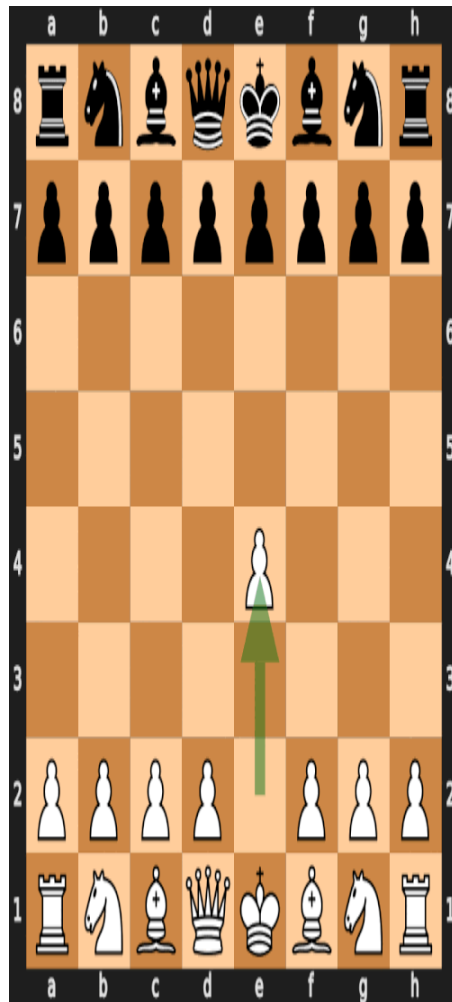


Build a Space Action Game in Python

You can build a complete space shooter game using Python and the Pygame library. This project teaches fundamental computer literacy concepts like coordinate systems, loops, and object-oriented programming.



1. Setup Your Environment

To run the game, you must install Python and the Pygame library on your system. Open your terminal (Linux) or Command Prompt/PowerShell (Windows) and run the correct command for your operating system:

For Fedora Linux

Fedora manages Python packages safely through system repositories. Use the built-in package manager:

bash

```
sudo dnf install python3-pygame
```

Use code with caution.

For Windows

Windows uses Python's standard package installer ([pip](#)). Run this command in your Command Prompt:

cmd

```
pip install pygame
```

Use code with caution.

2. The Game Code

Create a new text file on your computer, name it `space_game.py`, and paste the following code into it:

python

```
import pygame
import random
import sys

# Initialize Pygame
pygame.init()

# Game Constants
SCREEN_WIDTH = 800
SCREEN_HEIGHT = 600
FPS = 60

# Colors (RGB)
BLACK = (0, 0, 0)
WHITE = (255, 255, 255)
RED = (255, 0, 0)
```

```

GREEN = (0, 255, 0)

# Setup Game Window
screen = pygame.display.set_mode((SCREEN_WIDTH, SCREEN_HEIGHT))
pygame.display.set_caption("Space Action Shooter")
clock = pygame.time.Clock()

# Player Class
class Player(pygame.sprite.Sprite):
    def __init__(self):
        super().__init__()
        self.image = pygame.Surface((50, 40))
        self.image.fill(GREEN)
        self.rect = self.image.get_rect()
        self.rect.centerx = SCREEN_WIDTH // 2
        self.rect.bottom = SCREEN_HEIGHT - 10
        self.speed = 8

    def update(self):
        keys = pygame.key.get_pressed()
        if keys[pygame.K_LEFT] and self.rect.left > 0:
            self.rect.x -= self.speed
        if keys[pygame.K_RIGHT] and self.rect.right < SCREEN_WIDTH:
            self.rect.x += self.speed

    def shoot(self):
        laser = Laser(self.rect.centerx, self.rect.top)
        all_sprites.add(laser)
        lasers.add(laser)

# Enemy Class
class Enemy(pygame.sprite.Sprite):
    def __init__(self):
        super().__init__()
        self.image = pygame.Surface((30, 30))
        self.image.fill(RED)
        self.rect = self.image.get_rect()
        self.rect.x = random.randint(0, SCREEN_WIDTH - 30)
        self.rect.y = random.randint(-100, -40)
        self.speed = random.randint(2, 6)

    def update(self):
        self.rect.y += self.speed
        if self.rect.top > SCREEN_HEIGHT:
            self.rect.x = random.randint(0, SCREEN_WIDTH - 30)
            self.rect.y = random.randint(-100, -40)
            self.speed = random.randint(2, 6)

```

```

# Laser Class
class Laser(pygame.sprite.Sprite):
    def __init__(self, x, y):
        super().__init__()
        self.image = pygame.Surface((5, 15))
        self.image.fill(WHITE)
        self.rect = self.image.get_rect()
        self.rect.centerx = x
        self.rect.bottom = y
        self.speed = -10

    def update(self):
        self.rect.y += self.speed
        if self.rect.bottom < 0:
            self.kill()

# Sprite Groups
all_sprites = pygame.sprite.Group()
enemies = pygame.sprite.Group()
lasers = pygame.sprite.Group()

# Create Game Objects
player = Player()
all_sprites.add(player)

for i in range(8):
    enemy = Enemy()
    all_sprites.add(enemy)
    enemies.add(enemy)

# Game Score
score = 0
font = pygame.font.SysFont("Arial", 24)

# Main Game Loop
running = True
while running:
    clock.tick(FPS)

    # 1. Event Handling
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_SPACE:
                player.shoot()

```

```

# 2. Update Positions
all_sprites.update()

# 3. Collision Detection
hits = pygame.sprite.groupcollide(enemies, lasers, True, True)
for hit in hits:
    score += 10
    new_enemy = Enemy()
    all_sprites.add(new_enemy)
    enemies.add(new_enemy)

if pygame.sprite.spritecollideany(player, enemies):
    running = False

# 4. Drawing / Rendering
screen.fill(BLACK)
all_sprites.draw(screen)

score_text = font.render(f"Score: {score}", True, WHITE)
screen.blit(score_text, (10, 10))

pygame.display.flip()

pygame.quit()
sys.exit()

```

Use code with caution.

3. How to Launch and Play

Open your terminal or command prompt, navigate to the folder where you saved the file, and type:

```

bash
python3 space_game.py

```

Use code with caution.

(Note: On Windows, if `python3` does not work, simply type `python space_game.py`).

- Use the Left and Right Arrow Keys to move your green ship.

- Press the Spacebar to shoot lasers at the falling red targets.
-

4. Core Computer Literacy Concepts

- The Game Loop: Video games continuously process your inputs, update positions, check for hits, and redraw the screen 60 times every second (`FPS = 60`).
- The Coordinate System: Computer screens place `((0,0))` at the top-left corner. Moving right increases `(X)`, and moving down increases `(Y)`.
- Sprites and Collisions: Pygame uses "Sprites" to track objects. The code constantly checks if the boundaries (`rect`) of lasers and enemies overlap to trigger score updates or a game-over sequence.