10_lines_circles.py

```
# IMPORTANT: switch Mu mode to "Pygame Zero" or
# nothing will appear!

WIDTH = 500
HEIGHT = 500  # what are these units?

def draw():
    screen.clear()
    screen.draw.circle((250, 250), 50, "white")
    screen.draw.filled_circle((250, 100), 50, "red")
    screen.draw.line((150, 20), (150, 450), "purple")
    screen.draw.line((150, 20), (350, 20), "purple")

# TODO: draw a picture
# Make sure you understand (x,y) co-ordinates
# In maths this called a 'Cartesian coordinate system'
# and everything we do in Pygame Zero will use it
```

11_moving_boxes.py

```
# to make things move we need to add
# the update() function

WIDTH = 500

box = Rect((20, 20), (100, 100))

def draw():
    screen.clear()
    screen.draw.filled_rect(box, "red")

def update():
    box.x = box.x + 2
    if box.x > WIDTH:
        box.x = 0

#TODO
# make box move faster
# move in different direction
# have two boxes with different colours
```

12_sprites.py

```
# Sprites are very similiar to boxes!
# Click 'Images' to see images available
#   alien.png should already be there, but
#   for other images you must add them yourself

WIDTH = 500

alien = Actor('alien')
alien.x = 0
alien.y = 50

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

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

#TODO
# draw or download your own image to use instead of alien
```

13_keyboard_input.py

```
alien = Actor('alien')
alien.pos = (0,50)

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

def update():
    if(keyboard.right):
        alien.x = alien.x + 2
    elif(keyboard.left):
        alien.x = alien.x - 2

#TODO
# make the alien move up and down as well as left and right
# use the += operator to change the alien.x more concisely
# use the 'or' operator to allow WASD keys to move the alien
#   in addition to the cursor keys
# make alien wrap around when he moves off edge of screen
```

14_joystick_input.py

```
# Some game controllers have different inputs and some are not be compatible so don’t be
# surprised if this doesnt quite work properly!  Use joystick_tester.py to test yours.

import pygame

joystick = pygame.joystick.Joystick(0)
joystick.init()

alien = Actor("alien")
alien.pos = (0, 50)

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

def update():
    if keyboard.right or joystick.get_axis(0) > 0.1:
        alien.x = alien.x + 2
    elif keyboard.left or joystick.get_axis(0) < -0.1:
        alien.x = alien.x - 2

# TODO
# make the alien move up and down as well as left and right
```

15_collisions.py

```
# Most of this code is copied from programs 11 and 13

WIDTH = 500
HEIGHT = 500

alien = Actor("alien")
alien.pos = (0, 50)
box = Rect((20, 20), (100, 100))

def draw():
    screen.clear()
    screen.draw.filled_rect(box, "red")
    alien.draw()

def update():
    if keyboard.right:
        alien.x = alien.x + 2
    elif keyboard.left:
        alien.x = alien.x - 2
    box.x = box.x + 2
    if box.x > WIDTH:
        box.x = 0
    if alien.colliderect(box):
        print("hit")


# TODO
# joystick input (again), vertical movement (again)
# make the box chase the alien
# print number of times hit (the score)
```

16_sound.py

```
# Most of this code is copied from program 15
WIDTH = 500
alien = Actor("alien")
alien.pos = (0, 50)
box = Rect((20, 20), (100, 100))

def draw():
    screen.clear()
    screen.draw.filled_rect(box, "red")
    alien.draw()
def update():
    if keyboard.right:
        alien.x = alien.x + 2
    elif keyboard.left:
        alien.x = alien.x - 2
    box.x = box.x + 2
    if box.x > WIDTH:
        box.x = 0
# PLAY SOUND AND SHOW IMAGE WHEN HIT
    if alien.colliderect(box):
        alien.image = 'alien_hurt'
        sounds.eep.play()
    else:
        alien.image = 'alien'
# TODO
# Record your own sound effect
# Add more boxes or sprites that move in different ways to avoid
# Add a second alien controlled by different keys or gamepad
```

17_mouse_input.py

```
# changes: box has been removed
# mouse function for clicking on alien
# score display

WIDTH = 500
HEIGHT = 500

alien = Actor("alien")
alien.pos = (0, 50)
score = 0

def draw():
    screen.clear()
    alien.draw()
    screen.draw.text("Score "+str(score), (0,0))

def update():
    if keyboard.right:
        alien.x = alien.x + 2
    elif keyboard.left:
        alien.x = alien.x - 2
    alien.image = 'alien'

def on_mouse_down(pos, button):
    global score
    if button == mouse.LEFT and alien.collidepoint(pos):
        alien.image = 'alien_hurt'
        sounds.eep.play()
        score = score + 1
```

18_mouse_movement.py

```
alien = Actor("alien")

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

def on_mouse_move(pos):
    alien.pos = pos

#TODO:
# what happens if you delete line 8 and replace it with this:
#
#   animate(alien, pos=pos, duration=1, tween=’bounce_end’)
#
# what happens if you change on_mouse_move to on_mouse_down?
# can you make a game with one alien controlled by mouse
#    and another controlled by keyboard?
```

19_joystick_tester.py

```
import pygame

def update():
    screen.clear()
    joystick_count = pygame.joystick.get_count()
    y = 0
    for i in range(joystick_count):
        joystick = pygame.joystick.Joystick(i)
        joystick.init()
        name = joystick.get_name()
        axes = joystick.get_numaxes()
        buttons = joystick.get_numbuttons()
        hats = joystick.get_numhats()
        screen.draw.text(
            "Joystick {} name: {} axes:{} buttons:{} hats:{}".format(
                i, name, axes, buttons, hats
            ),
            (0, y))
        y += 14
        for i in range(axes):
            axis = joystick.get_axis(i)
            screen.draw.text("Axis {} value: {:>6.3f}".format(i, axis), (20, y))
            y += 14
        for i in range(buttons):
            button = joystick.get_button(i)
            screen.draw.text("Button {:>2} value: {}".format(i, button), (20, y))
            y += 14
        for i in range(hats):
            hat = joystick.get_hat(i)
            screen.draw.text("Hat {} value: {}".format(i, str(hat)), (20, y))
            y += 14
```

20_loops.py

```
WIDTH = 500
HEIGHT = 500
def draw():
    screen.clear()
    for x in range(0, WIDTH, 40):
        screen.draw.filled_circle((x, 20), 20, "red")

    for x in range(0, WIDTH, 40):
        for y in range(60, HEIGHT, 40):
            screen.draw.filled_circle((x, y), 10, "green")

#TODO:
# import random and make the positions random
# learn about RGB colour and make random colours
# create a timer variable and change colours
#   based on time
```

21_arrays.py

```
WIDTH = 500

aliens = []
for i in range(0,10):
    aliens.append(Actor(’alien’, (i*30, i*30)))

def draw():
    screen.clear()
    for alien in aliens:
        alien.draw()

def update():
    for alien in aliens:
        alien.x += 2
        if alien.x > WIDTH:
            alien.x = 0

def on_mouse_down(pos, button):
    aliens.append(Actor(’alien’, pos))

#TODO:
# Go back to your previous game (program 15)
# make an array of bullets that shoot when you
# press the space bar
```