EXP-02	Comparative Analysis of Naïve Prompting versus Basic
DATE:	Prompting Using ChatGPT Across Various Test Scenarios

Prethiveerajan P -212221230079

Aim:

- To test how ChatGPT responds to naïve prompts (broad or unstructured) versus basic prompts (clearer and more refined) across multiple scenarios, analyzing the quality, accuracy, and depth of the generated responses.

Instructions:

1. **Define the Two Prompt Types:**
   - **Naïve Prompts:** Broad, vague, or open-ended prompts with little specificity.
   - **Basic Prompts:** Clear, detailed, and structured prompts that give specific instructions or context to guide the model.
2. **Prepare Multiple Test Scenarios:**
   - Select various scenarios such as:
     - Generating a creative story.
     - Answering a factual question.
     - Summarizing an article or concept.
     - Providing advice or recommendations.
     - Or Any other test scenario
   - For each scenario, create both a naïve and a basic prompt. Ensure each pair of prompts targets the same task but with different levels of structure.
3. **Run Experiments with ChatGPT:**
   - Input the naïve prompt for each scenario and record the generated response.
   - Then input the corresponding basic prompt and capture that response.
   - Repeat this process for all selected scenarios to gather a full set of results.
4. **Evaluate Responses :**

   Compare how ChatGPT performs when given naïve versus basic prompts and analyze the output based on Quality,Accuracy and Depth. Also analyse does ChatGPT consistently provide better results with basic prompts? Are there scenarios where naïve prompts work equally well?

Deliverables:

- A table comparing ChatGPT's responses to naïve and basic prompts across all scenarios.
- Analysis of how prompt clarity impacts the quality, accuracy, and depth of ChatGPT's outputs.
- Summary of findings with insights on structuring prompts for optimal results when using ChatGPT.

The definitions for the two prompt types:

Naïve Prompts

Naïve prompts are characterized by their broad, vague, or open-ended nature, providing minimal specificity. They may lack clarity and structure, making it difficult for the model to understand the intended context or focus. Examples include questions like "Tell me about technology" or "What is your opinion on books?"

Basic Prompts

Basic prompts are clear, detailed, and structured, offering specific instructions or context to guide the model effectively. They are designed to elicit precise responses and often include relevant details or parameters. Examples include prompts like "Explain the benefits of renewable energy sources in detail" or "Summarize the main themes of 'Pride and Prejudice' in three paragraphs."

Few-Shot Pattern:

In Few-Shot Prompting, you put examples of what you want the GenAI to do in your prompt[1][2][3]. These examples are also called shots[4], which is how the technique gets its name.

Few-shot prompting builds upon the idea that large language models can learn new information from their prompt[5]. By providing a few demonstrations of the desired task or output format within the prompt itself, you guide the model to produce more accurate and consistently formatted responses.

Few-shot prompting sits on a spectrum of "shot" techniques:

1. Zero-shot prompting: No examples are provided
2. One-shot prompting: A single example is given
3. Few-shot prompting: Multiple examples (typically 2+) are included

Basic Prompts:

Make it in proper order Software Engineer - Python specialist needed at TechCorp. 5+ years of experience required. Salary range $90,000 - $120,000. Remote work available. Apply by June 30, 2024.

Chat GPT-Output:

Job Title: Software Engineer - Python Specialist

Company: TechCorp

Experience Required: 5+ years

Salary Range: $90,000 - $120,000

Location: Remote work available

Application Deadline: June 30, 2024

Naïve Prompts

INPUT: Software Engineer - Python specialist needed at TechCorp. 5+ years experience required. Salary range $90,000 - $120,000. Remote work available. Apply by June 30, 2024.

OUTPUT:

Position: Software Engineer

Specialization: Python

Company: TechCorp

Experience Required: 5+ years

Salary Range: $90,000 - $120,000

Work Type: Remote

Application Deadline: June 30, 2024 INPUT: Marketing Manager for GlobalBrand. MBA preferred. 3-5 years in consumer goods marketing. $75K-$95K DOE. Hybrid work model. Applications close July 15, 2024.

OUTPUT:

Position: Marketing Manager

Company: GlobalBrand

Education: MBA preferred

Experience Required: 3-5 years

Industry: Consumer goods

Salary Range: $75,000 - $95,000

Work Type: Hybrid

Application Deadline: July 15, 2024 INPUT: Data Scientist wanted at AI Innovations Ltd. PhD in Computer Science or related field. Minimum 2 years industry experience. Competitive salary €60,000 - €80,000 based on experience. On-site work in Berlin office. Apply by August 31, 2024.

