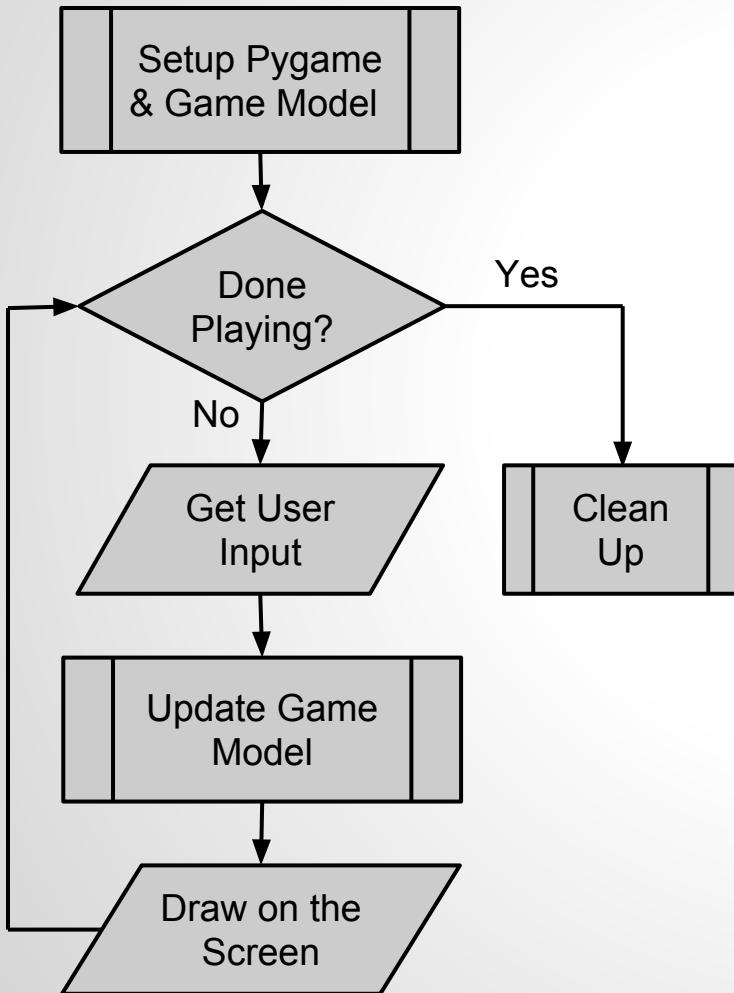


Using Pygame

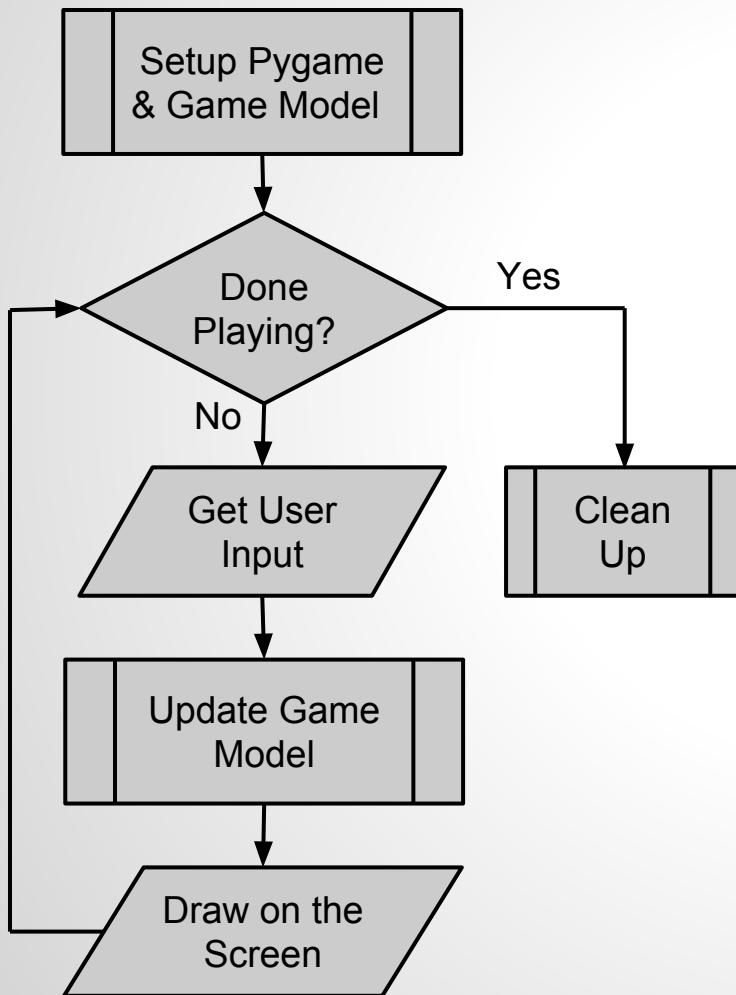
CSCI1300 - Fall 2013

How Games Work - High Level



- **Initialization**
 - Game Engine (pygame)
 - Game Model (your stuff)
- **Game / Event Loop**
 - Run Until User Quits
 - Get User Input
 - Update Game Model
 - Draw Stuff On Screen
 - Repeat
- **Clean Up**

How Pygame Works



```
# Setup Pygame
import pygame

playing = False
screen = pygame.display.set_mode((640,480))

# Setup Game Model (YOUR WORK)

while playing:      # Done Playing?
    # Get User Input
    user_input = pygame.events.get()

    # Update Game Model (YOUR WORK)

    # Draw on the Screen
    screen.blit(your_image, (x,y))
    pygame.display.flip()

pygame.quit()          # Clean Up
```

Drawing on the Screen

- Create a screen / window

```
screen = pygame.display.set_mode( (WIDTH, HEIGHT) )
```

How big a screen you want?

- Load an image

```
cat_image = pygame.image.load("research_cat.png")
```

What's your filename?

