**Проект «2д-шутер»**

Утепкали Махамбет

Тыныштык Дархан

**О проекте:**

Это приложение – игра, в жанре «шутер», категории «2д с обзором сверху».

**Цель проекта:**

Создать игру с помощью библиотеки pygame и реализовать все взаимодействия спрайтов внутри игры, тем самым обеспечив качественную и комфортную игру.

**Техническая часть:**

Программный код проекта состоит из главного игрового цикла и множества других игровых циклов отдельных вкладок.

Список функции вкладок:

**main\_menu()** – функция, показывающая главное меню. Она будет вызывать другие функции, т.е. открывать другие вкладки в зависимости от действий игрока.

**levels()** – функция, показывающая меню выбора уровней. Он состоит из трех кнопок выбора уровней и кнопки НАЗАД. При нажатии кнопки уровней в первую очередь будет проходить проверка, прошел ли игрок предыдущие уровни. Если да будет вызвана функция загрузки уровней и в ней загрузятся данные об уровне. Затем вызывется функция самой игры.

**game(image, pos, level, enemy\_pos)** – в эту функцию передаются такие параметры: image – карта уровня, pos – позиция игрока на карте, level – номер уровня, enemy\_pos – позиции врагов. В этой функции происходит главный процесс игры. Здесь размещаются спрайты врагов, стен и игрока, а также происходит их движение и взаимодействие.

**pause()** – меню паузы. Здесь игра временно останавливается, и открывается меню паузы. Здесь 4 кнопки: кнопка возобновления игры, которая останавливает эту функцию и возобновляет игру, кнопка настроек, которая вызывает функцию settings(), т.е. открывает меню настроек, кнопка выхода в главное меню, которая завершает игру и открывает главное меню и кнопка выхода из программы, которая закрывает программу.

**settings()** – меню настроек. Здесь есть единственная кнопка – кнопка сброса игры. Здесь можно сбросить прогресс и начать игру с нуля. При нажатии кнопки файл с прогрессом игры обнуляется.

**you\_win()** – конечная заставка победителя. Она выходит, когда игрок победил всех врагов. В зависимости от пройденного уровня будут появляться разные сообщения. Также в конечной заставке будут выводиться количество очков игрока.

**game\_over()** – конечная заставка проигравшего. Она появляется, когда игрок умер. Тогда показывается соответствующее изображение и надпись.

**Спрайты**

В игре есть 6 классов спрайтов: стена, пуля, кнопка, игрок, враг, кровь.

Спрайт **игрока** может принимать множество различных изображений – положение игрока при повороте на все 360 градусов по 10 градусов на каждый кадр и изображения умершего игрока, которые меняются после смерти игрока. При инициализации в объект передаются аргументы позиции игрока и группы спрайтов, к которым принадлежит этот экземпляр. Затем координаты спрайта меняются, и он размещается в заданном нами месте. Управлять игроком можно с помощью клавиш WASD(движение), стрелок вправо-влево (поворот) и кнопки ПРОБЕЛ или кнопки мыши(выстрел).

При нажатии клавиш движения положение игрока меняется по одному из осей, в зависимости от нажатой клавиши. А при нажатии клавиш поворота, изображение игрока меняется на соответствующее изображение, только немного повернутое в определенном направлении и градусе. При отпускании этих кнопок, смена кадров или движение останавливается. При нажатии кнопок для выстрела, вызывается функция выстрела, в которой происходит проверка, жив ли игрок, если да то создается спрайт пули в местоположении игрока *(подробнее о спрайте вы сможете узнать в следующем абзаце)*. Каждое мгновение происходит обновление спрайта. В функции обновления происходит проверка, умер ли игрок, или жив. Если он жив, то изображение меняется в изображение, заданное в других функциях (поворот). Если игрок мертв, вызывается функция смерти игрока. В данной функции активируется псевдо-таймер, и через определенный промежуток времени создается эффект крови или меняется изображение мертвого игрока. Если прошло достаточное количество времени, объект убирается. Проверка мертв ли игрок или жив, проверяется с помощью функции проверки статуса. В ней происходит проверка, пересекаются ли пули противника с игроком, и если да то установить флажок смерти на True.

