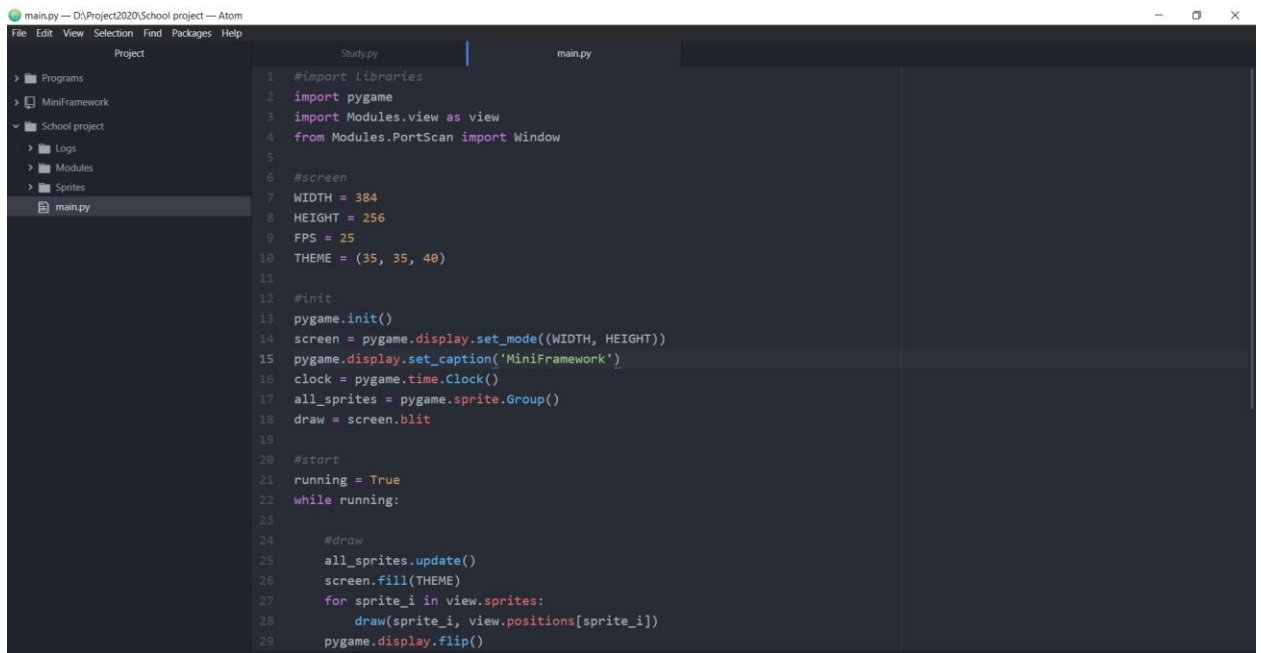


Сборник советов по написанию игр и приложений на Python

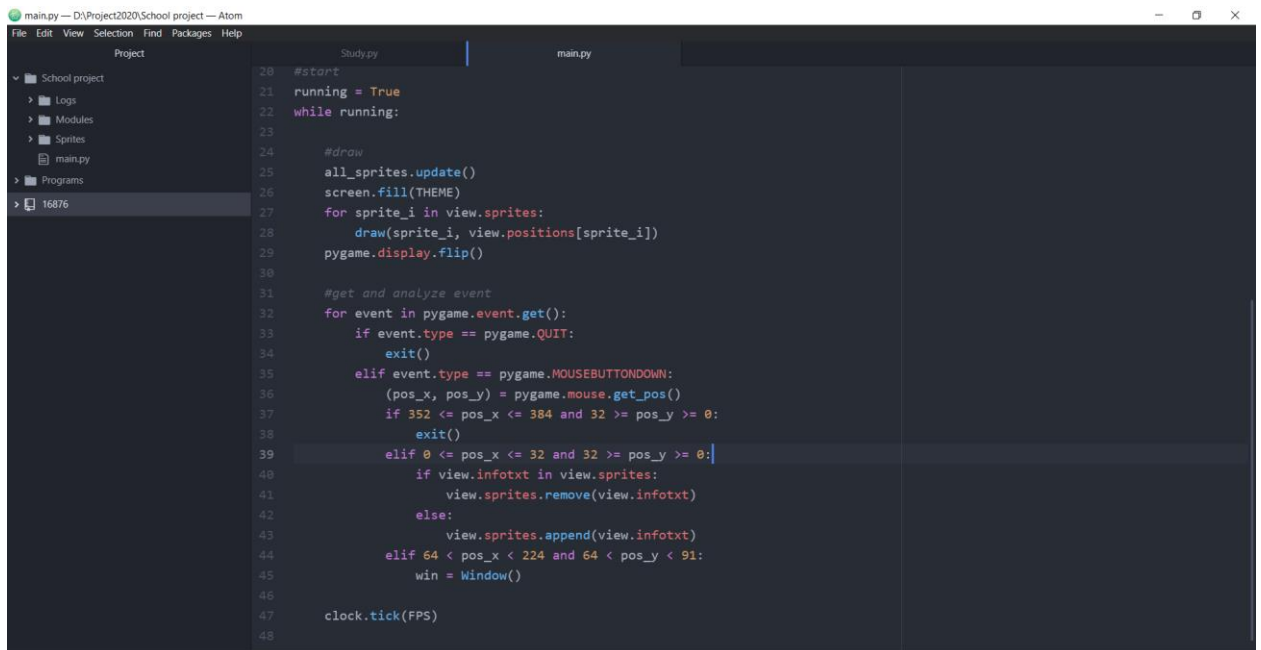
Весь код (и этот сборник) можно найти по ссылке:

