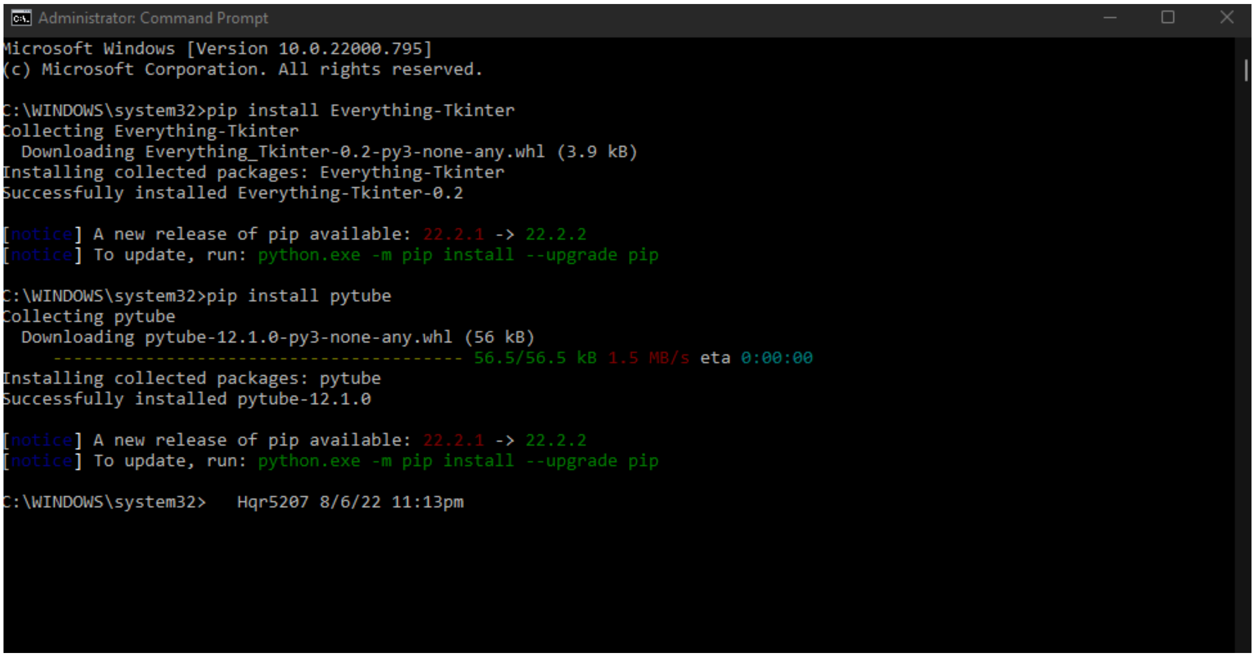
Henley Ramoodit
Final Individual Take home Project
Cyber 100 Section 001

