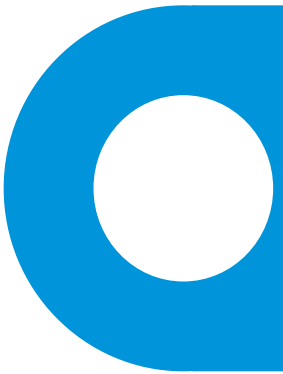


# Python & Pygame

# Python & Pygame

- The Python interpreter software can be downloaded from the official website of the Python programming language,

  http://www.python.org

  - A video tutorial of how to install Python is available from this book's website at at http://invpy.com/installing

- You will have to download and install Pygame, which is as easy as downloading and installing the Python interpreter.
  - In a web browser, go to the URL

    http://pygame.org
  - A video tutorial of how to install Pygame is available from this book's website at http://invpy.com/videos.

- http://inventwithpython.com/pygame/
  - You can read the book for free
  - You can download all the available related materials for free

# Important when you using book's code

- Be sure to enter the code exactly as it appears.
- Notice that some of the lines don't begin at the leftmost edge of the page, but are indented by four or eight or more spaces.
- Be sure to put in the correct number of spaces at the start of each line.

# PYGAME BASICS

- The Pygame framework includes several modules with functions for drawing graphics, playing sounds, handling mouse input, and other things.
- If you have trouble with some of the programming concepts
  - “Invent Your Own Computer Games with Python”
  - http://invpy.com/book
  - The “Invent with Python” book also has a few chapters covering Pygame

    http://invpy.com/chap17.
  - This book is aimed at complete beginners to programming, and also has a few chapters covering Pygame.

# GUI vs. CLI

- The Python programs only deal with text through the print() and input() functions.
  - Your program can display text on the screen and let the user type in text from the keyboard.
  - This type of program has a command line interface, or CLI.
- Pygame provides functions for creating programs with a graphical user interface, or GUI.
  - Instead of a text-based CLI, programs with a graphics-based GUI can show a window with images and colors.

# Hello World with Pygame

- …..the most boring video game
  - Calling the print() function to make text appear in the window won't work because print() is a function for CLI programs.
  - The same goes for input() to get keyboard input from the user.

