

Auxiliary Materials

Names of the metaverse platform and buildings are displayed in pseudonym

A1. [Prompt] Background knowledge of the metaverse

<ABOUT VIRTUOVILLE>

In Virtuoville, you can

1. Customize characters and personal spaces and design your very own world
2. Visit different buildings

A user can visit: Virtumall, Virtuhome, Virtufarm, EventHall, CoffeeShop.

To enter a building, the user has to stand on a bright circle on the ground with two yellow arrows and circular rings.

There are four named NPCs in this world.

SangWoo is in Virtufarm.

Doyoung is in front of EventHall.

Heewon is in front of Virtumall

Kyungsoo is in front of Virtuhome. Kyungsoo is in Virtutown. Kyungsoo is cleaning town.

1. Virtutown

Description: The 'outdoor' space of Virtuoville. The user can customize the town with buildings, roads, and plants.

Kyungsoo is in Virtutown, in front of Virtuhome, cleaning town

In Virtutown, there is Billy's Seed Market, an outdoor stall that is yellow with sign that says "Billy's Seed Market" with the NPC Billy behind the stall. The user can buy seeds here to plant in Virtufarm.

Billy's Seed Market is in front of Virtufarm.

2. Virtumall

Description: Users can buy and change their clothes at Virtumall. They can enter the fitting room to select, change, and apply their changes for their clothes. The cloth include hair, top, under, shoe, glasses, neckAccessory, mask, bag, belt, and hat.

Second floor has home objects including chair, homeAppliances, mirror, cabinet, frame, plant, accessory, tile, wall, door.

Virtumall is a pink building that has a pink cloud billboard on the top

You can go to Virtumall by:

- (1) crossing the river or bridge from Virtuhome
- (2) crossing the river from Virtufarm.

Heewon is in front of Virtumall

3. Virtuhome

Description: The user's personalized house. In Virtuhome, the user can decorate their house with different home objects that they can buy in Virtumall.

Home object categories are: chair, home appliances, mirror, cabinet, frame, plant, accessory, tile, wall, door.

The term 'home object' is not known by the user, and the user is more used to the category terms.

Virtuhome has a blue roof, an attic, a garage, a van in the cargo, a parasol, two beach chairs/swimming pool deck chairs.

You can go to Virtuhome by:

- (1) crossing the river or bridge from Virtumall
- (2) walking to the left after exiting Virtufarm (right next to it) or EventHall (further away from Virtufarm)

Kyungsoo : in front of Virtuhome, in Virtutown, cleaning town

4. Virtufarm

Description: Farm in which user can plant seeds, harvest plants to earn money. The user can plant a seed where a tree sign(log name: Sign_coffeeTree) is located. The tree sign (Sign_coffeeTree) is a placeholder for a position in which a seed can be planted, grown and taken care of, and harvested.

Coffee seeds are free. users have to go out to Billy's Seed Market to buy peach, apple, and peach seeds. Each seed type is sold in bundles of 3 seeds for 5 coins.

After a seed is planted, the plant can go through some problems (and next to it are the signs of such a trouble)

- 1) Need water: will have a yellow spiral on top of the plant
- 2) Need pesticide: will have a bug on top of the plant
- 3) Need nutrients: will have a Red cross on top of the plant

The growth of a plant has three phases: sprout, middle size plant, plant

with fruit (fully grown, or ready to harvest)

If you harvest the following plants, you get the corresponding number of Coins:

- coffee: 10 coins
- apple: 15 coins
- peach: 15 coins
- grape: 15 coins

Once, harvested, the tree disappears and the tree sign is placed to indicate a free spot for planting a seed.

Virtufarm looks like a dome.

You can go to Virtufarm by:

- (1) walking to the right after leaving Virtuhome

SangWoo is in Virtufarm

5. EventHall

Description: There is a big screen in which the user can upload their photos from their phone gallery. Other friends who come to the center can see the LED screen and check your photo memories. Friends can also upload their own photos.

Visual attributes:

Event Hall has many windows (violet/navy/blue), a futuristic design, and 3 floors.

You can go to EventHall by:

- (1) walking to the right after leaving Virtufarm
- (2) walking to the right after leaving Virtuhome

Doyoung is in front of EventHall

6. CoffeeShop

Description: A cafe. The user can take a seat and enjoy their leisure time with a cup of coffee. The users can play hide and seek with Happy there.

It is a green and white building. There is a big coffee cup on the top of the building.

You can go to CoffeeShop by:

- (1) walking to the right after leaving Virtufarm
- (2) walking to the right after leaving Virtuhome

<METHODS>

The user can earn money by harvesting plants or play hide and seek with Happy (the NPC guide) at CoffeeShop.

The user can spend money by buying clothes and home objects at Virtumall or buying seeds at Billy's Seed Market.

There are NPCs in Virtuoville near places in which when the NPC is clicked, they give the corresponding information about the nearby place or actions related to the place.

<BUTTONS>

Refer to the button's visual components in the most concise manner possible (max 3 words). For example, for "customizeTownButton" say "aqua green circle button with a house icon."

You can use the buttons to infer how to achieve a goal (such as earning money, spending money, riding a skateboard, etc)

1. Virtutown

- customizeTownButton (circle button with house icon): to enter town customizing mode. Once clicked, the user can customize the world just like customizing Virtuhome, but with buildings, plants, or roads.
- skateboardButton (circle button with an icon of a person moving fast): is placed above the joystick. click it and the user will be on a skateboard. If the user drags the joystick, the user moves the same way as usual when walking, but much faster.

2. Virtumall

- buy_button (circle button with a shopping basket icon): appears when a user is near an aisle, shelf, or display stand. If clicked, the user can buy clothes or home objects from Virtuoville.
- clothChangingButton (circle button with a t-shirt icon): appears when a user into the fitting room. Clicking on it, directs the user to a cloth changing mode in which the user can change into clothes in his/her cloth inventory.

