Introduction to Game Development Using PyGame

Hackathon by the Sea 3.0

Uri Mann

Let’s get to know each other

- About me
  - My mane is Uri Mann (**Ooree** if you want to pronounce it correctly)
  - Live in Oak Park
  - Professionally developing software since 1985
  - My second involvement with Ventura County Hackathon
- About you
  - Did you write any programs in the past?
  - Which language?
  - Do you know Python?
  - Did you try to write any games? What language? Framework?

Agenda

- Demo
- Why Python and PyGame
- Basic Python coding
- Basic game development
- Build Your Game
- Adding sounds

game.py - D:\play\game\game.py (3.8.0)

File Edit Format Run Options Window Help

```
# Import everything we're going to use for the game
import pygame
import sys

# Intialize game environment
pygame.init()
pygame.display.init()
clock = pygame.time.Clock()
pygame.display.set_caption('Swiming Fish Game') # Set game window title

# Load background image
bg = pygame.image.load('underwater.png') # Screen background
bg_rect  = bg.get_rect()
screen = pygame.display.set_mode((bg_rect.width, bg_rect.height))
# Load animation images
fishs = (pygame.image.load('fish1.png'), pygame.image.load('fish2.png'))
fish_rect = fishs[0].get_rect()
no_fish = len(fishs)

# Misc. variables
VELOCITY = 5
step = 0
flip = False

# Main game loop
while pygame.QUIT not in [event.type for event in pygame.event.get()]:
    clock.tick(20)

    # Check keyboard input
    keys = pygame.key.get_pressed()
    if keys[pygame.K_LEFT]: # Q: Did the user pressed the left arrow?
        if fish_rect.x > 0: # Q: Is the fish all the way to the left?
            fish_rect.x = fish_rect.x - VELOCITY
        step = step + 1
        flip = False # Moving to the left the image stays the same
    elif keys[pygame.K_RIGHT]: # Q: Did the user pressed the right arrow?
        if fish_rect.x + fish_rect.width < bg_rect.width: # Q: Is the fish all t
            fish_rect.x = fish_rect.x + VELOCITY
        step = step + 1
        flip = True # Moving to the right the image must be flipped

    fish = pygame.transform.flip(fishs[step % no_fish], flip, False)
    # Update the screen
    screen.blit(bg, bg_rect)        # Paint the backgound image
    screen.blit(fish, fish_rect)    # Paint the fish over the backgound
    pygame.display.update()         # Update the screen
    # end of game loop

# Shutdown everything in reverse order of initialization
pygame.display.quit()
pygame.quit()
sys.exit(0)
```

Ln: 1 Col: 0

Game Demo

Why Python and PyGame

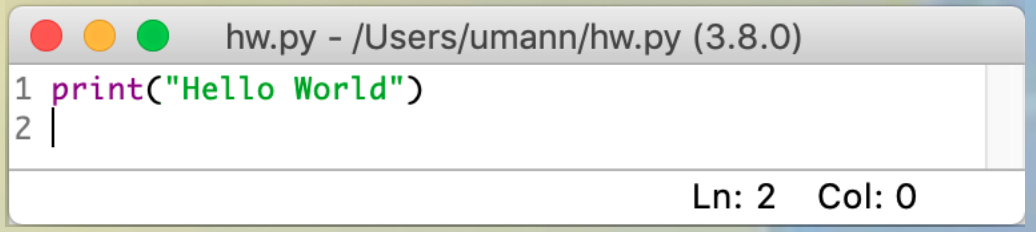
- Python is a general-purpose programming language
- Invented by Guido van Rossum
- Free to use by everyone (open source), continuously improved
- Easy to learn, widely used, portable
- Interpreted language facilitating interactive development
- Many modules (libraries) for almost any purpose
- PyGame is cross-platform game library (also open source)
- Written by Pete Shinners
- Based on SDL (Simple DirectMedia Library)

Python IDLE

- IDLE (Interactive Development and Learning Environment) is simple but powerful tools to start having fun with Python
- Allows writing code and run it and debug it interactively
- Two main windows
  - Editor to write and save your work
  - Python Shell to display program execution and experiment with code snippets