Chat GPT-Output:

Position: Data Scientist

Company: AI Innovations Ltd

Education: PhD in Computer Science or related field

Experience Required: Minimum 2 years

Salary Range: €60,000 - €80,000 (based on experience)

Work Type: On-site (Berlin office)

Application Deadline: August 31, 2024

Persona Pattern:

<u>Basic Prompt:</u>

Input - Design and Develop a game

Output Given -

To design and develop a game, let's break down the process into manageable steps. Since you're experienced with game development and Unreal Engine, we can use that for the development environment. Here's an outline:

Step 1: Conceptualization

- Genre: Choose the genre of the game (platformer, puzzle, action-adventure, etc.).
- Storyline/Objective: Create a basic storyline or objective (e.g., rescue a character, collect items).
- Game Mechanics:
  - Player movement (running, jumping, shooting, etc.).
  - Interaction with objects (collecting, pushing, destroying).
  - Enemy behaviors (AI, attack, defense).
- Level Design: Map out how levels or game environments will progress.

Example Concept

- Genre: Platformer
- Objective: Help the player collect hidden gems while avoiding enemies and solving puzzles to reach the next level.
- Mechanics: The player can jump, crouch, attack enemies, and move objects to solve puzzles.

Step 2: Game Design Document (GDD)

Create a Game Design Document that outlines:

- Game World: Environment, visual style, and background story.
- Player Abilities: Skills or actions the player can take.
- Enemies/Challenges: Define types of enemies or puzzles and how they interact with the player.

- Art and Sound Design: The visual and sound effects needed to bring the game to life.

Step 3: Development

Tools:

- Engine: Unreal Engine (or Unity if preferred).
- Programming Languages: Blueprint scripting or C++ (for Unreal Engine).
- Art: 2D/3D models, textures, sprites (use tools like Blender or Maya for 3D, Photoshop for 2D).
- Sound: Create or source sound effects and background music (Audacity for editing sounds).

Process:

- Prototype: Create a simple prototype to test core mechanics.
- Level Design: Develop initial game levels with obstacles, platforms, or challenges.
- Player Control & Interaction:
- Set up the character controller with movement (e.g., WASD keys or gamepad).
- Add jumping, crouching, and interaction mechanics.
- Enemy AI: Program basic AI for enemies (e.g., patrol, chase, attack behaviors).
- UI Design: Add score, health, and a main menu screen using Unreal's UMG for UI.

Step 4: Testing

- Conduct playtests to gather feedback.
- Fix bugs and improve the gameplay experience based on player input.

Step 5: Deployment

- Package and export the game for different platforms (PC, mobile, console).

Input:

Consider you as a Game developer and I am your client where you need to develop a game for me from scratch and you need to ask questions one by one about how the game should be designed with its difficulty level, obstacles, and other stuff

Output:

Alright! Let's start with the basics.

Question asked from Chat-GPT:
What type of game are you looking for? Is it an adventure, puzzle, strategy, racing, or some other genre?

Input Given:
Adventure

Question asked from Chat-GPT:
Great! For the adventure game, do you have a preferred setting or theme? For example, should it be set in a fantasy world, post-apocalyptic world, jungle, outer space, or something else?

Input Given:
outer space

Question asked from Chat-GPT:
Awesome! An outer space setting sounds exciting. Do you envision the main character as a human astronaut, alien, robot, or something entirely different?

Input Given:
something entirely different

Question asked from Chat-GPT:
Got it! Something entirely different can open up a lot of creative possibilities. Can you describe the main character's unique traits or abilities? What makes them stand out in the outer space adventure?

Input Given:
with some super abilities like Alien X

Question asked from Chat-GPT:
Interesting choice! So, we'll have a character with super abilities akin to Alien X.

What kind of super abilities do you envision for this character? For example, should they have powers like shape-shifting, time manipulation, telekinesis, or something else?

Input Given:
time manipulation

Question asked from Chat-GPT:
What type of game are you looking for? Is it an adventure, puzzle, strategy, racing, or some other genre?

Input Given:
gradually increase in difficulty

Question asked from Chat-GPT:
Great choice! A gradual increase in difficulty will help players learn and adapt as they progress.

Now, let's discuss obstacles. What kind of obstacles do you want the players to encounter? For example, should they face enemies, environmental hazards, puzzles, or a mix of these?