<https://github.com/RuyCode/16876>

PyGame:



```
1 #import libraries
2 import pygame
3 import Modules.view as view
4 from Modules.PortScan import Window
5
6 #screen
7 WIDTH = 384
8 HEIGHT = 256
9 FPS = 25
10 THEME = (35, 35, 40)
11
12 #init
13 pygame.init()
14 screen = pygame.display.set_mode((WIDTH, HEIGHT))
15 pygame.display.set_caption('MiniFramework')
16 clock = pygame.time.Clock()
17 all_sprites = pygame.sprite.Group()
18 draw = screen.blit
19
20 #start
21 running = True
22 while running:
23
24     #draw
25     all_sprites.update()
26     screen.fill(THEME)
27     for sprite_i in view.sprites:
28         draw(sprite_i, view.positions[sprite_i])
29     pygame.display.flip()
```

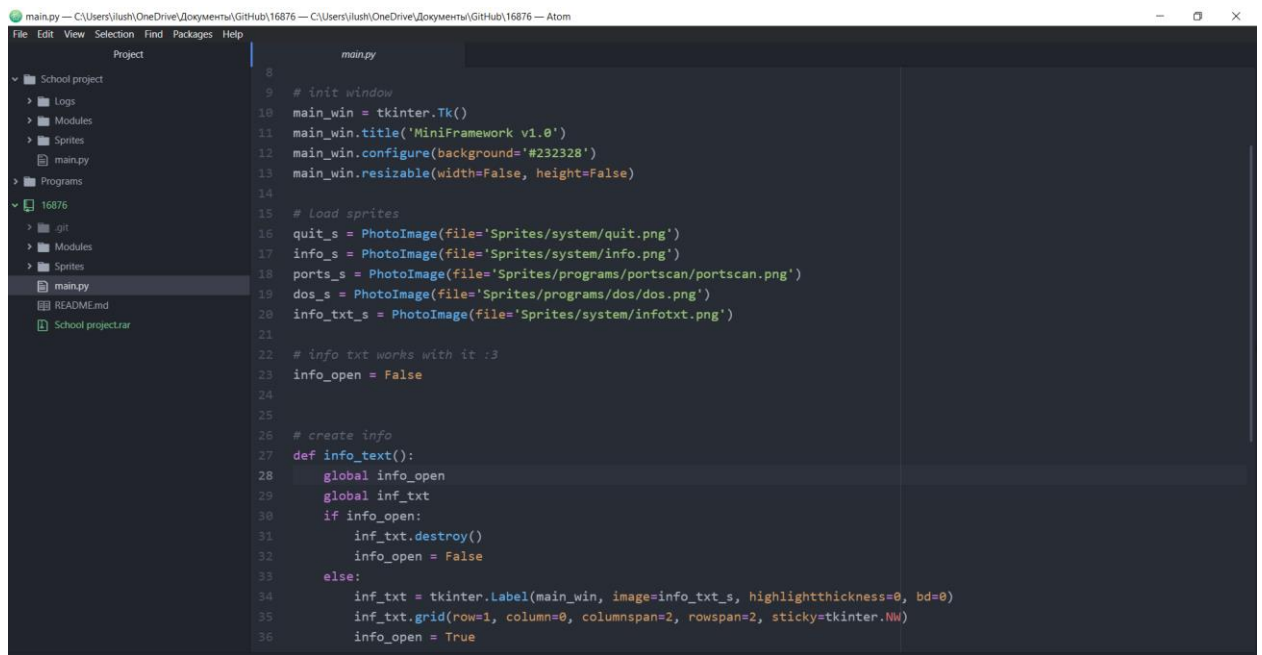


```
20 #start
21 running = True
22 while running:
23
24     #draw
25     all_sprites.update()
26     screen.fill(THEME)
27     for sprite_i in view.sprites:
28         draw(sprite_i, view.positions[sprite_i])
29     pygame.display.flip()
30
31     #get and analyze event
32     for event in pygame.event.get():
33         if event.type == pygame.QUIT:
34             exit()
35         elif event.type == pygame.MOUSEBUTTONDOWN:
36             (pos_x, pos_y) = pygame.mouse.get_pos()
37             if 352 <= pos_x <= 384 and 32 >= pos_y >= 0:
38                 exit()
39             elif 0 <= pos_x <= 32 and 32 >= pos_y >= 0:
40                 if view.infotxt in view.sprites:
41                     view.sprites.remove(view.infotxt)
42                 else:
43                     view.sprites.append(view.infotxt)
44             elif 64 < pos_x < 224 and 64 < pos_y < 91:
45                 win = Window()
46
47     clock.tick(FPS)
48
```

