Attendance:
tiny.cc/event-attendance

App Dev League

Day 3: Pygame Basics

Agenda

1. Intro to Pygame and Pygame Basics
2. TicTacToe

Intro to Pygame

1

What is Pygame?

- Pygame is a python library that makes it easy to create games
- We will be using pygame to create the rest of games during this course

Advantages of Pygame

- Much more graphics oriented
    - Displays, windows
- Compatible with the many other python modules
    - Random, time etc.
- Easy to use within Object Oriented Programming

Basic Pygame Constructs

- Display
  - Windows, Screens
  - Updating
- Events
  - Detecting key, mouse presses etc.
- Timing
  - Delays for animations etc.

Basic Pygame Structure

- Initialize Pygame canvas
- While True / game loop with exit condition
- display.update() ⇒ to update the display for animations etc.
- Some kind of delay to help with graphics updating
    - Most of these use the time module

Example Starter Pygame Code

Initializing pygame

Game loop

```
# Import and initialize the pygame library
import pygame
pygame.init()

# Set up the drawing window (500x500)
screen = pygame.display.set_mode([500, 500])

# Run until the user asks to quit
running = True
while running:

    # Did the user click the window close button?
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

    # Fill the background with white
    screen.fill((255, 255, 255))

    # Flip the display
    pygame.display.update()

# Done! Time to quit.
pygame.quit()
```

Setting up the window

Setting up the Screen

- “import pygame” is crucial to accessing the PyGame libraries
- Tell the program to start using PyGame with “pygame.init()” and set a caption with “pygame.display.set_caption()”
- Set up the window with “pygame.display.set_mode”
  - You feed this function a size comprised of the WIDTH and HEIGHT you want

```
import pygame
pygame.init()
pygame.display.set_caption("Name of Game")

WIDTH = 400
HEIGHT = 400
size = (WIDTH, HEIGHT)
screen = pygame.display.set_mode(size)
```

What are sprites?

- A sprite is any image that is drawn to the 2D screen in a video game
- Sprites are useful because they represent the player character as well as everything present in an image
- Sprites must be drawn to the screen in **every frame**

In this image nearly everything is a Sprite. Mario, the pipes, and the blocks are all examples of images that are updated and repeatedly drawn to the screen

Creating FPS with Clock

- Start off code with “import time”
- Set your fps with a variable “fps = 30”
- Initialize the clock with “clock = pygame.time.Clock()”
- At the end of your game loop use “clock.tick(fps)”

```
import time
fps = 30
clock = pygame.time.Clock()

while(running):
    #Code for game loop
    pygame.display.update()
    clock.tick(fps)
```

This is what the general structure of your code should look like if you are implementing FPS

Keyboard Presses

- **You can keep track of keyboard events with a for loop**
  - **for event in pygame.event.get():**
- **Quitting out of the game with the x button**
  - **if event.type == pygame.QUIT:**
- **Pressing a key down**
  - **if event.type == pygame.KEYDOWN:**
    - **if event.key == pygame.K_LEFT:**
- **https://www.pygame.org/docs/ref/key.html**

```
while not done:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done = True
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT:
                #insert code for moving left
            if event.key == pygame.K_RIGHT:
                #insert code for moving right
            if event.key == pygame.K_UP:
                #insert code for moving up
            if event.key == pygame.K_DOWN:
                #insert code for moving down
```

Pygame stores your keyboard and mouse actions as events every frame

Mouse Presses

- **You can keep track of mouse presses using event listeners too**
- **if event.type == pygame.MOUSEBUTTONDOWN:**
  - **After this you can perform some click function**

```
while not done:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done = True
        if event.type == pygame.MOUSEBUTTONDOWN:
            click()
```

This code handles mouse clicks, remember that you must create your own click function (we will get to that later)

Boundaries and Collision Detection

- When moving your sprites around it is important to make sure they don't go past the boundaries
- We can do this by adding a small check to our move functions to see if an action would bring a sprite off of the screen
- This same code works for collision detection!
  - Simply replace the WIDTH and HEIGHT variables with whatever the coordinates of the sprite we want to check for collision are

```
#say we have variables player.width, player.height, WIDTH, HEIGHT
#we will also need the position of the player as: player.x and player.y

#when we want to move left
if player.x < 0:
    player.x = 0

#when we want to move right
if player.x > WIDTH - player.width:
    player.x = WIDTH - player.width

#when we want to move up
if player.y < 0:
    player.y = 0

#when we want to move down
if player.y > HEIGHT - player.height:
    player.y = HEIGHT - player.height
```

Score

- ➔ **Keep a variable called score that starts off equal to 0**
- ➔ **Every time an action occurs that increases score, update this variable**
- ➔ **Use this show_score function to print the score to the screen on every frame**

```
# displaying Score function
def show_score(choice, color, font, size):

    # creating font object score_font
    score_font = pygame.font.SysFont(font, size)

    # create the display surface object
    # score_surface
    score_surface = score_font.render('Score : ' + str(score), True, color)

    # create a rectangular object for the text
    # surface object
    score_rect = score_surface.get_rect()

    # displaying text
    game_window.blit(score_surface, score_rect)
```

Tic Tac Toe

2

Project Showcase

- We will be building Tic Tac Toe
- Open up repl.it and follow along with me

THANKS!

ANY QUESTIONS?

You can find more info @

- https://www.appdevleague.org
- https://linktr.ee/AppDevLeague