Input Given:
both should face enemies and environmentail hazards

Question asked from Chat-GPT:
Perfect! We'll include both enemies and environmental hazards to keep the gameplay engaging.

For the enemies, do you have any specific types in mind? For instance, should they be robotic foes, hostile aliens, or perhaps space creatures?

Input Given:
space creatures and hostile aliens

Question asked from Chat-GPT:
Got it! We'll include both space creatures and hostile aliens as enemies.

Now, regarding environmental hazards, what kind of hazards do you envision? For example, should there be black holes, asteroid fields, energy storms, or something else?

Input Given:
asteroid fields

Question asked from Chat-GPT:
Asteroid fields will definitely add an exciting challenge!

Next, let's talk about the game's objective. What is the main goal for players in this adventure? For instance, should they be trying to complete missions, collect resources, save another character, or do something else entirely?

Input Given:
trying to create a new universe

Question asked from Chat-GPT:
Creating a new universe is an intriguing objective!

Now, let's discuss the game's progression. Would you like players to level up their character's abilities, unlock new skills, or gather resources to improve their equipment as they progress?

Input Given:
unlock new skills by defeating the enemies

Question asked from Chat-GPT:
Unlocking new skills by defeating enemies adds a great incentive for players to engage in combat!

Finally, how do you envision the game's visual style? Should it be realistic, cartoonish, pixel art, or something else?

Input Given:
realistic

Question asked from Chat-GPT:
A realistic visual style will enhance the immersion in the outer space adventure.

To wrap things up, do you have any specific music or sound design preferences for the game? Should it be orchestral, electronic, or atmospheric sounds to complement the gameplay?

Input Given:
atmospheric sounds

FINAL OUTPUT:
Atmospheric sounds will create an immersive experience for players!

To summarize our game design:

- **Genre:** Adventure
- **Setting:** Outer Space
- **Main Character:** Unique character with time manipulation abilities
- **Difficulty Level:** Gradually increases
- **Obstacles:** Enemies (space creatures and hostile aliens) and environmental hazards (asteroid fields)
- **Objective:** Create a new universe
- **Progression:** Unlock new skills by defeating enemies
- **Visual Style:** Realistic
- **Sound Design:** Atmospheric sounds

Conclusion:

Analysis of How Prompt Clarity Impacts Quality, Accuracy, and Depth

- **Quality Impact:**
  - Naïve prompts result in responses that are often general and lack focus. The information provided may be correct but is often too broad to be actionable.
  - Basic prompts lead to high-quality responses, as they specify the topic and context, allowing ChatGPT to provide relevant and detailed information.
- **Depth Impact:**
  - The depth of information is significantly greater with basic prompts. They typically cover multiple aspects of a topic, providing a thorough understanding of the subject matter.
  - Naïve prompts tend to produce superficial answers that do not delve into the intricacies of the topic, missing out on essential details.
- **Accuracy Impact:**
  - Clarity in prompts directly correlates with the accuracy of responses. Basic prompts provide enough context and detail, enabling ChatGPT to generate precise information tailored to the user's needs.
  - Naïve prompts often lead to less accurate information as they lack specificity, making it difficult for the model to discern what the user is seeking.

Context Pattern:

Basic Prompt:

Input - Hardware used in mobiles

Output From Chat-GPT:

Mobile phones consist of several key hardware components that enable their functionality. Here are some of the most important ones:

1. **System on Chip (SoC):** Combines the CPU, GPU, RAM, and other essential processing units into one chip. Common examples are Qualcomm's Snapdragon, Apple's A-series, and Samsung's Exynos.
2. **Central Processing Unit (CPU):** The brain of the phone, responsible for executing instructions and managing tasks. Modern mobile CPUs are multi-core for better performance and efficiency.
3. **Graphics Processing Unit (GPU):** Handles rendering images, animations, and videos, especially in games and graphics-intensive applications.
4. **Display:** The screen, typically an OLED or LCD panel. It shows the user interface, videos, and apps. Most modern phones use high-resolution displays like AMOLED.
5. **Memory (RAM):** Used by the system to run applications and manage active processes. Typical modern smartphones have between 4GB and 12GB of RAM.
6. **Storage:** Used for storing apps, media, and the operating system. Flash storage (UFS) is commonly used, with sizes ranging from 32GB to 512GB or more.
7. **Battery:** Powers the phone, usually a lithium-ion battery. Modern smartphones come with fast charging, wireless charging, and various battery optimization technologies.
8. **Camera System:** Includes multiple sensors for capturing images and videos. Common sensors are wide-angle, ultra-wide, and telephoto lenses, with megapixel counts ranging from 12MP to 108MP.
9. **Modem:** Facilitates wireless communication, such as 4G LTE, 5G, Wi-Fi, and Bluetooth.
10. **Sensors:** Various sensors like accelerometer, gyroscope, proximity sensor, ambient light sensor, and fingerprint scanner add functionality like orientation detection, screen brightness adjustment, and biometric security.
11. **Speakers and Microphones:** Used for audio output and input, enabling phone calls, music playback, and voice commands.
12. **Ports and Connectors:** Includes USB-C or Lightning ports for charging and data transfer, as well as SIM card slots, and possibly headphone jacks.