Для создания окна необходимо сначала запустить pygame командой “pygame.init()”. Окну можно установить размеры

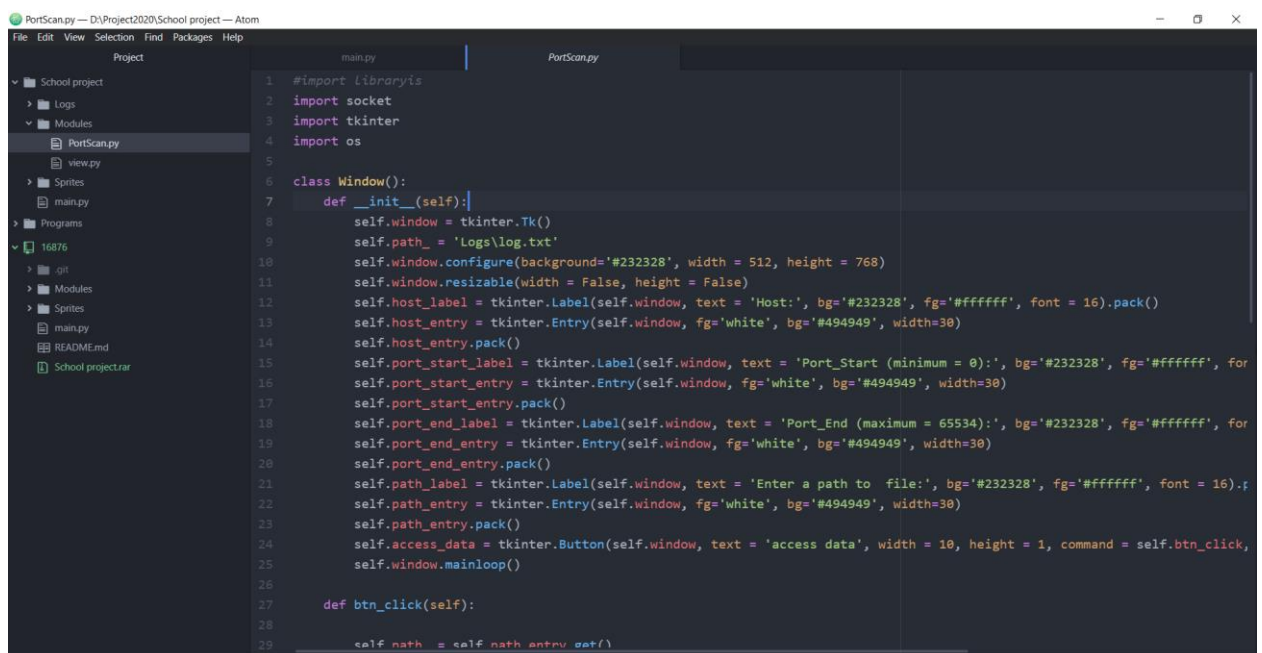
“pygame.display.set_mode((WIDTH, HEIGHT))”, название
“pygame.display.set_caption(NAME)”. “pygame.sprite.Group().update()” для
обновления всех спрайтов. “Screen.fill(THEME)” – заполняет все окно
указанным цветом в текущем кадре. “screen.blit(sprite_i,
view.positions[sprite_i])” отрисовывает спрайт на экране каждый кадр.
“pygame.display.flip()” обновляет окно, если этого не делать то новые кадры не
будут появляться. Далее мы получаем текущие события и в зависимости от его
типа, совершаем определенное действие (механизм работы можно видеть на
скриншотах сверху).

