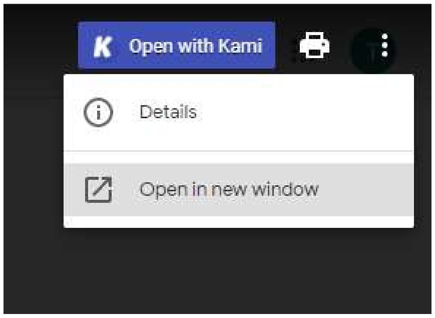
Chapter 3 Practice Directions

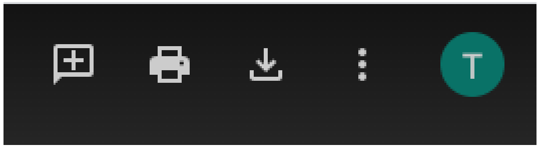
1. Before opening up IDLE, we need to save the game assets that we will be using this chapter to our game directory folder. Assets are images, videos, or other media that are not code but are part of your game. Navigate out to Google Classroom and find the assignment for this Chapter.

2. Select the restaurant.jpg 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.

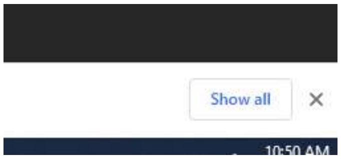


4. In the new window that opens up, click the Download button in the top right corner to download the image.



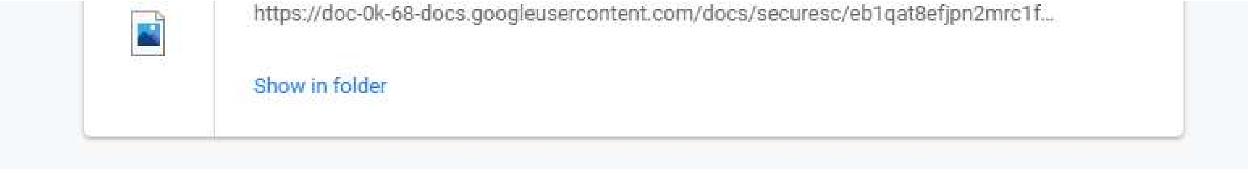
5. You will see your downloaded image appear in a task bar along the bottom of your browser window.

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”.

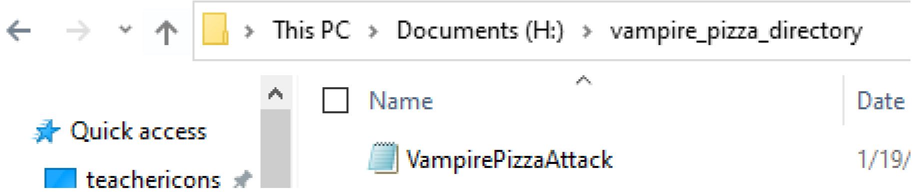


https://doc-0k-68-docs.googleusercontent.com/docs/securesc/eb1qat8efjpn2mrc1f...

