Name: Sasi Visvan C
Roll No: CH.SC.U4.AIE24084

Image Steganography and Tinker GUI

Code explanation:

1. **Have 2 functions Encode() and Decode(). Encode() takes 3 arguments input_file_path, secret message, output_file path.while Decode() take only input_file_path.**
2. **Using tinker GUI create**
   a. **1 file selector and saves the file path to a global variable.**
   b. **Text input field to get message as input(secret_message).**
   c. **3 buttons. Each button calls the respective functions**
      i. **1)for Encoding the text in the text field**
      ii. **2)decoding the text in the selected img file**
      iii. **3)exit button that quits the app**

Code:

```
from PIL import Image
# import all components
# from the tkinter library
from tkinter import *
import os
# import filedialog module
from tkinter import filedialog
from PIL import Image

def hide_text(image_path, secret_text, output_path=""):
        output_path = image_path[0:len(image_path)-4] + "_encoded.jpg"
        filename2 = output_path
        print("encode text to:",output_path)
        # Open the image
        image = Image.open(image_path)
        # Convert the secret text to binary
        binary_secret_text = ''.join(format(ord(char), '08b') for char in
secret_text)

        # Check if the image can accommodate the secret text
        image_capacity = image.width * image.height * 3
        if len(binary_secret_text) > image_capacity:
                raise ValueError("Image does not have sufficient capacity to hide
the secret text.")

        # Hide the secret text in the image
        pixels = image.load()
        index = 0
        for i in range(image.width):
                for j in range(image.height):
                        r, g, b = pixels[i, j]

                        # Modify the least significant bit of each color channel
                        if index < len(binary_secret_text):
                                r = (r & 0xFE) | int(binary_secret_text[index])
                                index += 1
                        if index < len(binary_secret_text):
                                g = (g & 0xFE) | int(binary_secret_text[index])
                                index += 1
                        if index < len(binary_secret_text):
                                b = (b & 0xFE) | int(binary_secret_text[index])
                                index += 1

                        pixels[i, j] = (r, g, b)

        # Save the image with the hidden secret text
        image.save(output_path)

def extract_text(image_path):
   print("extract text at:",image_path)
   # Open the image
   image = Image.open(image_path)

   # Extract the secret text from the image
```

```
    pixels = image.load()
    binary_secret_text = ""
    for i in range(image.width):
        for j in range(image.height):
            r, g, b = pixels[i, j]

            # Extract the least significant bit of each color channel
            binary_secret_text += str(r & 1)
            binary_secret_text += str(g & 1)
            binary_secret_text += str(b & 1)

    # Convert the binary text to ASCII
    secret_text = ""
    for i in range(0, len(binary_secret_text), 8):
        char = chr(int(binary_secret_text[i:i+8], 2))
        secret_text += char
    return secret_text

def browseFiles():
        global filename2
        filename = filedialog.askopenfilename(initialdir = "",
                                                                          title = "Select a
File",
                                                                          filetypes =
(("Text files",

                    "*.jpg*"),

        ("all files",

                    "*.*")))
        filename2=filename
        label_file_explorer.configure(text="File Opened: "+ filename)


# Create the root window
window = Tk()

# Set window title
window.title('File Explorer')

# Set window size
window.geometry("700x500")
```

```
#Set window background color
window.config(background = "white")

# Create a File Explorer label
label_file_explorer = Label(window,
                                                                text = "File Explorer using
Tkinter",
                                                                width = 100, height = 4,
                                                                fg = "blue")


button_explore = Button(window,
                                                        text = "Browse Files",
                                                        command = browseFiles)
# adding Entry Field
secret_text = Entry(window, width=10)
secret_text.grid(column =1, row =3)


# function to display user text when
# encoded button is clicked
def encode():
        print("Running encoding:")
        image_path = filename2
        output=""
        hide_text(image_path, secret_text.get(),output)
        print("encoded text:",secret_text.get())

def decode():
        image_path = filename2
        secret_text_from_img = extract_text(image_path)
        print("decoded text:",secret_text_from_img)

button_encode = Button(window,
                                                text = "Encode",
                                                command = encode)

button_decode = Button(window,
                                                text = "Decode",
                                                command = decode)

button_exit = Button(window,
                                                text = "Exit",
                                                command = exit)
```

```
# Grid method is chosen for placing
# the widgets at respective positions
# in a table like structure by
# specifying rows and columns

label_file_explorer.grid(column = 1, row = 1)
button_explore.grid(column = 1, row = 2)
button_encode.grid(column = 1,row = 4)
button_decode.grid(column = 1,row = 5)
button_exit.grid(column = 1,row = 6)

# Let the window wait for any events
window.mainloop()
```

Algorithm Used: Least Significant Bit (LSB)

Input Image:

Output Image:

You can see there isn't much difference because it uses least significant bits to encode so not much difference.

Learned:

1. **To use TINKER GUI**
    a. **Making buttons that calls a command**
    b. **Making a file selector**
    c. **Making a text input field**
2. **To encode text in an image by taking all the least significant bits pixel values and changing them a bit thus encoding a message.**