***Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования***

|  |  |
| --- | --- |
|  | ***«Московский государственный технический университет  имени Н.Э. Баумана»***  ***(МГТУ им. Н.Э. Баумана)*** |

**Факультет ИУ**

**Кафедра ИУ10**

**Отчет**

**по Лабораторной работе 3**

**Дисциплина: Информатика**

**Тема: Git и GitHub**

Работу выполнила: Быстрицкая Ксения Александровна ИУ10-23

Проверил: Буркацкий Кирилл Александрович

Москва, 2023

Оглавление

[**Отчет** 1](#_Toc130744977)

[**Цель работы:** 2](#_Toc130744978)

[**Ход работы** 2](#_Toc130744979)

[Задача №1 2](#_Toc130744980)

[*Ход решения задачи* 2](#_Toc130744981)

[Задача №2 3](#_Toc130744982)

[*Ход решения задачи* 3](#_Toc130744983)

[Задача №3 6](#_Toc130744984)

[*Ход решения задачи* 6](#_Toc130744985)

**Цель работы:** Нарисовать с помощью модуля pygame.draw различные картинки

# **Ход работы**

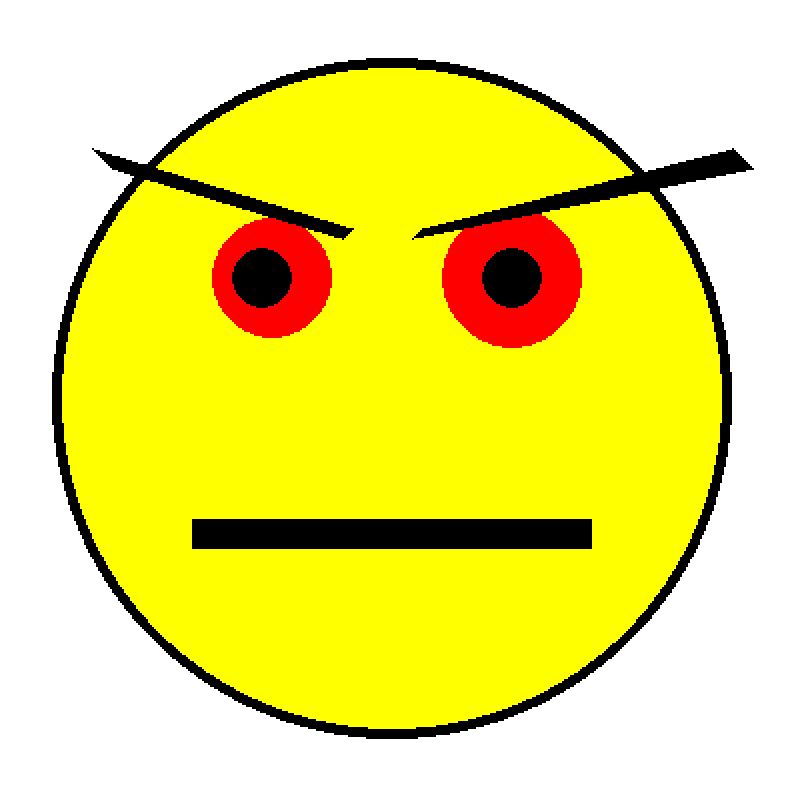
## Задача №1

*Цель:* Нарисовать злой смайлик

### *Ход решения задачи*

1. Написание программы
2. import pygame  
   from pygame.draw import \*  
     
   pygame.init()  
   a=1  
   FPS = 30  
   def smail():  
    screen = pygame.display.set\_mode((400, 400))  
    screen.fill((255, 255, 255))  
    circle(screen, (255, 255, 0), (200, 200), 170)  
    circle(screen, (0, 0, 0), (200, 200), 170, 5)  
    circle(screen, (255, 0, 0), (140, 140), 30)  
    circle(screen, (0, 0, 0), (135, 140), 15)  
    circle(screen, (255, 0, 0), (260, 140), 35)  
    circle(screen, (0, 0, 0), (260, 140), 15)  
    polygon(screen, (0, 0, 0), [(60, 85), (50, 75), (180, 115), (175, 120)])  
    polygon(screen, (0, 0, 0), [(210, 120), (215, 115), (370, 75), (380, 85)])  
    rect(screen, (0, 0, 0), (100, 260, 200, 15))  
     
