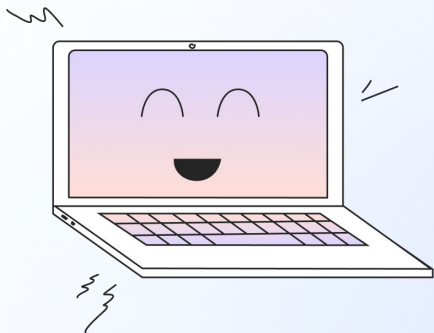


Программирование на Python

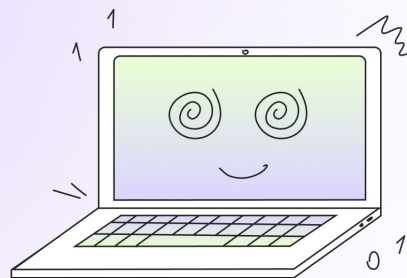
Урок 14

Групповая разработка. Создание классов. Игрок





**Ставь + в чат,
если хорошо видно и слышно**



Давайте вспомним предыдущий урок =)

Переходим на сайт с викториной по **ссылке**,
которую отправит преподаватель





Что будет на уроке сегодня?

- Создадим новый проект
- Добавим общий шаблон игры
- Разделимся на команды
- Каждая команда напишет свои классы



**Создаем
и настраиваем
новый проект**

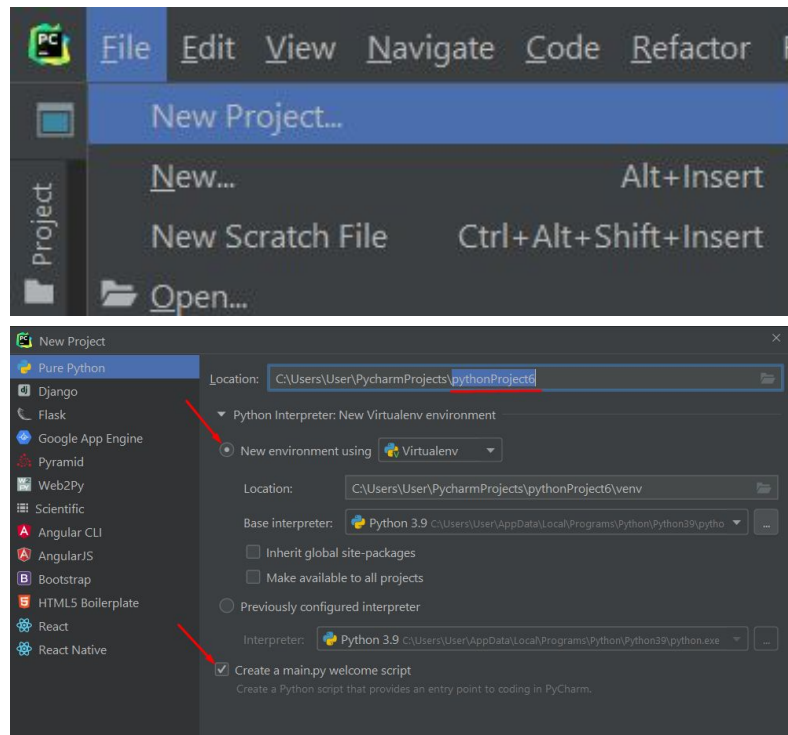




Создаем новый проект

Для новой игры нам потребуется новый проект. Создаем его выбрав пункт **File – New Project**.

Убеждаемся, что выбраны все необходимые пункты. Можно также переименовать его любым именем.





Создаем новый проект

После создания проекта, необходимо сразу установить модуль `pygame` Точно таким же способом как и в предыдущем проекте

```
pip install pygame
```

```
Terminal: Local x Local (2) x +
Microsoft Windows [Version 10.0.19042.867]
(c) Корпорация Майкрософт (Microsoft Corporation), 2019. Все права защищены.

(venv) D:\Projects\Python\LongPython>

(venv) D:\Projects\Python\LongPython>pip install pygame
Collecting pygame
  Downloading pygame-2.0.1-cp39-cp39-win_amd64.whl (5.2 MB)
    |██████████████████████████████████████████████████████████████████████████████| 5.2 MB 1.6 MB/s
Installing collected packages: pygame
Successfully installed pygame-2.0.1

(venv) D:\Projects\Python\LongPython>
```



Создаем новый проект

Дальше внутри файла `main.py` удаляем весь стандартный код и вставляем на его место код из шаблона.

