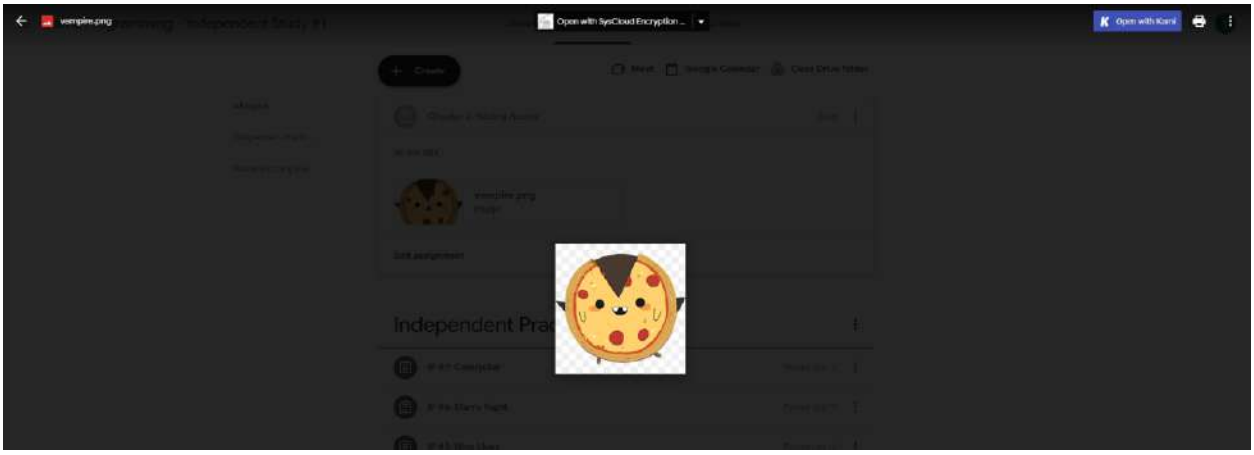
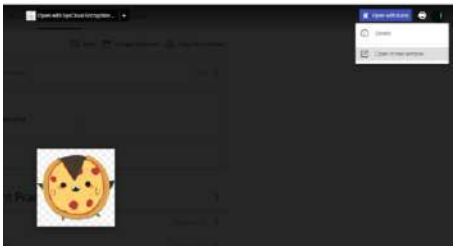
Chapter 2 Practice Directions

1. Before opening up IDLE, we need to save the game assets that we will be using this chapter to our game directory folder. Assets are images, videos, or other media that are not code but are part of your game. Navigate out to Google Classroom and find the assignment for this Chapter.

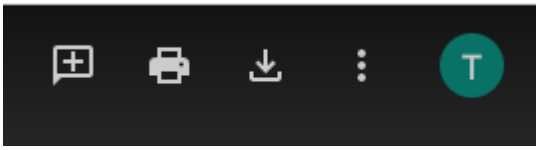
2. Select the vampire.png image to view the preview screen for the image.



3. Click the three dots at the top right corner and select the “Open in new window” option from the list.



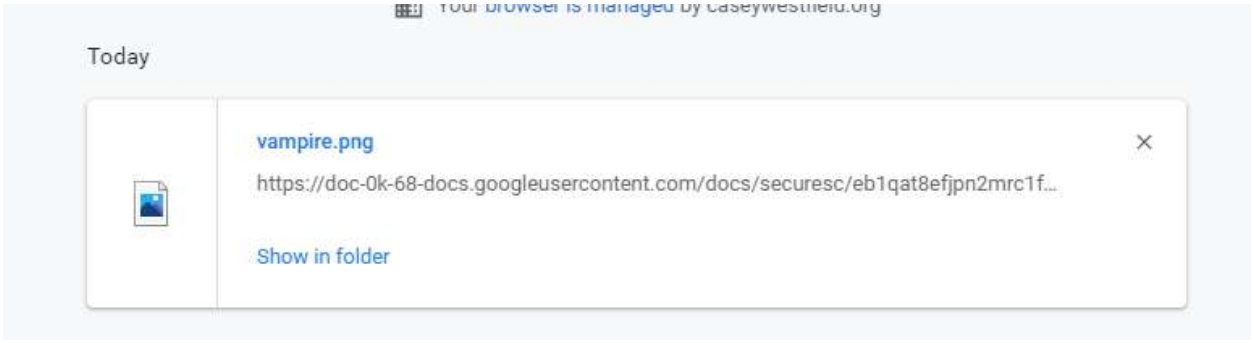
4. In the new window that opens up, click the Download button in the top right corner to download the image.