     
     
     
   if a==1:  
    smail()  
     
   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

Тестирование



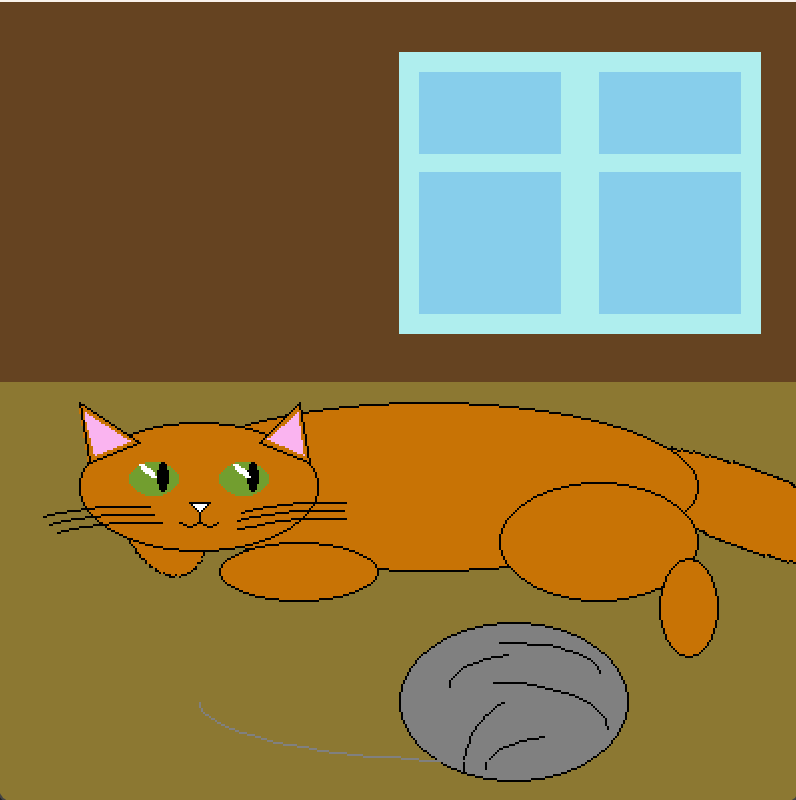
## Задача №2

*Цель:* Нарисуйте на экране картинку кошки

### *Ход решения задачи*

1. Написание программы
2. import pygame  
   from pygame.draw import \*  
   from math import \*  
   pygame.init()  
     
   FPS = 30  
   screen = pygame.display.set\_mode((400, 400))  
   screen.fill((101, 67, 33))  
     
   def draw\_ellipse\_angle(surface, color, rect, angle, width=0):  
    target\_rect = pygame.Rect(rect)  
    shape\_surf = pygame.Surface(target\_rect.size, pygame.SRCALPHA)  
    pygame.draw.ellipse(shape\_surf, color, (0, 0, \*target\_rect.size), width)  
    rotated\_surf = pygame.transform.rotate(shape\_surf, angle)  
    surface.blit(rotated\_surf, rotated\_surf.get\_rect(center = target\_rect.center))  
     
   # экран+окно  
   polygon(screen, (175, 238, 238), [(200, 25), (380, 25), (380, 165), (200, 165)])  
   polygon(screen, (135, 206, 235), [(210, 35), (280, 35), (280, 75), (210, 75)])  
   polygon(screen, (135, 206, 235), [(300, 35), (370, 35), (370, 75), (300, 75)])  
   polygon(screen, (135, 206, 235), [(300, 85), (370, 85), (370, 155), (300, 155)])  
   polygon(screen, (135, 206, 235), [(210, 85), (280, 85), (280, 155), (210, 155)])  
   polygon(screen, (140, 120, 50), [(0, 190), (400, 190), (400, 400), (0, 400)])  
     