Input - Hardware is used in mobiles with its architecture, design, and biggest brand.

Output From Chat-GPT:

Mobile hardware consists of several critical components, each with its own architecture, design, and leading brands. Let's break them down:

1. System on Chip (SoC):

- **Architecture**: SoCs integrate the CPU, GPU, memory controller, modems, and other components on a single chip. They use **ARM architecture** because it's energy-efficient, crucial for mobile devices.
- **Design**: SoCs are designed to be compact and efficient, enabling multitasking, gaming, and communications to function together without overheating or draining battery.
- **Biggest Brands**:
  - **Qualcomm** (Snapdragon series)
  - **Apple** (A-series, M-series chips)
  - **Samsung** (Exynos series)
  - **MediaTek** (Dimensity series)

2. Central Processing Unit (CPU):

- **Architecture**: Most mobile CPUs are based on ARM architecture (e.g., ARM Cortex-A series), which prioritizes low power consumption. CPU cores are often arranged in **big.LITTLE architecture**, where high-performance cores handle intensive tasks, and power-efficient cores handle lighter tasks.
- **Design**: CPUs in mobile SoCs are designed with multi-core configurations (usually between 6 and 8 cores). They focus on optimizing both performance (for tasks like gaming) and efficiency (for everyday use).
- **Biggest Brands**:
  - **ARM** (Cortex-A series)
  - **Apple** (A14, A15 Bionic)
  - **Qualcomm** (Kryo cores in Snapdragon chips)

3. Graphics Processing Unit (GPU):

- **Architecture**: Mobile GPUs are designed to handle graphic rendering and AI tasks. They typically use **ARM's Mali**, **Qualcomm's Adreno**, or **Apple's proprietary GPU**. Their architecture ensures smooth video playback, gaming, and graphics-intensive applications.
- **Design**: GPUs in mobile SoCs are designed for real-time rendering with minimal power consumption, supporting high-resolution displays and augmented reality applications.
- **Biggest Brands**:
  - **ARM** (Mali series)
  - **Qualcomm** (Adreno series)
  - **Apple** (custom GPUs in A-series chips)
  - **Imagination Technologies** (PowerVR)

4. Memory (RAM):

- **Architecture**: Mobile RAM is usually based on **LPDDR (Low-Power DDR)** architecture. LPDDR is optimized for energy efficiency compared to traditional desktop RAM, while still delivering fast performance.
- **Design**: LPDDR RAM is designed in **dual-channel** or higher configurations for better data throughput, ensuring that the phone can handle multitasking without lag.
- **Biggest Brands**:
  - **Samsung**
  - **SK Hynix**
  - **Micron Technology**

5. Storage (Flash Memory):

- **Architecture**: Mobile devices use **NAND flash storage**, typically in the form of **Universal Flash Storage (UFS)**, which offers higher read/write speeds compared to older standards like eMMC.
- **Design**: UFS is designed to handle fast sequential read/write speeds and better power efficiency, enabling quicker app launches and smoother multitasking.
- **Biggest Brands**:
  - **Samsung**
  - **Western Digital**
  - **Kioxia** (formerly Toshiba Memory)

6. Display:

- **Architecture**: Mobile displays are typically **AMOLED** (Active Matrix Organic Light-Emitting Diode) or **LCD** (Liquid Crystal Display) technology. AMOLED is preferred for its deep blacks, vibrant colors, and energy efficiency.
- **Design**: AMOLED displays are flexible and can be designed to be curved or folded. Displays also incorporate high refresh rates (90Hz, 120Hz, etc.) for smoother scrolling and better gaming experiences.
- **Biggest Brands**:
    - **Samsung Display**
    - **LG Display**
    - **BOE Technology Group**