Спрайт **пули** также как и спрайт игрока, принимает множество одних и тех же картинок, только в повернутом на какой-то градус виде. В инициализатор передаются такие параметры как позиция изображения, т.е. в какое направление должна быть повернута пуля, координаты пули – местоположение и группы спрайтов. В зависимости от направления, движение пули будет меняться. Например, если пуля направлена вправо, то его движение будет равно такому – (1, 0). Первое значение движение по оси х, а второе по оси у. Так же как и все спрайты, он будет обновляться постоянно. При обновлении происходит проверка, попала ли по цели. Если да то убрать его. Иначе дальше осуществлять движение и проверку статуса. В проверке статуса проверяется, пересекается ли пуля с каким-то объектом, если да, то установить флажок смерти.

В отличие, от других спрайтов, спрайт **стены** состоит лишь из одного невидимого изображения. Преимущества такого способа в том, что можно изменять лишь изображение карты уровня, а не добавлять новые изображения. При контакте со стеной пуля исчезает, а игрок не может войти в него.

В классе **кнопки** есть множество изображений всех кнопок. При инициализации передается название кнопки и по этому названию загружаются изображения определенной кнопки. Также передаются координаты объекта, по которым размещается этот объект. Также постоянно происходит обновление. В обновлении проверяется, пересекается ли мышь с этой кнопкой, если да, то сменить изображение, иначе вернуть начальное изображение. При нажатии на эту кнопку активировать псевдо-таймер. Если прошло определенное время, то установить флажок. Этот отсчет нужно для того чтобы одновременно не считалось клик несколько кнопок.

Класс **крови** содержит несколько одинаковых изображений разных размеров и при инициализации выбирается случайный из них и устанавливается движение по осям по заданным аргументам и действие гравитации. Также устанавливаются границы, внутри которой должны находиться эти эффекты. Если они выйдут за пределы этой границы, они исчезают.

Спрайт **врага** может принимать множество различных изображений – положение врага при повороте на все 360 градусов по 10 градусов на каждый кадр и изображения умершего врага, которые меняются после его смерти. В инициализатор передаются такие переменные: pos, hurt\_count, group. Pos – позиции по которым враг должен перемещаться. Hurt\_count – необходимое количество ранений для убийства врага. Group – группы, к которым относится этот экземпляр. В начале по позициям, по которым враг должен перемещаться, определяются направления движения. Например, (1, 0) – чтобы двигаться вправо. Затем устанавливается положение игрока(поворот на какой-то угол) по переданной переменной pos. Затем объект размещается также по переменной pos. В функции обновления происходит проверка, умер ли объект, если да вызвать функцию смерти. Иначе происходит смена кадров, движение, смена направления, поворот, выстрел, проверка статуса и другие события. В функции выстрела происходит отсчет псевдо-таймера, и если прошло определенное время и объект еще жив, происходит выстрел. В функции проверки статуса проходит проверка, попала ли пуля игрока по объекту и это не та же пуля что и предыдущая, и если да, то увеличиваем количество ранений. Если количество ранений – достаточное чтобы умереть, активируем флажок смерти. В функции смерти также проходит отсчет псевдо-таймера, и через какое-то время вызываем эффект крови и меняем кадр. Если прошло достаточное количество времени, удаляем объект.

**Структура файлов уровней**

Все данные об уровнях хранятся в текстовых файлах. Формат текстовых файлов таков:

В первой строке записаны координаты прямоугольников всех стен, разделенные пробелами. А эти прямоугольники разделены символами “|”. Вот пример:

42 10 529 8|204 16 9 93|359 16 10 207|33 60 10 50|47 103 96 5|32 112 5 379

Во второй строке записаны координаты игрока, разделенные пробелом. Например, «0 0».

В остальных строках записаны координаты четырехугольника и положение врага при каждом повороте, по которым враг должен перемещаться. Вот пример:

100 140 0|301 140 9|301 401 18|100 401 27

Все точки разделены “|”. Каждая точка содержит три элемента: координаты х, у и положение врага в этой точке.

Прогресс игрока хранится в текстовом файле, который содержит единственно целое число – сколько уровней игрок прошел. Оно равно 1, если игрок еще не прошел ни единого уровня. Оно равно 2, если игрок прошел первый уровень. Оно равно 3, если игрок прошел 1 и 2 уровень. Оно равно 4, если игрок прошел все уровни. По этому файла выполняется проверка и открывается или закрывается доступ к какому-то уровню.