   # передние лапы  
   draw\_ellipse\_angle(screen, (200, 115, 5), [46, 232, 71, 43], -70, 0)  
   draw\_ellipse\_angle(screen, (0, 0, 0), [46, 232, 71, 43], -70, 1)  
     
     
     
   # ноги и хвост  
   draw\_ellipse\_angle(screen, (200, 115, 5), [300, 233, 150, 40], -200, 0)  
   draw\_ellipse\_angle(screen, (0, 0, 0), [300, 233, 150, 40], -200, 1)  
     
     
     
   # кот  
   ellipse(screen, (200, 115, 5), (80, 200, 270, 85))  
   ellipse(screen, (0, 0, 0), (80, 200, 270, 85), 1)  
   ellipse(screen, (200, 115, 5), (40, 210, 120, 65))  
   ellipse(screen, (0, 0, 0), (40, 210, 120, 65), 1)  
     
     
   #уши  
   polygon(screen, (200, 115, 5), [(45, 230), (40, 200), (70, 220)])  
   polygon(screen, (0, 0, 0), [(45, 230), (40, 200), (70, 220)], 1)  
   polygon(screen, (250, 180, 240), [(48, 226), (43, 205), (65, 219)])  
   polygon(screen, (200, 115, 5),[(130, 220), (150, 200), (155, 230)])  
   polygon(screen, (0, 0, 0), [(130, 220), (150, 200), (155, 230)], 1)  
   polygon(screen, (250, 180, 240), [(134, 218), (149, 205), (152, 226)])  
     
   #глаза  
   ellipse(screen, (114, 158, 47), (65, 230, 25, 17))  
   ellipse(screen, (114, 158, 47), (110, 230, 25, 17))  
   ellipse(screen, (0, 0, 0), (124, 230, 6, 15))  
   ellipse(screen, (0, 0, 0), (79, 230, 6, 15))  
   draw\_ellipse\_angle(screen, (255, 255, 255), [69, 233, 10, 3], -220, 0)  
   draw\_ellipse\_angle(screen, (255, 255, 255), [116, 233, 10, 3], -220, 0)  
     
     
     
   #ноги задние  
   ellipse(screen, (200, 115, 5), (250, 240, 100, 60))  
   ellipse(screen, (0, 0, 0), (250, 240, 100, 60), 1)  
     
   ellipse(screen, (200, 115, 5), (330, 278, 30, 50))  
   ellipse(screen, (0, 0, 0), (330, 278, 30, 50), 1)  
     
   ellipse(screen, (200, 115, 5), (110, 270, 80, 30))  
   ellipse(screen, (0, 0, 0),(110, 270, 80, 30), 1)  
     
     
   #нос и усы  
   polygon(screen, (255, 255, 255), [(95, 250), (105, 250), (100, 255)])  
   polygon(screen, (0, 0, 0), [(95, 250), (105, 250), (100, 255)], 1)  
   line(screen,(0, 0, 0), [100, 255], [100, 260], 1)  
   arc(screen,(0, 0, 0), (100, 258, 10, 4), pi,15 \* pi / 8 )  
   arc(screen,(0, 0, 0), (90, 258, 10, 4), pi,15 \* pi / 8 )  
     
     
     
   # усы  
   arc(screen, (0, 0, 0), [20, 250, 300, 200], 1.55, 1.9)  
   arc(screen, (0, 0, 0), [17, 254, 305, 205], 1.55, 1.9)  
   arc(screen, (0, 0, 0), [15, 258, 310, 210], 1.55, 1.9)  
     
   arc(screen, (0, 0, 0), [-80, 252, 305, 200], 1.55, 1.9)  
   arc(screen, (0, 0, 0), [-76, 256, 300, 200], 1.55, 1.9)  
   arc(screen, (0, 0, 0), [-72, 260, 300, 200], 1.55, 1.9)  
     