Tkinter:



```
main.py — C:\Users\Iush\OneDrive\Документы\GitHub\16876 — C:\Users\Iush\OneDrive\Документы\GitHub\16876 — Atom
File Edit View Selection Find Packages Help
Project
  School project
    > Logs
    > Modules
    > Sprites
    > main.py
  Programs
  16876
    > .git
    > Modules
    > Sprites
    > main.py
    README.md
    School project.rar

main.py
8
9 # init window
10 main_win = tkinter.Tk()
11 main_win.title('MiniFramework v1.0')
12 main_win.configure(background='#232328')
13 main_win.resizable(width=False, height=False)
14
15 # Load sprites
16 quit_s = PhotoImage(file='Sprites/system/quit.png')
17 info_s = PhotoImage(file='Sprites/system/info.png')
18 ports_s = PhotoImage(file='Sprites/programs/portscan/portscan.png')
19 dos_s = PhotoImage(file='Sprites/programs/dos/dos.png')
20 info_txt_s = PhotoImage(file='Sprites/system/infotxt.png')
21
22 # info txt works with it :3
23 info_open = False
24
25
26 # create info
27 def info_text():
28     global info_open
29     global inf_txt
30     if info_open:
31         inf_txt.destroy()
32         info_open = False
33     else:
34         inf_txt = tkinter.Label(main_win, image=info_txt_s, highlightthickness=0, bd=0)
35         inf_txt.grid(row=1, column=0, columnspan=2, rowspan=2, sticky=tkinter.NW)
36         info_open = True
```



```
PortScan.py — D:\Project2020\School project — Atom
File Edit View Selection Find Packages Help
Project
  School project
    > Logs
    > Modules
    > PortScan.py
    > view.py
  Programs
  16876
    > .git
    > Modules
    > Sprites
    > main.py
    README.md
    School project.rar

main.py
1 #import librarys
2 import socket
3 import tkinter
4 import os
5
6 class Window():
7     def __init__(self):
8         self.window = tkinter.Tk()
9         self.path = 'Logs\log.txt'
10        self.window.configure(background='#232328', width = 512, height = 768)
11        self.window.resizable(width = False, height = False)
12        self.host_label = tkinter.Label(self.window, text = 'Host:', bg='#232328', fg='ffffff', font = 16).pack()
13        self.host_entry = tkinter.Entry(self.window, fg='white', bg='#494949', width=30)
14        self.host_entry.pack()
15        self.port_start_label = tkinter.Label(self.window, text = 'Port_Start (minimum = 0):', bg='#232328', fg='ffffff', for
16        self.port_start_entry = tkinter.Entry(self.window, fg='white', bg='#494949', width=30)
17        self.port_start_entry.pack()
18        self.port_end_label = tkinter.Label(self.window, text = 'Port_End (maximum = 65534):', bg='#232328', fg='ffffff', for
19        self.port_end_entry = tkinter.Entry(self.window, fg='white', bg='#494949', width=30)
20        self.port_end_entry.pack()
21        self.path_label = tkinter.Label(self.window, text = 'Enter a path to file:', bg='#232328', fg='ffffff', font = 16).;
22        self.path_entry = tkinter.Entry(self.window, fg='white', bg='#494949', width=30)
23        self.path_entry.pack()
24        self.access_data = tkinter.Button(self.window, text = 'access data', width = 10, height = 1, command = self.btn_click,
25        self.window.mainloop()
26
27 def btn_click(self):
28
29     self.path = self.path_entry.get()
```

Сначала мы объявляем главное окно “`main_win = tkinter.Tk()`”. И задаем ему название “`main_win.title('MiniFramework v1.0')`”, а также устанавливаем цвет фона “`main_win.configure(background='#232328')`”. Можно запретить изменять размер окна по ширине и высоте командой “`main_win.resizable(width=False, height=False)`”. Загрузка спрайта (к примеру спрайта “`quit_s`”) осуществляется командой “`quit_s = PhotoImage(file='Sprites/system/quit.png')`”. Для удаления окна или компонента используется команда “`main_win.destroy()`”. В Tkinter

есть несколько видов «основных ячеек» (строка с примером из скриншота выше указана после определения):

- Label – текст или картинка (12),
- Entry – ввод текста (13),
- Button – кнопка (24), в значение `command` пишется функция, исполняемая при нажатии.

Чтобы разместить элемент можно воспользоваться методом `“pack()”` – тогда он разместится в свободном месте. Или `“grid()”`, но в данном случае создается сетка, а координаты указываются от левого верхнего угла ячейки в сетке. Для получения значения из Entry используется метод `“get()”`.