ТЗ

Дизайн

Итак, наш проект “***СуперДуперМегаОмегаИграБога***” встречает игрока невероятным меню. В нем вы узрите кнопку “В бой”, “Апок” (завершение игры) и, конечно же, “НАСТРОЙКИ”. Кнопки расположены на светлом свитке. Во вкладке настроек можно выбрать методом нажатия на песню саундтрек, также можно регулировать громкость всех аудио эффектов и музыки при помощи небольшого треугольного микшера. Переходим к самому игровому процессу. На игровом поле в центре расположено клеточное поле, оно сформировано из слотов для карт светлого оттенка. Поле занимает 2 строки (союзная и вражеская) и по 4 слота на каждую строку. Снизу находится еще одно клеточное поле, которое аналогично одной строке игрового клеточного поля, но в этом поле находятся не слоты, а карты. Карты представляют собой игровых персонажей, картинки для которых взяты и Бестиария D&D, сверху карты будет расположена картинка, а под ней название, способность персонажа и характеристики. Чуть правее клеточных полей будут находиться небольшие сферы, на которых будет отрисовываться общий урон. Во время анимации атаки каждой из карт должен быть прорисован взмах мечом, а после призыва культиста должен быть прорисован призыв пентаграммы поверх карты.

Код

Используемые библиотеки:

Sqlite3 – используется для работы с базой данных, которая обеспечивает сохранение результатов матчей, состояния карт (все карты, которые находятся на столе, в руке у игроков, в колоде у игроков), данных колод, карт.

Random – используется для написания искусственного интеллекта противника, если быть конкретнее, то для выбора карт, с которыми будет взаимодействовать оппонент - бот.

PyGame – основная библиотека для написания игры, весь проект создается в первую очередь благодаря ей. Она позволяет отрисовывать каждый кадр по заданным параметрам.

Вся игра базируется на цикле while, который будет повторять цикл на постоянной основе, пока не будет завершена работа программы. Всего будет несколько циклов (меню, настройки, игра, …), все они отвечают за прорисовку разных окон, и все они помещены в различные функции, вызов которых будет запускать новое окно в игре. В цикле, который отведен под меню, находятся функции для отрисовки кнопок (см. ниже), в цикле, который заточен под сам игровой процесс, находятся функции для отрисовки игрового поля и кнопок завершения игры и пропуска хода (см. ниже).

Меню. Отрисовывается просто, при помощи screen.blit накладывается задний фон, а при помощи собственных функций создаются и отрисовываются кнопки (так же при помощи screen.blit). Кнопки работают также незамысловато, каждый кадр идет проверка значений (pygame.mouse.get\_pressed()) нажатия на мышку и, если оно равно 1, то считываются координаты курсора (pygame.mouse.get\_pos()), если он находится в области кнопки, то запускается функция, к которой она привязана. Кнопки в меню запускают цикл игрового стола, настроек, выхода и т.д.

Игровой стол. Здесь уже сложнее, отрисовывается задний фон, кнопки выхода и пропуска хода, которые работают так же, как и в меню, и клетчатое поле. Клетчатое поле отрисовывается циклами for путем наложения моделек для слотов функцией screen.blit. При помощи обращения к базам данных стола \*база данных\*.execute(\*что сделать\*) идет просмотр значений, которые относятся к определённым слотам на клетчатом поле. При значениях, неравных нулю, отрисовывается карта, которая имеет соответствующее значение (Все карты и их значения находятся в отдельной базе данных), в ином случае отрисовывается пустой слот. По принципу, как с кнопками, мы можем выкладывать карты на стол, после проверки значения нажатия на мышку (pygame.mouse.get\_pressed()) если значение равно 1, то карта, которую вы взяли, будет принимать координаты нахождения относительно курсора (pygame.mouse.get\_pos()), а на ее прежнем месте будет пустой слот. После того, как вы отпустите мышку, если вы наведетесь на пустой слот, то карта будет положена на стол, в базе данных значение слота в руке будет равно 0, а на столе значению карты. Сразу, как вы положите карту на стол, активируется ее способность, если это возможно, и произойдет обращение к функции со списком способностей. Все анимации, которые будут воспроизводиться, воспроизводятся при помощи проигрывания цикла for, он поочередно обрисовывает все кадры анимации, кроме анимации смерти персонажа, пока что, карта просто сужается в центр (pygame.transform.scale()) с каждым кадром после своей смерти, и через какое-то время она окончательно пропадает.

Настройки и Звук. Вкладка настроек отрисовывает, конечно же, задний фон, текст, при помощи отдельно созданной функции, в которую поступают все необходимые значения, и она выводит на экран поступившие символы (pygame.font.Font, \*шрифт\*.render(), screen.blit()), кнопка в виде текста для смены аудиодорожки и кнопка в виде 5 треугольников для смены громкости звука. Кнопка для смены дорожки работает, как обычная, но она просто меняет проигрываемую фоновую музыку на другую (pymixer.load(), pymixer.play()). Кнопка для смены громкости звука при нажатии на нее, как на обычную кнопку, левой кнопкой мыши делает звук громче (pymixer.set\_volume()), а правой – делает звук тише.