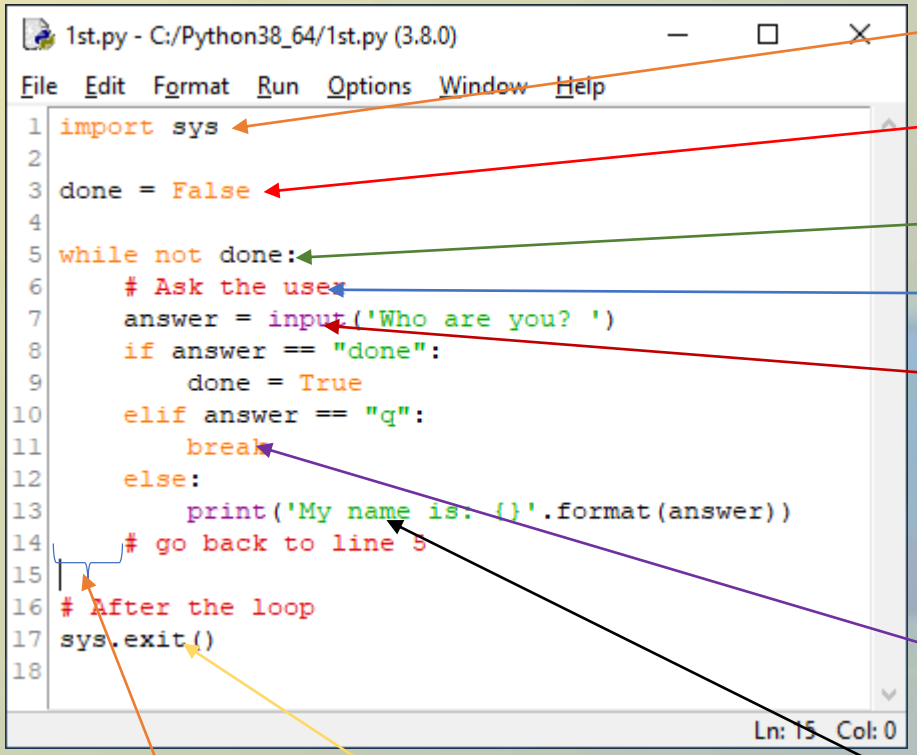
Your first program

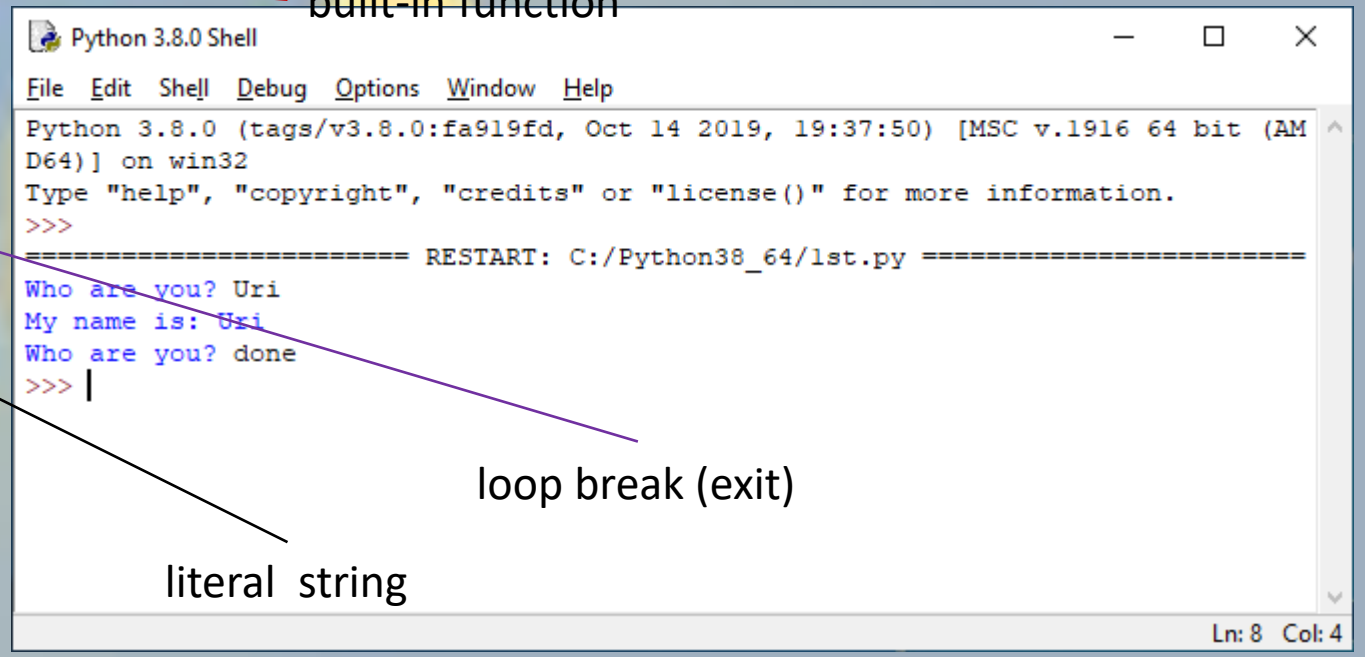
- Traditionally the first program is “Hello World!”