Скопировать его можно по ссылке ниже:

<https://gist.githubusercontent.com/ronmount/90a4e651dd41955687979503ad026395/raw/eb92103985c837fe9662455ec2568cac38f7ba6f/%25D1%2588%25D0%25B0%25D0%25B1%25D0%25BB%25D0%25BE%25D0%25BD%2520%25D0%25B8%25D0%25B3%25D1%2580%25D1%258B>

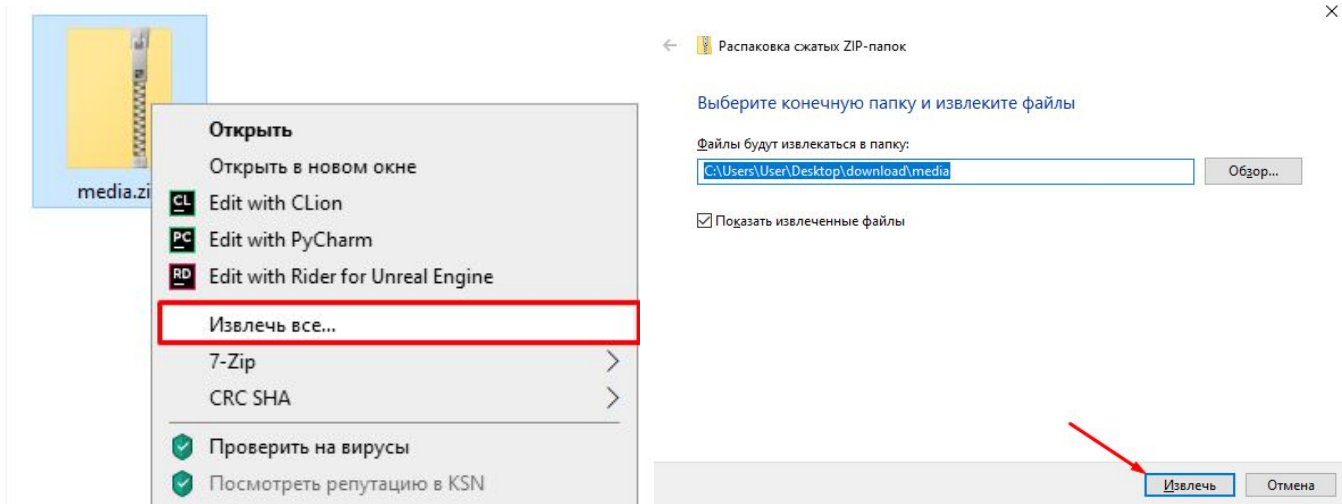


Добавляем ресурсы в проект

Необходимо скачать архив с ресурсами по ссылке:

https://drive.google.com/file/d/1DBsN2oS_gAct1iX0pM-AcSx-KqQVLBFC/view

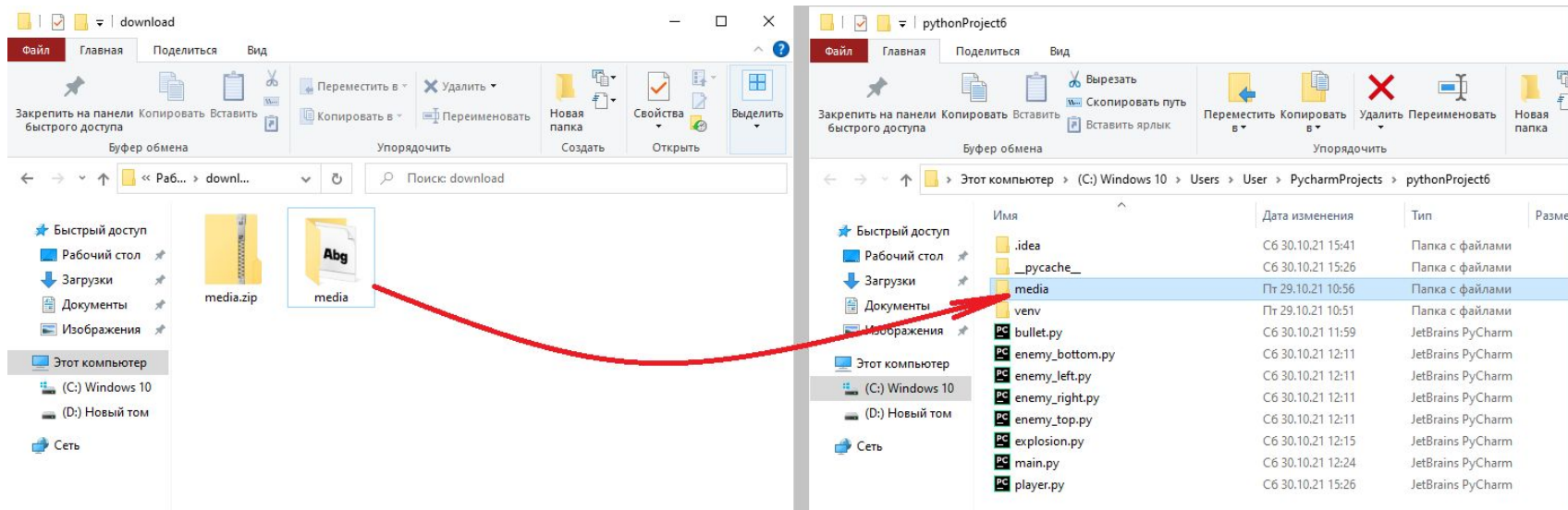
Распаковываем наш архив:





Добавляем ресурсы в проект

Переносим распакованную папку в проект с игрой





Добавляем ресурсы в проект

Теперь необходимо создать переменные, в которых будут храниться данные пути.

Создаем их до игрового цикла:

```
snd_dir = 'media/snd/'           # Путь до папки со звуками
img_dir = 'media/img/'          # Путь до папки со спрайтами
```

Протестируем наши пути, изменив иконку игры:

```
icon = pygame.image.load(img_dir + 'icon.png')  # Загружаем файл с иконкой
pygame.display.set_icon(icon)                   # Устанавливаем иконку в окно
```

 Racing





Создаем группу спрайтов

Спрайтов в игре может быть очень много, то управлять ими со временем становится не удобно. Чтобы решить эту проблему, разработчики решили объединить все спрайты в одну группу. И давать команды не каждому спрайту, а сразу целым группам спрайтов. Создадим общую группу спрайтов. Сделать это нужно до игрового цикла:

```
all_sprites = pygame.sprite.Group() # Создаем группу для спрайтов
```

Затем запустим выполнение действий у всех спрайтов:

```
while run: # Начинаем бесконечный цикл  
    timer.tick(fps) # Контроль времени (обновление игры)  
    all_sprites.update() # Выполняем действия всех спрайтов в группе
```



Создаем группу спрайтов

Затем рисуем сразу все спрайты в группе. Добавим команду после заливки экрана:

```
screen.fill(GREEN)           # Заливка заднего фона
all_sprites.draw(screen)     # Отрисовываем все спрайты
pygame.display.update()     # Обновляем экран
```