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

10. Copy or cut the restaurant.jpg 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.

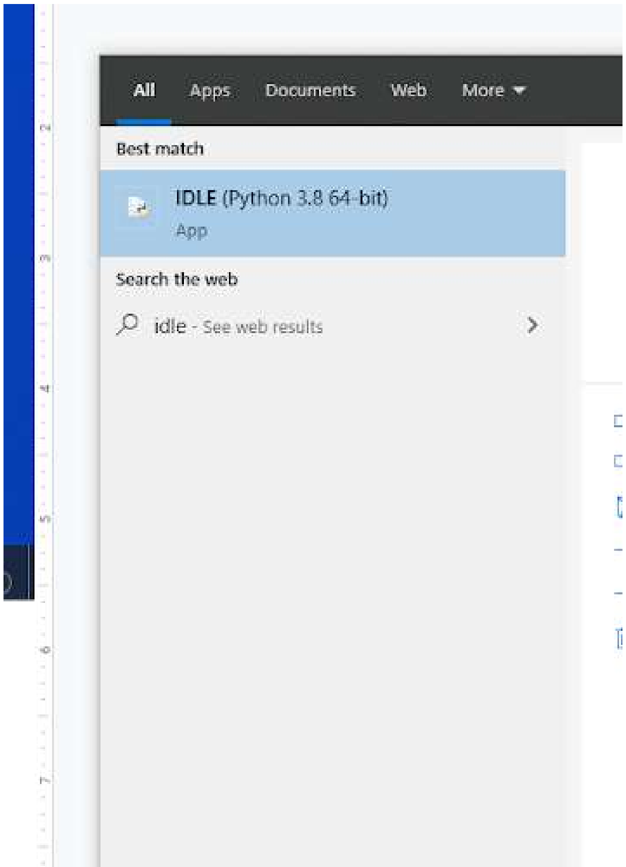


This PC > Documents (H:) > vampire_pizza_directory

12. Paste the image file into your vampire_pizza_directory folder.

13. You can close out of your Windows Explorer window after you have made your folder.

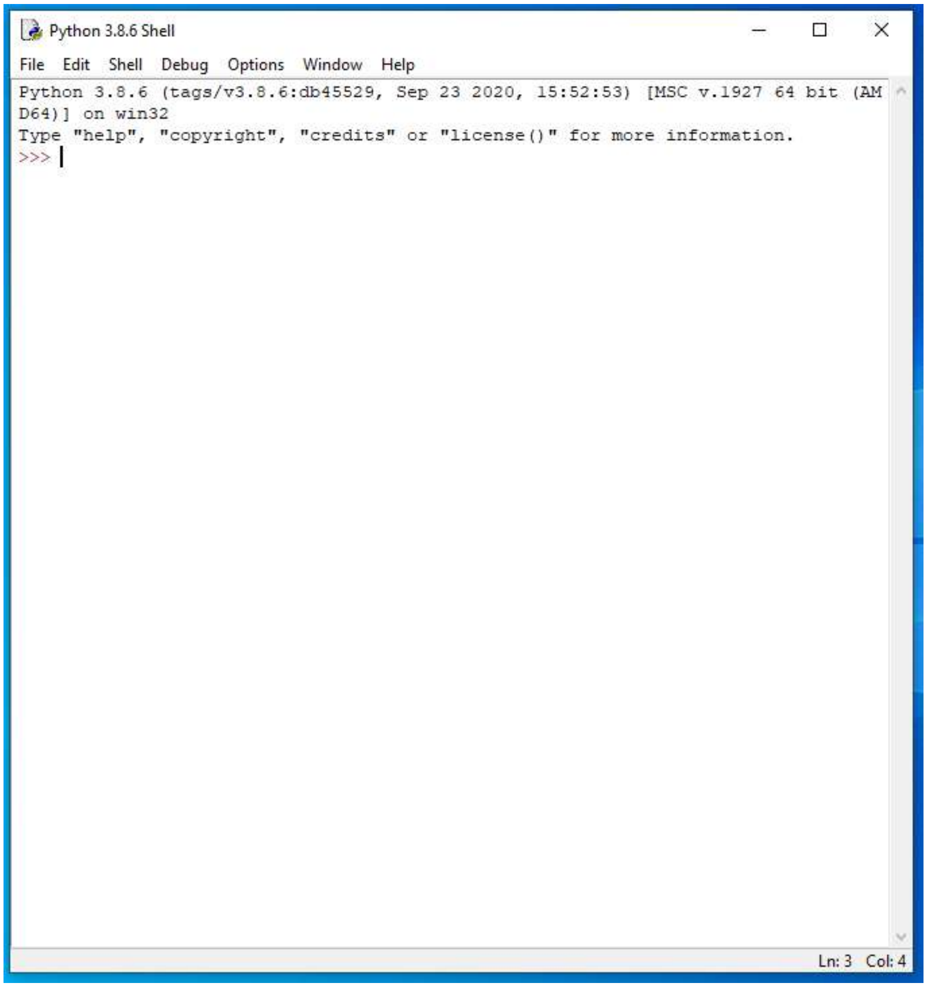
14. We are now ready to start adding code to our VampirePizzaAttack.py file. Using your Windows button menu, find and launch your IDLE program.



