#2 - Создание игрока и его передвижение

<https://www.pygame.org/wiki/GettingStarted>

PyGame позволяет рисовать различные объекты на сцене окна приложения. В видео мы создадим главного игрока, создадим основной цикл, научимся отслеживать различные события от пользователя, а также научим персонажа передвигаться по сцене.

Библиотека PyGame содержит встроенный функционал для отслеживания действий пользователя, а также рисовать всевозможные объектов на сцене игры.

Для отслеживания действий (нажатие кнопки, закрытие приложения и так далее) необходимо создать главный цикл, который будет проверять все события в нашей игре. Такой цикл должен быть бесконечным, поэтому для него стоит использовать следующую структуру:

play = True

while play:

# Цикл выполняется пока переменная равна True

Когда мы захотим закрыть программу, то все что необходимо будет сделать, так это присвоить значение «False» к переменной play.

В цикле можно брать все происходящие события и проверять их. Когда будет нажата какая-либо из кнопок, то мы сможем это отследить и выполнить какой-либо кусочек кода.

Также цикл позволяет рисовать объекты на сцене. Для каждого объекта существует свой метод, например:

circle - создание круга;

rect - создание квадрата;

line - создание линии.

Помимо этих методов есть и другие, но эти являются наиболее популярными и часто используемыми.

В конце цикла всегда добавляйте обновление сцены игры, чтобы на ней постоянно были отрисованы новые кадры:

pygame.display.update()

Если запустить цикл как есть, то его выполнение будет слишком быстрым и играть будет невозможно. Чтобы такого не произошло стоит в конце цикла отправлять прогарммку в «сон», хотя бы на 0.1 секунды.

<https://itproger.com/course/pygame/2>

#3 - Прыжки и добавление границ

В уроке мы научим нашего персонажа прыгать. Прыжок мы сделаем похожим на настоящий, то есть с учетом притяжения объектов. Также мы добавим границы, чтобы игрок не выходил за пределы экрана и всегда оставался в поле зрения.

<https://itproger.com/course/pygame/3>

Парабола для прыжка:

480-50 = 430 /10

430 - 40,5 = 389,5 /9

389,5 - 32 = 357,5 /8

357,5 - 24,5 = 333 /7

333 - 18 = 315 /6

315 - 12,5 = 302,5 /5

302,5 - 8 = 294,5 /4

294,5 - 4,5 = 290 /3

290 - 2 = 288 /2

288 - 0,5 = 287,5 /1

287,5 - 0 = 287,5 /0

287,5 + 0,5 = 288 /-1

288 + 2 = 290 /-2

290 + 4,5 = 294,5 /-3

294,5 + 8 = 302,5 /-4

302,5 + 12,5 = 315 /-5

315 + 18 = 333 /-6

333 + 24,5 = 357,5 /-7

357,5 + 32 = 389,5 /-8

389,5 + 40,5 = 430 /-9

430 + 50 = 480 /-10

384x240

878x550

коэфф = 2,2

#4 - Анимация объектов и спрайты

В уроке мы добавим картинки в игру, а также поработаем над анимацией. Мы создадим персонажа, который будет анимировано передвигаться по сцене, а также добавим задний фон.

Для добавления изображений через библиотеку PyGame можно воспользоваться встроенными функциями. Одной из функций является функция «load», что позволяет подгружать спрайты (картинки) внутрь игры.

Для использования спрайтов необходимо добавить метод blit(). Он принимает два параметра:

1. Картинку для отображения;
2. Позиция в которую будет помещена картинка.

Для анимации картинок и создания эффекта передвижения можно использовать фреймы, которые срабатывают одинаковое количество раз за секунду. В нашем случае мы использовали значение 30 фреймов, что говорит о частоте обновления, равной 30 кадрам за секунду.

Для установки значения можно использовать класс pygame.time.Clock и его метод tick().

<https://itproger.com/course/pygame/4>

<https://www.gamedevmarket.net/> - [diana\_ust@mail.ru](mailto:diana_ust@mail.ru) - euflfqvtkjlb.8

#5 - Стрельба снарядами

В уроке мы добавим возможность стрелять снарядами для нашего персонажа. Игрок всегда будет стрелять не более чем 5 снарядами, а также он может это делать в разные стороны.

<https://itproger.com/course/pygame/5>