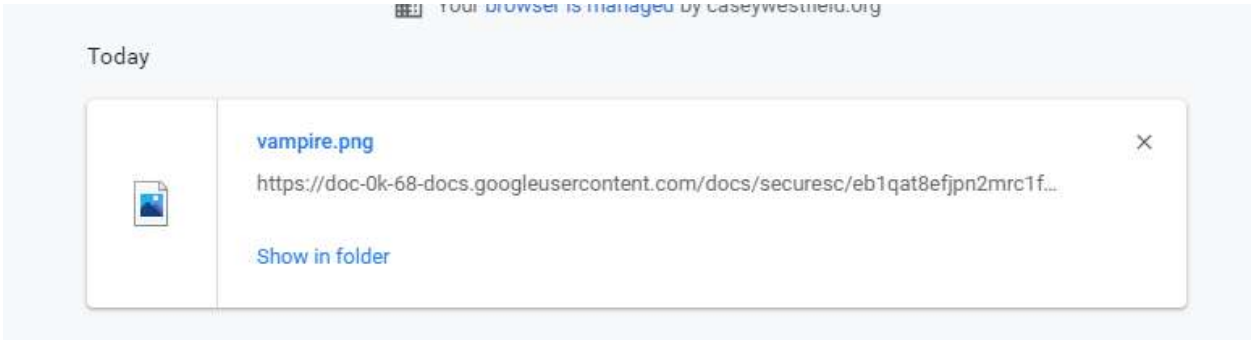
5. You will see your downloaded image appear in a task bar along the bottom of your browser window.

6. Our game assets must be saved in the same folder on the PC that our actual game file is saved in. That is why we created the game directory folder last chapter. Click the “Show All” button in the bottom right corner on the task bar.

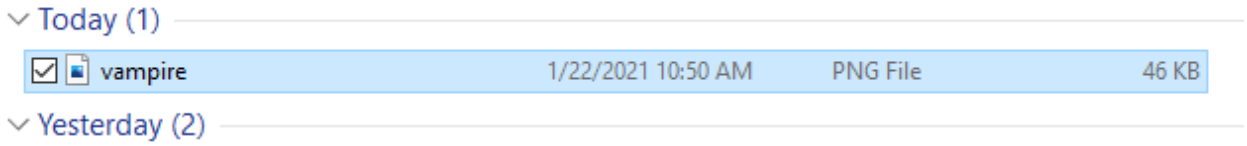
7. A list of all of your Downloads will open up.