- Hit F5 key
- It’s that simple!

Python programming primer





imported module

boolean variable

while loop

comment

built-in function

loop break (exit)

literal string

sys module function called

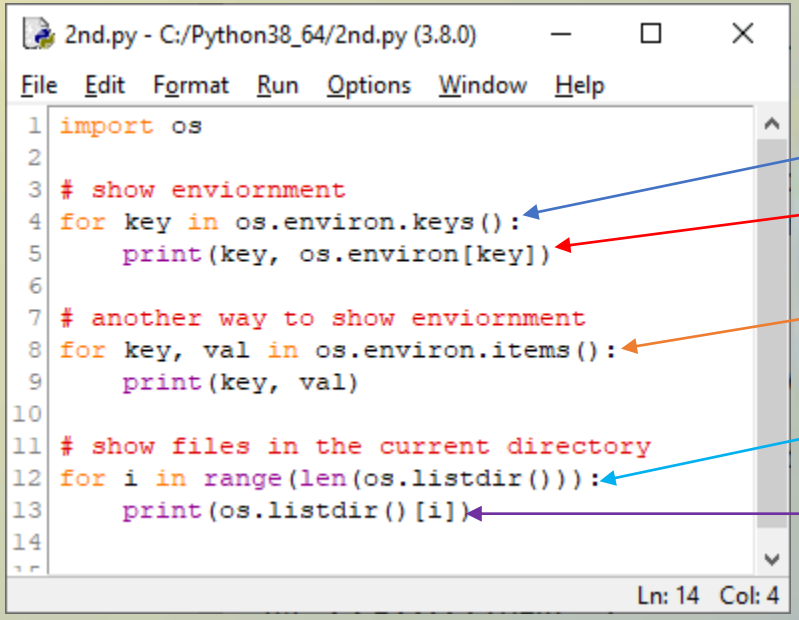
control block indentation (required)

Collections (tuple, list, dict)

- Lists may hold any type of objects (may be non-homogenous)
- Lists may be inserted into, appended or shrunk
- Accessed by index
- Tuple – read-only array of items
  - Delimited by parenthesis (e.g. (1,’2’,3))
- List - Modifiable
  - Delimited by square braces (e.g. [‘this’, ’is’, ’a’, ‘list’, ‘of’, 5, (‘i ’, ‘t ’, ‘ e ’, ‘ m ’, ‘ s’)])
- Dict – searchable list
  - Key, value, collection delimited by curly braces (e.g. {1: “one”, 2: [1,2,3]})
  - Keys must be “hashable”
  - Values may be modified but not the keys

For Loops

- Lists may be accessed sequentially (by iteration)



get list of keys from the os.environ dict(ionary)

use the key as index for the value

get key, value pair

get list of files in the current directory
(len() returns the number of items)

access list using index

Anatomy of PyGame Game

- Game has three parts
  - Game initialization
    - Importing required modules (one of them being **import pygame**)
    - Initializing PyGame framework
    - Acquiring game assets (graphics, sounds, etc.)
  - Processing main game loop
    - Processing events (Window events, keyboard, mouse, etc.)
    - Repainting the screen to show game progress
  - Game teardown
    - Releasing assets
    - PyGame framework shutdown
    - Terminating python game script

