**Лабораторная работа 2. Рисование графических объектов на языке Python**

Для установки библиотеки следуйте инструкциям на pygame.org

Чтобы импортировать возможности библиотеки Pygame в вашей программе недостаточно одной инструкции import, нужны ещё некоторые дополнительные действия:

|  |
| --- |
| import pygame  # После импорта библиотеки, необходимо её инициализировать:  pygame.init()  # И создать окно:  screen = pygame.display.set\_mode((300, 200))  # здесь будут рисоваться фигуры  # ...  # после чего, чтобы они отобразились на экране, экран нужно обновить:  pygame.display.update()  # Эту же команду нужно будет повторять, если на экране происходят изменения.  # Наконец, нужно создать основной цикл, в котором будут отслеживаться  # происходящие события.  # Пока единственное событие, которое нас интересует - выход из программы.  while True:  for event in pygame.event.get():  if event.type == pygame.QUIT:  pygame.quit() |

Помимо команды import pygame для более удобного доступа к функциям рисования, можно дополнительно прописать ещё одну строку импорта:

|  |
| --- |
| import pygame  from pygame.draw import \* |

Это позволит вместо pygame.draw.rect(...) писать просто rect(...).

Также хорошей практикой является добавление небольшой задержки в главный цикл программы, чтобы не заставлять ее работать "вхолостую", постоянно считывая события, которых, скорее всего, нет. Для этого в pygame есть специальный модуль time. До начала главного цикла создаем объект Clock:

|  |
| --- |
| clock = pygame.time.Clock() |

После этого в главном цикле добавляем строку:

|  |
| --- |
| clock.tick(30) |

Здесь 30 – это максимальный FPS, быстрее которого программа работать не будет. Естественно, можно указать и любое другое значение (которое, кстати, есть смысл записать в отдельную переменную для легкого доступа).

Пример №1

Выведем простую картинку. Создайте файл 1\_draw.py, скопируйте туда текст примера №1 и запустите.

|  |
| --- |
| import pygame  from pygame.draw import \*  pygame.init()  FPS = 30  screen = pygame.display.set\_mode((400, 400))  rect(screen, (255, 0, 255), (100, 100, 200, 200))  rect(screen, (0, 0, 255), (100, 100, 200, 200), 5)  polygon(screen, (255, 255, 0), [(100,100), (200,50),  (300,100), (100,100)])  polygon(screen, (0, 0, 255), [(100,100), (200,50),  (300,100), (100,100)], 5)  circle(screen, (0, 255, 0), (200, 175), 50)  circle(screen, (255, 255, 255), (200, 175), 50, 5)  pygame.display.update()  clock = pygame.time.Clock()  finished = False  while not finished:  clock.tick(FPS)  for event in pygame.event.get():  if event.type == pygame.QUIT:  finished = True  pygame.quit() |

**Пример №2**

Для создания штриховок можно использовать циклы:

|  |
| --- |
| import pygame  from pygame.draw import \*  pygame.init()  FPS = 30  screen = pygame.display.set\_mode((400, 400))  x1 = 100; y1 = 100  x2 = 300; y2 = 200  N = 10  color = (255, 255, 255)  rect(screen, color, (x1, y1, x2 - x1, y2 - y1), 2)  h = (x2 - x1) // (N + 1)  x = x1 + h  for i in range(N):  line(screen, color, (x, y1), (x, y2))  x += h  pygame.display.update()  clock = pygame.time.Clock()  finished = False  while not finished:  clock.tick(FPS)  for event in pygame.event.get():  if event.type == pygame.QUIT:  finished = True  pygame.quit() |

Все функции модуля pygame.draw в качестве первого аргумента принимают экран, на котором нужно рисовать (приложение может открывать и несколько окон, нужно точно знать, на каком рисовать). Второй аргумент - цвет, заданный кортежем из трех чисел от 0 до 255 в формате RGB. Также возможно наличие четвертого числа - прозрачности. После этого следуют координаты фигуры (для каждой фигуры свой формат задания координат), далее - параметр width. Если передать в этот параметр положительное значение, оно будет означать толщину линии. Если параметр равен 0 (значение по умолчанию), фигура будет полностью закрашеной. Полное описание функций модуля pygame.draw вы можете найти в официальной документации.

**Задание на 2 лабораторную работу:**

2.1 Нарисуйте картинку. Задание индивидуальное. Картинка выдается преподавателем.