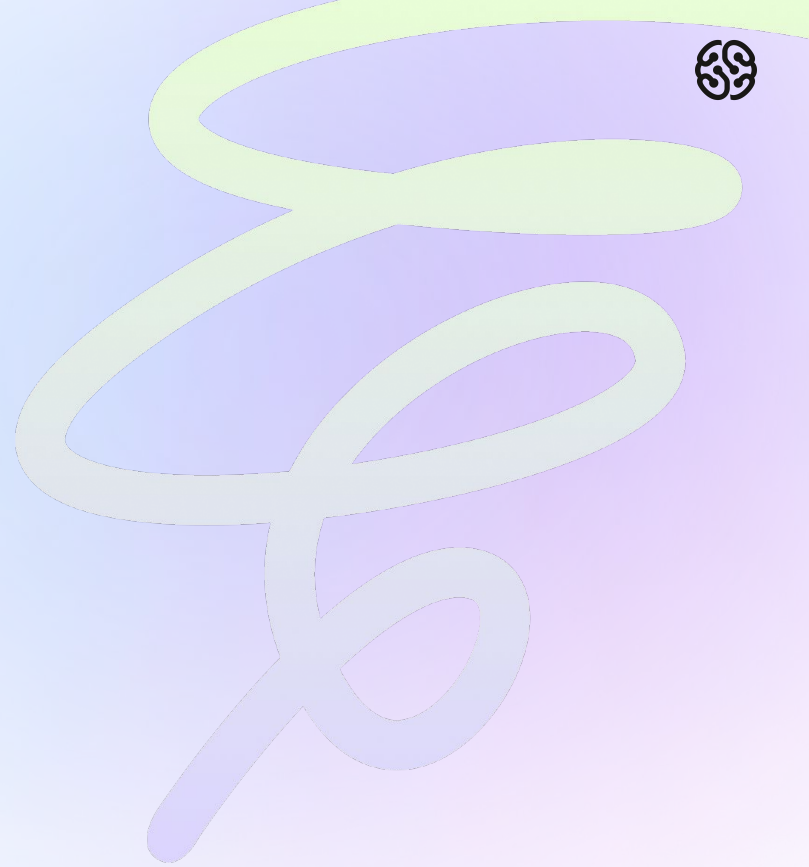


Перерыв

10 мин



**Добавляем
класс игрока**



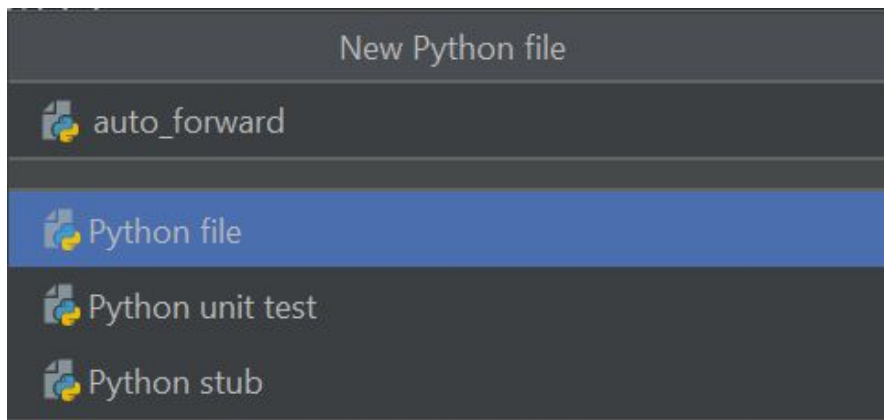
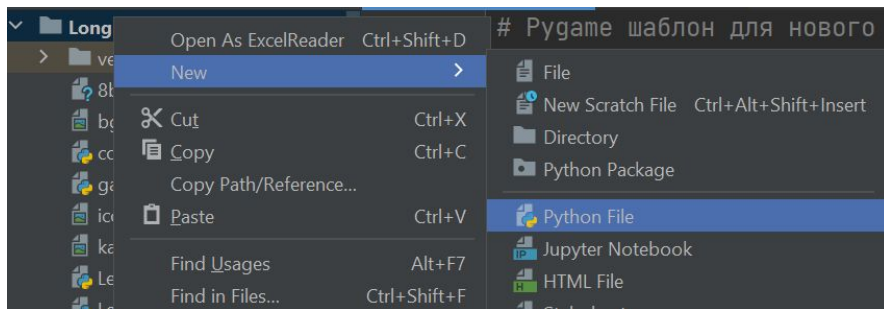


Создаем класс игрока

Создадим отдельный файл в нашем проекте. Кликнем правой кнопкой мыши по названию проекта в дереве и выберем пункты

New — Python File.