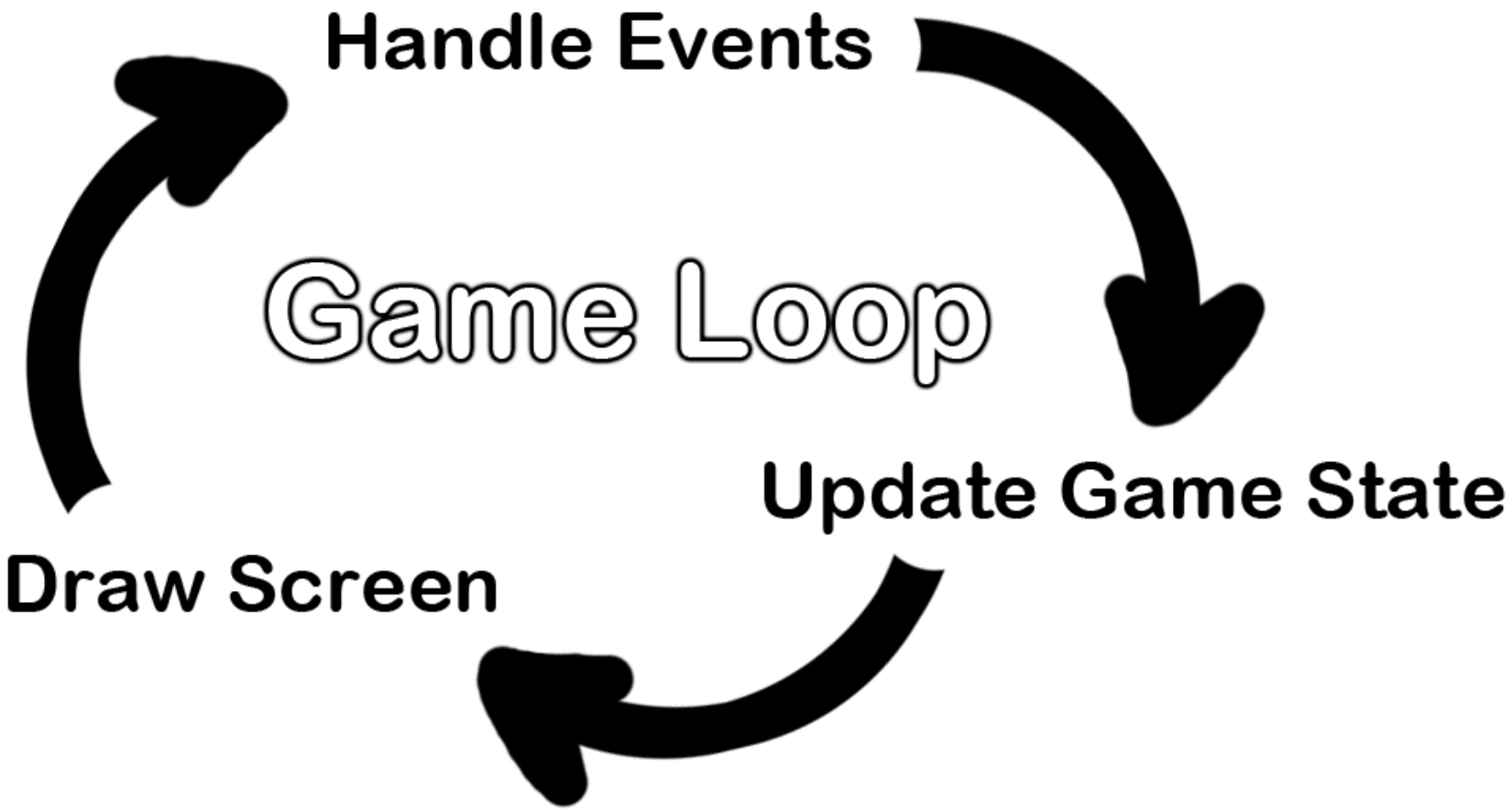
# Hello World with Pygame

```
import pygame, sys
from pygame.locals import *

pygame.init()
DISPLAYSURF = pygame.display.set_mode((400, 300))
pygame.display.set_caption('Hello World!')
while True: # main game loop
    for event in pygame.event.get():
        if event.type == QUIT: pygame.quit() sys.exit()
    pygame.display.update()
```

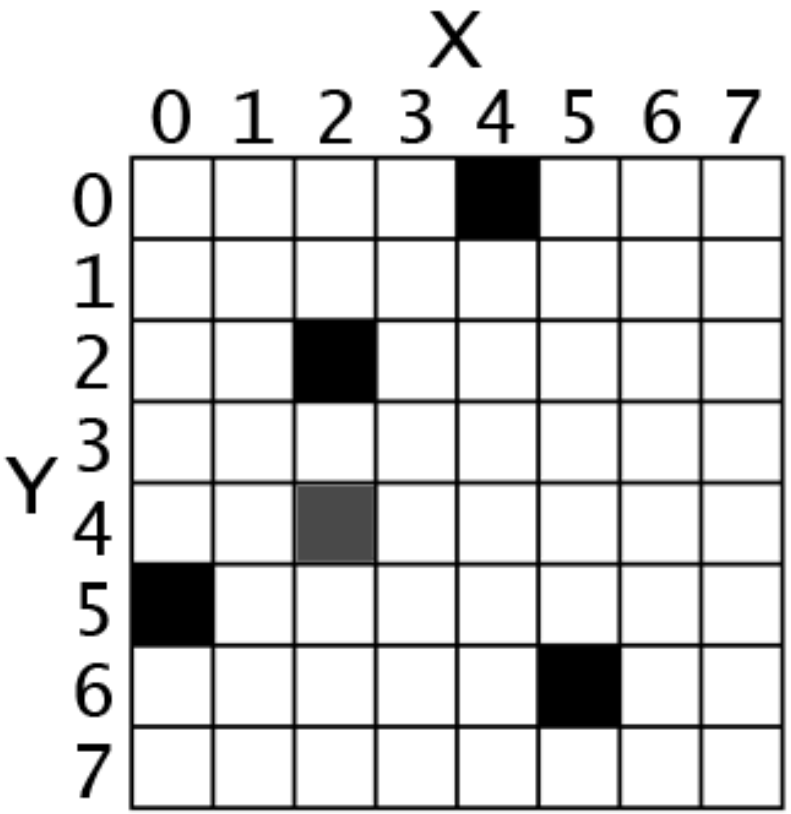
# Main loop

- The games in this book all have these while True loops in them along with a comment calling it the —main game loopll. A **game loop** (also called a **main loop**) is a loop where the code does three things:
- Handles events.
- Updates the game state.
- Draws the game state to the screen.