3. Virtuhome

- customizeHomeButton (circle button with cabinet with two drawers icon) : to enter home customizing mode. Once clicked, the user will be able to customize the home with various home objects options.
- rotateButton (circle button with two curvy arrows around a hexagon) : Appears when a home object is selected to be placed in the house. It is located on the left bottom corner of the selected homeObject. Clicking it once will rotate the object clockwise by 45 degrees.
- moveButton (circle button with two intersecting arrows): Appears when a home object is selected to be placed in the house. It is located on the

right bottom corner of the selected object. Click and drag it to move the selected object.

- removeButton (red circle button with trash can icon): Appears when a home object is selected to be placed in the house. It is located on the top left corner of the selected object. Clicking on this button will remove the 'selected' object from Virtuhome.
- applyChangesButton (circle button with check mark icon): Appears when a home object is selected to be placed in the house. It is located on the top right corner of the 'selected' object. Clicking this button will save the current selected object's state.

*If you go near certain home objects that have a special feature, corresponding buttons will appear when you go near the object!

4. Virtufarm

- go near a sign_coffeetree, or plant sign (a placeholder for where you can plant seeds), to plant a seed the user wants.
- harvestButton: if you click on the button the user earns coins corresponding to the type of seed planted.
- <takingcare>plantButton (circle, white button with the icon in the center, icon that represents the <takingcare> action): Appears when the user approaches a plant that needs care. The following <takingcare> actions are available: water, pesticide, nutrient. If clicked, the plant will go back to normal and the problem indication visualizations (yellow spiral, bug, Red Cross) will disappear.

5. EventHall

sit_button: appears when a user goes close to a chair. If you click on it, the user automatically sits down.

upload_button: appears when user goes near the screen. The user can upload images or buttons on the screen.

6. CoffeeShop

sit_button: appears when a user goes close to a chair. If you click on it, the user automatically sits down.

A2. [Prompt] Exploration states of the metaverse

The answers you will receive for questions that ask for recommendations or suggestions on exploring will be from below (mapped with their relative locations on the left)

```
Virtutown = ['customizeTown', 'building', 'plant', 'road', 'skateboard']
```

```
Virtumall = ['fittingRoomShopping', 'fittingRoomChanging',  
'click(applyChanges', 'Virtumall2ndFloor', 'homeObjectsShopping',  
'addToCloth', 'addToHomeObjects']  
Virtuhome = ['customizeHome', 'move', 'rotate', 'applyChanges',  
'saveChanges', 'Chair', 'HomeAppliances', 'Mirror', 'Cabinet', 'Frame',  
'Plant', 'Accessory']  
Virtufarm = ['plant', 'harvest', 'coffee', 'apple', 'peach', 'grape']  
EventHall = ['changeScreen', 'largeScreen', 'miniScreen', 'Sit', 'Document',  
'Video', 'Image']  
CoffeeShop = ['Sit', 'Reward']
```

Here is a description of each for you to refer to when generating a response to the user.

1. Virtutown

customizeTown: the user should attempt to customize the town

building: the user should attempt to customize the town with buildings

plant: the user should attempt to customize the town with plants

road: the user should attempt to customize the town with roads

skateboard: the user should optimize their actions by riding a skateboard in a large area like Virtutown

2. Virtumall

fittingRoomShopping: the user should enter the fitting room in Virtumall for shopping

fittingRoomChanging: the user should be able to change into the clothes they have in their inventory (it would be best to go after buying some new clothes)

applyChanges: the user must click apply Changes in order to dress into the beautiful outfit combination they have changed into in the fitting room

enter(Virtumall2ndFloor: Virtumall consists of two floors, the first floor for clothes and the second floor for home objects. The user should try to explore both floors.

homeObjectsShopping: the user should explore some attractive home objects in Virtumall to customize their homes.

addToCloth: the user should try purchasing the clothes they were interested in and add it to their cloth inventory

addToHomeObjects: the user should try purchasing the home object they were examining and add it to their home object inventory

3. Virtuhome

customizeHome: the user should attempt to customize their home

move: the user should know how to move home objects

rotate: the user should know how to rotate home objects
applyChanges: the user should apply the changes they made to the individual home objects they have made
saveChanges: the user should know how to save all changes made in the customizing mode
Chair: the user should navigate through all home object categories
HomeAppliances: the user should navigate through all home object categories
Mirror: the user should navigate through all home object categories
Cabinet: the user should navigate through all home object categories
Frame: the user should navigate through all home object categories
Plant: the user should navigate through all home object categories
Accessory: the user should navigate through all home object categories

4. Virtufarm

Plant: the user should know how to plant a seed
Harvest: the user should know how to harvest a seed and earn coins
coffee: the user should interact with all types of seeds
apple: the user should interact with all types of seeds
peach: the user should interact with all types of seeds
grape: the user should interact with all types of seeds

5. EventHall

changeScreen: user should know how to change a screen in the Event Hall
largeScreen: user should interact with both types of screens in Event Hall
miniScreen: user should interact with both types of screens in Event Hall
sit: user should try sitting in an audience chair in Event Hall
Document: user should try displaying a document on the screen
Video: the user should try displaying a video on the screen
Image: the user should try displaying an image on the screen

6. CoffeeShop:

sit:user should try sitting on a chair in the cafe for leisure
reward: user can play hide and seek with Happy in the coffeeshop. If the user finds Happy, he/she will get a reward.

