Обработка событий в pygame. Создание спрайтов и группы спрайтов в парадигме объектно-ориентированного программирования. Коллизия объектов.

1. Описать процесс гравитации (методы, классы).

В момент нажатия на пробел или «w» или стрелку вверх срабатывает метод jump класса Actor(), который проверяет, находится ли игрок на платформе, и, в случае нахождения, устанавливает вертикальную скорость (-7). В файле constants.py есть переменная GRAVITY = 0.15, задающий уровень гравитации. В файле base\_sprites.py в классе Character() идет присвоение значения атрибуту heavy. В этом же классе есть метод gravitate(), который увеличивает вертикальную скорость персонажа на величину heavy

1. Описать процесс движения стрелы (методы, классы).

Стрела считается таким же игровым персонажем (объектом класса Character() файла base\_sprites.py). В файле game\_sprites.py есть класс Fire(), в котором любой снаряд (в том числе и стрела) двигаются, как платформы. Скорость полета стрелы – 7, дистанция полета – 300 (прописано в файле constans.py)

В момент выстрела срабатывает метод fire() класса Actor(), который создает прямоугольник, помещает его с нужной стороны от стрелка (слева или справа) и создает объект класса Fire(), который в свою очередь создает объект класса Character()

1. Описать процесс коллизии объектов (методы, классы).

Коллизия описана в методе update() класса Character() файла base\_sprites.py. Более подробно каждый метод описан при помощи комментариев в файле