart определена только анимация выстрела у дула танка. В классе ShotEnd определено достаточно много: анимация попадания, нанесение урона, обсчет очков для игрока. Все объекты этих классов одноразовые, т.е. удаляются, как только выполнили свою функцию.

В функциях файла main.py, не принадлежащих классам, определен UI, игровой цикл, обработка уровней.

В файл system\_functions.py вынесены некоторые специальные функции (экстренное завершение, загрузка и первичная обработка картинок, загрузка уровней).

В файле init.py определены почти все глобальные переменные для программы main.py (мы решили, что так будет удобнее редактировать численные константы, вроде скорости, наносимого урона и т.п.), включая загружаемые в начале картинки.

Используются следующие библиотеки: pygame (по требованиям к проекту) sys (экстренное завершение работы), os (работа с файловой системой), random (достаточно много случайных событий в игре).

Скриншоты