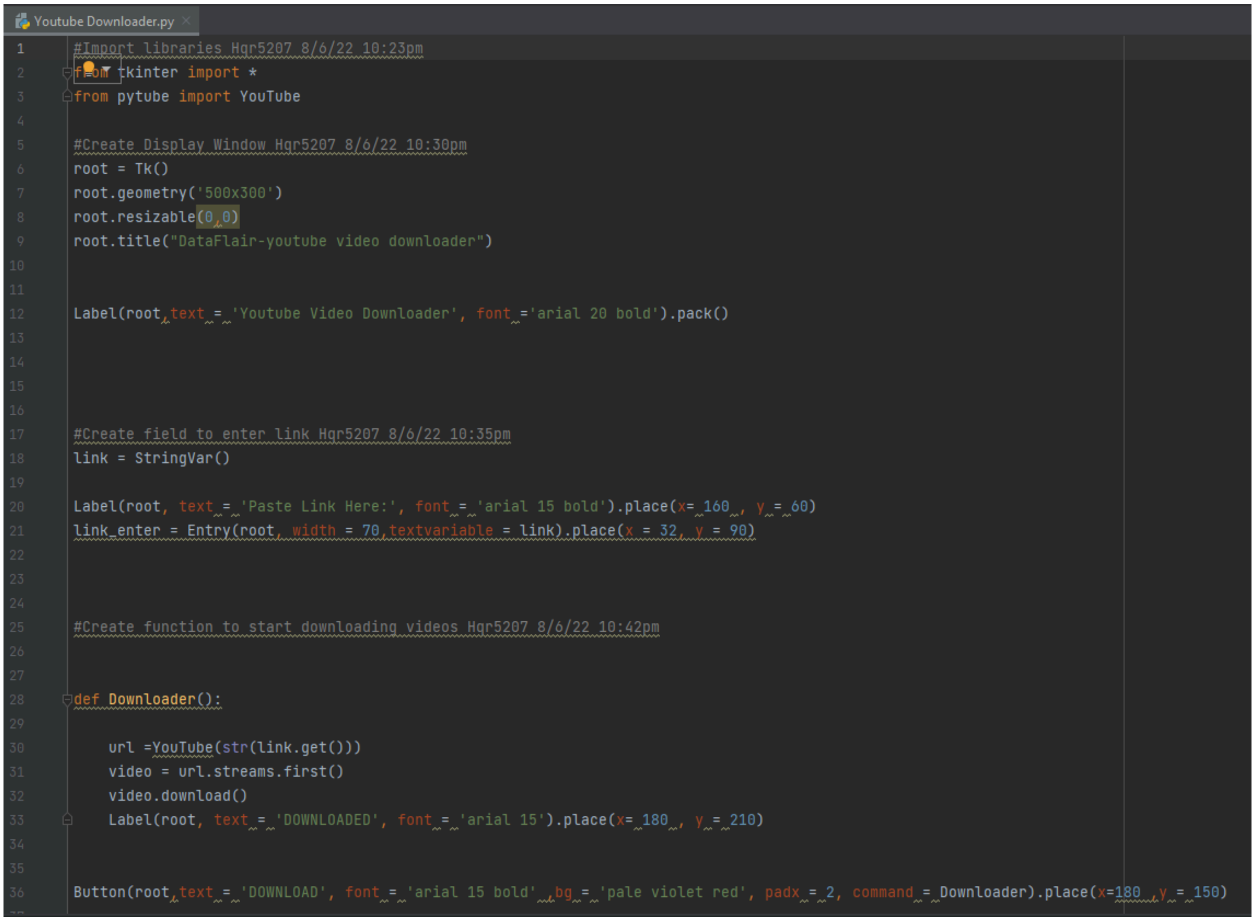
Basic - Python Youtube Downloader

The goal of this python project is for the user to be able to download any youtube video by pasting the link into the GUI application we are going to create. For us to be able to complete this project, we require two python modules. Tkinter, a GUI library, will allow us to make a GUI for the youtube downloader easily. And pytube, which will enable us to download videos from youtube. To install these modules, you will have to run the pip installer commands in your command line, as I did in Figure 1.1 below. After that, the first step for the youtube downloader is to import the modules and create the display window. We will use commands Tk() to create a display window using Tkinter and geometry() to set the width and height of the window. You will also use the title() command to give the window a title. Next, you will use commands like Label(), root, font, and more to establish how you want the GUI application to look. After that is done, you are ready to create a field where the user can input the youtube link of the video they want to download. You will use commands such as link, a string type variable that will store the youtube video link entered, and Entry() which will allow us to create an input text field. Lastly, you have to make the function to start downloading youtube videos. The get() function will get the youtube link, and str() will convert this link into a string datatype. Next is the stream.first() method, which will download the video. Figure 1.2 shows the source code and Figure 1.3 shows the downloader in use.



