פרויקט חדש:

Import necessary libraries#

Pip install Pillow pygame

Import random

From PIL import Image, ImageTK

Import tkinter as tk

Import pygame

Import os

#Initialize pygame for sound

Pygame.init()

#Define a dictionary of social situations and their correct answers

Correct\_answers = { "You're at a birthday party, and it's time to sing 'Happy Birthday.'": ["Sing along and clap."], "You're playing with your friend at the playground, and someone new wants to join.": ["Invite the new person to play."], "You're in a classroom, and the teacher asks a question.": ["Raise your hand and wait to be called on."], "You're in classroom and you saw an empty seat next to a friend. You wanted to sit but were told it was reserved for someone else.": ["Find another free seat for you and take advantage of the opportunity to meet new friends."], "You heard some of your classmates are meeting this afternoon at the mall, and they forgot to invite you.": [ "Ask one of the children if it is possible to join? find out where to meet and at what time."]}

Define a dictionary of social situation and their corresponding image#

Situation and \_image = {"You're at a birthday party, and it's time to sing 'Happy Birthday.'": "birthday\_party\_image.jpg", "You're playing with your friend at the playground, and someone new wants to join.": "playground\_image.jpg", "You're in a classroom, and the teacher asks a question.": "classroom\_image.jpg", "You're in classroom and you saw an empty seat next to a friend. You wanted to sit but were told it was reserved for someone else.": "classroom2\_image.jpg", "You heard some of your classmates are meeting this afternoon at the mall, and they forgot to invite you.": "friends\_image.jpg"}

#Initialize the step counter and points

Step\_counter = 0

Points = 0

#Load success sound

Success\_sound = pygame.mixer.Sound('success.wav')

#Create a tkinter window for displaying images

Window =tk.TK()

Window.title("Social Comprehesion")

#Load default image

default\_image = Image.open("default\_image.jpg")

default\_image = ImageTK.PhotoImage(default\_image)

image\_label = tk.Label(window, image=default\_image)

image\_label.pack()

#Define a function to check the correctness of the user's response

def is\_correct\_answer(situation, answer):

#Check if the situation exists in the dictionary

If situation in correct\_answers:

#Get the list of correct answers for the situation

Correct\_answers\_list = correct\_answers[situation]

#Check if the user's answer matches any correct answer for the situation

If answer in correct\_answers\_list:

return True

else:

return False

else:

#Situation not found in the dictionary, handle as needed

Return False

#Define a function to display success feedback

Def show\_success\_feedback():

Global points

Points += 10

#Play success sound

Success\_sound.play()

#Load and display a picture of fireworks

fireworks\_image = Image.open("fireworks\_image.jpg")

fireworks\_image =ImageTK.PhotoImage(fireworks\_image)

image\_label.configure(image=fireworks\_image)

image\_label.update()

#Display a success message

Success\_message = tk.Label(window, text= "Congratulations! You earned 10 points!", font = ("Calibri", 14))

Success\_message.pack()

After a few seconds, reset the feedback#

Def reset\_feedback():

#Load and display the default image

default\_image = Image.open("default\_image.jpg")

default\_image = ImageTK.PhotoImage(default-image)

image\_label.configure(image=default\_image)

#Clear the success message

Success\_message.pack\_forget()

#Define a function to present a social comprehension exercise with an image

Def social\_comprehension\_exercise():

Global step\_counter #Declare step\_counter as a global variable

#Randomly select a social situation

Situation = random.choice(list(correct\_answers.keys()))

#Display the social situation to the user

Print("Social Situation:")

Print(situation)

#Load and display the corresponding image for the situation

Image\_filename = situation\_image.get(situation, "default\_image.jpg")

Image = Image.open(image\_filename)

Image = ImageTK.PhotoImage(image)

Image\_label.configure(image=image)

#Ask a question related to the situation

Question = input("what should you do in this situation?")

#Check the correctness of the user's answer

If is\_correct\_answer(situation, question):

Print("Correct! You handled the situation well.")

Step\_counter += 1 #Increment the step counter on a correct answer

Show\_success\_feedback()

Else:

Print("Oops! That's not the best way to handle this situation.")

#Give a clue

#Define a diction of social situations and their correct answers

Situations\_and\_answers = {

"You're at birthday party, and it's time to sing 'Happy Birthday.' ": {

"Question" : "what should you do in this situation?",

"Choices" : [

,"Sing along and clap"

, "Sit quietly and watch"

, "Start dancing wildly"

."Leave the party"

, [

"Correct": "Sing along and clap"

,{

}: "You're playing with your friend at the playground, and someone new wants to join"

, "Question": "What should you do in this situation?"

] : "Choices"

"Invite the new person to play"

"Tell the new person to go away"

"Ignore the new person"

"Leave the playground"

, [

"Correct": "Invite the new person to play."

, {

}: "You're in a classroom, and the teacher asks a question"

"Question": "What should you do in this situation?"

] : "Choices"

"Raise your hand and wait to be called on"

"Yell out the answer as loudly as possible"

"Run out of the classroom"

"Hide under your desk"

,[

"Correct": "Raise your hand and wait to be called on"

,{

"You're in classroom and you saw an empty seat next to a friend. You wanted to sit but were told it was reserved for someone else.":{

'"Question" : "what should you do in this situation?",

Choises" :["

Find another free seat and sit there without asking anyone""

Ask your friend if it's okay to sit in the reserved seat anyway""

Find another free seat and take advantage of the opportunity to meet new friends.""

"Leave the classroom and go somewhere else"

],

"Correct": "Correct": "Find another free seat for you and take advantage of the opportunity to meet new friends."

},

"You heard some of your classmates are meeting this afternoon at the mall, and they forgot to invite you.":{

"Question" : "what should you do in this situation?",

"Choises" :[

Don't do anything and feel upset about not being invited""

"Approach one of your classmates, ask if you can join, and find out where and when they are meeting."

Decide to organize your own gathering and invite some classmates.""

Confront your classmates about not being invited and demand an explanation." "

"Correct": " Approach one of your classmates, ask if you can join, and find out where and when they are meeting."

,{

}

Define a function to present a social comprehension exercise with answer choices#

def social\_comprehension\_exercise():

global step\_counter

# Declare step\_counter as a global variable

Randomly select a social situation#

situation = random.choice(list(situations\_and\_answers.keys()))

Display the social situation to the user#

print("Social Situation:")

print(situation)

Load and display the corresponding image for the situation#

image\_filename = "default\_image.jpg" # Default image

if situation in situation\_images:

image\_filename = situation\_images[situation]

image = Image.open(image\_filename)

image = ImageTk.PhotoImage(image)

image\_label.configure(image=image)

#Display the question and answer choices

question = situations\_and\_answers[situation]["Question"]

choices = situations\_and\_answers[situation]["Choices"]

print(question)

for i, choice in enumerate(choices, start=1):

print(f"{i}. {choice}")

Ask the user to select an answer#

user\_choice = int(input("Enter the number of your choice (1-4): "))

Check the correctness of the user's answer#

: if 1 <= user\_choice <= 4 and is\_correct\_answer(situation, choices[user\_choice - 1])

print("Correct! You handled the situation well.")

step\_counter += 1 # Increment the step counter on a correct answer

show\_success\_feedback()

: else

print("Oops! That's not the best way to handle this situation.")

print(f"The correct answer is: {situations\_and\_answers[situation]['Correct']}")