

Pygame lesson 4

In this lesson you are going to learn a bit about working with collections of objects. We will start with a reasonably simple one, and make snow fall from the top of the screen. I am modifying the “Animating Snow” from this source:

http://programarcadegames.com/index.php?chapter=introduction_to_animation&lang=en

Here is some code to get you started:

```
# Animating snow
# Source:
http://programarcadegames.com/index.php?chapter=introduction_to_animation&lang=en

# Imports
import pygame
import random

# Initialize the game engine
pygame.init()

BLACK = [0, 0, 0]
WHITE = [255, 255, 255]

# Set the height and width of the screen
SIZE = [400, 400]

screen = pygame.display.set_mode(SIZE)
pygame.display.set_caption("Snow Animation")

# Create an empty array
snow_list = []

# Loop 50 times and add a snow flake in a random x,y position
for i in range(50):
    x = random.randrange(0, 400)
    y = random.randrange(0, 400)
    snow_list.append([x, y])

clock = pygame.time.Clock()

# Loop until the user clicks the close button.
done = False
while not done:

    for event in pygame.event.get(): # User did something
        if event.type == pygame.QUIT: # If user clicked close
            done = True # Flag that we are done so we exit this loop

    # Set the screen background
    screen.fill(BLACK)
```

```
# Process each snow flake in the list
for i in range(len(snow_list)):

    # Draw the snow flake
    pygame.draw.circle(screen, WHITE, snow_list[i], 2)

    # Move the snow flake down one pixel
    snow_list[i][1] += 1

# Go ahead and update the screen with what we've drawn.
pygame.display.flip()
clock.tick(20)
```

pygame.quit()

A few notes:

```
snow_list=[]

for i in range(50):
    x = randrange(0,400)
    y = randrange(0,400)
    snow_list.append([x,y])
```

Create a list for all the snowflakes. This means you don't have to come up with tons of silly names. It also makes it much easier to iterate through all the snowflakes and check if they have fallen off the screen

Put each snowflake in a random position on the screen. Add each to the list

```
# Process each snow flake in the list
for i in range(len(snow_list)):

    # Draw the snow flake
    pygame.draw.circle(screen, WHITE, snow_list[i], 2)

    # Move the snow flake down one pixel
    snow_list[i][1] += 1
```

Iterate through each snowflake in the list, draw it, then move it ready for the next cycle through the program.

You will have seen that your snowflakes disappear off the bottom and don't come back. This looks a bit silly.

Challenge: make your snowflakes appear back at the top of the screen if they disappear. Spoiler – there is a solution at the top of the next page:

Finishing off the snowflake challenge:

```
# Move the snow flake down one pixel
snow_list[i][1] += 1

# If the snow flake has moved off the bottom of the screen
if snow_list[i][1] > 400:
    # Reset it just above the top, try setting y to 0
    # to see why this looks so much better
    y = random.randrange(-50, -10)
    snow_list[i][1] = y
    # Give it a new x position
    x = random.randrange(0, 400)
    snow_list[i][0] = x
```

The method we have just used of creating a list for all the snowflakes works because each snowflake is virtually identical to every other one:

- They move the same amount in the same direction
- They are the same colour
- They are the same size

If we want something more complicated than this, we really need to use object orientation.