- Blit it on your screen

```
screen.blit(cat_image, (x, y) )
```

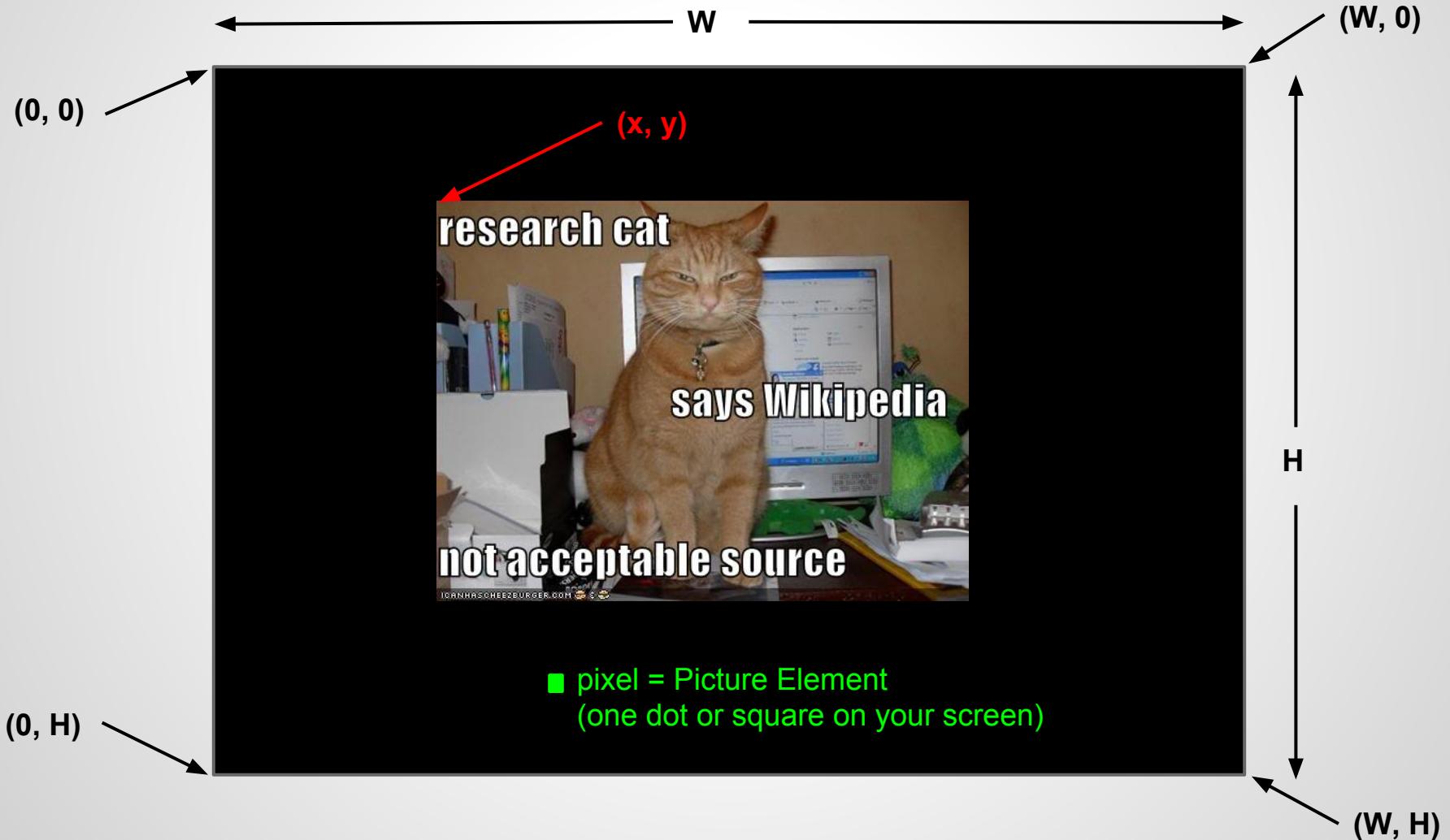
Where do you want this?

- Nothing yet? Flip the display

```
pygame.display.flip()
```

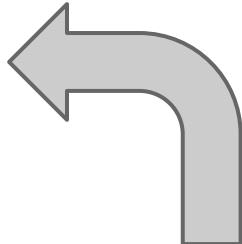
- TADA!

Drawing on the Screen

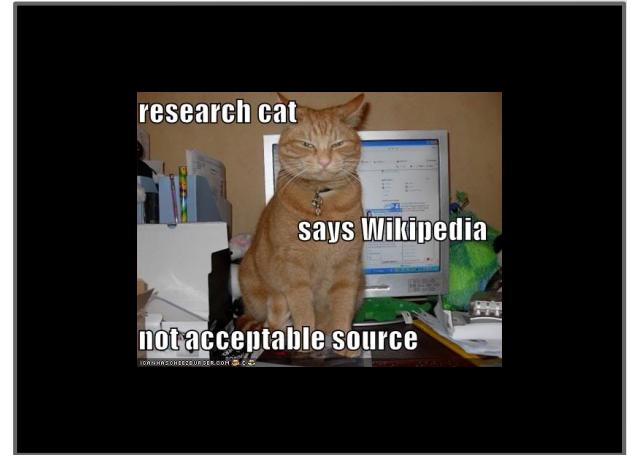


Flip, Huh?

- What's this pygame.display.flip() anyway?