8. Click “Show in folder”.

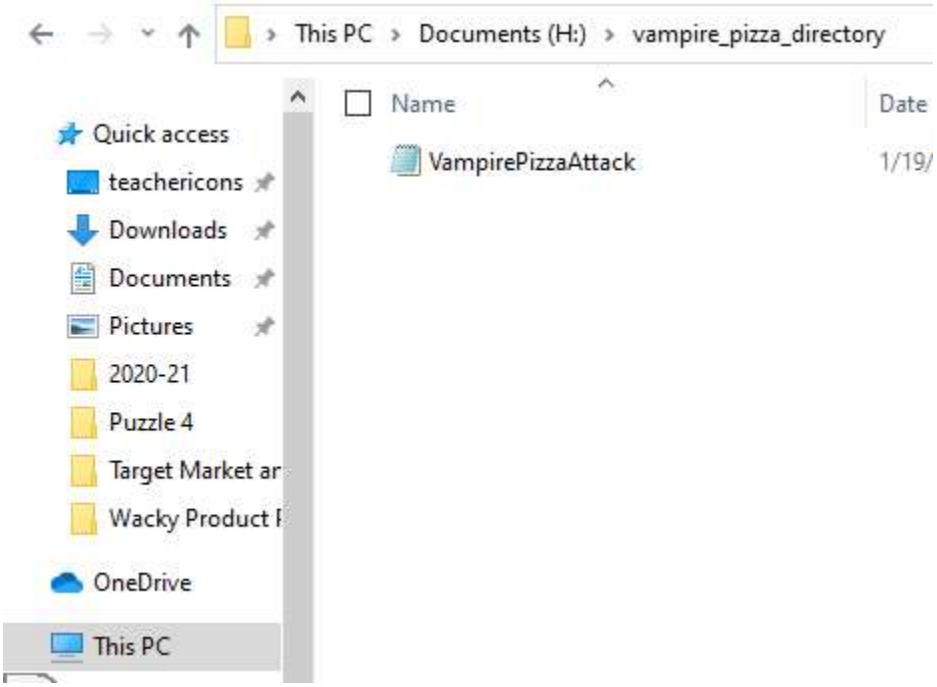


9. This will take you to your Downloads folder on your PC.

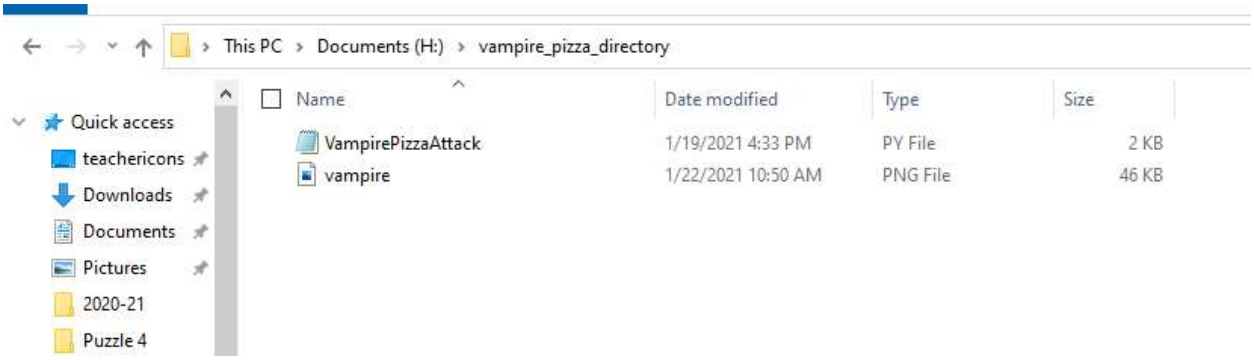


10. Copy or cut the vampire.png file.

11. Navigate to your vampire_pizza_directory folder on your H: or V: drive. You may have to click the “This PC” link in the menu at the left and then find your student drive and the appropriate folder.

