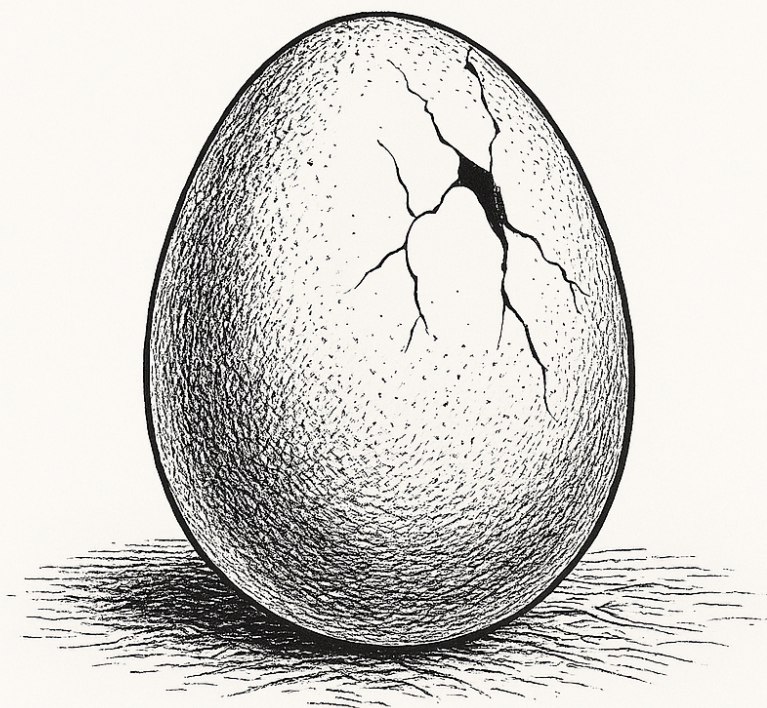


PYTHON

# PyGame

легко и просто



автор:  
Сергей Самойлов

# Введение в PyGame --- Урок 1

## Слайд 1: Заголовок

### **Введение в PyGame**

Создание простой игры с карточками

Целевая аудитория: Подростки 12-14 лет

## Слайд 2: Что такое PyGame?

- Библиотека для создания компьютерных игр на Python.
- Позволяет рисовать графику, воспроизводить звук и обрабатывать события (клавиши, мышь).
- Устанавливается через команду: `pip install pygame`

---

PyGame — это библиотека, которая помогает нам создавать простые игры на Python. Она позволяет рисовать графику, воспроизводить звуки и реагировать на действия игрока, такие как нажатия клавиш или движения мыши. Чтобы начать работать с PyGame, нужно установить её с помощью команды: `pip install pygame`.

## Слайд 3: Основные понятия PyGame

- **Окно игры** — область, где отображается наша игра.
- **Цикл игры** — главный цикл, который управляет игрой.
- **События** — действия пользователя (нажатия клавиш, закрытие окна и т.д.).
- **Объекты** — элементы игры (карточки, текст, кнопки).

---

Для создания игры в PyGame нам нужно понимать несколько ключевых понятий: - **Окно игры** — Это место, где мы будем рисовать все элементы нашей игры. - **Цикл игры** — Главный цикл, который повторяется снова и снова, пока игра не закончится. - **События** — Это действия игрока, такие как нажатие клавиш или закрытие окна. - **Объекты** — Всё, что отображается на экране: карточки, текст и другие элементы.

## Слайд 4: Создание окна

```
import pygame

pygame.init()
mw = pygame.display.set_mode((500, 500))
pygame.display.set_caption('Учебная Игра')
```

- `pygame.init()` — инициализация PyGame.
- `pygame.display.set_mode()` — создание окна.
- `pygame.display.set_caption()` — заголовок окна.

---

Этот код создаёт окно для нашей игры. Сначала мы запускаем PyGame командой `pygame.init()`. Затем создаём окно размером 500 на 500 пикселей с помощью `pygame.display.set_mode()`. Мы также устанавливаем название окна через `pygame.display.set_caption()`. Это окно — наш холст, на котором мы будем рисовать.

