

Lesson 4: Introduction to Pygame

Objectives:

- Set up a Pygame window and create simple visualizations.
- Use Pygame to create basic animations related to space science.

Lesson Plan

1. Introduction (10 minutes)

Engagement:

- Discuss how visualizations and simulations help scientists understand complex space phenomena.
- Show a simple example of a Pygame window to get students excited about creating their own visualizations.

2. Explanation (15 minutes)

Setting Up Pygame:

For this, I highly recommend using the Pygame sandbox in CodeHS – easy to set up and use.

Example Code:

```
import pygame
from pygame.locals import *
import os

# Set SDL audio driver to avoid driver error in console
os.environ['SDL_AUDIODRIVER'] = 'dsp'

# Initialize Pygame
pygame.init()

# Get the display information and set up the window
screen_info = pygame.display.Info()
WIDTH = screen_info.current_w
HEIGHT = 350 # Fixed height for the window
screen = pygame.display.set_mode((WIDTH, HEIGHT), pygame.RESIZABLE)
pygame.display.set_caption("Space Visualization")

# Main loop
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        elif event.type == KEYDOWN and event.key == K_ESCAPE:
            running = False
```

```

screen.fill((0, 0, 0)) # Fill the screen with black

# Draw a white rectangle in the middle of the screen
rect_width = 100
rect_height = 100
rect_x = (WIDTH - rect_width) // 2
rect_y = (HEIGHT - rect_height) // 2
pygame.draw.rect(screen, (255, 255, 255), (rect_x, rect_y, rect_width,
rect_height))

# Draw some text to confirm the window is updating
font = pygame.font.Font(None, 36)
text = font.render('Space Visualization', True, (255, 255, 255))
text_x = (WIDTH - text.get_width()) // 2
text_y = rect_y - text.get_height() - 10 # Positioning text above the
rectangle
screen.blit(text, (text_x, text_y))

pygame.display.flip() # Update the display

pygame.quit()

```

Drawing Shapes:

- Show how to draw basic shapes (e.g., circles for planets) on the Pygame window.

Example Code:

```

import pygame
from pygame.locals import *
import os

# Set SDL audio driver to avoid driver error in console
os.environ['SDL_AUDIODRIVER'] = 'dsp'

# Initialize Pygame
pygame.init()

# Set window dimensions
WIDTH = 800
HEIGHT = 600
screen = pygame.display.set_mode((WIDTH, HEIGHT))
pygame.display.set_caption("Space Visualization")

# Main loop
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        elif event.type == KEYDOWN and event.key == K_ESCAPE:
            running = False

    screen.fill(("orange")) # Fill the screen with orange
    pygame.draw.circle(screen, (255, 255, 0), (WIDTH // 2, HEIGHT // 2), 50) #
    Draw the Sun
    pygame.display.flip() # Update the display

pygame.quit()

```

3. Hands-On Activity (20 minutes)

Task:

Students will create their own Pygame window and draw additional shapes representing planets or moons.

Worksheet (below):

- Practice setting up Pygame and creating a window.
- Exercises on drawing shapes and creating basic animations.

Example Worksheet Tasks:

Example Code:

```
import pygame
from pygame.locals import *
import os

# Set SDL audio driver to avoid driver error in console
os.environ['SDL_AUDIODRIVER'] = 'dsp'

# Initialize Pygame
pygame.init()

# Get the display information and set up the window
screen_info = pygame.display.Info()
WIDTH = screen_info.current_w
HEIGHT = 350 # Fixed height for the window
screen = pygame.display.set_mode((WIDTH, HEIGHT), pygame.RESIZABLE)
pygame.display.set_caption("Solar System Visualization")

# Main loop
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        elif event.type == KEYDOWN and event.key == K_ESCAPE:
            running = False

    screen.fill((0, 0, 0)) # Fill the screen with black
    pygame.draw.circle(screen, (255, 255, 0), (WIDTH // 2, HEIGHT // 2), 50) #
    Draw the Sun
    pygame.draw.circle(screen, (0, 0, 255), (WIDTH // 2 + 200, HEIGHT // 2), 20) #
    Draw the Earth
    pygame.display.flip() # Update the display

pygame.quit()
```

4. Review (10 minutes)

Q&A:

- Address any questions students might have about variables, data types, or the example code.