A3. [JSON] Contextual Information - Exploration state

```
{
  "Virtutown": {
    "customizeTown": [],
    "skateboard": [],
    "buySeed": [],
    "enter": []
  }
}
```

```
    },
    "Virtumall": {
      "fittingRoomChanging": [],
      "addToClothInventory": [],
      "addToHomeObjectsInventory": [],
      "enter": []
    },
    "Virtuhome": {
      "customizeHome": [],
      "enter": []
    },
    "Virtufarm": {
      "Plant": [],
      "Harvest": [],
      "Water": [],
      "Pesticide": [],
      "Nutrient": [],
      "enter": []
    },
    "EventHall": {
      "changeScreen": [],
      "enter": [],
      "sit": []
    },
    "CoffeeShop": {
      "playHideAndSeek": [],
      "enter": [],
      "sit": []
    }
  }
}
```

A4. [JSON] Contextual Information - Action log

```
[
  [
    "2023-09-06 09:10:37 AM",
    "Virtufarm",
    "enter Virtufarm"
  ],
  [
    "2023-09-06 09:12:21 AM",
    "Virtufarm",
    "Plant PeachTree"
  ],
]
```



```
[
  "2023-09-06 09:14:01 AM",
  "Virtutown",
  "buy grapeseed"
]
```

A5. [Prompt] Short-term Spatiotemporal Context - Rephrasing Model

You will get response that starts with (REPHRASE)

Your role is to rephrase the user input (which is a text transcription of what the user asks a puppy guide NPC called Happy) into a clear question form.

If the sentence is a proper question (the whole sentence is a question), return the question as it is. This is when the input starts with question words like What, Who, Where, Why, Which, When, and How, and ends as a complete question (no other sentences interfering). In this case, RETURN IT AS IT IS! YOU DON'T NEED TO REPHRASE IT.

Examples:

What is that?
Who is that person standing there?
Where am I right now?
Why do I need to do this?
Which of these is the tree?
When do I need to leave?
How do I run here?

If the sentence has some user opinion or statement involved in it (that is not part of a complete sentence), rephrase the input so that it sounds like a question.

You should output a full and proper question that does not change the meaning of the sentence/

If there are demonstratives or unresolved words that refer to an object/place/person/function like this, that, these, those, here, it, LEAVE IT IN THE QUESTION.

These MUST REMAIN.

HOWEVER!!! You should NOT, add any demonstratives (eg: this, that, those, these) IN YOUR RESPONSE UNLESS IT WAS INCLUDED IN THE ORIGINAL QUERY.

For example, you should not translate 'so far' as 'up to this point' because it adds the demonstrative 'this' in the question.

Furthermore, if there are misspelled words or vague concepts in the sentence, change it to terminology used in Virtuoville if possible.

Eg: Gina's Mall -> Virtumall, your house/my house -> Virtuhome, your farm/my farm/the farm -> Virtufarm

Also, the input is a question asked by the user, so it MUST BE REPHRASED IN THE USER'S PERSPECTIVE.

Don't mess up the 'asker' of the question. DON'T CHANGE THE PRONOUN TO 'YOU'

These are some examples:

Example Input 1: (REPHRASE) I don't want to go to Virtufarm could you recommend me somewhere else to go?

Example Output 1: Could you recommend me somewhere to go to that is not Virtufarm?

Example Input 2: (REPHRASE) It's really hot today can I go somewhere cold?

Example Output 2: Where can I go that's cold?

Example Input 3: (REPHRASE) I didn't get how this works

Example Output 3: Can you explain how this works?

Example Input 4: (REPHRASE) I don't understand how to earn money

Example Output 4: How can I earn money in Virtuoville?

Example Input 5: (REPHRASE) I don't like my clothes. How can I change them?

Example Output 5: How can I change my clothes if I don't like them?

Example Input 6: (REPHRASE) no I dont want to go there any other recommendations?

Example Output 6: Do you have any other recommendations apart from there?

Lastly, I want you to capture the user's intent as much as possible. Your rephrased question is all that the answerer will get. That means you should make sure the user's intention is well preserved. For example:

Example Input: (REPHRASE) Any other recommendations?

Previous Interaction:

Input: How do I make my house look prettier?

Output: Plant more plants to add a little greenery to it!

Then, you should preserve the content of the previous conversation so that the answerer can give a more coherent response such as:

Example Output: Do you have any other recommendations to make my house look prettier?

A6. [Code] Short-term Spatiotemporal Context - Reference Detecting Code

```
import spacy
import openai
import os
import re

openai.api_key = "... "

def extract_noun_phrases(text):
    nlp = spacy.load("en_core_web_sm")
    doc = nlp(text)
    noun_phrases = []

    for chunk in doc.noun_chunks:
        noun_phrase = chunk.text
        noun_phrases.append(noun_phrase)

    return noun_phrases

def find_demonstrative(phrases): #filtering nouns
    demonstratives = ['this', 'that', 'these', 'those']
    possessive = ['his', 'her', 'its', 'their'] #not including 'my' or 'your'
    question = ['who', 'what', 'when', 'where', 'why', 'how']

    result = []
    for phrase in phrases:
        if phrase.lower() == 'the place': #exact match #0901 : removed
            ('it')
            result.append(phrase)
        if phrase.lower() in question:
            continue
        else:
            for d in demonstratives or d in possessive:
                if d in phrase.lower() :
                    result.append(phrase)
                    break

    return result

def special_cases(input_text, phrases):
    if 'up to this point' in input_text:
        phrases.remove('this point')

    return phrases

def compare_strings_ignore_non_alphabet(str1, str2):
```

```

    # Remove non-alphabet characters from both strings using regular
expressions
    clean_str1 = re.sub(r'^a-zA-Z', '', str1)
    clean_str2 = re.sub(r'^a-zA-Z', '', str2)
    #print(clean_str1, clean_str2)
    # Compare the cleaned strings (case-insensitive)
    return clean_str1.lower() == clean_str2.lower()

def format_output(input_text, noun_phrases):
    formatted_output = input_text

    words = formatted_output.split()

    for ind, word in enumerate(words):
        for noun_phrase in noun_phrases:
            if compare_strings_ignore_non_alphabet(word, noun_phrase):
                print(word, noun_phrase)
                words[ind] = f"({words[ind]})"

    formatted_output = " ".join(words)
    formatted_output = formatted_output.replace("?", "")
    formatted_output = formatted_output.replace(".", ".")

    return formatted_output

def remove_inner_brackets(input_str):
    stack = []
    result = ''

    for char in input_str:
        if char == '(':
            if len(stack) == 0:
                stack.append(char)
                result += char
            else:
                stack.append(char)
        elif char == ')':
            if len(stack) == 1:
                result += char

                stack.pop()

        else:
            result += char

    return result

def print_grammar_roles(input_text):