## Слайд 5: Работа с цветами и шрифтами

```
BLACK = (0, 0, 0)
WHITE = (255, 255, 255)
LIGHT_BLUE = (200, 200, 255)

font = pygame.font.Font(None, 25)
```

- Цвета задаются с помощью кортежей (R, G, B).
- Шрифты создаются через `pygame.font.Font()`.

---

Цвета в PyGame задаются с помощью трёх чисел: Красный (R), Зелёный (G) и Синий (B). Например, (0, 0, 0) — это чёрный цвет, а (255, 255, 255) — белый. Мы также создаём шрифты, чтобы отображать текст, используя команду `pygame.font.Font()`.

## Слайд 6: Создание области с текстом (TextArea)

```
class TextArea:
    def __init__(self, x, y, width, height, color):
        self.rect = pygame.Rect(x, y, width, height)
        self.fill_color = color
```

- Создаём класс для рисования прямоугольников и текста.

---

Мы создаём класс `TextArea`, который позволяет нам рисовать прямоугольники с текстом. В этом классе мы задаём размеры и цвет области, а также создаём прямоугольник, где будет отображаться текст.

## Слайд 7: Установка текста

```
def set_text(self, text, fsize, text_color):  
    font = pygame.font.Font(None, fsize)  
    self.image = font.render(text, True, text_color)
```

- Метод для добавления текста в область.
- Шрифт автоматически уменьшается, если текст не помещается.

---

В классе `TextArea` есть метод `set_text()`, который позволяет добавлять текст в прямоугольник. `PyGame` автоматически уменьшает размер шрифта, если текст не помещается в выделенную область. Это помогает нам красиво отображать длинные строки.



## Слайд 8: Отрисовка области

```
def draw(self):  
    pygame.draw.rect(mw, self.fill_color, self.rect)  
    mw.blit(self.image, (text_x, text_y))
```

- Рисование прямоугольника и текста на экране.

---

Метод `draw()` рисует прямоугольник и отображает текст на экране. Сначала мы рисуем прямоугольник, затем помещаем текст в его центр. Это важно, чтобы всё выглядело аккуратно.

## Слайд 9: Основной цикл игры

```
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
```

- Основной цикл отвечает за работу игры.
- Проверяем события и реагируем на них.

---

В играх есть так называемый «цикл игры», который постоянно выполняется, пока игра не завершена. В этом цикле мы проверяем все события, например, нажатия клавиш или закрытие окна. Если игрок нажимает на крестик, игра заканчивается.

## Слайд 10: Обработка нажатий клавиш

```
if event.type == pygame.KEYDOWN:
    if event.key == pygame.K_q:
        # Показать вопрос
    if event.key == pygame.K_a:
        # Показать ответ
```

- Проверка нажатий клавиш: Q — Вопрос, A — Ответ.

---

Мы добавляем возможность игроку нажимать клавиши Q и A. Если нажата Q, то случайным образом появляется вопрос. Если нажата A, то появляется ответ. Этот код использует команду `randint()`, чтобы случайным образом выбирать текст из списка.

## Слайд 11: Обновление экрана

```
mw.fill(WHITE)
quest_card.draw()
ans_card.draw()
pygame.display.update()
```

- Очистка экрана и отрисовка элементов.
- `pygame.display.update()` — обновление экрана.

---

Каждый раз, когда что-то изменяется на экране, мы должны очистить его и заново нарисовать все объекты. Команда `pygame.display.update()` обновляет экран и показывает нам результат наших действий.

## Слайд 12: Завершение игры

```
pygame.quit()
```

- Закрытие окна и завершение PyGame.

---

Когда основной цикл игры заканчивается, мы используем команду `pygame.quit()`, чтобы закрыть окно и завершить работу PyGame.

## Слайд 13: Задание для учеников

- Улучшите игру, добавив больше вопросов и ответов.
- Сделайте так, чтобы вопросы не повторялись дважды подряд.
- Добавьте счётчик правильных ответов.

---

Сейчас ваша задача — улучшить игру! Вы можете добавить больше вопросов и ответов, сделать так, чтобы они не повторялись дважды подряд. Попробуйте добавить счётчик правильных ответов. Это поможет сделать игру более интересной!

## Слайд 14: Вопросы и обсуждение

- Понравилось ли вам работать с PyGame?
- Что было самым сложным?
- Хотите ли вы создать свою игру?