     
   # клубок и нитки  
   ellipse(screen, (128, 128, 128), (200, 310, 115, 80))  
   ellipse(screen, (0, 0, 0), (200, 310, 115, 80), 1)  
   arc(screen, (0, 0, 0), [200, 320, 100, 30], 0, pi / 2)  
   arc(screen, (0, 0, 0), [190, 340, 115, 47], 0, pi / 2)  
   arc(screen, (0, 0, 0), [243, 365, 118, 36], 2 \* pi / 3, pi)  
   arc(screen, (0, 0, 0), [232, 345, 80, 80], 2 \* pi / 3, pi)  
   arc(screen, (0, 0, 0), [225, 324, 118, 36], 2 \* pi / 3, pi)  
     
   arc(screen, (128, 128, 128), (100 , 320, 300 , 60 ), pi, 3\*pi / 2)  
     
   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()

Тестирование



## Задача №3

*Цель:* улучшить программу и нарисовать много кошек.

### *Ход решения задачи*

1. Написание программы

Используем функции с параметром. Т. е. каждую координату умножим на величину i.

Используем два файла. В одном будут все функции, другой- будет их вызывать

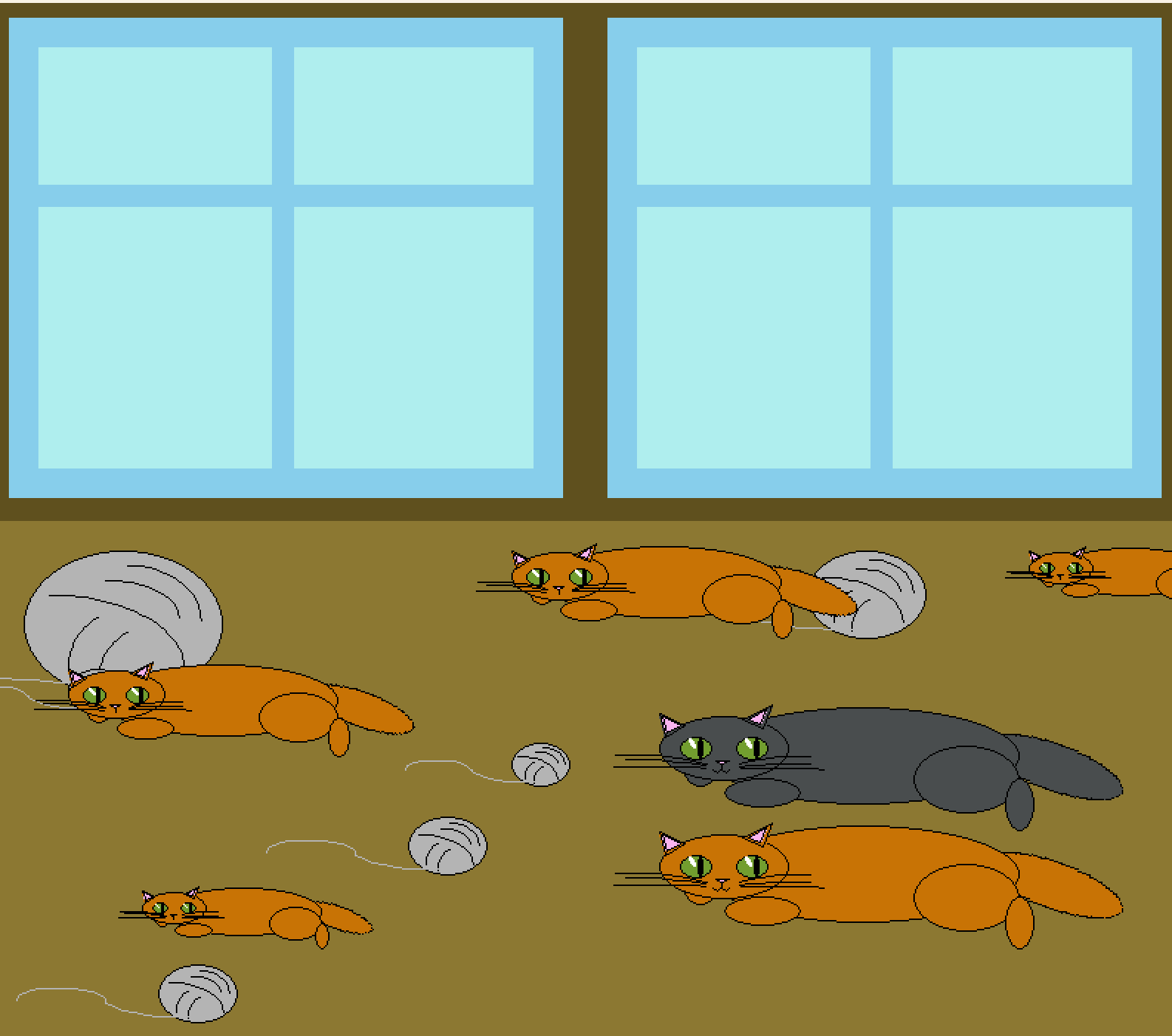
Файл mnogokoteika:

import pygame  
from pygame.draw import \*  
from math import \*  
pygame.init()  
  
  
FPS = 30  
  
def draw\_ellipse\_angle(surface, color, rect, angle, width=0):  
 target\_rect = pygame.Rect(rect)  
 shape\_surf = pygame.Surface(target\_rect.size, pygame.SRCALPHA)  
 pygame.draw.ellipse(shape\_surf, color, (0, 0, \*target\_rect.size), width)  
 rotated\_surf = pygame.transform.rotate(shape\_surf, angle)  
 surface.blit(rotated\_surf, rotated\_surf.get\_rect(center=target\_rect.center))  
  
  
def window(x, y, i, screen):  
 i = i / 10  
 x0 = 415  
 y0 = 10  
 x1 = - x0 \* i + x  
 y1 = - y0 \* i + y  
 rect(screen, (175, 238, 238), (415 \* i + x1, 10 \* i + y1, 375 \* i, 325 \* i))  
 rect(screen, (135, 206, 235), (415 \* i + x1, 10 \* i + y1, 375 \* i, 325 \* i), 20)  
 line(screen, (135, 206, 235), (415 \* i + x1, 130 \* i + y1), (785 \* i + x1, 130 \* i + y1), 15)  
 line(screen, (135, 206, 235), (600 \* i + x1, 10 \* i + y1), (600 \* i + x1, 325 \* i + y1), 15)  
  
  
def koteika(x, y, i, screen, color):  
 i = i / 10  
 x0 = 20  
 y0 = 370  
 x1 = - x0 \* i + x  
 y1 = - y0 \* i + y  
  
 draw\_ellipse\_angle(screen, color, [563 \* i + x1, 420 \* i + y1, 250 \* i, 80 \* i], -20, 0)  
 draw\_ellipse\_angle(screen, (0, 0, 0), [563 \* i + x1, 420 \* i + y1, 250 \* i, 80 \* i], -20, 1)  
  
 ellipse(screen, color, (100 \* i + x1, 360 \* i + y1, 530 \* i, 165 \* i))  
 ellipse(screen, (0, 0, 0), (100 \* i + x1, 360 \* i + y1, 530 \* i, 165 \* i), 1)  
  
 ellipse(screen, color, (60 \* i + x1, 430 \* i + y1, 60 \* i, 65 \* i))  
 ellipse(screen, (0, 0, 0), (60 \* i + x1, 430 \* i + y1, 60 \* i, 65 \* i), 1)  
  
 ellipse(screen, color, (20 \* i + x1, 375 \* i + y1, 220 \* i, 110 \* i))  
 ellipse(screen, (0, 0, 0), (20 \* i + x1, 375 \* i + y1, 220 \* i, 110 \* i), 1)  
  
 ellipse(screen, color, (130 \* i + x1, 480 \* i + y1, 130 \* i, 50 \* i))  
 ellipse(screen, (0, 0, 0), (130 \* i + x1, 480 \* i + y1, 130 \* i, 50 \* i), 1)  
  
 ellipse(screen, color, (450 \* i + x1, 425 \* i + y1, 180 \* i, 115 \* i))  
 ellipse(screen, (0, 0, 0), (450 \* i + x1, 425 \* i + y1, 180 \* i, 115 \* i), 1)  
  