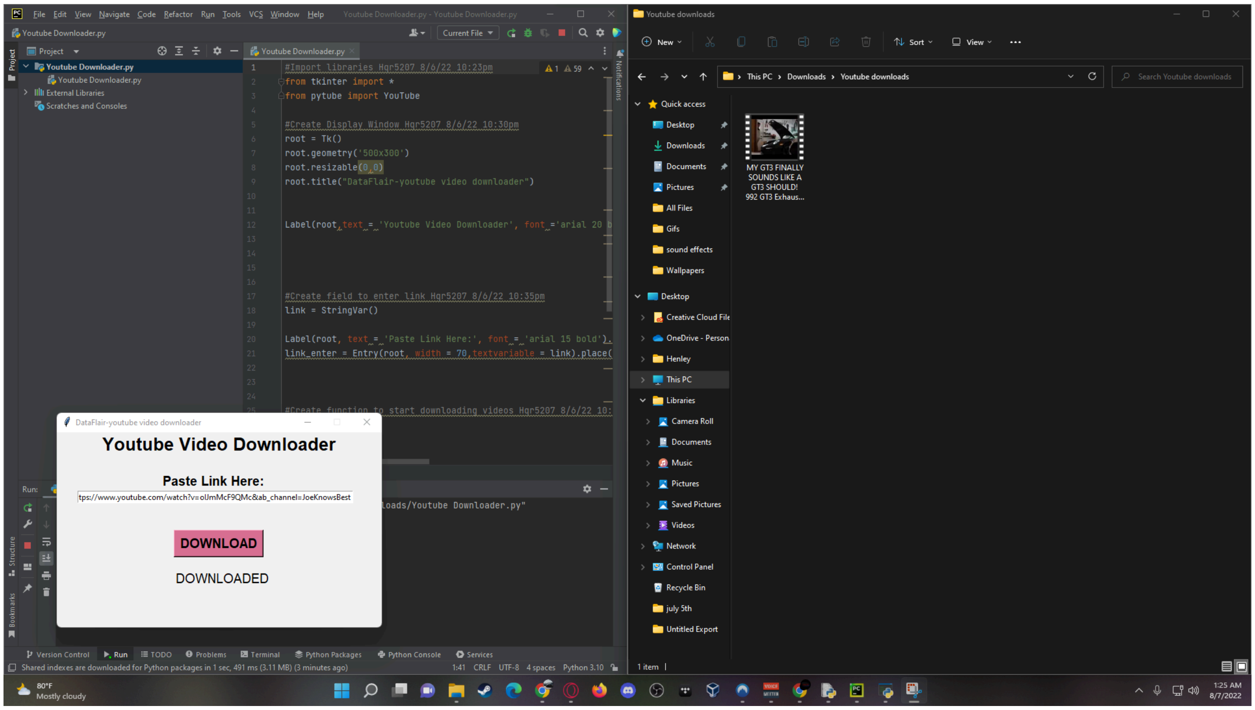
```
Microsoft Windows [Version 10.0.22000.795]
(c) Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>pip install Everything-Tkinter
Collecting Everything-Tkinter
  Downloading Everything_Tkinter-0.2-py3-none-any.whl (3.9 kB)
Installing collected packages: Everything-Tkinter
Successfully installed Everything-Tkinter-0.2

[notice] A new release of pip available: 22.2.1 -> 22.2.2
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\WINDOWS\system32>pip install pytube
Collecting pytube
  Downloading pytube-12.1.0-py3-none-any.whl (56 kB)
     ---------------------------------------- 56.5/56.5 kB 1.5 MB/s eta 0:00:00
Installing collected packages: pytube
Successfully installed pytube-12.1.0

[notice] A new release of pip available: 22.2.1 -> 22.2.2
[notice] To update, run: python.exe -m pip install --upgrade pip

C:\WINDOWS\system32>   Hqr5207 8/6/22 11:13pm
```

Figure 1.1: Installation of the Tkinter and Pytube modules in the Windows CMD.



```python
#Import libraries Hqr5207 8/6/22 10:23pm
from tkinter import *
from pytube import YouTube

#Create Display Window Hqr5207 8/6/22 10:30pm
root = Tk()
root.geometry('500x300')
root.resizable(0,0)
root.title("DataFlair-youtube video downloader")


Label(root,text = 'Youtube Video Downloader', font ='arial 20 bold').pack()




#Create field to enter link Hqr5207 8/6/22 10:35pm
link = StringVar()

Label(root, text = 'Paste Link Here:', font = 'arial 15 bold').place(x= 160 , y = 60)
link_enter = Entry(root, width = 70,textvariable = link).place(x = 32, y = 90)



#Create function to start downloading videos Hqr5207 8/6/22 10:42pm


def Downloader():

    url =YouTube(str(link.get()))
    video = url.streams.first()
    video.download()
    Label(root, text = 'DOWNLOADED', font = 'arial 15').place(x= 180 , y = 210)


Button(root,text = 'DOWNLOAD', font = 'arial 15 bold' ,bg = 'pale violet red', padx = 2, command = Downloader).place(x=180 ,y = 150)
```