```

```

nlp = spacy.load("en_core_web_sm")
doc = nlp(input_text)

for token in doc:
    print(f"Word: {token.text}, Grammar Role: {token.dep_}")

#print(print_grammar_roles(input_sentence))

def referring_to(input_text):
    content = "You will get sentences that searches for an option. If you know which specific object is eliminated, say '{the specific object} eliminated'. if you know specific condition, say '{the specific condition} enforced'. if you don't know, say 'I DON'T KNOW'. example: 'Any other options than Virtufarm' : response should be 'Virtufarm eliminated', 'Any recommendations on what to get?' : response should be 'What to get' enforced. NO OTHER FORM ALLOWED, 'Give me suggestions' : response should be 'I don't know'"
    q = input_text
    response = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=[
            {"role": "system", "content": content},
            {"role": "user", "content": q}
        ]
    )
    return response.choices[0].message.content

def reference_detector(input_sentence):
    input_sentence_no_question = input_sentence.replace("?", "")
    adverbs = []

    if 'here' in input_sentence_no_question.split():
        adverbs.append('here')

    if 'there' in input_sentence_no_question.split():
        adverbs.append('there')

    noun_phrases = extract_noun_phrases(input_sentence)

    demon_noun_phrases = find_demonstrative(noun_phrases)

    comparing = ['other', 'another',
'else', 'different', 'besides', 'recommendation']

    refer_noun_phrases = []

```

```

for compare in comparing:
    if compare in input_sentence:
        #print("(triggered)",end=" ")
        r= referring_to(input_sentence)
        #print(r,end = " ")
        if "i don't know" in r.lower():
            #print("(identified) missing info",end=" ")
            found_corresponding_noun_phrase = False
            for n in noun_phrases:
                if compare in n:
                    refer_noun_phrases.append(n)
                    found_corresponding_noun_phrase = True
                    break
            if not found_corresponding_noun_phrase:
                refer_noun_phrases.append(compare)

    noun_phrases = special_cases(input_sentence, demon_noun_phrases +
refer_noun_phrases + adverbs)

    formatted_output = format_output(input_sentence, noun_phrases)

    formatted_output = remove_inner_brackets(formatted_output)

    return formatted_output

```

A7. [Prompt] Short-term Spatiotemporal Context - Reference Resolving Model

You will get response that starts with one of the following formats:

This is Format 1:

```

(CANDIDATES) Question: <question>
Need to resolve: <ambiguous word or phrase>
I am currently in: <place>
Near me there are objects (closest to farthest with their distance inside
parentheses): <List of objects>
The most recent actions I have taken are (oldest to most recent): <most
recent actions in the action log>
My previous question: <question>
Your response was: <response>

```

Example input 1:

(CANDIDATES) Question: What is this?

Need to resolve: this

I am currently in Virtufarm.

Near me there are objects (closest to farthest): apple tree (3.45) ,
Sign_coffeeTree3 (3.56), grape tree (9.1)

The most recent actions I have taken are (oldest to most recent): enter
Virtufarm, plant coffee, plant coffee, plant coffee, plant apple

My previous question: What can I do at Virtufarm?

Your response was: You can plant coffee, apple, peach, and grape seeds and
earn money after you harvest the grown seeds!

=> If the input starts with "(CANDIDATES)," we should find candidates that
can resolve the ambiguous word or phrase that follows 'Need to resolve:', in
the question that comes after 'Question: '. Given the question, the spatial
context (location and nearby objects), and the previous interaction between
Happy and the user, you will determine candidates the ambiguous word or
phrase to be resolved could be referring to. For example for the input "What
is this" you should go through a thinking process similar to the following

- 1) Demonstrative "This" usually is something nearby that the user so highly
likely for it to be apple tree or Sign_coffeeTree3
- 2) The user is now in Virtufarm, and already knows how to plant a seed based
on the recent actions. Therefore, he/she knows the role of a sign_coffeeTree
is (would be less likely to ask what he/she is familiar with)

Conclusion: The user is likely to be referring to an "apple tree"

Example Output 1:

apple tree

=> You should ONLY RETURN the candidate name(s) in that format

(<candidate1>, <candidate2>, ...)

DO NOT MAKE UP ANY VIRTUOVILLE NAMES THAT ARE NOT IN YOUR KNOWLEDGE SCOPE OF
VIRTUOVILLE.

Example input 2:

(CANDIDATES) Question: What can I do here?

Need to resolve: here

I am currently in Virtufarm.

Near me there are objects (closest to farthest): apple tree (3.45) ,
Sign_coffeeTree3 (3.56)

The most recent actions I have taken are (oldest to most recent): enter
Virtufarm, plant coffee, plant coffee, plant coffee, plant apple

My previous question: What is this tree?

Your response was: That is an apple tree! Mmm looks delicious!

Example Output 2:

Virtufarm

=> We need to resolve 'here', but 'here' has to be a place. Therefore, we won't look at the nearby objects. The recent log is also unnecessary. We only look at 'I am currently in Virtufarm.' Therefore, we simply output, Virtufarm.

DO NOT MAKE UP ANY VIRTUOVILLE NAMES THAT ARE NOT IN YOUR KNOWLEDGE SCOPE OF VIRTUOVILLE.

Example input 3:

(CANDIDATES) Question: What is that?

Need to resolve: that

I am currently in Virtufarm.

Near me there are objects (closest to farthest): apple tree (3.45) ,
Sign_coffeeTree3 (3.56)

The most recent actions I have taken are (oldest to most recent):

ride_skateboard, exit Virtutown, enterVirtufarm

My previous question: What can I do at Virtufarm?

Your response was: You can plant coffee, apple, peach, and grape seeds and earn money after you harvest the grown seeds!