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.:

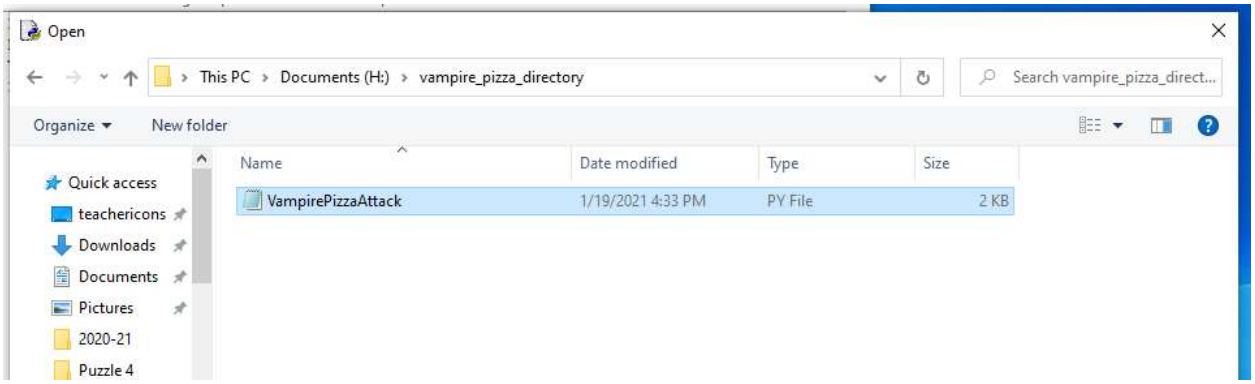


```
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:52:53) [MSC v.1927 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
```

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.

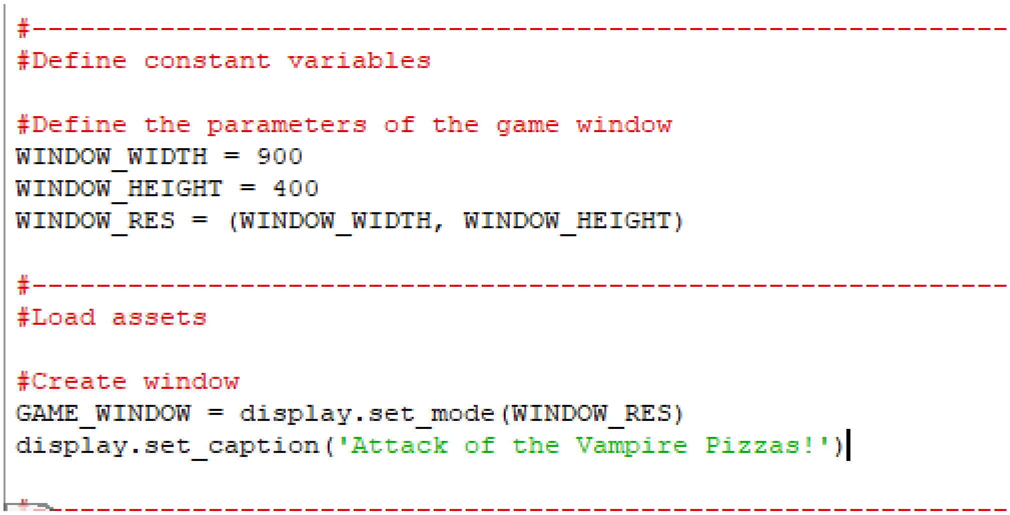


Name	Date modified	Type	Size
VampirePizzaAttack	1/19/2021 4:33 PM	PY File	2 KB

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.

