

Alien Game Introduction

```
In [ ]: alien = Actor('alien')

WIDTH = 500
HEIGHT = alien.height + 20

def draw():
    screen.clear()
    alien.draw()

def update():
    alien.left += 2
    if alien.left > WIDTH:
        alien.right = 0

def on_mouse_down(pos):
    if alien.collidepoint(pos):
        set_alien_hurt()

def set_alien_hurt():
    alien.image = 'alien_hurt'
    sounds.eep.play()
    clock.schedule_unique(set_alien_normal, 1.0)

def set_alien_normal():
    alien.image = 'alien'
```

Simple Game

Game Description

Name: Flappy Bird There is a yellow square on a blue background that is trying to maneuver out of the way of green pipes. There is a set gap between the pipes through which the square needs to pass through. The square's x position is set, and its y position decreases rapidly to simulate gravity. The pipes move to the left at a set rate.

Changes Made

- 1) I changed the speed of the pipes' leftward movement from 60 to 200. This greatly increased the pace of the game.
- 2) I changed the background color from blue to green and the pipes from green to red.
- 3) I greatly increased the bird speed on_key_down so that it shoots up faster at every click. To counter this, I also greatly increased the y_speed so that it falls much faster as well.

```
In [ ]: import random
```

```
bird_x = 62  
bird_width = 30  
bird_height = 25
```

```
playing_area_width = 300  
playing_area_height = 388
```

```
pipe_space_height = 100  
pipe_width = 54
```

```
def new_pipe_space_y():  
    pipe_space_y_min = 54  
    pipe_space_y = random.randint(  
        pipe_space_y_min,  
        playing_area_height - pipe_space_height - pipe_space_y_min  
    )  
  
    return pipe_space_y
```

```
def reset():  
    global bird_y  
    global bird_y_speed  
    global pipe_1_x  
    global pipe_1_space_y  
    global pipe_2_x  
    global pipe_2_space_y  
    global score  
    global upcoming_pipe  
  
    bird_y = 200  
    bird_y_speed = 0  
  
    pipe_1_x = playing_area_width  
    pipe_1_space_y = new_pipe_space_y()  
  
    pipe_2_x = playing_area_width + ((playing_area_width + pipe_width) / 2)  
    pipe_2_space_y = new_pipe_space_y()  
  
    score = 0  
  
    upcoming_pipe = 1
```

```
reset()
```

```
def update(dt):  
    global bird_y  
    global bird_y_speed  
    global pipe_1_x  
    global pipe_2_x  
    global pipe_1_space_y  
    global pipe_2_space_y  
  
    bird_y_speed += 900 * dt  
    bird_y += bird_y_speed * dt
```

```

def move_pipe(pipe_x, pipe_space_y):
    pipe_x -= 200 * dt

    if (pipe_x + pipe_width) < 0:
        pipe_x = playing_area_width
        pipe_space_y = new_pipe_space_y()

    return pipe_x, pipe_space_y

pipe_1_x, pipe_1_space_y = move_pipe(pipe_1_x, pipe_1_space_y)
pipe_2_x, pipe_2_space_y = move_pipe(pipe_2_x, pipe_2_space_y)

def is_bird_colliding_with_pipe(pipe_x, pipe_space_y):
    return (
        # Left edge of bird is to the left of the right edge of pipe
        bird_x < (pipe_x + pipe_width)
        and
        # Right edge of bird is to the right of the left edge of pipe
        (bird_x + bird_width) > pipe_x
        and (
            # Top edge of bird is above the bottom edge of first pipe segment
            bird_y < pipe_space_y
            or
            # Bottom edge of bird is below the top edge of second pipe segment
            (bird_y + bird_height) > (pipe_space_y + pipe_space_height)
        )
    )

if (
    is_bird_colliding_with_pipe(pipe_1_x, pipe_1_space_y)
    or is_bird_colliding_with_pipe(pipe_2_x, pipe_2_space_y)
    or bird_y > playing_area_height
):
    reset()

def update_score_and_closest_pipe(this_pipe, pipe_x, other_pipe):
    global score
    global upcoming_pipe

    if (
        upcoming_pipe == this_pipe
        and bird_x > (pipe_x + pipe_width)
    ):
        score += 1
        upcoming_pipe = other_pipe

    update_score_and_closest_pipe(1, pipe_1_x, 2)
    update_score_and_closest_pipe(2, pipe_2_x, 1)

def on_key_down():
    global bird_y_speed

    if bird_y > 0:
        bird_y_speed = -300

def draw():

```

```

screen.fill((0, 0, 0))

screen.draw.filled_rect(
    Rect(
        (0, 0),
        (playing_area_width, playing_area_height)
    ),
    color=(32, 200, 80)
)

screen.draw.filled_rect(
    Rect(
        (bird_x, bird_y),
        (bird_width, bird_height)
    ),
    color=(224, 214, 68)
)

def draw_pipe(pipe_x, pipe_space_y):
    screen.draw.filled_rect(
        Rect(
            (pipe_x, 0),
            (pipe_width, pipe_space_y)
        ),
        color=(220, 20, 70)
    )

    screen.draw.filled_rect(
        Rect(
            (pipe_x, pipe_space_y + pipe_space_height),
            (pipe_width, playing_area_height - pipe_space_y - pipe_space_height)
        ),
        color=(220, 20, 70)
    )

draw_pipe(pipe_1_x, pipe_1_space_y)
draw_pipe(pipe_2_x, pipe_2_space_y)

screen.draw.text(str(score), (15, 15))

```

```

WIDTH = 300
HEIGHT = 388

```