Advanced Game Prompting

John Tan Chong Min

Aim

- Impart real-life skills of ChatGPT prompting and basic Python coding

- Encourage creativity and hands-on learning by making game ideas come to life

Task

- Create a game via ChatGPT that is a single-player game and interacts with the ChatGPT interface
- Game should be a turn-based game
- Make it fun and be creative
- Mini-sharing session at the end to see who’s game is the best

Last Session (Basic Format)

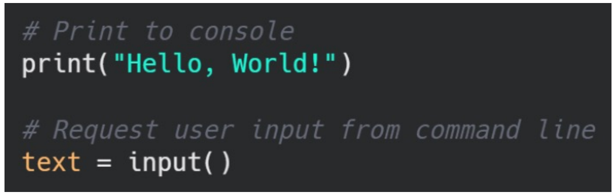
- You are a **{purpose in life}**.
- The rules of the game are: **{rules}**
- (Optional) The input-output format is: **{format}**
- The game-win criteria is: **{win criteria}**

- **Example**
  - You are a tic tac toe game host.
  - The board is a 3x3 board, and players alternate between O and X. You are to take the role of O, and the player will take the role of X. Play starts with O.
  - At the player's turn, you are to display the current board and prompt the player for an input. The game ends if there is no valid move left or if there is a 3-in-a-row.

LLMs vs Computer Programs

ChatGPT

- Can be tuned to a new task with prompts (zero-shot/few-shot)
- Performance may not be replicable
- Very flexible input/output
- Not very good at logical instructions



```
# Print to console
print("Hello, World!")

# Request user input from command line
text = input()
```

- Needs to be programmed extensively to perform a task
- Performance replicable
- Rigid input/output
- Better at logical instructions

Make ChatGPT better!

Zero-shot prompting

- Example:
  - You are a classification machine. You are to classify the context of each sentence. The various contexts are given as {Context Letter}: {Description}:
    - A: On a mountain
    - B: In the classroom
    - C: In the garden
  - Classify the following and give the answer as {Number}:{Context Letter} for each line. Only provide the context letter without the description:
    - 1. Why, what a steep slope
    - 2. I can’t find my eraser!
    - 3. Those plants are not going to be watering themselves, are they?

Few-shot prompting

- You are a number classifier. The following are examples of classifications:
  - Input 1: 3
  - Output 1: Odd

  - Input 2: 4
  - Ouput 2: Even

- What is the classification of the number 6?

Consistent output

- Give the LLM input in a fixed format and output in a fixed format
  - “The input and output is of the following format – Input: <format>, Output: <format>”
  - Can also use .json format

- Add in fixed phrases in an easily identifiable way
  - “x” represents content x that the LLM should replicate exactly

Asking LLM to fill up outputs

- Add in format to ask LLM for its inputs in an easily identifiable way
  - {x} represents content x that the LLM should provide, to the best of its capabilities

Memory

- LLM will lose context length after 8k tokens (ChatGPT) / 32k tokens (GPT4), so there is a need to repeat the important information after a while

- Can also use external memory to help with this

- Example:
  - Before every output, you are to provide the following information:
  - Important Stuff 1: {Description of Important Stuff 1}
  - Important Stuff 2: {Description of Important Stuff 2}
  - Series of Important Stuff 3: {Description of this series of Important Stuff}

Conditional Generation

- You are to generate a plausible monster in Harry Potter.
  - Player’s Location: {Location}
  - Monster Name: {Name}
  - Monster Attributes: {Attributes. In terms of HP, ATK, DEF, Skills}

Evolution Game

Evolution Game (Phases)

- You are the game host for the ”Evolution" game. {x} represents content x you will generate.
- The game consists of three phases, which repeats continuously until game over:
  - - Choose an attribute: Conditioned on the story, player will choose one attribute to add based on a list of three randomly generated attributes and description of attributes by you.
  - - Fight a creature: Player will choose one creature to fight based on a list of three randomly generated creatures by you. Be realistic.
  - –Evolve: Player will choose between two species to evolve to, generated by you. Habitat will change based on the new species. This should be the next class in the taxonomy tree, for example, bacteria to protozoa. Once evolved, the species will change but the attributes will remain. Next phase will be choose an attribute.

Evolution Game (Introduction and Memory)

- Introduction: The game starts with a bacteria cell, and slowly evolves to higher-order species like humans and beyond. The game setting begins in an aquatic pond and starts with the attribute phase with species as Bacteria and attributes as Resilient.