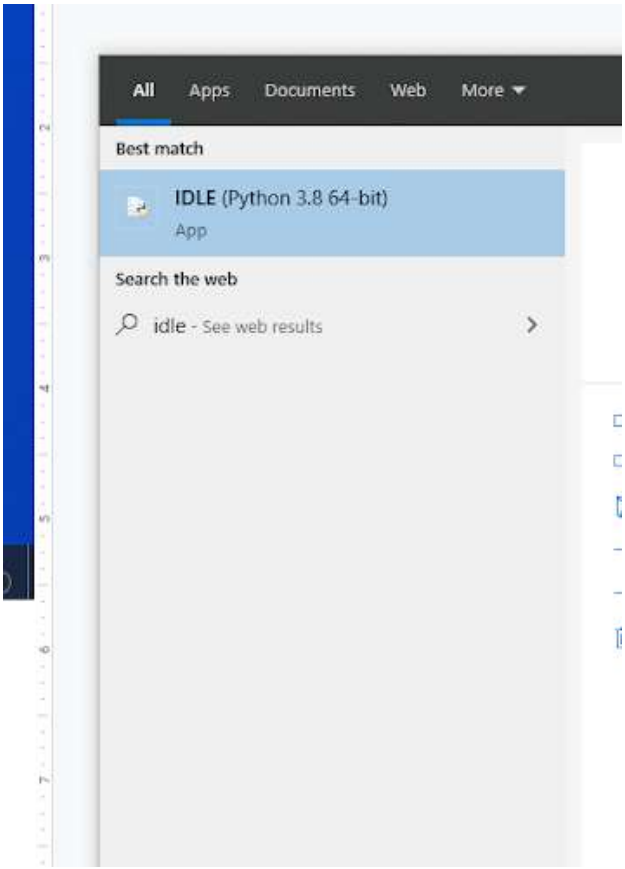


12. Paste the image file into your vampire_pizza_directory folder.



13. You can close out of your Windows Explorer window after you have saved your image to the folder.

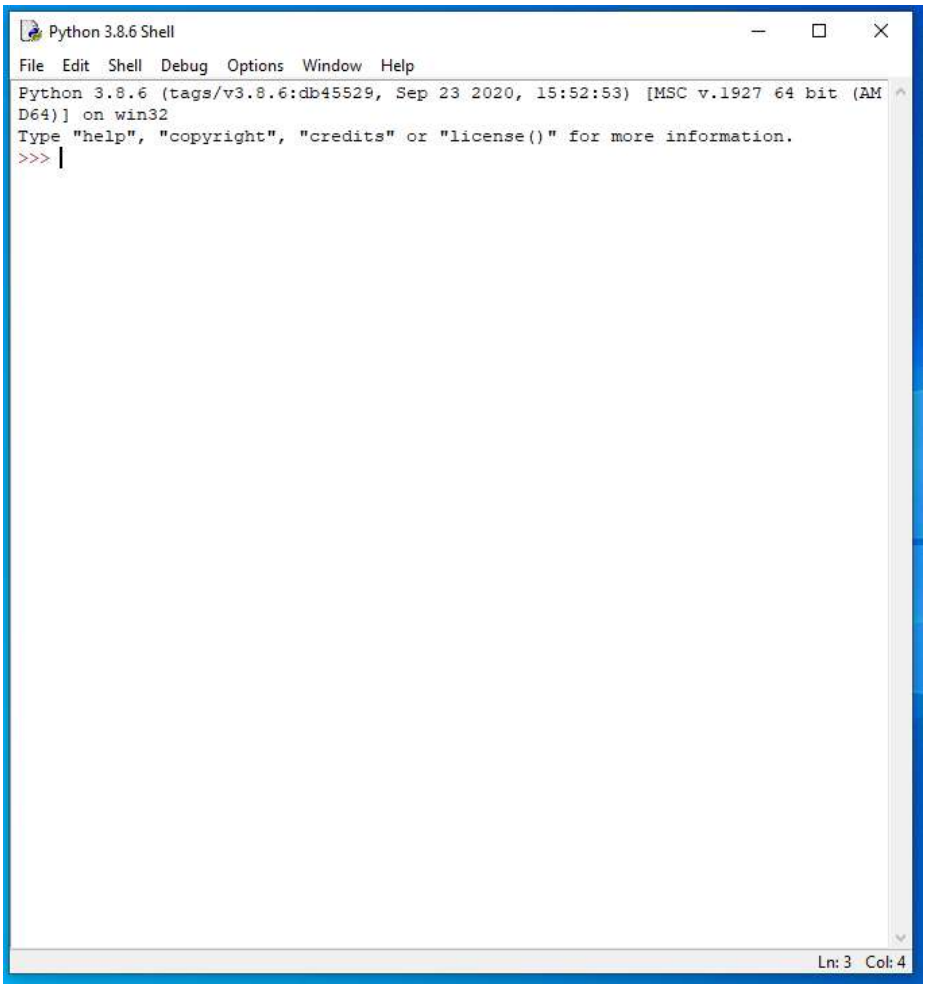
14. We are now ready to start adding code to our file that we created in Chapter 1. Using your Windows button menu, find and launch your IDLE program.



IDLE is the integrated development environment associated with Python. It is made up of a code editor where you type your code along with other helpful tools that allow you to write, save, and test run programs.

IDLE is designed to recognize Python code, compile Python code, and provide basic debugging tips to programmers if there are problems with their code.