Назовем наш файл **player.**





Создаем класс игрока

Добавим в файл строчки, которые будут у всех классов:

```
import pygame
width = 1200           # ширина игрового окна
height = 600          # высота игрового окна
img_dir = 'media/img/' # папка с картинками
snd_dir = 'media/snd/' # папка со звуками
```

Ниже создадим сам класс и сразу напишем строчки, которые нужны всем классам:

```
class Player(pygame.sprite.Sprite):
    def __init__(self):           # Функция, где указываем что будет у игрока
        pygame.sprite.Sprite.__init__(self)
```



Создаем класс игрока

Самостоятельное задание:

1. Загрузите картинку игрока
2. Получите рамку игрока
3. Перенесите вашего игрока в центр экрана по горизонтали и вниз по вертикали
4. Добавьте игрока в группу ко всем спрайтам в основном файле.
Убедитесь, что он появился в нужном месте.

Застряли? Посмотрите подсказку в [9-ой презентации](#)

Добавляем класс спидометра



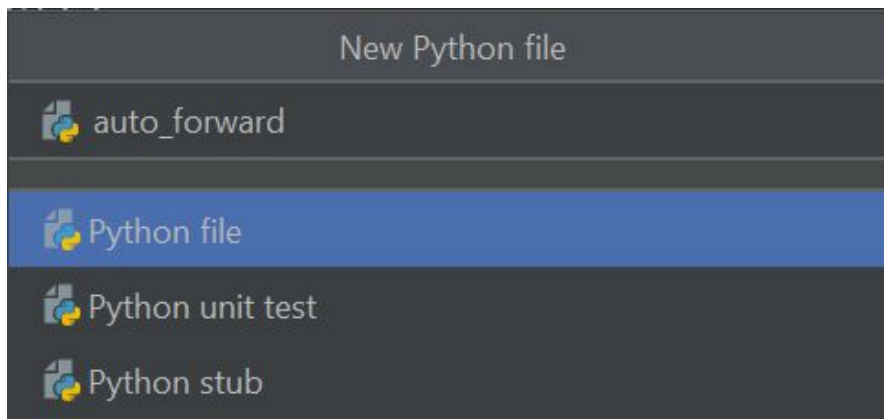
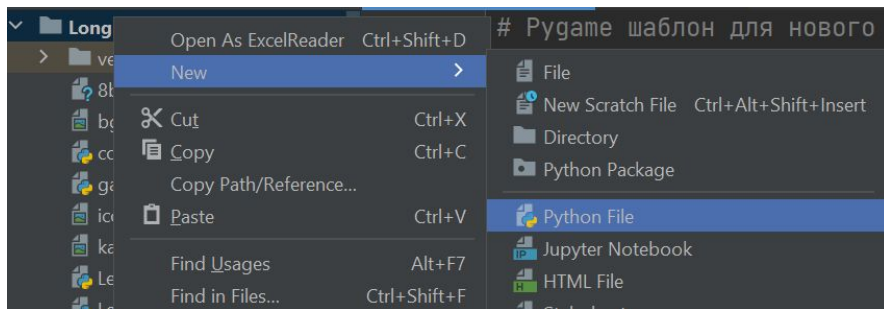


Создаем класс спидометра

Создадим отдельный файл в нашем проекте. Кликнем правой кнопкой мыши по названию проекта в дереве и выберем пункты

New — Python File.

Назовем наш файл **speedometr**.





Создаем класс спидометра

Добавим в файл строки, которые будут у всех классов:

```
import pygame
width = 1200           # ширина игрового окна
height = 600          # высота игрового окна
img_dir = 'media/img/' # папка с картинками
snd_dir = 'media/snd/' # папка со звуками
```

Ниже создадим сам класс и сразу напишем строки, которые нужны всем классам:

```
class Speedometr(pygame.sprite.Sprite):
    def __init__(self):           # Функция, где указываем что будет у спрайта
        pygame.sprite.Sprite.__init__(self)
```



Создаем класс спидометра

Самостоятельное задание:

1. Загрузите картинку спидометра
2. Получите рамку спидометра
3. Перенесите спидометр в нижний левый угол используя переменные `width` и `height`.
4. Добавьте спидометр в группу ко всем спрайтам в основном файле.
Убедитесь, что он появился в нужном месте

Застряли? Посмотрите подсказку в [9-ой презентации](#)

**Добавляем
класс стрелки**

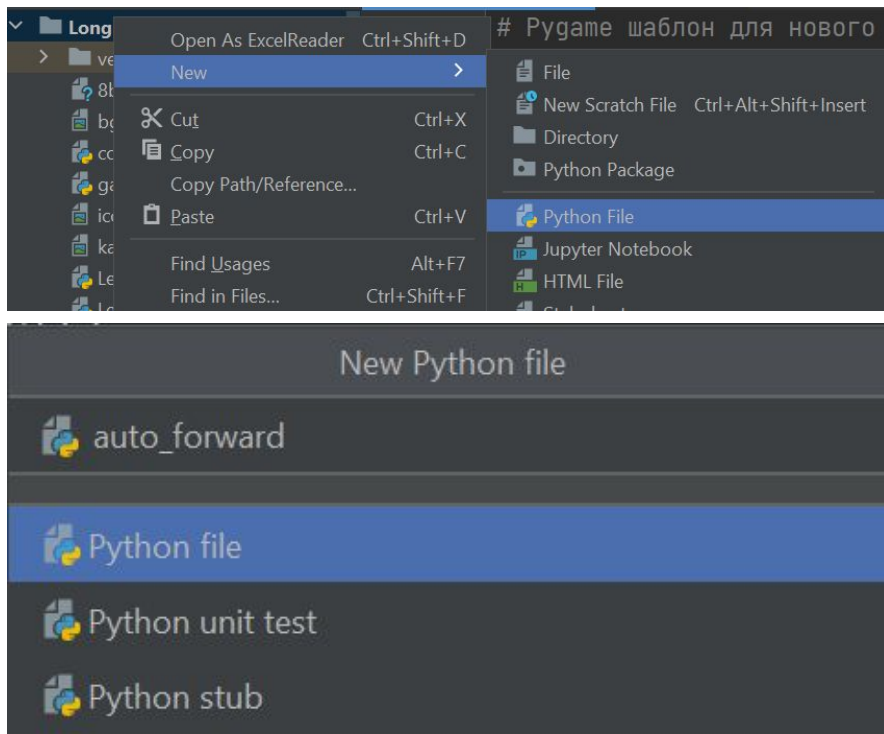




Создаем класс стрелки

Создадим отдельный файл в нашем проекте. Кликнем правой кнопкой мыши по названию проекта в дереве и выберем пункты **New — Python File**.

Назовем наш файл **arrow**.





Создаем класс стрелки

Добавим в файл строчки, которые будут у всех классов:

```
import pygame
width = 1200           # ширина игрового окна
height = 600          # высота игрового окна
img_dir = 'media/img/' # папка с картинками
snd_dir = 'media/snd/' # папка со звуками
```

Ниже создадим сам класс и сразу напишем строчки, которые нужны всем классам:

```
class Arrow(pygame.sprite.Sprite):
    def __init__(self):           # Функция, где указываем что будет у спрайта
        pygame.sprite.Sprite.__init__(self)
```



Создаем класс спидометра

Самостоятельное задание:

1. Загрузите картинку стрелки
2. Получите рамку стрелки
3. Перенесите стрелку в нижний левый угол используя переменные width и height.
4. Добавьте стрелку в группу ко всем спрайтам в основном файле.
Убедитесь, что она появилась в нужном месте.

Застряли? Посмотрите подсказку в [9-ой презентации](#)



Результат

Вы сами творите свой результат :)





Итоги

- Создали новый проект
- Добавили шаблонный код игры
- Создали классы игрока, спидометра и стрелки





На следующем занятии:

- Добавим свойства скорости тем классам, где они необходимы
- Настроим перемещение игрока
- Настроим поворот стрелки спидометра в зависимости от развитой скорости
- Ограничим движение игрока и повороты стрелки





Домашнее задание





Ваши вопросы





**Спасибо
за внимание**