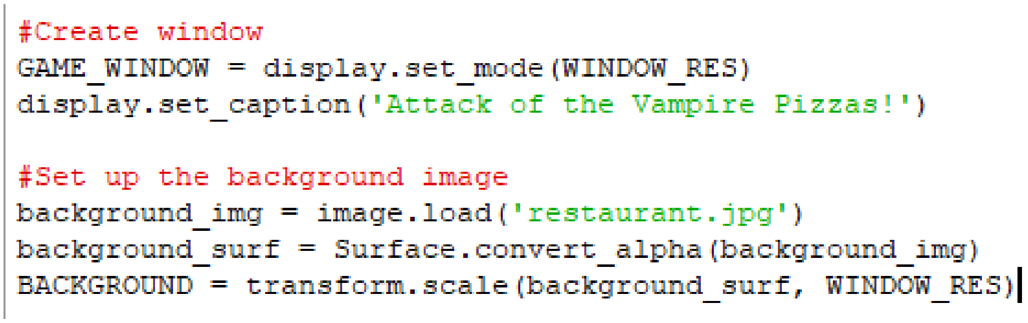
19. Click at the end of Line 20 after the code to set the display caption.



```
#---------------------------------------------------------------
#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 – 25 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 background image
23  background_img = image.load('restaurant.jpg')
24  background_surf = Surface.convert_alpha(background_img)
25  BACKGROUND = transform.scale(background_surf, WINDOW_RES)
```

Line 22 creates a comment describing what the next block of code does.

Lines 23 – 25 repeat a process that we went through in the last chapter using the vampire.png image to load the background image into the game.

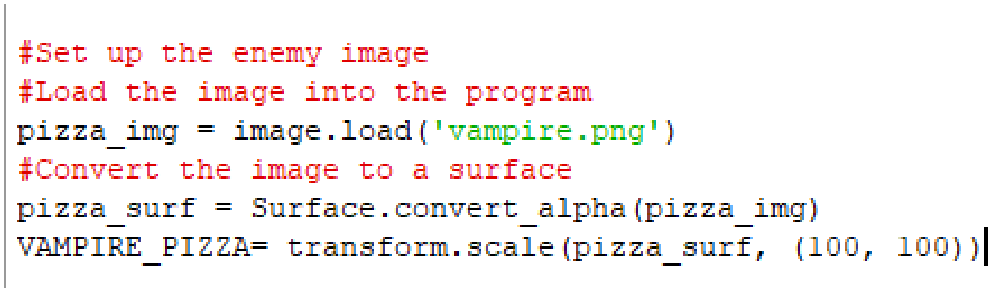
Line 23 creates the background_img variable and loads the restaurant.jpg image as the background.

Line 24 converts the restaurant.jpg image into a surface and assigns that image surface to the background_surf variable.

Line 25 creates a constant variable called BACKGROUND and resizes the background_surf image to be the same size as the game window's resolution.

22. Press ENTER twice.

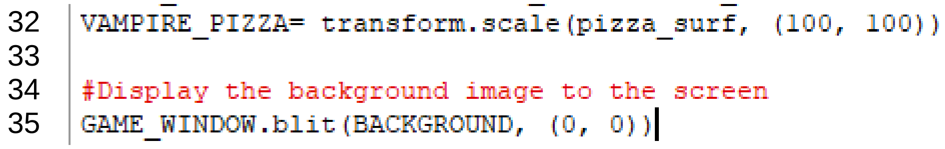
23. Click at the end of Line 32 (after you set the scale of the pizza_surf image and assign it to the VAMPIRE_PIZZA variable) and press ENTER twice again



```
26
27  #Set up the enemy image
28  #Load the image into the program
29  pizza_img = image.load('vampire.png')
30  #Convert the image to a surface
31  pizza_surf = Surface.convert_alpha(pizza_img)
32  VAMPIRE_PIZZA= transform.scale(pizza_surf, (100, 100))
```