Handle Events
Game Loop
Update Game State
Draw Screen

# Pixel Coordinates

- We can refer to a specific pixel by using a Cartesian Coordinate system.
  - Each column of the X-axis and each row of the Y-axis
  - XY coordinates are also called points.
  - The Y-axis starts at 0 at the top and then increases going down.
- The Pygame framework represents Cartesian Coordinates as a tuple of two integers
  - (4, 0) or (2, 2)
  - Cartesian Coordinates are covered in more detail in chapter 12 of “Invent Your Own Computer Games with Python” at http://invpy.com/chap12



# Surface Object and Window

- Surface objects are objects that represent a rectangular 2D image
- The window border, title bar, and buttons are not part of the display Surface object
- the Surface object returned by pygame.display.set_mode() is called the display Surface
- Anything that is drawn on the display Surface object will be displayed on the window when the pygame.display.update() function is called

# Colors

- There are three primary colors of light: red, green and blue.
  - By combining different amounts of these three colors you can form any other color.
- In Pygame, colors are represented with tuples of three integers.
  - The first value in the tuple is how much red is in the color.
  - An integer value of 0 means there is no red in this color, and a value of 255 means there is the maximum amount of red in the color.
  - The second value is for green and the third value is for blue.
- These tuples of three integers used to represent a color are often called RGB values.
  - Pygame can draw 16,777,216 different colors (that is, 256 x 256 x 256 colors).
  - If try to use a number larger than 255 or a negative number, you will get an error that looks like “ValueError: invalid color argument”.
- For example,
  - we will create the tuple (0, 0, 0) and store it in a variable named BLACK.
  - With no amount of red, green, or blue, the resulting color is completely black.
  - The color black is the absence of any color.

# Transparent colors

- When you look through a glass window that has a deep red tint, all of the colors behind it have a red shade added to them.
- You can mimic this effect by adding a fourth integer value to your color values.
- This value is known as the alpha value.
- It is a measure of how opaque (that is, not transparent) a color is.
- When you draw a pixel onto a surface object, the new color completely replaces whatever color was already there.
- But with colors that have an alpha value, you can instead just add a colored tint to the color that is already there.
- An alpha value of
  - 255 means the color is completely opaque (that is, not transparency at all)
  - 0 means the color is completely transparent.
- In order to draw using transparent colors, you must create a Surface object with the convert_alpha() method.
  - For example,

```
anotherSurface = DISPLAYSURF.convert_alpha()
```

# Basic shapes in Pygame

- Pygame provides several different functions for drawing different shapes onto a surface object.
- These shapes such as rectangles, circles, ellipses, lines, or individual pixels are often called drawing primitives.

# Animation in Pygame

- The trick to making believable looking animation is to have your program draw a picture to the window, wait a fraction of a second, and then draw a slightly different picture.

# Fonts in Pygame

- There are six steps to making text appear on the screen:
  - Create a pygame.font.Font object
  - Create a Surface object with the text drawn on it by calling the Font object's render() method.
  - Create a Rect object from the Surface object by calling the Surface object's get_rect() method.
    - This Rect object will have the width and height correctly set for the text that was rendered, but the top and left attributes will be 0.
  - Set the position of the Rect object by changing one of its attributes.
  - Blit the Surface object with the text onto the Surface object returned by pygame.display.set_mode().
  - Call pygame.display.update() to make the display Surface appear on the screen.

# Sounds

- Pygame can load WAV, MP3, or OGG files
- Example

```
soundObj = pygame.mixer.Sound('beeps.wav')
soundObj.play()
import time time.sleep(1) # wait and let the sound play for 1 second
soundObj.stop()
```

- Example - background music (WAV, MP3, or MIDI)

```
pygame.mixer.music.load(‘backgroundmusic.mp3')
pygame.mixer.music.play(-1, 0.0)
```