



Introduction to pygame

Greg Corbett, Scientific Computing
Department

What is pygame for?

We'll use it to make a simple game

- Draw rectangles
- Move rectangles
- Fire them across the screen
- Aim of the game: hit one rectangle with another



The Basics

1. Import and initialise pygame
2. Create a game window
3. Add a Game Loop



Before we start

Create a new python file for this session

Don't call your python file 'pygame.py'

- Any ideas why this might be a problem?



Step 1: Import + Initialize

Copy this code:

```
import sys
import pygame

pygame.init()
```

What does `pygame.init()` do?

- Initializes all imported pygame modules. You can always initialize individual modules manually, but `pygame.init()` is a convenient way to get everything started.



Step 2: Create a Game Window

Add to the bottom of your code:

```
BLACK = (0, 0, 0)
```

```
WINDOWWIDTH = 400
```

```
WINDOWHEIGHT = 300
```

```
WINDOW_SURFACE = pygame.display.set_mode((WINDOWWIDTH,  
WINDOWHEIGHT))
```



Step 3: Game Loop

Every game has a loop that goes round and round.

Each time the pygame game loop goes round it:

- MUST check for user input
- MAY process game logic
 - E.g. if one rectangle hits another, end the game
- MAY update the position of any objects
- MAY redraw objects



Step 3: Our Game Loop

while True:

```
    for event in pygame.event.get():
```

```
        # Check to see if the "Close" button has been pressed
```

```
        if event.type == pygame.QUIT:
```

```
            pygame.quit()
```

```
            sys.exit(0)
```

```
    WINDOW_SURFACE.fill(BLACK)
```

```
    pygame.display.update()
```

Our simple game loop checks if the game should be closed and redraws the screen. Run your program and see what happens.



Step 4: Hello World!

Drawing Text

```
TEXT = BASICFONT.render('Hello world!', True, WHITE, BLACK)
```

Arguments of the text are:

- String
- Anti-aliasing
- Text colour
- Background colour



Put this before your game loop:

```
BASIC_FONT = pygame.font.SysFont(None, 48)  
TEXT = BASIC_FONT.render('Hello world!', True, WHITE, BLACK)
```

And this in your game loop:

```
WINDOW_SURFACE.fill(BLACK)  
text_rect = TEXT.get_rect()  
text_rect.centerx = WINDOW_SURFACE.get_rect().centerx  
text_rect.centery = WINDOW_SURFACE.get_rect().centery  
WINDOW_SURFACE.blit(TEXT, text_rect)  
pygame.display.update()
```

Remember, you'll
need to define
WHITE



Step 5: Drawing other objects

```
pygame.draw.line(windowSurface, COLOUR, (xstart, ystart), (xend, yend))
```

```
pygame.draw.circle(windowSurface, COLOUR, (xcentre, ycentre), radius)
```

PyGame has lots of drawing functions – experiment with them!

- `pygame.draw.rect(...)`
- `pygame.draw.ellipse(...)`
- All the `pygame.draw` methods have slightly different signatures, documentation is your friend:
<https://www.pygame.org/docs/ref/draw.html>



RGB values

Color RGB Values

- Black (0, 0, 0)
- Blue (0, 0, 255)
- Gray (128, 128, 128)
- Green (0, 128, 0)
- Lime (0, 255, 0)
- Purple (128, 0, 128)
- Red (255, 0, 0)
- Teal (0, 128, 128)
- White (255, 255, 255)
- Yellow (255, 255, 0)