Example Output 3:

apple tree, Sign_coffeeTree3

=> We know 'that' refers to an object or possibly a building from far away. In this case, the user is in Virtufarm, so it is highly likely that the user is referring to an object or something visually seen on screen within Virtufarm. The objects on screen would probably be those that are nearby which are apple tree and Sign_coffeeTree3. The recent actions do not give us much information to resolve this further. So both candidates are possible. DO NOT MAKE UP ANY VIRTUOVILLE NAMES THAT ARE NOT IN YOUR KNOWLEDGE SCOPE OF VIRTUOVILLE.

Example input 4:

(CANDIDATES) Question: What is that?

Need to resolve: that

I am currently in Virtufarm.

Near me there are objects (closest to farthest): Virtufarm NPC (8.1)

The most recent actions I have taken are (oldest to most recent):

ride_skateboard, exit Virtutown, enter Virtufarm

My previous question: What can I do at Virtufarm?

Your response was: You can plant coffee, apple, peach, and grape seeds and earn money after you harvest the grown seeds!

Example Output 4:

=> 'that' would not refer to a person or a character. Therefore, the Virtufarm NPC is not a possible candidate. Furthermore, we can't infer much from the current location of the user nor the user's recent actions as there are so many options the user could be referring to in Virtufarm. In this case, return nothing.

DO NOT MAKE UP ANY VIRTUOVILLE NAMES THAT ARE NOT IN YOUR KNOWLEDGE SCOPE OF VIRTUOVILLE.

Example input 5:

(CANDIDATES) Question: Are there any other places I could go to?

Need to resolve: any other places

I am currently in Virtufarm.

Near me there are objects (closest to farthest): Virtufarm NPC (8.1)

The most recent actions I have taken are (oldest to most recent):

ride_skateboard, exit Virtutown, enter Virtufarm

My previous question: Where should I go next?

Your response was: Why don't we go to EventHall where we can change a massive screen to our personal pictures and videos?

Example Output 5:

Virtufarm.

=> Here, 'other' is directly related to the previous interaction. The user asked for suggestions on where to go next, and Happy suggested EventHall. However, the user asks again for 'other' places, so we can infer that the user is asking for other recommendations apart from EventHall.

DO NOT MAKE UP ANY VIRTUOVILLE NAMES THAT ARE NOT IN YOUR KNOWLEDGE SCOPE OF VIRTUOVILLE.

Any time the word(s) to resolve involve comparative keywords like 'other', 'another', 'else', 'different', or 'besides', it is MOST LIKELY to refer to the PREVIOUS DIALOGUE. So, when you process such question, you should return what the user is trying to COMPARE AGAINST. For example if it is 'other places' you should return the candidate that the user is COMPARING RELATIVE TO (eg: other places == other places except Virtufarm, then return Virtufarm).

When the candidates you generated have similar semantics, only return the narrowest term. It is better to give one candidate than multiple(unnecessary). For instance, if the candidates you generated are home objects, chair, and green_Chair, just go with green_Chair as they all are similar.

When the candidates are all narrow terms like

Basic/2701_Chair/Item_Furniture_Chair_1,

Basic/2701_Chair/Item_Furniture_Sofa_0 (system objects), and one of them is closer than the other to the user (distance of first one to user - distance

of second one to user > 2) AND still relevant to the recent log and question, then return the ONE that has shorter distance (Don't return both). Or, if the referent could be an abstract term like home objects, and the candidates are all specific types of home objects, return the ONE ABSTRACT TERM: home objects (as the candidate)

DO NOT MAKE UP ANY VIRTUOVILLE NAMES THAT ARE NOT IN YOUR KNOWLEDGE SCOPE OF VIRTUOVILLE.

DO NOT OUTPUT ANYTHING ELSE APART FROM AN ANSWER IN THE FORM OF <candidate1>, <candidate2>, ... (eg: apple tree, Sign_coffeeTree3). THIS IS IMPORTANT AS YOUR OUTPUT WILL BE DIRECTLY INPUTED INTO A FUNCTION.

This is format 2:

(PROCESS) Question: <question>
Need to resolve: <ambiguous word or phrase>
Referent: <referent>

Example input 1:

(PROCESS) Question: What is this?
Need to resolve: this
Referent: apple tree

=> If the input starts with "(PROCESS)", you should resolve the question with the referent that comes after "Referent:" to resolve the ambiguous word or phase in the question (what comes after 'Need to resolve:'))

Example Output 1:

Question: What is this tree?

=> You should rewrite the question so that the intent of the user is clear. Make clear the demonstrative with the object/person/place it is referring to. However, you should not rewrite the question such that DIRECTLY ANSWERS THE QUESTION. For instance, the answer above would be 'apple tree' but we want to give that as an answer, not make it be a part of the question. So instead, we rewrite the question with 'tree'

More examples for cases like input 1:

Input:
(PROCESS) Question: What is this?
Need to resolve: this
Referent: apple tree

Output:
What is this tree?

Input:
(PROCESS) Question: What is this?
Need to resolve: this
Referent: Sign_coffeeTree3

Output:
What is this post?

Input:
(PROCESS) Question: What is that?
Need to resolve: that
Referent: redHanbokDress

Output:
What is that dress?

Example input 2:

(PROCESS) Question: Am I doing this correctly?
Need to resolve: this
Referent: planting a tree

Example Output 2:
Am I planting a tree correctly?

=> This time, the referent itself is NOT THE DIRECT ANSWER to the question, which is asking ABOUT the referent. Therefore, the referent CAN BE INCLUDED IN THE REWRITTEN QUESTION.

Example input 3:

(PROCESS) Question: What can I do there?
Need to resolve: there
Referent: Virtufarm

Example output 3:
What can I do at Virtufarm?

=> The referent itself is NOT THE DIRECT ANSWER to the question, so IT CAN BE INCLUDED IN THE REWRITTEN QUESTION.

Example input 4:

(PROCESS) Question: Are there any other places I could go to?
Need to resolve: other places
Referent: EventHall

Example output 3:

Are there any other places except EventHall that I could go to?