24. Press ENTER twice.

25. Type the code that you see on Lines 34 – 35 of the screenshot below:

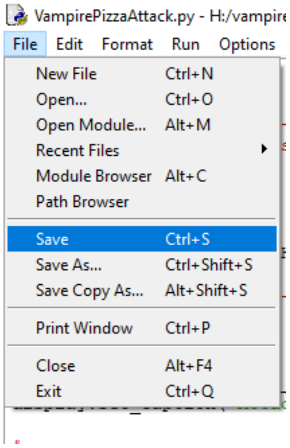


```
32  VAMPIRE_PIZZA= transform.scale(pizza_surf, (100, 100))
33
34  #Display the background image to the screen
35  GAME_WINDOW.blit(BACKGROUND, (0, 0))
```

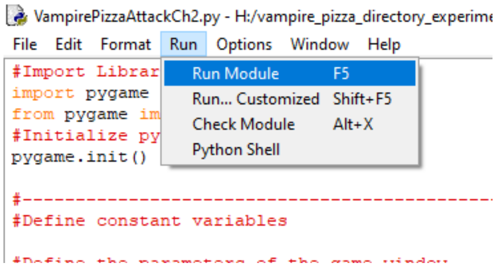
Line 34 contains a comment describing what the next line of code will do.

Line 35 blits the background surface image to the game window and places the image at the coordinates of 0, 0 (or at the center of the game window). This ensures that the image is centered in the game window.

