Пояснительная записка

Stronghold

**Введение:**  
В качестве проекта Pygame я создала простую игру-стратегию, неподобие серии игр Stronghold.

Её суть в том, что бы создать деревню и армию и с их помощью выигать как можно больше сражений.

Идея проекта принадлежит В. Воронцову.

**Реализация:**

Проект сосотоит из двух “.py” файлов: “village.py” - непосредственно игра и ***“create settings.py”,*** отвечающий за запись настроек, используемых в игре, в файл. Таких как время, за которое происходят различные события и монеты, получаемые в результате них. Благодаря отдельнуму файлу можно легко изменить скорость и сложность игры (заданные в нем параметры).

В папке ***data*** храняться изображения и текстовые файлы, как с настройками, так и с информацией, необходимой в игре (правила и сообщения). Плюс файл, в который записан рекорд – лучшее количество выигранных боев за всю историю игры.

Помимо стандартной функции для загрузки картинок, у меня есть функции, загружающие поле из текстого файла, сообщения из таких же вайлов и настройки из csv файла, чтобы легко поместить их в словарь.

Игра имеет де «локации»: деревня и поле боя. Их названия заданы константами для использования в игровом цикле.

***Деревня*.** Изначально на поле находится казарма. С нее тут же необходимо собрать доход, чтобы начать строить другие сдания. Жилой дом строится при нажатии на левую кнопку мыши, стоит 20 монет и заселяется 5 жителями, из которых можно сделать воинов, нажав правой кнопкой мыши на казарму (правой собираются монеты). Кроме того можно построить ферму, которая пассивно приносит еду. Все ресурсы заданны словарем, где «фермеры», «войны» и «свободные люди» - списки объектов соответствующих классов. Воины заданы отдельной группой, чтобы была возможность рисовать их отдельно, когда место сменется на поле боя. Больше о правилах игры можно узнать из файла rules.txt.

Все события происходят по времени. В игровом цикле я считаю колличство тикой и когда оно становится кратно определенной цифре, вызывается соответствующее событие.

***Поле боя****.* В определенный момент игроку предлагается поучаствовать в битве. Это произойдет только в том случае, если количество воинов у него больше двух. (Подробнее в battlefield\_rules.txt). Создается враг (класс Enemey), генерирующий почти случайное количество воинов: от 1 до количества, на один большего количества воинов у игрока. На поле боя изображения воинов анимированные. Так как удалось найти только одну подходящую картинку война, армия противника отличается от армии игрока мной лично нарисованными красными поясами. Каждый спрайт воина имеет маску. Когда два противника пересекаются, один ударяет другого (метод go(self, en)) со случайной силой (параметр жизни уменьшается на силу удара). Как только этот параметр достигает 0, соответствующий спрайт уничтожается, с помощью метода kill() и удаляется из списка воинов (в ресурсах у игрока или соответствующего списка в классе, задающем врага)

Когда развитие игры невозможно (все поле застроенно, свободных жителей нет, войны погибли или пользователь совершил действие, которое привело к минусовому бюжету), игра заканчивается, соотвествующее сообщение выводится на консоль и в том случае, если количество выигранных боев больше предыдущего рекорда (записан в файле), об этом сообщается и новый результат записывается в файл.

Всего в программе, задающей игру, реализованны 11 классов. Два из них (Building(pygame.sprite.Sprite)

И Hero(pygame.sprite.Sprite)) являются безовыми для основных игровых объектов: зданий и героев.

InfoButton(pygame.sprite.Sprite) – спрайт, выполняющий функцию кнопки. Отвечает за вывод информайии.

Collector(pygame.sprite.Sprite) показывает, когда здение приносит доход. При нажатии исчезает, а соответствующая сумма добавляется в ресурсы. ControlPanel показывет все ресурсы игрока в деревне и «жизни» воинов на поле боя.

Так же мной реализовыны функции перевода координат и проверки их на корректность для удобства работы.

**Технологии:**

* Чтение и запись в текстовый и csv файл
* Спрайты, пересечение спрайтов по маске
* Анимация спрайтов
* Часы
* Отлавливанние ошибок, вызванных некорректностью действий пользователя.

**Идеи усовершенствования:**

* Возможность при нажатии на ферму заменять денги на пропитание для войнов
* Возможность использовать фермеров в качестве воинов, когда не занятых жителей нет
* Усовершенствование начального экрана

**Заключение:**

Не смотря на достаточно большой объем кода, поект оказался довольно простым в реализации. В нем были использованны основы Pygame, задействована работа с файлами и существует. Клетчатое поле представляет из себя деревня, где каждое здание занимает свою клетку. На поле боя все взаимодействия происходят на уровне спрайтов для удобства реализации.