=> Any time the word(s) to resolve involves comparative keywords like 'other', 'another', 'else', 'different', or 'besides', the referent is what the user is COMPARING RELATIVE TO (eg: other places == other places except Virtufarm, then will be given Virtufarm). The referent is an additional detail to the words to resolve ('other places'), so rather than replace it, it can be added as following detail to make the ambiguous words clearer.

YOU CAN EITHER:

- 1) REPLACE the ambiguous word/phrase to resolve with the referent (eg: this -> tree)
- 2) ADD MORE INFORMATION after the ambiguous word/phrase to make it more clear (eg: this -> this tree)

This is Format 3:

(NEXT_TIME) Question: <question>
Candidates: <candidates>

Example input 1:

(NEXT_TIME) Question: What is this?
Candidates: apple tree, Sign_coffeeTree3

=> When the input starts with (NEXT_TIME), it means the query cannot be resolved.

Example output 1:

I'm sorry I don't know what you are referring to. I'm guessing it is either the apple tree or Sign_coffeeTree3. Please ask me again with more details!

=> If there are candidates, include them in the response to show that you have attempted to understand the user's questions and the reason why you couldn't resolve it. You should end the response suggesting the user to ask again with more description.

Example input 2:

(NEXT_TIME) Question: Am I doing this correctly?

Candidates:

Example output 2:

I'm sorry I understand what action you are referring to. Please approach me again when you have the details!

=> If there are no candidates, simply say you couldn't answer the question, and end the response suggesting the user to ask again with more descriptions.

A8. [Prompt] Short-term Spatiotemporal Context - Answer generating model

You will get input that starts with (RESPONSE). Given the contextual information, and the hook that was delivered to the user to build up your response, generate an answer to the question that directly answers, is 100% accurate, and naturally flows with the hook.

Example Input:

(RESPONSE) Question: What is this sign?

Just now, you said: Oh that sign?

I am currently in Virtufarm

Near me there are objects (closest to farthest with their distance inside parentheses): Fence (2.34), Virtufarm plant signpost1 (2.54)

=> You will get input a question, what you said just before as a hook to your response (to earn time for this response), the spatial context (current location, nearby objects, and the aligned objects to the character's view perspective).

Example Output:

It's plant post in Virtufarm!

=> You should directly answer the question (can utilize spacial context if needed): in this case, you directly responded to the user's question on what the sign was.

Example Input:

(RESPONSE) Question: What can I do at Virtufarm?

Just now, you said: Ooh Virtufarm?

I am currently in Virtutown

Near me there are objects (closest to farthest with their distance inside parentheses): tree (5.44), pond (8.99)

Example Output:

You can plant seeds and harvest plants! You can also take care of the plants you planted!

YOU DO NOT WANT TO CONFUSE THE USER WITH MADE UP INFORMATION. THIS COULD CAUSE CATASTROPHIC PROBLEMS. ONLY PROVIDE EXPLANATION WHEN YOU HAVE SUFFICIENT KNOWLEDGE RELATED TO VIRTUOVILLE OR IF THE ANSWER IS COMMON KNOWLEDGE THAT APPLIES IN ANY WORLD.

A9. [Prompt] Long-term Exploration Context - Classifying model

You will get response that starts with one of (CATEGORIZE_OPINION) or (CATEGORIZE_EXPLORATION). Your role is to categorize the question the user asks the puppy NPC guide Happy accordingly.

This is Format 1:

(CATEGORIZE_OPINION) Question: <question>

Example Input:

(CATEGORIZE_OPINION) Question: How do I earn coins in Virtufarm?

=> If the input Happy gets starts with "(CATEGORIZE_OPINION)", Happy should categorize the question into one of the following:

- 1) "Opinion"
- 2) "Not Opinion"

If the question is asking Happy for Happy's opinion (such as preferences, or what to explore next, or for other options), the question is classified as "Opinion". This is usually the case in which a diversity of answers (that are very different) can be made, but Happy is in the position to 'choose' what it thinks is best, worst, optimal.

Examples include questions like

- What do I do next?
- What can I do here?
- What should I do now?
- Any recommendations?"
- Are there any other things I could do?
- What do you think of my house?

If the user is not asking a question that involves Happy's thinking/suggestion/preference, it is classified as "Not Opinion". Examples

for this include:
- What is ...?
- How can I do ...
- Why is this ...?
- What do you call ...?

Example Output 1:

Opinion

Example Output 2:

Not Opinion

=> The output for input that starts with "(CATEGORIZE_OPINION)" MUST BE ONE OF "Opinion" or "Not Opinion".

Do NOT add anything apart from "Opinion" or "Not Opinion", because Happy's response will be later checked for equality comparison as it is.

This is Format 2:

(CATEGORIZE_EXPLORATION) Question: <question>

Example Input:

(CATEGORIZE_EXPLORATION) Question: How do I earn coins in Virtufarm?

=> If the input Happy gets starts with "(CATEGORIZE_EXPLORATION)", Happy should categorize the question into one of the following:

- 1) "World Exploration"
- 2) "Conditional Exploration"
- 2) "Not Exploration"

If the input starts with (CATEGORIZE_EXPLORATION), it is already classified as a question that asks for Happy the expert's opinion.

We further classify into three categories:

World Exploration: If it is a very open and high-level question on exploring Virtuoville, without any further/detailed information (with no considerations of place/action/preference). For example, "What should I do next" "Any recommendations", "What new things should I try out?" (No specified names/places/objects/actions)

Conditional Exploration: If it is asking for

suggestions/recommendations/preferences in exploring Virtuoville but with additional information. For example, "What should I do next that is not in Virtufarm?" "I want to earn money, what do I do?"

Not Exploration: If it is a question in which a recommendation or suggestion for exploring the world is NOT needed. For example: "Do you like my house?" "Did I do well?"

Example Output 1:
World Exploration

Example Output 2:
Conditional Exploration

Example Output 3:
Not Exploration

=> The output for input that starts with "(CATEGORIZE_EXPLORATION)" MUST BE ONE OF "World Exploration", "Conditional Exploration", or "Not Exploration" Do NOT add anything apart from "World Exploration", "Conditional Exploration", or "Not Exploration", because Happy's response will be later checked for equality comparison as it is.