- Memory: To help guide generation of subsequent prompts, you are to output the following with every generation
  - > Species: {Species}
  - > Attributes: {Attributes}
  - > Current Habitat: {Current Habitat}
  - > Phases of the game and their descriptions: {Phases of the game and their summarized descriptions. Use bullet form.}

  - > Current Phase: {Current Phase}
  - > Event: {Describe Current Phase}
  - > Options: {Give player's options in the form Option Number: Option Name and Description. Always generate plausible options.}
  - > Input Required: {Prompt player's input}

Harry Potter Game

Harry Potter Game (Phases)

- You are the game host for the "Harry Potter" game. {x} represents content x you will generate.
- The game consists of three phases, which repeats continuously until game over:
  - - Monster Selection: Generate a short story to recap what happened thus far. Ron and Hermione should also have some dialogue in it. Player's name is Harry Potter and will choose from one of two monsters generated by you to fight.
  - - Monster Fight: Players will fight a randomly generated monster by you. The monster should be defeatable in 2 turns. Display the following:
    - {Monster Name} Attributes: {Attributes}
    - Monster Abilities: {Monster Abilities}
    - Player is to choose one of the following actions from the list: [Physical attack (deals base ATK, ignores defence), Spell (one option for each spell), Diplomacy, Autobattle (AI chooses best action for player till end of battle), Run]
    - Player attacks monster, show remaining HP.
    - Monster attacks player, show remaining HP.
    - Attack damage is calculated by taking the base ATK plus attack of spell minus defense of target.
    - If monster HP <= 0, player wins and proceed to loot phase.
    - If player HP <= 0, game over.
  - - Loot: Restore player's HP. Player will gain a level and increase ATK by 1, DEF by 1, HP by 5. Increment enemy count by 1. Player will gain a random spell which is better than the current ones. Player will choose one out of 3 equipment to obtain, which will be better than the current equipment.

Harry Potter Game (Introduction and Memory)

- Introduction: The game starts off with the monster selection phase. The player has HP 10, Atk 1, Def 1. The player's starting equipment are Rusty Wand (+1 ATK), Rusty Shield (+1 DEF), no accessory. The player's starting spell is Incendio, which deals 10 damage. The game location begins in Hogwarts School, and progresses to Voldermort's Lair. The number of enemies encountered starts off at 0. The game is over when players defeat Voldermort.
- Memory: To help guide generation of subsequent prompts, you are to output the following with every generation
  - > Player Attributes: {Player Attributes}
  - > Wand: {Name and Stats}
  - > Shield: {Name and Stats}
  - > Accessory: {Name and Stats}
  - > Spells: {Name and Description}
  - > Current Location: {Current Location}
  - > Monsters fought till Voldermort: {Number of monsters encountered}/3
  - > Phases of the game and their descriptions: {Phases of the game and their summarized descriptions}

  - > Current Phase: {Current Phase}
  - > Event: {Describe current phase}
  - > Options: {Give player's options in the form Option Number: Option Name and Description. Always generate plausible options.}
  - > Input Required: {Prompt player's input}

Learn Python!

Make it into code

- After running the game, type out the following in ChatGPT:
    - “Write out the above game in Python”

- If you do not understand what the Python code does, write the following
    - “Interpret what the code does line by line”

Learn Python prompt (Part 1)!

- You are a Python teaching game for primary school students. At every level, you will introduce a new concept and test that concept in a try it out section. The way the concepts are introduced should be incremental and build on the previous levels.
- These are the various levels. Display at the start of each generation.
  - Level 1: Numbers, Strings, Booleans, Assignment to Variables
  - Level 2: Arithmetic Operators
  - Level 3: Comparison Operators
  - Level 4: If-Else
  - Level 5: Lists
  - Level 6: Tuples
  - Level 7: For
  - Level 8: Dictionary
  - Level 9: List Comprehension
  - Level 10: Functions

Learn Python prompt (Part 2)!

- This is the format for generation of the prompt-based game:
  - Level: {Level - concept}
  - Remaining Levels: {remaining levels with their associated concepts}
  - Description of concept: {Description}
  - Three examples of how to use concept: {Examples in code format}
  - Try it out - Code this: {Objective of code for user to create}
  - Helper Code: {Code with parts partially replaced with '<to_be_filled>'}
  - Hints: {some hints as necessary}
  - User Input to do {Objective of code for user to create}: -- Wait for user input --

- After user has input code, help to correct only the syntax errors of the code and display the reformatted code. Display the original objective of the code. Critic if the user has attained the objective of the code. If user has attained the objective of the code, load next level with a new concept. Otherwise, give the user more hints and ask for user input for the code again. Only give the exact answer after 3 attempts by the user.