 ellipse(screen, color, (607 \* i + x1, 480 \* i + y1, 50 \* i, 90 \* i))  
 ellipse(screen, (0, 0, 0), (607 \* i + x1, 480 \* i + y1, 50 \* i, 90 \* i), 1)  
  
 polygon(screen, color,[(x, y), (63 \* i + x1, 392 \* i + y1), (30 \* i + x1, 410 \* i + y1)])  
 polygon(screen, (0, 0, 0), [(x, y), (63 \* i + x1, 392 \* i + y1), (30 \* i + x1, 410 \* i + y1)], 1)  
  
 polygon(screen, (250, 180, 240),[(27 \* i + x1, 377 \* i + y1), (55 \* i + x1, 392 \* i + y1), (34 \* i + x1, 403 \* i + y1)])  
 polygon(screen, (0, 0, 0), [(27 \* i + x1, 377 \* i + y1), (55 \* i + x1, 392 \* i + y1), (34 \* i + x1, 403 \* i + y1)],1)  
  
 polygon(screen, color, [(210 \* i + x1, 355 \* i + y1), (197 \* i + x1, 395 \* i + y1),(160 \* i + x1, 382 \* i + y1)])  
 polygon(screen, (0, 0, 0),[(210 \* i + x1, 355 \* i + y1), (197 \* i + x1, 395 \* i + y1), (160 \* i + x1, 382 \* i + y1)], 1)  
  
 polygon(screen, (250, 180, 240), [(202 \* i + x1, 363 \* i + y1), (192 \* i + x1, 388 \* i + y1), (172 \* i + x1, 381 \* i + y1)])  
 polygon(screen, (0, 0, 0),  
 [(202 \* i + x1, 363 \* i + y1), (192 \* i + x1, 388 \* i + y1), (172 \* i + x1, 381 \* i + y1)], 1)  
  
 ellipse(screen, (114, 158, 47), (55 \* i + x1, 410 \* i + y1, 55 \* i, 40 \* i))  
 ellipse(screen, (0, 0, 0), (55 \* i + x1, 410 \* i + y1, 55 \* i, 40 \* i), 1)  
  
 ellipse(screen, (114, 158, 47), (150 \* i + x1, 410 \* i + y1, 55 \* i, 40 \* i))  
 ellipse(screen, (0, 0, 0), (150 \* i + x1, 410 \* i + y1, 55 \* i, 40 \* i), 1)  
  
 ellipse(screen, (0, 0, 0), (85 \* i + x1, 410 \* i + y1, 10 \* i, 40 \* i))  
  
 ellipse(screen, (0, 0, 0), (180 \* i + x1, 410 \* i + y1, 10 \* i, 40 \* i))  
  
 draw\_ellipse\_angle(screen, (255, 255, 255), [73 \* i + x1, 410 \* i + y1, 8 \* i, 20 \* i], 220, 0)  
  
 draw\_ellipse\_angle(screen, (255, 255, 255), [168 \* i + x1, 410 \* i + y1, 8 \* i, 20 \* i], 220, 0)  
  
 polygon(screen, (250, 180, 240),[(127 \* i + x1, 450 \* i + y1), (139 \* i + x1, 450 \* i + y1), (127 \* i + x1, 457 \* i + y1),(115 \* i + x1, 450 \* i + y1)]) # нос  
 polygon(screen, (0, 0, 0),[(127 \* i + x1, 450 \* i + y1), (139 \* i + x1, 450 \* i + y1), (127 \* i + x1, 457 \* i + y1),(115 \* i + x1, 450 \* i + y1)], 1)  
  
 line(screen, (0, 0, 0), (127 \* i + x1, 457 \* i + y1), (127 \* i + x1, 467 \* i + y1), 1)  
  
 arc(screen, (0, 0, 0), [127 \* i + x1, 462 \* i + y1, 16 \* i, 10 \* i], pi, 15 \* pi / 8)  
 arc(screen, (0, 0, 0), [111 \* i + x1, 462 \* i + y1, 16 \* i, 10 \* i], pi, 15 \* pi / 8)  
  