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

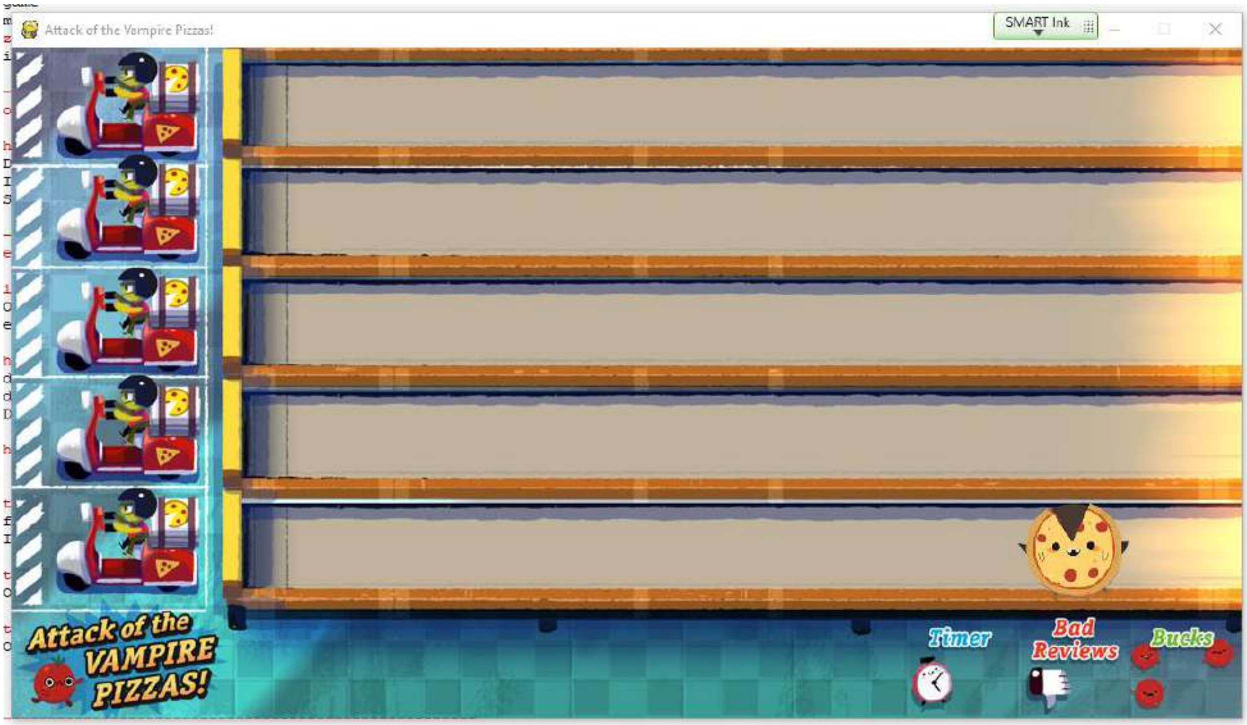


27. Now, go to Run > Run Module.


VampirePizzaAttackCh2.py - H:/vampire_pizza_directory_experime
File
Edit
Format
Run
Options
Window
Help
Run Module F5
Run... Customized Shift+F5
Check Module Alt+X
Python Shell