Figure 1.2: Above is the source code for the Youtube Downloader. My PSU ID, date, and time of completion are shown as well.



```
#Import libraries Hqr5207 8/6/22 10:23pm
from tkinter import *
from pytube import YouTube

#Create Display Window Hqr5207 8/6/22 10:30pm
root = Tk()
root.geometry('500x300')
root.resizable(0,0)
root.title("DataFlair-youtube video downloader")


Label(root,text = 'Youtube Video Downloader', font ='arial 20 b




#Create field to enter link Hqr5207 8/6/22 10:35pm
link = StringVar()

Label(root, text = 'Paste Link Here:', font = 'arial 15 bold').
link_enter = Entry(root, width = 70,textvariable = link).place(



#Create function to start downloading videos Hqr5207 8/6/22 10:
```

DataFlair-youtube video downloader

Youtube Video Downloader

Paste Link Here:

tps://www.youtube.com/watch?v=oIJmMcF9QMc&ab_channel=JoeKnowsBest

DOWNLOAD

DOWNLOADED

```
loads/Youtube Downloader.py"
```

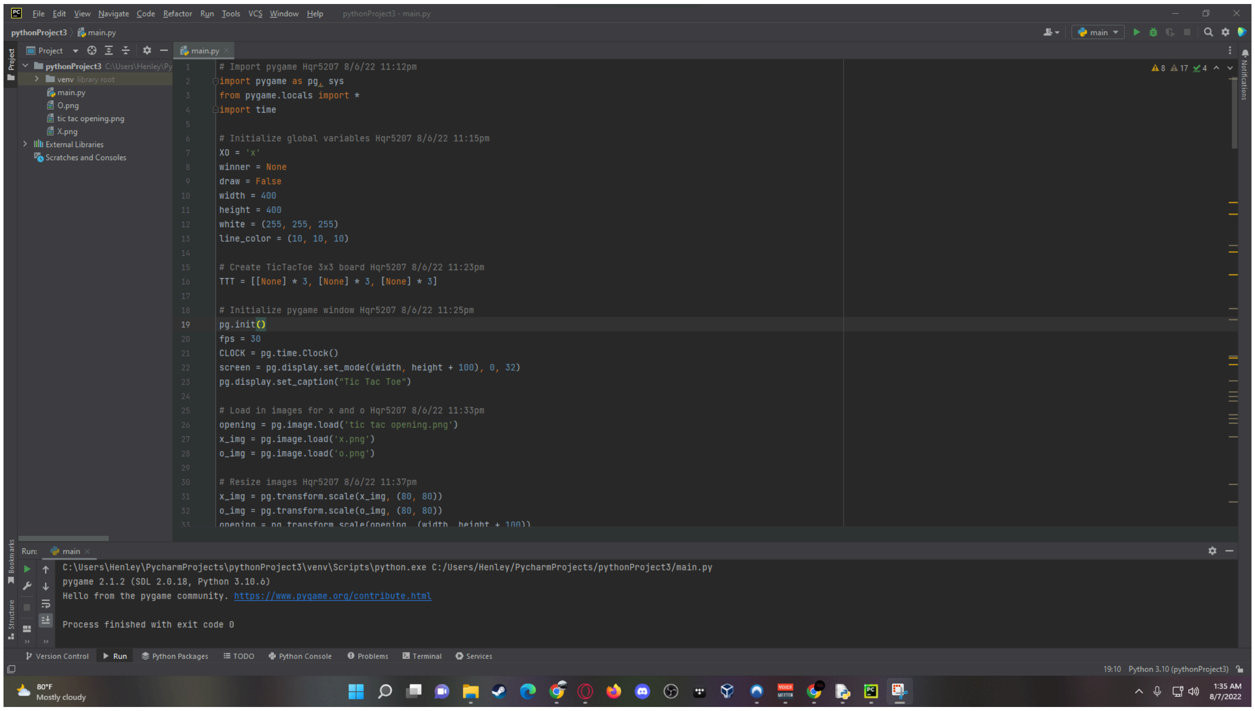
Shared indexes are downloaded for Python packages in 1 sec, 491 ms (3.11 MB) (3 minutes ago)

MY GT3 FINALLY SOUNDS LIKE A GT3 SHOULD! 992 GT3 Exhaus...