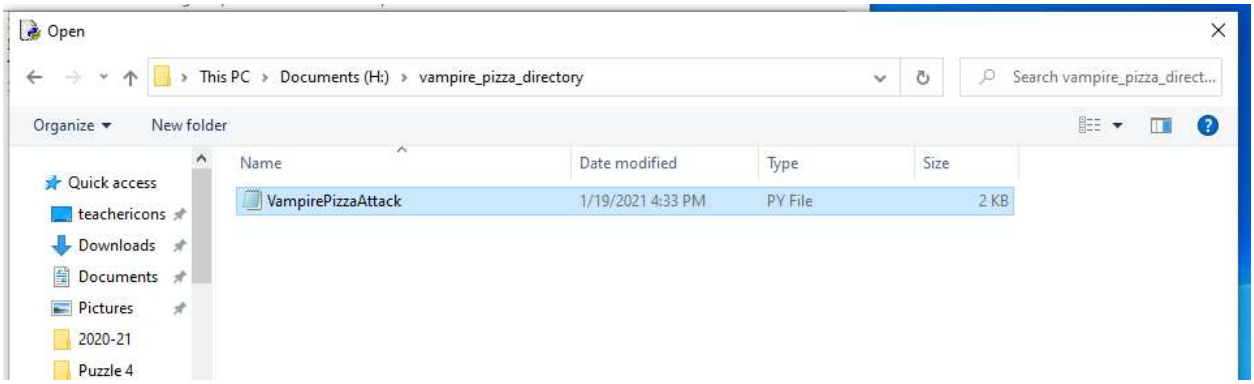
15. Your IDLE window should look something like this once it has launched.:



On Startup, IDLE will display the Python Shell, which can be used to give commands to the computer's operating system. Since we are viewing the shell through IDLE and not the actual command prompt window, the commands that we type into the Shell will not communicate directly with our operating system. However, you can type similar commands in the Python Shell directly from the Python program (not through IDLE) and, if you have permission to access the operating system's commands, you can communicate with the computer's operating system that way.

In IDLE, the shell is mainly used as a launching screen for other activities that we will do, like writing code for our game or debugging a file.

16. Go to File > Open and then browse to find your VampirePizzaAttack file that we created last chapter and open it.



17. Your Python file and code from last chapter will open up.

18. We will now begin to code the next part of our game. I like to make my coding window larger so that I can see all of my code a bit better, but that is a personal decision. Remember as we move through these exercises that your spelling, capitalization, and indentation should match. If it doesn't, your program likely won't work.

19. Click at the end of Line 20 after the code to set the display caption.

```
#-------------------------------------------------------------
#Define constant variables

#Define the parameters of the game window
WINDOW_WIDTH = 900
WINDOW_HEIGHT = 400
WINDOW_RES = (WINDOW_WIDTH, WINDOW_HEIGHT)

#-------------------------------------------------------------
#Load assets

#Create window
GAME_WINDOW = display.set_mode(WINDOW_RES)
display.set_caption('Attack of the Vampire Pizzas!')

#-------------------------------------------------------------
```

20. Press ENTER twice.

21. Type the code that you see on Lines 22 – 24 of the screenshot below:

```
18  #Create window
19  GAME_WINDOW = display.set_mode(WINDOW_RES)
20  display.set_caption('Attack of the Vampire Pizzas!')
21
22  #Set up the enemy image
23  #Load the image into the program
24  pizza_img = image.load('vampire.png')
25  #-------------------------------------------------------------
```

Lines 22 and 23 contain comments describing what this block of code does.

Line 24 creates a new variable called pizza_img. The variable's value is set to be equal to the vampire.png image. Remember, all image files must be saved in the same folder as your code file or it won't work. The image.load() command is used to load an image into your game.

22. Press ENTER.

23. Type the code that you see on Lines 25 – 28 of the screenshot below:

```
20 display.set_caption('Attack of the Vampire Pizzas!')
21
22 #Set up the enemy image
23 #Load the image into the program
24 pizza_img = image.load('vampire.png')
25 #Convert the image to a surface
26 pizza_surf = Surface.convert_alpha(pizza_img)
27 VAMPIRE_PIZZA= transform.scale(pizza_surf, (100, 100))
28 GAME_WINDOW.blit(VAMPIRE_PIZZA, (900, 400))
29 #------------------------------------------------------------
```

Line 25 contains another comment explaining what the next few lines of code will do.

Line 26 creates a variable called pizza_surf and sets its value to be the result of the Surface.convert.alpha() method. The Surface.convert.alpha method will convert our pizza_img variable image (in other words, our vampire.png image) to be a surface in our game instead of an actual image. What this means is that the image will be treated the same way as other background elements in our game, like background pictures. This will prevent problems later. If you don't want your user to be able to interact with something in the game, then you should change it to a surface.

So, Line 26 converts the vampire.png image to a surface and assigns that surface to the pizza_surf variable.