Player Sees This

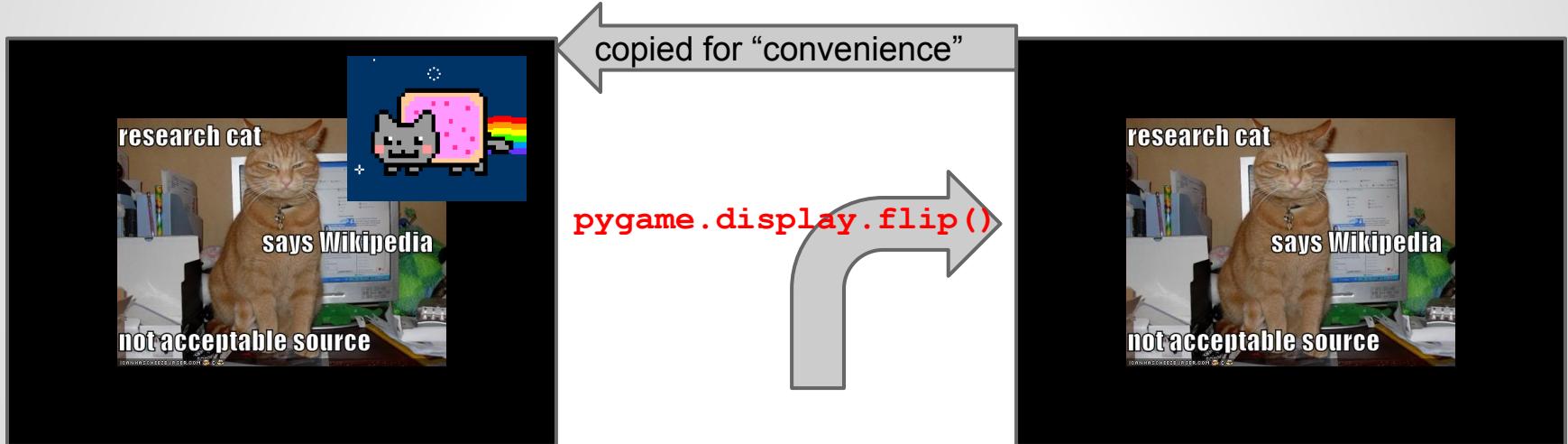


You Draw Here



Flip, Huh?

- What's this pygame.display.flip() anyway?



Now, You Draw Here!



Player Sees This

How About Input?

- User input is called “events” in Pygame
 - key presses, mouse movement and clicks, joysticks...
- First, we ask Pygame what’s happened

```
user_input = pygame.event.get()
```

- Is a key on the keyboard down?

```
pressed_keys = pygame.key.get_pressed()    # List of key
if pressed_keys[K_m]:                      # Asking about
    'm'
    print("The 'm' key is pressed!")
```

How About Input?

- Or, ask about the mouse
- Where is the mouse?

```
mouse_position = pygame.mouse.get_pos()  
print("The mouse is at the point", mouse_position)
```

- Which buttons are down?

```
mouse_buttons = pygame.mouse.get_pressed() # Just like  
key!  
  
if mouse_buttons[0]: # Asking  
about  
  
    print("The left mouse button is down") # left button
```

Collision

- Games usually require checking if something hits something else
 - Did the player hit an enemy?
 - Did the player click on something?
- Images are called “sprites”
- To check for collision, draw a “bounding box” around the sprite



```
lolcat_box = Rect( (x, y), lolcat.get_size())
```

Upper left of the bounding box

Width and height of the image

Collision

- Did lolcat hit nyan_cat? Use `colliderect`

```
lolcat_box = Rect( (x1, y1), lolcat.get_size())
nyan_cat_box = Rect( (x2, y2), nyan_cat.get_size())
if lolcat_box.colliderect(nyan_cat_box):
    print("Meow!")
```

- Did mouse click lolcat? Use `collidepoint`

```
mouse_position = pygame.mouse.get_pos()
if lolcat_box.collidepoint(mouse_position):
    print("Om Nom!")
```

All Done!

- Remember, clean up your program!
`pygame.quit() # Outside the event loop`
- Just like `my_file.close()`