Figure 1.3: Shown above are the results after running and using the downloader. The GUI application can be seen in the bottom left with a youtube link pasted in it. On the right side of the screenshot, you can see file manager is open to where the youtube video was downloaded and saved.

Intermediate - Python Tic-Tac-Toe Game

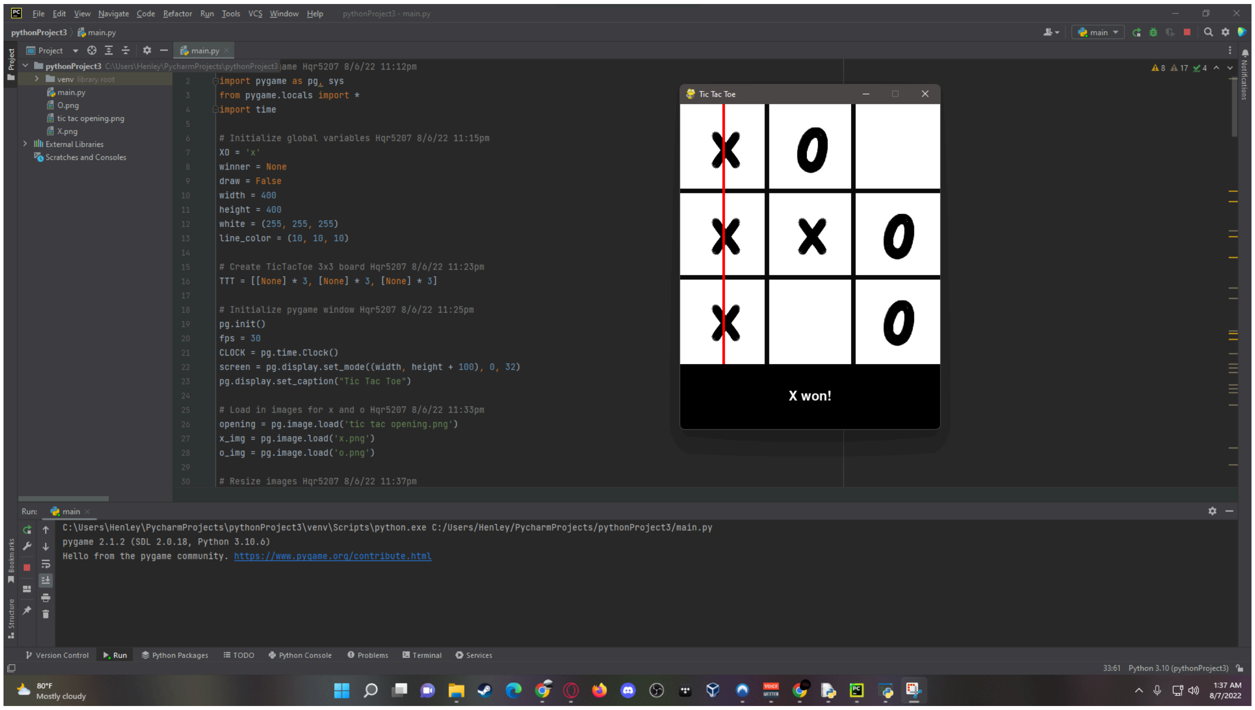
Using the Pygame library module, we will create a Tic-Tac-Toe game where we can make the window for the game and draw images and shapes to play the game. To use Pygame, you must install it through your command line before using its modules. The procedure for this project requires us to import the pygame library and the time library to start our game creation. Next, we must use pygame to create a window for us to play our Tic-Tac-Toe game. We will use commands like pg.init() and pg.display.set_mode() to initialize the game and set the window size. Next, we must load the x and o images that will be drawn when a user makes a move. Next, we will open the images provided and resize them with the pg.transform.scale command. After that is complete, we can start defining the functions that will start the game. We will use the function blit() to draw an image on top of another image and pg.display.update() to update the display. Next, we use the pg.draw.line commands to make the 3x3 grid for Tic-Tac-Toe. The draw_status function we will use next will show us which player's turn it is and whether the game ends or draws. Next, we establish a check_win() function to check the board to see the X's and O's. The function will calculate if a player wins or loses the game. We will also use functions such as drawXO(row,col) to draw the X and O where they are played, and the userClick() function for every time the user presses their mouse button. Lastly, we use the rest_game function to restart the game and rest the variables when one game concludes. We also use the game_opening() function to run an infinite loop to check for events made by the user, such as a mouse click on a certain column or row. After all of this is done, you are ready to play Tic-Tac-Toe. Figure 2.1 shows some source code for the Tic-Tac-Toe game while Figure 2.2 shows the game being played.



```python
# Import pygame Hqr5207 8/6/22 11:12pm
import pygame as pg, sys
from pygame.locals import *
import time

# Initialize global variables Hqr5207 8/6/22 11:15pm
XO = 'x'
winner = None
draw = False
width = 400
height = 400
white = (255, 255, 255)
line_color = (10, 10, 10)

# Create TicTacToe 3x3 board Hqr5207 8/6/22 11:23pm
TTT = [[None] * 3, [None] * 3, [None] * 3]

# Initialize pygame window Hqr5207 8/6/22 11:25pm
pg.init()
fps = 30
CLOCK = pg.time.Clock()
screen = pg.display.set_mode((width, height + 100), 0, 32)
pg.display.set_caption("Tic Tac Toe")

# Load in images for x and o Hqr5207 8/6/22 11:33pm
opening = pg.image.load('tic tac opening.png')
x_img = pg.image.load('x.png')
o_img = pg.image.load('o.png')

# Resize images Hqr5207 8/6/22 11:37pm
x_img = pg.transform.scale(x_img, (80, 80))
o_img = pg.transform.scale(o_img, (80, 80))
opening = pg.transform.scale(opening, (width, height + 100))
```

```
C:\Users\Henley\PycharmProjects\pythonProject3\venv\Scripts\python.exe C:/Users/Henley/PycharmProjects/pythonProject3/main.py
pygame 2.1.2 (SDL 2.0.18, Python 3.10.6)
Hello from the pygame community. https://www.pygame.org/contribute.html

Process finished with exit code 0
```

