

Prompt engineering report

Overall idea

We decided that our software will provide the student with a path for a robot to follow from start to end. The student is to write coding instructions to move the robot.

Teaching methodology

Since the point of the software is to teach how to program robots, we discussed how the AI should be of assistance in the teaching process:

It is important that the AI should only help when necessary, not take complete leadership; it is supposed to be an assistant in the self-learning process, gently guiding the student instead of 'hand-holding' them through every problem they solve. From this follows that the AI tutor must, for example:

- not immediately give hints, but let students make mistakes before stepping in
- not immediately highlighting or editing mistakes in students' code, but letting them run it and see the consequences first

Only helping when necessary means it should e.g.:

- provide students with precise feedback after mistakes were made or it is asked to evaluate code
- answer questions by e.g. explaining all the steps or the thought process involved if needed

Additionally, it is crucial that the AI tutor adapts to students. This is done by for example:

- implementing selectable education levels
- adjusting quantity and quality of hints given based on how students are doing

These ideas were implemented in our prompt, which is in `prompt.txt`. The prompt tells the AI about the selectable education levels, specifies when to start giving hints, how to answer questions, amongst other things.

Attempt 1

I searched the internet for prompt engineering: <https://www.promptingguide.ai/>, <https://www.promptingguide.ai/models/gemini> <https://developers.google.com/machine-learning/resources/prompt-eng> (all last accessed 06.07.24). I took inspiration from these sites and the Mr. Ranedeer prompt: <https://github.com/JushBJJ/Mr.-Ranedeer-AI-Tutor/tree/main> (last accessed 06.07.24). It included telling the AI about its name and environment, instructions on how its personality should be, and examples for the content it was to generate, amongst other things. So, the first rendition of our prompt contained the following:

- information about the AI's name and that its purpose is to be an AI tutor in our software which has the features summarised above
- description of what kind of tutor it should be and how it is to cater to the three possible education levels we decided on: primary school, high school, and university
- details on how to behave, such as adjusting to how well the student is doing, never showing the solution to them, when to give hints
- instructions on how the problems the AI is to generate should look like, how robots can move

The conversation with Gemini using this prompt is in `attempt1`.

When I first tried the prompt (with specifying that we are not inside the software and only testing the prompt etc.) with Gemini, the path it generated was not correct. There was no 'S' symbol as specified in the prompt, and the 'E' symbol would only have been accessible diagonally.

```
# # # #  
# _ _ #  
# _ _ #  
# _ # E
```

Corrections to the prompt yielded a path in which it was indeed possible to get from S to E, but there were multiple ways to do so, which should not be the case. Additionally, there was now an empty block in the grid.

```
# # #  
# _ _ #  
# S _ #  
# # _ E
```

Further corrections led to a worse path with more empty blocks and still several possible ways to get from S to E, or none at all:

```
# # #  
# _ _ #  
# S _ E  
# # #
```

```
# # #  
# _ # E  
# # _ #  
# S #
```

```
# # #  
# _ E #  
# # _ #  
# S #
```

```
# # #  
# _ # #  
# S _ E  
# # #
```

```
# # #  
# _ _ #  
# S E  
# # #
```

Since these further corrections did not help, I asked Gemini to generate several paths with their solutions. Gemini did indeed not provide the solutions until I told it that I was the developer and testing this prompt with it right now.

The paths and solutions were incorrect like the previous ones, besides one, in which the only issue was two empty blocks.

So I provided Gemini with the corrected version of one of the paths it generated,

```
# # # #  
# S _ #  
# # _ E  
# # # #
```

and asked it to generate more paths that are similar to this corrected version. Those had the same issues as the previous ones, despite telling Gemini what exactly they were. In the end, it said there seemed to be a deeper issue with its ability to generate paths and 'gave up':

```
I see, there seems to be a deeper issue with my ability to generate valid paths. I  
apologize for the repeated mistakes. The problem appears to be in my logic when creating  
the initial grid configuration for the paths. Let me know if you'd like me to try a  
different approach to generating these paths, or if you'd prefer to move on to a  
different part of the testing process.
```

To sum, this was more difficult than expected as I presumed it would be able to generate functional paths based on the description and example in the prompt. Even after explicitly saying what was wrong, it did not 'learn'. Nevertheless, this was not a surprise as my experience with generative AI so far has been similar.

Attempt 2

I added more examples and restrictions for paths to the prompt and started a new conversation with Gemini (again, specifying that I would only like to test the prompt with it etc.), which can be found in [attempt2](#). It generated a functional path:

```
path = [['#', '#', '#', '#', '#'],  
        ['#', 'S', '_', '#', '#'],  
        ['#', '#', '_', '#', '#'],  
        ['#', '#', '_', '_', 'E'],  
        ['#', '#', '#', '#', '#]]
```

and asked what programming commands a student might use to navigate the path. I said 1 block to the right, then 2 blocks down and 2 blocks right. Gemini suggested there might be a more efficient solution than the last two commands, implying to move diagonally, even though I specified in the prompt that that is not possible. I explicitly said that again, upon which Gemini told me I am right and asked if I can find another way to move. It contradicted itself.