Exit Ticket:

1. What is Pygame used for?

- A. Writing and executing basic Python scripts
- B. Creating games and visual simulations
- C. Managing files and directories
- D. Performing mathematical calculations

Answer: B

2. What function is used to create a Pygame window?

- A. `pygame.init()`
- B. `pygame.display.set_caption()`
- C. `pygame.display.set_mode((width, height))`
- D. `pygame.quit()`

Answer: C

3. Which of the following Python programs sets up a Pygame window and draws a circle in the center?

- A.
- ```
import pygame
pygame.init()

screen = pygame.display.set_mode((800, 600))
pygame.display.set_caption("Draw a Circle")

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

 screen.fill((0, 0, 0))
 pygame.draw.circle(screen, (255, 255, 255), (400, 300), 50)
 pygame.display.flip()

pygame.quit()
```
- B.
- ```
import pygame
pygame.init()

screen = pygame.display.set_mode((800, 600))
pygame.display.set_caption("Draw a Circle")

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

    screen.fill((0, 0, 0))
    pygame.draw.circle(screen, (255, 0, 0), (400, 300), 50)
    pygame.display.flip()

pygame.quit()
```

```
C. import pygame
pygame.init()

screen = pygame.display.set_mode((800, 600))
pygame.display.set_caption("Draw a Circle")

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

    screen.fill((0, 0, 0))
    pygame.draw.circle(screen, (255, 255, 255), (400, 300), 100)
    pygame.display.flip()

pygame.quit()
```

Answer: A

Worksheet 4: Introduction to Pygame

Section 1: Setting up Pygame

1. Write a Python program that sets up a Pygame window with a black background.

Section 2: Drawing Shapes

2. Write a Python program that sets up a Pygame window and draws a circle in the center.
3. Write a Python program that draws a representation of the Sun and the Earth.

ANSWER KEY:

Section 1: Setting up Pygame

1. Write a Python program that sets up a Pygame window with a black background.

```
import pygame
from pygame.locals import *
import os

# Set SDL audio driver to avoid driver error in console
os.environ['SDL_AUDIODRIVER'] = 'dsp'

# Initialize Pygame
pygame.init()

# Get the display information and set up the window
screen_info = pygame.display.Info()
WIDTH = screen_info.current_w
HEIGHT = 350 # Fixed height for the window
screen = pygame.display.set_mode((WIDTH, HEIGHT), pygame.RESIZABLE)
pygame.display.set_caption("My Pygame Window")

# Main loop
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        elif event.type == KEYDOWN and event.key == K_ESCAPE:
            running = False

    screen.fill((0, 0, 0)) # Fill the screen with black
    pygame.display.flip() # Update the display

pygame.quit()
```

Section 2: Drawing Shapes

2. Write a Python program that sets up a Pygame window and draws a circle in the center.

```
import pygame
from pygame.locals import *
import os

# Set SDL audio driver to avoid driver error in console
os.environ['SDL_AUDIODRIVER'] = 'dsp'

# Initialize Pygame
pygame.init()

# Get the display information and set up the window
screen_info = pygame.display.Info()
WIDTH = screen_info.current_w
HEIGHT = 350 # Fixed height for the window
screen = pygame.display.set_mode((WIDTH, HEIGHT), pygame.RESIZABLE)
pygame.display.set_caption("Draw a Circle")
```

```

# Main loop
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        elif event.type == KEYDOWN and event.key == K_ESCAPE:
            running = False

    screen.fill((0, 0, 0)) # Fill the screen with black
    pygame.draw.circle(screen, (255, 255, 255), (WIDTH // 2, HEIGHT // 2), 50) #
    Draw a white circle in the center
    pygame.display.flip() # Update the display

pygame.quit()

```

3. Write a Python program that draws a representation of the Sun and the Earth.

```

import pygame
from pygame.locals import *
import os

# Set SDL audio driver to avoid driver error in console
os.environ['SDL_AUDIODRIVER'] = 'dsp'

# Initialize Pygame
pygame.init()

# Get the display information and set up the window
screen_info = pygame.display.Info()
WIDTH = screen_info.current_w
HEIGHT = 350 # Fixed height for the window
screen = pygame.display.set_mode((WIDTH, HEIGHT), pygame.RESIZABLE)
pygame.display.set_caption("Solar System Visualization")

# Main loop
running = True
while running:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        elif event.type == KEYDOWN and event.key == K_ESCAPE:
            running = False

    screen.fill((0, 0, 0)) # Fill the screen with black
    pygame.draw.circle(screen, (255, 255, 0), (WIDTH // 2, HEIGHT // 2), 50) #
    Draw the Sun
    pygame.draw.circle(screen, (0, 0, 255), (WIDTH // 2 + 200, HEIGHT // 2), 20) #
    Draw the Earth

    font = pygame.font.Font(None, 36)
    text = font.render('Space Visualization', True, (255, 255, 255))
    text_x = (WIDTH - text.get_width()) // 2
    text_y = HEIGHT - text.get_height() - 10 # Positioning text above the
    rectangle
    screen.blit(text, (text_x, text_y))

    pygame.display.flip() # Update the display

pygame.quit()

```