Figure 2.1: Above is some source code for the Tic-Tac-Toe game. As well as my PSU ID, date, and time this was completed.



```python
ame Hqr5207 8/6/22 11:12pm
import pygame as pg, sys
from pygame.locals import *
import time

# Initialize global variables Hqr5207 8/6/22 11:15pm
XO = 'x'
winner = None
draw = False
width = 400
height = 400
white = (255, 255, 255)
line_color = (10, 10, 10)

# Create TicTacToe 3x3 board Hqr5207 8/6/22 11:23pm
TTT = [[None] * 3, [None] * 3, [None] * 3]

# Initialize pygame window Hqr5207 8/6/22 11:25pm
pg.init()
fps = 30
CLOCK = pg.time.Clock()
screen = pg.display.set_mode((width, height + 100), 0, 32)
pg.display.set_caption("Tic Tac Toe")

# Load in images for x and o Hqr5207 8/6/22 11:33pm
opening = pg.image.load('tic tac opening.png')
x_img = pg.image.load('x.png')
o_img = pg.image.load('o.png')

# Resize images Hqr5207 8/6/22 11:37pm
```

```
C:\Users\Henley\PycharmProjects\pythonProject3\venv\Scripts\python.exe C:/Users/Henley/PycharmProjects/pythonProject3/main.py
pygame 2.1.2 (SDL 2.0.18, Python 3.10.6)
Hello from the pygame community. https://www.pygame.org/contribute.html
```

Figure 2.2: Above shows the Tic-Tac-Toe game running, and an example of a user winning as X.