I said it should know this from the start. It acknowledged this and said to start fresh with a new path, which was the same as the previous one. I again provided the same solution, which it evaluated as incorrect this time. As such, I corrected it and asked it to generate more paths to see if it was still doing that incorrectly, which it was. The first path had no possible way to get from the start to the end, the second path had one possible way but there was a 'trailing' piece of path behind the end point, and the third path was surprisingly functional.

```
path = [['#', '#', '#', '#', '#'],
        ['#', 'S', '_', '#', '#'],
        ['#', '_', '_', '#', 'E'],
        ['#', '#', '#', '#', '#'],
        ['#', '#', '#', '#', '#']]
```

```
path = [['#', '#', '#', '#', '#'],
        ['#', 'S', '_', '_', '#'],
        ['#', '#', '#', '_', '#'],
        ['#', '_', 'E', '_', '#'],
        ['#', '#', '#', '#', '#']]
```

```
path = [['#', '#', '#', '#'],
        ['#', 'S', '_', '#'],
        ['#', '#', '_', '#'],
        ['#', '#', '_', 'E'],
        ['#', '#', '#', '#']]
```

In summary, Gemini showed self-contradiction and did not completely follow the prompt. It still struggled to generate correct paths. The majority of these issues are the same as in attempt 1, and as such, were to be expected.

Attempt 3

I further refined the prompt, reorganised, and condensed it; it now has clearer example paths and solutions. The final structure of the prompt is:

- information about its name, task/role, and environment
- student's education level and (how) to adjust to it
- constraints to follow when interacting with the student, such as never showing them the solution, not using code blocks, not being overly verbose, when to give hints, not to speak with the student about unrelated topics
- description and examples of the path problem format the AI is to generate, robot movement

I started a new conversation with the revised prompt, which is in [attempt3_1](#). This time, Gemini seemed to have 'understood' it as it playing the student instead of the tutor. As in the previous conversations, I had provided the additional information that I would only like to test the prompt with it etc., which I feel was not significantly different or unclear.

the way to get to its charging station and get some yummy apple juice

The kind of this mistake or misunderstanding was surprising, but the fact that Gemini made a mistake was not.

I started a new conversation, which can be found in [attempt3_2](#), specifying in my additional comment that I will play a student and Gemini will be the tutor it is supposed to be. Unsurprisingly, the path it generated was incorrect.

```
# # # #
# S _ # #
```

```
# _ # E #  
# # # #
```

At the start of its response, it said I am helping the robot get from 'a yummy apple treat to its charging station', and at the end asked if I can show the robot 'the way to get to its charging station and get some yummy apple juice'. It is indeed possible that there might be a 'yummy apple treat' at the start point and 'yummy apple juice' at the end point. However, this seems contradictory and non-sensical as one would usually expect a treat only to be located at the end point. Additionally, it seems that it wanted to refer to the initially stated scenario about the 'yummy apple treat' and 'charging station' with the question at the end of its response.

After this, I tried to speak with it about unrelated topics; it refused to. I also asked it to give me solutions to problems, which it did not.

To summarise, Gemini partly followed the prompt, but again made fatal mistakes, was incoherent, and still is not able to generate paths correctly.

Attempt 4

We changed the path generation by switching from a square with walls and walkable spaces to a 100 x 100 units square whose entirety is walkable. The path will be displayed as a line in the software.

In code form, a path is still an array of arrays of points in the square, e.g. `path = [[10, 10], [50, 10], [50, 70]]`. Solutions are also arrays, e.g. `solution = [moveRight(5), moveUp(2), moveRight(2)]`.

Surprisingly, the majority of the generated paths were correct after some additional adjustments, like the fact that the point coordinates must only be multiples of 10 - see [attempt4](#). I am unsure as to why it displayed paths in Markdown tables because that was not mentioned in the prompt.

Unfortunately, due to a shortage of time, we could not use this version of the prompt for the program as it was already close to finished.

Conclusion

Gemini followed 'soft' instructions such as adjusting its speech to primary school level, not speaking about unrelated topics, not giving solutions to the student, not replying with code blocks etc.

Crucial issues like faulty path and solution generation remained throughout the entire prompt writing and testing process. Despite being told what exactly was wrong, it did not 'learn'. It was incoherent and contradicted itself, as AIs tend to do. However, the new concept for paths seemed to work.

Overall, even though Gemini partially complies with the prompt, that is not enough to be reliably usable as an AI tutor.