7. Camera System:

- **Architecture**: Mobile camera systems use **CMOS (Complementary Metal-Oxide-Semiconductor)** sensors. Modern cameras use multiple lenses and computational photography techniques to improve image quality.
- **Design**: Mobile cameras integrate several lenses (wide-angle, telephoto, macro) and advanced AI-driven algorithms to enhance photo quality, focus speed, and night mode performance.
- **Biggest Brands**:
    - **Sony** (IMX series)
    - **Samsung** (ISOCELL series)
    - **OmniVision**

8. Battery:

- **Architecture**: Mobile devices use **Lithium-ion (Li-ion)** or **Lithium Polymer (Li-Po)** batteries. These batteries have high energy density and a relatively low self-discharge rate, making them ideal for mobile use.
- **Design**: Batteries are designed to be thin and light while offering higher capacities (3000-5000 mAh is typical). Modern batteries support **fast charging** (via USB-PD or proprietary technologies) and **wireless charging**.
- **Biggest Brands**:
    - **Samsung SDI**
    - **LG Chem**
    - **Panasonic**

9. Modem (Cellular & Connectivity Chipset):

- **Architecture**: Mobile modems are based on technologies like **4G LTE**, **5G NR**, **Wi-Fi**, and **Bluetooth**. These modems follow standards defined by organizations like **3GPP** (for cellular).
- **Design**: Modern modems support dual-SIM, carrier aggregation, and advanced MIMO (multiple input, multiple output) techniques for faster data transfer.
- **Biggest Brands**:
  - **Qualcomm** (X series modems)
  - **Samsung**
  - **MediaTek**

10. Sensors:

- **Architecture**: Sensors like accelerometers, gyroscopes, proximity sensors, and ambient light sensors are based on MEMS (Micro-Electro-Mechanical Systems) architecture.
- **Design**: These sensors are designed to be ultra-compact and power-efficient, allowing them to continuously track motion, orientation, and environmental changes with minimal battery impact.
- **Biggest Brands**:
  - **Bosch Sensortec**
  - **STMicroelectronics**
  - **Invensense** (TDK)

11. Audio (Speakers & Microphones):

- **Architecture**: Mobile speakers use **dynamic drivers** or **balanced armature drivers**, while microphones typically use MEMS-based designs for better sound pickup and noise cancellation.
- **Design**: Mobile audio systems are designed to provide high-quality sound in small form factors. Dolby Atmos or similar technologies are often integrated for enhanced audio experience.
- **Biggest Brands**:
  - **AAC Technologies**
  - **Knowles Corporation**
  - **Goertek**

Conclusion:
Analysis of How Prompt Clarity Impacts Quality, Accuracy, and Depth

1. **Depth of Responses:**
   - Naïve prompts lead to broader, more generic responses. These responses tend to cover basic concepts without diving into specific details or niche applications.
   - Basic prompts provide clarity, guiding ChatGPT to focus on more relevant details and nuances. The responses tend to be more structured and comprehensive, offering in-depth insights on the subject.

2. **Accuracy and Relevance:**
   - Naïve prompts may produce accurate information but can lack specificity. For example, asking a vague question like "Tell me about SoC" may not yield responses that align with the user's underlying intent (e.g., understanding SoC in the context of mobile hardware).
   - Basic prompts improve accuracy and relevance by narrowing the focus. By specifying architecture, design, and major brands, for example, the responses contain precise information that aligns closely with the user's needs.

3. **Contextual Understanding:**
   - Naïve prompts sometimes fail to provide enough context, leading to a more generalized response. For instance, "What's the weather like in Paris?" could return current or average weather without addressing specific timeframes.
   - Basic prompts clarify the user's intentions and provide ChatGPT with context to generate responses that are more appropriate. Asking for "current weather and forecast" ensures the response is timely and focused.

4. **Tailored Recommendations:**
   - Naïve prompts often return surface-level recommendations. For instance, a request to "Plan a trip to France" might result in suggestions of popular tourist destinations without accounting for timeframes or preferences.
   - Basic prompts that specify preferences or time constraints (e.g., "10-day trip focusing on historical sites") yield tailored and highly actionable responses.

5. **Technical Precision:**
   - Naïve prompts can lead to broad technical responses that may lack specific examples, explanations, or contrasting concepts (e.g., "Tell me about functions in Python").
   - Basic prompts often yield technically precise explanations, comparing specific types (e.g., regular vs. lambda functions) with examples that directly address the user's query.