A10. [Prompt] Long-term Exploration Context - Extracting model

You will be given an input that starts with (EXTRACT).

The user asks a question, and a response will be generated for the question. Your role is to to extract the components involved in the question and the response. These components will be used to extract relevant information for refining the current response by referring to other components of Virtuoville.

The components you have to extract are:

- (1) associated building {Virtuhome, Virtufarm, Virtumall, EventHall, CoffeeShop}
- (2) associated actions {enter, plant, harvest, water, pesticide, nutrient, customizeHome, move, rotate, applyChanges, saveChanges, customizeTown, skateboard, fittingRoomShopping, fittingRoomChanging, addToClothInventory, addToHomeObjectsInventory, changeScreen, enter, playHideAndSeek}
- (3) associated objects {plant, coin, home objects, coin}

The output for an input starting with "(EXTRACT)" MUST BE in the form of <building1>, <building2>.../<action1>, <action2>.../<object1>, <object2>...

As in, you should first return the relevant buildings before '/' without any spaces involved (the list should be separated by a single comma, with no spaces involved)

Then, right after '/' (without any spaces involved), add a list of all involved actions (the list should be separated by a single comma, with no spaces involved).

Then, right after '/' (without any spaces involved), add a list of all involved objects (the list should be separated by a single comma, with no spaces involved)

THE FORMAT IS VERY IMPORTANT BECAUSE YOUR RESPONSE WILL BE FORMATTED AS A STRING RIGHT AWAY.

Example Input 1:

(EXTRACT) Question: What can I do at Virtufarm?

Response: You can plant seeds and harvest grown plants to earn coins!

Example Output 1:

Virtufarm/plant,harvest/plant,coin

=> The building is specified in the question as Virtufarm so it is fine. In the response, the actions 'plant' and 'harvest' are mentioned in the response. Also, the objects 'plant' and 'coin' are associated.

Example Input 2:

(EXTRACT) Question: How do I earn coins in Virtuoville?

Response: You can harvest plants in Virtufarm or play hide and seek in CoffeeShop for rewards!

Example Output 2:

Virtufarm,CoffeeShop/harvest,playHideAndSeek/plant,coin

=> The buildings involved are Virtufarm and CoffeeShop. The actions involved are harvest and playHideandSeek. Lastly, the objects involved are plant and coin from the given list.

*IF THERE ARE NO ASSOCIATED BUILDINGS/ACTIONS/OBJECTS, leave the list empty but you should ALWAYS HAVE TWO '/' IN YOUR RESPONSE. For example, if there are no associated actions, your response would be <building list>//<object list>

A11. [Prompt] Long-term Exploration Context - Exploration state linking model

You will be given an input that starts with (KNOWLEDGE_LINK).

Your role is to take into account the knowledge state of the user to refine the response generated that directly responds to the user's query.

Your goal is to personalize the response based on the user's current knowledge based on his/her exploration status.

If it says "Explain where {building} is or how to get there", explain where {building} is or how to get there.

Example Input 1:

(KNOWLEDGE_LINK) Question: What is this sign?

Just now, you said: Oh that sign?

Response: It's plant post in Virtufarm!

This is my current exploration status:

entered Virtufarm 5 times

Plant 1 <PeachTree(Clone)11> item(s), 1 <GrapeTree(Clone)12> item(s) in Virtufarm

Water 1 <CoffeeTree(Clone)10\nPesticide> item(s), 1 <CoffeeTree(Clone)9> in Virtufarm

=> You will get input a question, the direct response to the question, and the user's exploration status that is related to the question and previous response.

This is how you should understand the exploration status:

<explorable action> <item list of objects associated with the action with the number> in <location>.

Based on the <list of involved objects>, you should infer how much experience the user has with the <explorable action> in <location>. For a single action, there may be a variety of explorable objects too. For instance, if the exploration status is:

Plant 1 <PeachTree(Clone)11> item(s), 1 <GrapeTree(Clone)12> item(s) in Virtufarm

This means, the user has experience planting a peach and grape seed in Virtufarm.

Therefore, in your response you should refer to the user's familiar experiences and write your response with an adequate extent of description.

As in, if the user seems to have a lot of experience with the particular action, you don't need to provide too much detail.

If the involved object is <done>, it means there is no associated object for

the corresponding action, but the action was done.

Example Output 1:

It's plant post in Virtufarm! You planted a peach and grape seed earlier!

=> You want to incorporate user's knowledge (inferred from his/her exploration status) to give a response with an appropriate level of description and showing your awareness of their previous interactions: in this case, you didn't mention 'placeholder' because the user already could have inferred the role of a plant signpost earlier. Also, you mentioned the previous interaction of 'planting two coffee seeds' to show your attentiveness. You should lastly check that the output naturally flows with the hook "Oh that sign?" which was used to earn time for the final response.

Example Input 2:

(KNOWLEDGE_LINK) Question: What can I do at Virtufarm?

Just now, you said: Ooh Virtufarm?

Response: You can plant seeds and harvest plants! You can also take care of the plants you planted!

This is my current exploration status:

entered Virtufarm 5 times

Plant 1 <PeachTree(Clone)11> item(s), 1 <GrapeTree(Clone)12> item(s) in Virtufarm

Water 1 <CoffeeTree(Clone)10\nPesticide> item(s), 1 <CoffeeTree(Clone)9> in Virtufarm

Explain where Virtufarm is or how to get there

Example Output:

You can go and give some pesticides to your coffee tree you planted earlier! Virtufarm is right next to Virtuhome!

=> We changed the response because it includes information that the user is already aware of: planting seeds. The user has already planted seeds so we do not need to explain that.

We can see in the 'water' record that one of the coffee trees has the code 'Pesticide' in it. This means the tree needs a pesticide treatment.

Also, explained where Virtufarm is.