PyGame Game Initialization

game.py - D:\play\game\game.py (3.8.0)

File Edit Format Run Options Window Help

```
1  # Import everything we're going to use for the game
2  import pygame
3  import sys
4
5  # Intialize game environment
6  pygame.init()
7  pygame.display.init()
8  clock = pygame.time.Clock()
9  pygame.display.set_caption('Swiming Fish Game') # Set game window title
10
11 # Load background image
12 bg = pygame.image.load('underwater.png') # Screen background
13 bg_rect  = bg.get_rect()
14 screen = pygame.display.set_mode((bg_rect.width, bg_rect.height))
15 # Load animation images
16 fishs = (pygame.image.load('fish1.png'), pygame.image.load('fish2.png'))
17 fish_rect = fishs[0].get_rect()
18 no_fish = len(fishs)
19
20 # Misc. variables
21 VELOCITY = 5
22 step = 0
23 flip = False
24
25 # Main game loop
26 while pygame.QUIT not in [event.type for event in pygame.event.get()]:
27     clock.tick(20)
```

Ln: 1 Col: 0

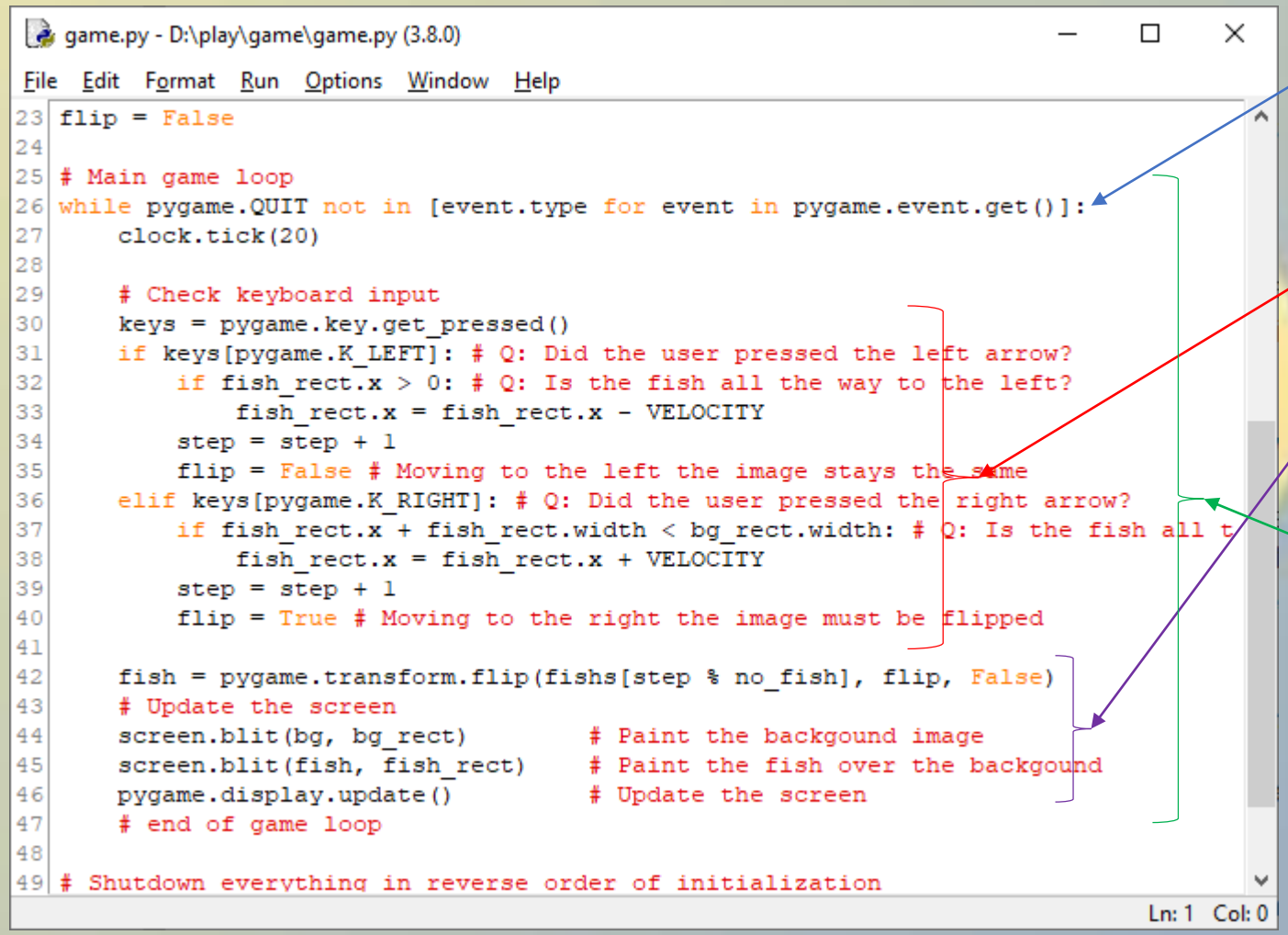
imports section

PyGame framework setup

graphical assets

miscellaneous other stuff

Game Loop



check if player wants to exit the game

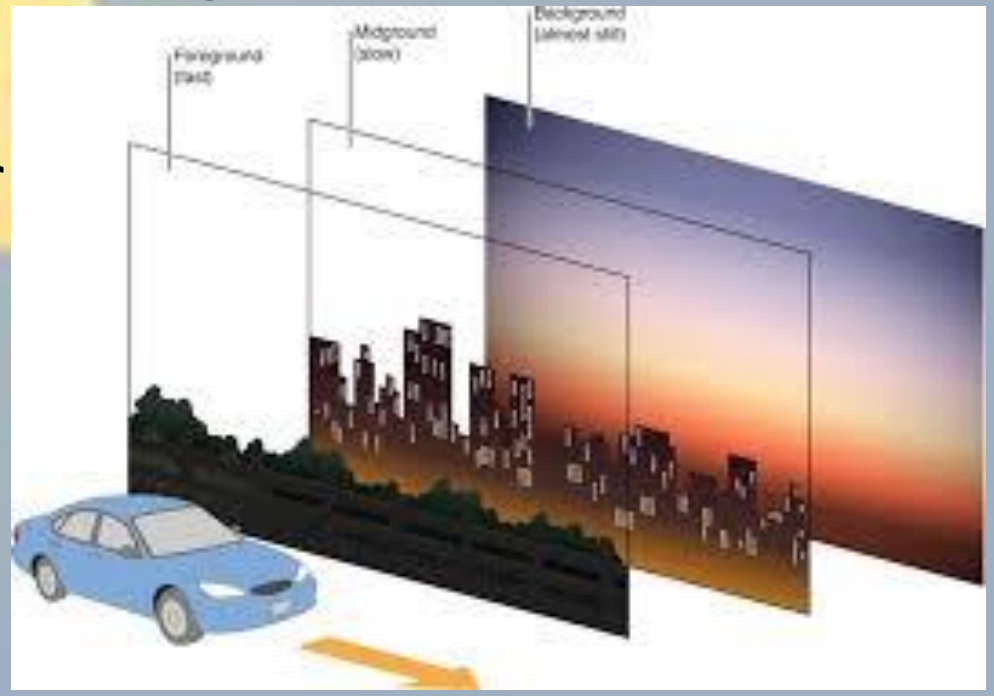
process keyboard input from player

repaint the screen

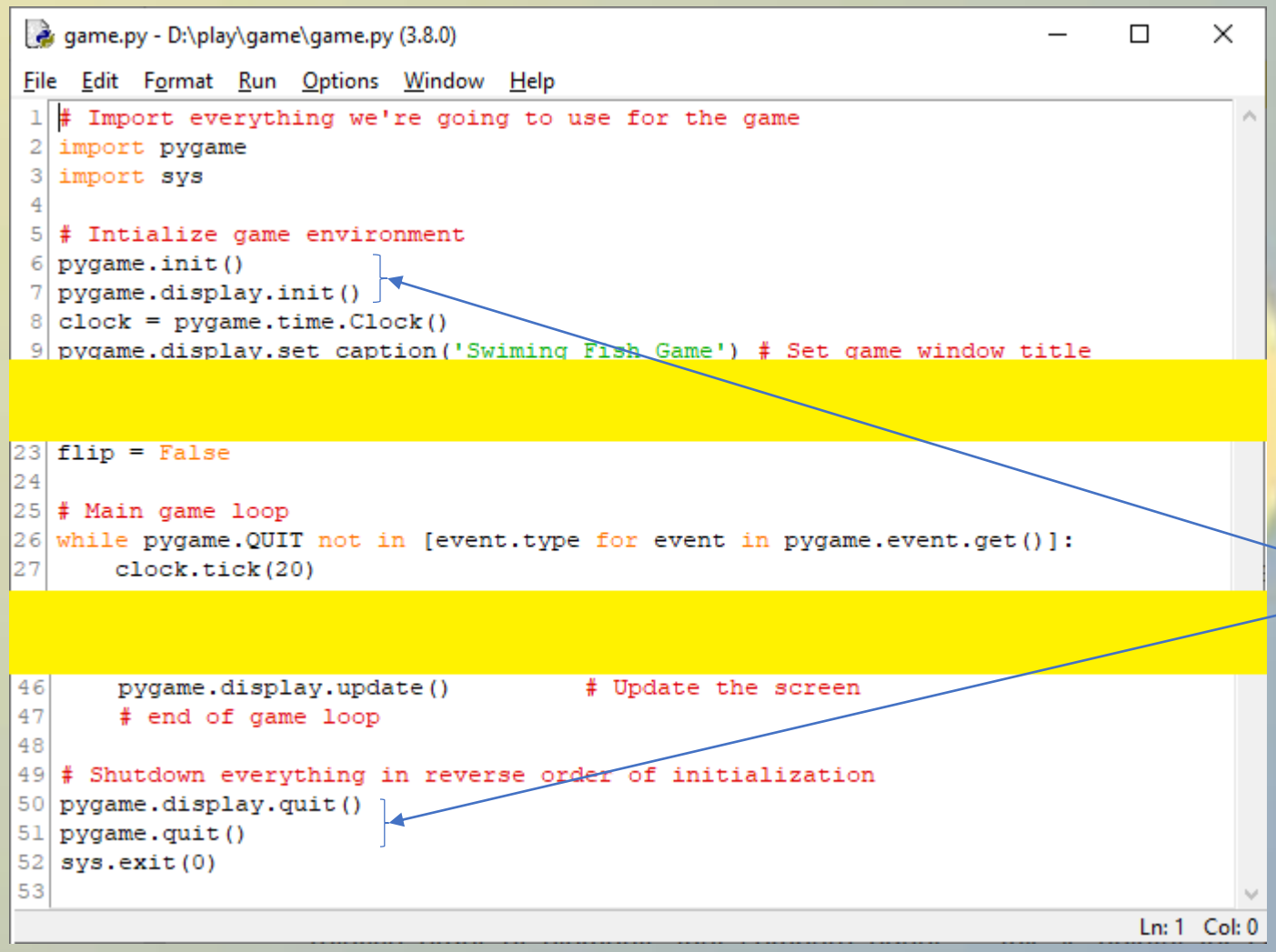
rinse and repeat

Painting the Screen

- Compose the scene form back to front. Far background to closer background to game characters
- Same rules apply to animation (parallax scrolling)
- Step one sprite image at a time
- PyGame compose each step into a buffer
- After all composition is finished, screen is updated at once



Game Shutdown Sequence



reverse the order of initialization

It’s Now Your Turn

- Find replacement assets (background image and sprites)
- Recommended: https://opengameart.org/ and https://images.google.com/
- Search background according to your game theme (race trace, hunted castle, magic city, etc.)
- Search for sprites (race car, vampire, worrier, etc.). You may have more than two sprite images
- Open **game.py** in IDLE
- Replace **underwater.png** and **fish1.png**, **fish2.png** with path to your images
- Hit F5 and enjoy

Additional Resources

- Python https://www.python.org/: use **Download** tab to install
- PyGame https://www.pygame.org/: use **Getting Started**, **Projects** and **Docs** tabs
- Open Game Art https://opengameart.org/: use **Browse** tab
- Stack Overflow https://stackoverflow.com/: search for answers and post questions
- Advance tutorial https://techwithtim.net/tutorials/game-development-with-python/pygame-tutorial/

- My email: abba.mann@gmail.com