 # усы  
 arc(screen, (0, 0, 0), [-80 \* i + x1, 444 \* i + y1, 700 \* i, 500 \* i], 1.55, 1.9)  
 arc(screen, (0, 0, 0), [-78 \* i + x1, 455 \* i + y1, 700 \* i, 200 \* i], 1.55, 1.9)  
 arc(screen, (0, 0, 0), [-145 \* i + x1, 463 \* i + y1, 700 \* i, 200 \* i], 1.3, 1.7)  
  
 arc(screen, (0, 0, 0), [-375 \* i + x1, 440 \* i + y1, 700 \* i, 500 \* i], 1.2, 1.65)  
 arc(screen, (0, 0, 0), [-410 \* i + x1, 448 \* i + y1, 700 \* i, 200 \* i], 1.1, 1.5)  
 arc(screen, (0, 0, 0), [-345 \* i + x1, 460 \* i + y1, 700 \* i, 200 \* i], 1.3, 1.75)  
  
  
def ball(x, y, i, screen):  
 i = i / 10  
 x0 = 400  
 y0 = 560  
 x1 = - x0 \* i + x  
 y1 = - y0 \* i + y  
  
 ellipse(screen, (180, 180, 180), (400 \* i + x1, 560 \* i + y1, 135 \* i, 100 \* i))  
 ellipse(screen, (0, 0, 0), (400 \* i + x1, 560 \* i + y1, 135 \* i, 100 \* i), 1)  
 arc(screen, (0, 0, 0), (420 \* i + x1, 570 \* i + y1, 100 \* i, 75 \* i), 0, pi / 2)  
 arc(screen, (0, 0, 0), (405 \* i + x1, 580 \* i + y1, 100 \* i, 50 \* i), 0, pi / 2)  
 arc(screen, (0, 0, 0), (325 \* i + x1, 590 \* i + y1, 185 \* i, 100 \* i), 0, pi / 2)  
 arc(screen, (0, 0, 0), (430 \* i + x1, 600 \* i + y1, 80 \* i, 80 \* i), 2 \* pi / 3, pi)  
 arc(screen, (0, 0, 0), (450 \* i + x1, 610 \* i + y1, 80 \* i, 80 \* i), 2 \* pi / 3, pi)  
  
 arc(screen, (180, 180, 180), (310 \* i + x1, 590 \* i + y1, 300 \* i, 60 \* i), pi, 3 \* pi / 2)  
 arc(screen, (180, 180, 180), (160 \* i + x1, 600 \* i + y1, 150 \* i, 40 \* i), 0, pi)  
  
  
  
  
def mnogokoteik():  
 screen = pygame.display.set\_mode((800, 700))  
 screen.fill((95, 80, 30))  
 rect(screen, (140, 120, 50), (0, 350, 800, 350))  
 window(415, 10, 10, screen)  
 window(10, 10, 10, screen)  
 ball(20, 370, 10, screen)  
 ball(50, 450, 3, screen)  
 ball(280, 550, 4, screen)  
 ball(111, 650, 4, screen)  
 ball(350, 500, 3, screen)  
 ball(550, 370, 6, screen)  
 koteika(450, 480, 4, screen, (73, 77, 78))  
 koteika(50, 450, 3, screen, (200, 115, 5))  
 koteika(100, 600, 2, screen, (200, 115, 5))  
 koteika(350, 370, 3, screen, (200, 115, 5))  
 koteika(700, 370, 2, screen, (200, 115, 5))  
 koteika(450, 560, 4, screen, (200, 115, 5))  
  
def pygame\_done():  
 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()

Файл lala, вывязывающий mnogokoteika

import pygame  
from pygame.draw import \*  
from mnogokoteik import \*  
pygame.init()  
FPS = 30  
mnogokoteik()  
pygame\_done()

Тестирование



Вывод: Мы написали программы, использующие модуль pygame, рисующие различные картинки.