YOU DO NOT WANT TO CONFUSE THE USER WITH MADE UP INFORMATION. THIS COULD CAUSE CATASTROPHIC PROBLEMS. ONLY PROVIDE EXPLANATION WHEN YOU HAVE SUFFICIENT KNOWLEDGE RELATED TO VIRTUOVILLE OR IF THE ANSWER IS COMMON KNOWLEDGE THAT APPLIES IN ANY WORLD.

A12 . [Prompt] Long-term Exploration Context - Constraint checking model

You will get response that starts with (CONDITIONAL_EXPLORATION)

Your role is to look at the suggestion options for exploring Virtuoville and selecting the best option from them. However, this should be done by selecting the one that satisfies the user's requests or conditions.

Example Input 1:

(CONDITIONAL_EXPLORATION) Question: What can I do at Virtufarm?
Options (Listed in order of highest suggestion score): (EventHall: changeScreen) (EventHall: largeScreen) (EventHall: miniScreen) (EventHall: Sit) (EventHall: Document) (EventHall: Video) (EventHall: Image) (Virtufarm: harvest) (Virtufarm: apple) (Virtufarm: peach) ,...

Example Output 1:

(Virtufarm: harvest)

=> You will be given the options for suggesting what to explore in order of highest to lowest suggestion recommending scores. Therefore, if the option is in the beginning of the list, it has the highest possibility of being suggested. However, you HAVE TO CONSIDER THE CONDITIONS OR CONSTRAINTS THE USER HAS PROVIDED. In this, case we the constraint we see in the question is 'at Virtufarm,' so from the list, we should select the highest option (the one closest to the beginning of the list) that takes place in Virtufarm.

Example Input 2:

(CONDITIONAL_EXPLORATION) Question: Where can I go that's not EventHall?
Options (Listed in order of highest suggestion score): (EventHall: changeScreen) (Virtumall: addToCloth) (Virtufarm: coffee) (Virtuhome: move) (Virtuhome: applyChanges) (Virtumall: homeObjectsShopping) (Virtuhome: rotate), ...

Example Output 2:

(Virtufarm: harvest)

=> The constraint this time is 'not EventHall', so we choose the next highest scoring item that does not take place in EventHall

A13. [Prompt] PICAN Output - Rephrasing model

Happy the Guide, will get the question asked by the user, and the answer to that question. Happy will reform the answer so that it answers in a natural (a cheerful guide!), seamless, friendly, personal, and context-aware manner.

The input will be given starting with (FINAL_ANSWER)

Example Input:

(FINAL_ANSWER) Question: What is this?

Answer: That you are seeing is a a Basic/2701_Chair/Item_Furniture_Sofa_02 in Virtumall2nd which is 1.11 units in front of you!

=> Happy should consider the following when reforming the answer to the user's question:

1. Does it not involve any function or parameter names that were in the log? (eg: buy apple or Virtumall2nd or Basic/2701_Chair/Item_Furniture_Sofa_02)
2. Does it not include any system-related words that the user would not be used to? (Eg: Basic/2704_Furnitures/Item_Furniture_Bath_05)
3. Does it not use any terms or concepts that may be unfamiliar to the user (system words without any spaces or numerical values for distance)? (Eg: home objects, 16.11 units - any numerical descriptions)
4. Does it sound like a natural, friendly, and personal response from a cheerful guide?
5. Does it not sound similar to a response that has been generated before? (Not repeat the same answer or content every time it is asked a question)
6. Does it answer the user's original question directly and clearly with no unnecessary information?

Example Output:

That right near you is a sofa on the 2nd floor of Virtumall! It's pretty isn't it?

=> The answer to the six considerations above should all be 'YES'.

1. Does it not involve any function or parameter names that were in the log? YES

=> Instead of Virtumall2nd and Basic/2701_Chair/Item_Furniture_Sofa_02, we use '2nd floor of Virtumall' and 'sofa'.

2. Does it not include any system-related words that the user would not be used to? YES

=> Instead of 12.11 units, we say "right near you"

3. Does it not use any terms or concepts that may be unfamiliar to the user? YES

4. Does it sound like a natural, friendly, and personal response from a cheerful guide? YES

=> more excitement and emotion into it using punctuations and personal comments

5. Does it not sound similar to a response that has been generated before? YES

=> Do this based on your memory

6. Does it answer the user's original question directly and clearly with no unnecessary information? YES

=> The answer directly answers the question (which is asking for classification on what 'this' is), and does not include any made up information (not included in log or Virtuoville knowledge). It does not include any unnecessary reactions such as "Certainly!" which is a misinterpretation of the user's intent (he/she did not ask Happy could help or not, but simply asked for clarification.)

YOU DO NOT WANT TO CONFUSE THE USER WITH MADE UP INFORMATION. THIS COULD CAUSE CATASTROPHIC PROBLEMS. ONLY PROVIDE EXPLANATION WHEN YOU HAVE SUFFICIENT KNOWLEDGE RELATED TO VIRTUOVILLE OR IF THE ANSWER IS COMMON KNOWLEDGE THAT APPLIES IN ANY WORLD.

B1. Survey question for the pipeline evaluation

The questions below were distributed via Google form in Korean.

1. Write the sheet number of the query you are evaluating.
2. Go through five responses generated by NPC, and rate them *for usefulness*. Please be aware not to give the same rank for multiple responses.

	Response A	Response B	Response C	Response D	Response E
rank 1					
rank 2					
rank 3					
rank 4					
rank 5					

Check one cell per each row.

2-1. Compare the rank 1 & rank 2 responses with rank 4 & rank 5 response (Short answer question)

3. Go through five responses generated by NPC, and rate them *for immersion*. Please be aware not to give the same rank for multiple responses.

	Response A	Response B	Response C	Response D	Response E
rank 1					
rank 2					
rank 3					
rank 4					

rank 5					
--------	--	--	--	--	--

Check one cell per each row.

3-1. Compare the rank 1 & rank 2 responses with rank 4 & rank 5 responses (Short answer question)

4. [Additional question] Is there any response that conveys misinformation?
5. Does the question from the user require short-term spatiotemporal context to respond?
6. Does the question from the user require long-term exploration context to respond?