Line 27 creates a constant variable called VAMPIRE_PIZZA. The transform.scale method that you see on Line 27 will scale the specified image/variable (in this case, our pizza_surf surface image) to the specified measurements. In this game, our vampire pizza surface image will be 100 pixels wide and 100 pixels high, which is why you see the scale settings on Line 27 as 100, 100.

Once the pizza_surf image is scaled down to the size we want, we assign that new image (the scaled down surface image) to the VAMPIRE_PIZZA variable.

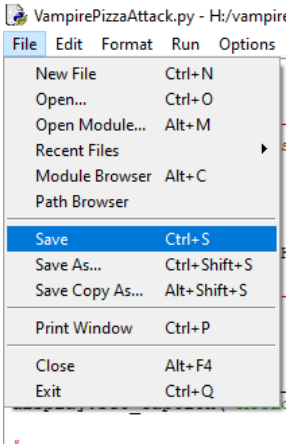
To do a quick review: Line 24 uploads the vampire pizza image into our game and assigns it to the variable of pizza_img. Line 26 takes the pizza_img image and turns it into a surface. Line 27 scales the surface image down and assigns the new, smaller image to the VAMPIRE PIZZA variable.

Line 28 uses the constant GAME_WINDOW variable to blit the VAMPIRE_PIZZA image variable. To blit an image means to display it on the screen in the game. The location coordinates of 900, 400 describe where the game window should blit the image. Remember, you've already scaled down the surface image size to what it needs to be. Now, Line 28 will let you choose where that surface image displays at in your game window.

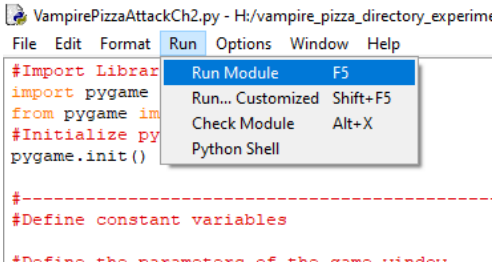
24. Before we go any further, we are going to change our game window size. Right now, with the image blitting at the coordinate of 900, 400, our game window is too small to see the image. At the top of your code (Lines 11 and 12), change the WINDOW_WIDTH variable to 1100 and the WINDOW_HEIGHT variable to 600.

```
pygame.init()

#---------------------------------------------------------------
#Define constant variables

#Define the parameters of the game window
WINDOW_WIDTH = 1100
WINDOW_HEIGHT = 600
WINDOW_RES = (WINDOW_WIDTH, WINDOW_HEIGHT)

#---------------------------------------------------------------
```

25. Go to File > Save. Before we test our game, we need to save it.



26. Now, go to Run > Run Module.



27. After giving your game window time to load, you should see your vampire pizza image in the bottom right corner of your window.

28. You can close out of the Python file. You can also close out of the Python Shell if you still have it open.

Final Code:

```
#Import Libraries
import pygame
from pygame import *
#Initialize pygame
pygame.init()

#-------------------------------------------------------------
#Define constant variables

#Define the parameters of the game window
WINDOW_WIDTH = 1100
WINDOW_HEIGHT = 600
WINDOW_RES = (WINDOW_WIDTH, WINDOW_HEIGHT)

#-------------------------------------------------------------
#Load assets

#Create window
GAME_WINDOW = display.set_mode(WINDOW_RES)
display.set_caption('Attack of the Vampire Pizzas!')

#Set up the enemy image
#Load the image into the program
pizza_img = image.load('vampire.png')
#Convert the image to a surface
pizza_surf = Surface.convert_alpha(pizza_img)
VAMPIRE_PIZZA= transform.scale(pizza_surf, (100, 100))
GAME_WINDOW.blit(VAMPIRE_PIZZA, (900, 400))
#-------------------------------------------------------------
#Start main game loop

#Game loop
game_running = True
while game_running:

#-------------------------------------------------------------
#Check for events

    #Checking for and handling events
    for event in pygame.event.get():
        #Exit loop on quit
        if event.type == QUIT:
            game_running = False

#-------------------------------------------------------------
    #Update display.
    display.update()

#Close main game loop
#-------------------------------------------------------------

#Clean up game
pygame.quit()
```