

نوٹي اب ى دق م Python Cafe

```
import pygame
import time
import random

pygame.init()

width, height = 800, 600
display = pygame.display.set_mode((width, height))
pygame.display.set_caption("Snake")

white = (255, 255, 255)
black = (0, 0, 0)
red = (255, 0, 0)
rose = (255, 0, 255)

snake_block = 12
snake_speed = 13
snake = [(width / 2, height / 2)] # Corrected initialization
snake_direction = "RIGHT"

apple = (
    round(random.randrange(0, width - snake_block) / 10.0) * 10.0,
    round(random.randrange(0, height - snake_block) / 10.0) * 10.0,
)

clock = pygame.time.Clock()

# Load sound effect
bite_sound = pygame.mixer.Sound("snake.MP3")

def draw_snake(snake):
    for i, block in enumerate(snake):
        pygame.draw.rect(display, rose, [block[0], block[1], snake_block,
snake_block])
        if i == 0:
            draw_snake_head(block[0], block[1], snake_direction)

def draw_snake_head(x, y, direction):
    if direction == "UP":
        pygame.draw.circle(display, white, (int(x + snake_block / 4), int(y
+ snake_block / 4)), 3)
```



```

        pygame.draw.circle(display, white, (int(x + 3 * snake_block / 4),
int(y + snake_block / 4)), 3)
        pygame.draw.rect(display, black, [x, y, snake_block, snake_block])

def draw_apple(apple):
    pygame.draw.rect(display, red, (apple[0], apple[1], snake_block,
snake_block))

def message(msg, color, pos, size=50):
    font = pygame.font.SysFont(None, size)
    screen_text = font.render(msg, True, color)
    display.blit(screen_text, pos)

import pygame
import time
import random

pygame.init()

width, height = 800, 600
display = pygame.display.set_mode((width, height))
pygame.display.set_caption("Snake")

white = (255, 255, 255)
black = (0, 0, 0)
red = (255, 0, 0)
rose = (255, 0, 255)

snake_block = 15
snake_speed = 15
snake = [(width / 2, height / 2)] # Corrected initialization
snake_direction = "RIGHT"

apple = (
    round(random.randrange(0, width - snake_block) / 10.0) * 10.0,
    round(random.randrange(0, height - snake_block) / 10.0) * 10.0,
)

clock = pygame.time.Clock()

# Load sound effect

```

```

bite_sound = pygame.mixer.Sound("snake.MP3")

def draw_snake(snake):
    for i, block in enumerate(snake):
        pygame.draw.rect(display, rose, [block[0], block[1], snake_block,
snake_block])
        if i == 0:
            draw_snake_head(block[0], block[1], snake_direction)

def draw_snake_head(x, y, direction):
    if direction == "UP":
        pygame.draw.circle(display, white, (int(x + snake_block / 4), int(y
+ snake_block / 4)), 3)
        pygame.draw.circle(display, white, (int(x + 3 * snake_block / 4),
int(y + snake_block / 4)), 3)
        pygame.draw.rect(display, black, [x, y, snake_block, snake_block])

def draw_apple(apple):
    pygame.draw.rect(display, red, (apple[0], apple[1], snake_block,
snake_block))

def message(msg, color, pos, size=50):
    font = pygame.font.SysFont(None, size)
    screen_text = font.render(msg, True, color)
    display.blit(screen_text, pos)

def get_user_name(width):
    user_name = ""
    input_rect = pygame.Rect(300, 200, 140, 32)
    label_font = pygame.font.SysFont(None, 30)
    label_text = label_font.render("Enter Your Name:", True, white)
    label_pos = label_text.get_rect(center=(width/2, 180))
    color_inactive = pygame.Color('lightskyblue3')
    color_active = pygame.Color('dodgerblue2')
    color = color_inactive
    active = True # Set active to True initially
    done = False

    # Simulate mouse click inside input rectangle
    pygame.event.post(pygame.event.Event(pygame.MOUSEBUTTONDOWN, {'pos':
input_rect.center}))

```

```

while not done:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            done = True
        if event.type == pygame.MOUSEBUTTONDOWN:
            if input_rect.collidepoint(event.pos):
                active = not active
            else:
                active = False
        color = color_active if active else color_inactive
        if event.type == pygame.KEYDOWN:
            if active:
                if event.key == pygame.K_RETURN:
                    done = True
                elif event.key == pygame.K_BACKSPACE:
                    user_name = user_name[:-1]
                else:
                    user_name += event.unicode

    display.fill((30, 30, 30))
    txt_surface = pygame.font.Font(None, 32).render(user_name, True,
color)
    width = max(200, txt_surface.get_width()+10)
    input_rect.w = width
    display.blit(label_text, label_pos)
    display.blit(txt_surface, (input_rect.x+5, input_rect.y+5))
    pygame.draw.rect(display, color, input_rect, 2)
    pygame.display.flip()
    clock.tick(30)
return user_name

def check_high_score(score):
    try:
        with open("high_score.txt", "r") as file:
            high_score = int(file.read())
    except FileNotFoundError:
        high_score = 0

    if score > high_score:

```

```

        with open("high_score.txt", "w") as file:
            file.write(str(score))

def get_high_score():
    try:
        with open("high_score.txt", "r") as file:
            high_score = int(file.read())
    except FileNotFoundError:
        high_score = 0
    return high_score

# Get user name
user_name = get_user_name(width)

score = 0

game_over = False

while not game_over:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            game_over = True
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT:
                snake_direction = "left"
            elif event.key == pygame.K_RIGHT:
                snake_direction = "right"
            elif event.key == pygame.K_UP:
                snake_direction = "up"
            elif event.key == pygame.K_DOWN:
                snake_direction = "down"

    if snake_direction == "right":
        snake[0] = (snake[0][0] + snake_block, snake[0][1])
    if snake_direction == "left":
        snake[0] = (snake[0][0] - snake_block, snake[0][1])
    if snake_direction == "up":
        snake[0] = (snake[0][0], snake[0][1] - snake_block)
    if snake_direction == "down":
        snake[0] = (snake[0][0], snake[0][1] + snake_block)

```

```

if (
    apple[0] <= snake[0][0] <= apple[0] + snake_block and
    apple[1] <= snake[0][1] <= apple[1] + snake_block
):
    apple = (
        round(random.randrange(0, width - snake_block) / 10.0) * 10.0,
        round(random.randrange(0, height - snake_block) / 10.0) * 10.0,
    )
    snake.append((-1, -1))
    score += 1
    bite_sound.play() # Play bite sound effect

if (
    snake[0][0] >= width
    or snake[0][0] < 0
    or snake[0][1] >= height
    or snake[0][1] < 0
):
    game_over = True

for block in snake[1:]:
    if block == snake[0]:
        game_over = True

snake = [snake[0]] + snake[:-1]

display.fill(black)

draw_snake(snake)
draw_apple(apple)
message(f"Player: {user_name}    Score: {score}    Hi-Score:
{get_high_score()}", white, (20, 20), size=30)

pygame.display.update()

clock.tick(snake_speed)

check_high_score(score)
message("Game Over", red, (width / 3, height / 3))
pygame.display.update()
time.sleep(3)

```

```
pygame.quit()
quit()

def check_high_score(score):
    try:
        with open("high_score.txt", "r") as file:
            high_score = int(file.read())
    except FileNotFoundError:
        high_score = 0

    if score > high_score:
        with open("high_score.txt", "w") as file:
            file.write(str(score))

def get_high_score():
    try:
        with open("high_score.txt", "r") as file:
            high_score = int(file.read())
    except FileNotFoundError:
        high_score = 0
    return high_score

# Get user name
user_name = get_user_name(width)

score = 0

game_over = False

while not game_over:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            game_over = True
        elif event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT:
                snake_direction = "left"
            elif event.key == pygame.K_RIGHT:
                snake_direction = "right"
            elif event.key == pygame.K_UP:
                snake_direction = "up"
```

```

        elif event.key == pygame.K_DOWN:
            snake_direction = "down"

    if snake_direction == "right":
        snake[0] = (snake[0][0] + snake_block, snake[0][1])
    if snake_direction == "left":
        snake[0] = (snake[0][0] - snake_block, snake[0][1])
    if snake_direction == "up":
        snake[0] = (snake[0][0], snake[0][1] - snake_block)
    if snake_direction == "down":
        snake[0] = (snake[0][0], snake[0][1] + snake_block)

    if (
        apple[0] <= snake[0][0] <= apple[0] + snake_block and
        apple[1] <= snake[0][1] <= apple[1] + snake_block
    ):
        apple = (
            round(random.randrange(0, width - snake_block) / 10.0) * 10.0,
            round(random.randrange(0, height - snake_block) / 10.0) * 10.0,
        )
        snake.append((-1, -1))
        score += 1
        bite_sound.play() # Play bite sound effect

    if (
        snake[0][0] >= width
        or snake[0][0] < 0
        or snake[0][1] >= height
        or snake[0][1] < 0
    ):
        game_over = True

    for block in snake[1:]:
        if block == snake[0]:
            game_over = True

    snake = [snake[0]] + snake[:-1]

    display.fill(black)

    draw_snake(snake)

```



```
draw_apple(apple)
message(f"Player: {user_name}    Score: {score}    Hi-Score:
{get_high_score()}", white, (20, 20), size=30)

pygame.display.update()

clock.tick(snake_speed)

check_high_score(score)
message("Game Over", red, (width / 3, height / 3))
pygame.display.update()
time.sleep(3)
pygame.quit()
quit()
```