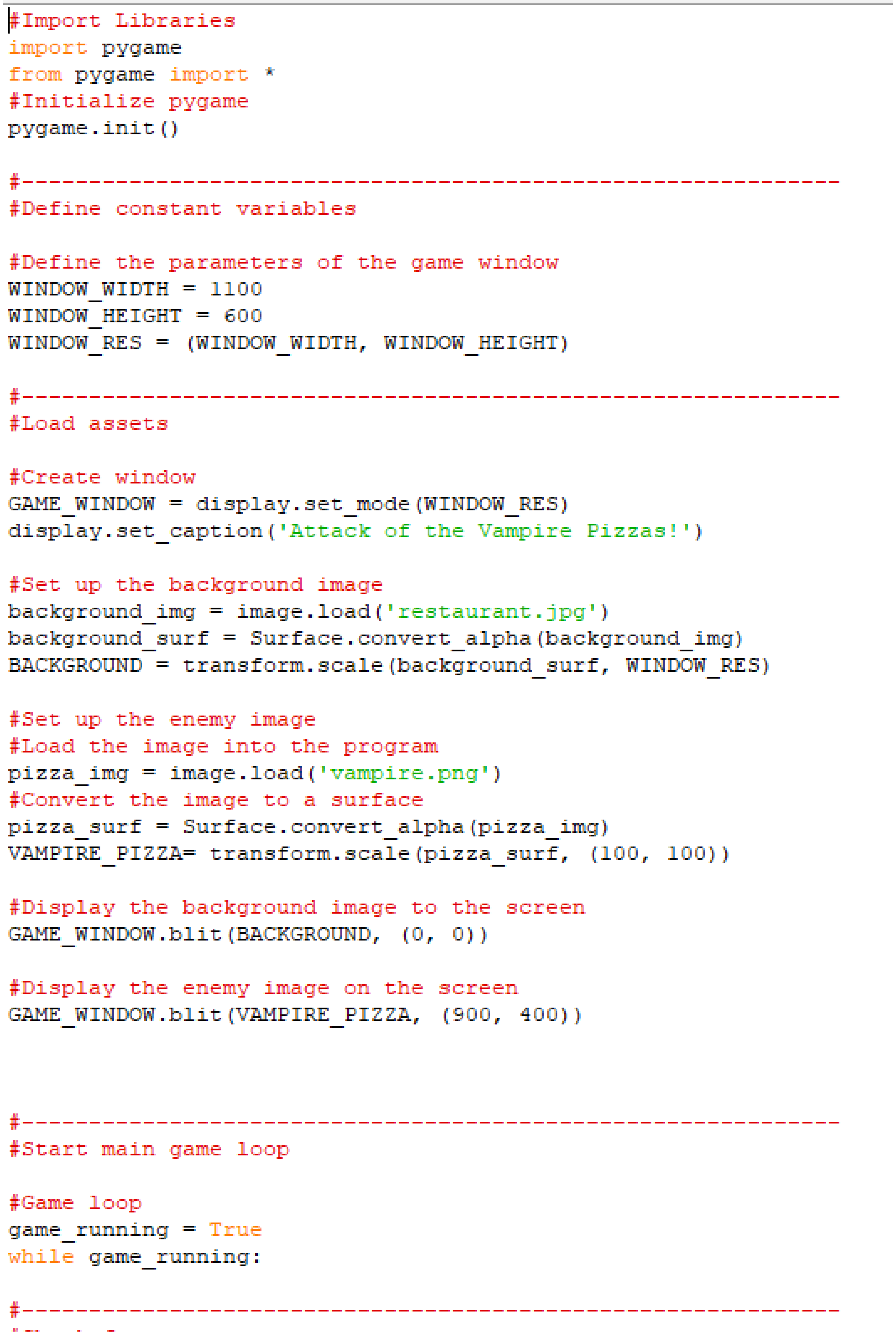
```
#Import Librar
import pygame
from pygame im
#Initialize py
pygame.init()

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

28. After giving your game window time to load, you should see your background image in your game window.

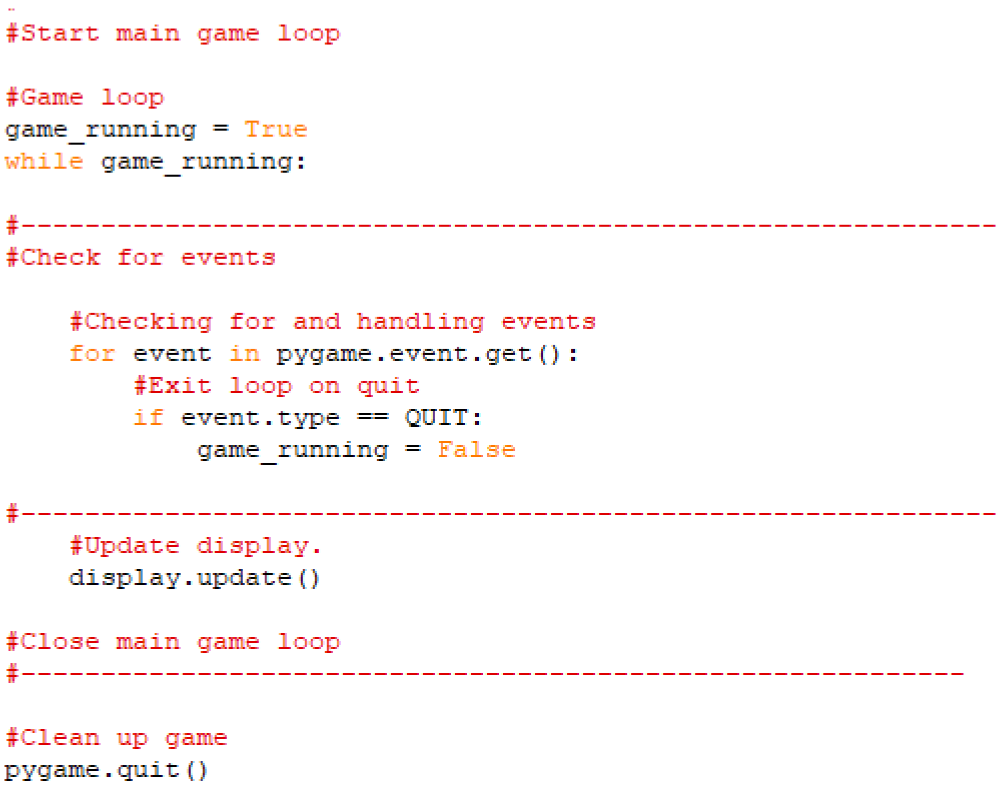


29. You can close out of the Python file. You can also close out of the Python Shell if you still have it open.

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 background image
background_img = image.load('restaurant.jpg')
background_surf = Surface.convert_alpha(background_img)
BACKGROUND = transform.scale(background_surf, WINDOW_RES)

#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))

#Display the background image to the screen
GAME_WINDOW.blit(BACKGROUND, (0, 0))

#Display the enemy image on the screen
GAME_WINDOW.blit(VAMPIRE_PIZZA, (900, 400))



#-------------------------------------------------------------
#Start main game loop

#Game loop
game_running = True
while game_running:

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



```
#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()
```