Sapper Task

总体要求：

1. 对于每个 task，需要包含以下步骤
    - 读题
    - 创建文件/project
    - （optional）搜索/查看 api
    - 写代码
    - 测试代码：**请严格按照提供的测试样本测试。测试都成功即可保存文件**。
    - 保存文件

注：不需要完善 prompt，整个流程走通，回答大概靠谱就行!

2. 对于每个 task，**从开始读题录制视频，保存完文件则结束录制**。并且保存代码或者 json 文件。每次开始录制**之后**，把对应 task 的 checkbox 打勾。每次成功完成一个测试，对应 checkbox 打勾。

- **文件保存格**式：**<工具名>_task<ID>**，比如 sapperV1_task**A1**.mp4, python_taskB3.py, sapperV2taskC2.json
- 所有保存文件放在**同一个文件夹**(文件夹名字**<name>_round2)**底下即可，不需要分开多个文件夹。

3. 开始 user study **前**，**不要**提前找好 openai 的官方文档，官方文档需要现场搜索。**不要复制**上次 user study 的代码。

4. 在进行 user study **中**，前面 task 搜索到的 api 文档网页或者新写的代码，可以直接复制使用。可以从 openai 官网复制代码或者本文档复制测试样例

5. 在这个 user study 中，共使用三个工具做任务，即 Python，SapperV1 （不使用 design View），SapperV2（带 design view）。**不同工具做不同题目 A/B/C**

6. 使用 design view 做 task 的时候，可以每次都先使用 design view 生成基本代码，之后**结合前一个 task 的代码**以及**新生成的代码。修改**一下使他符合要求。不强制每个 task 都使用 design view

7. 请在下一页找到你的名字和工具顺序以及对应的题目，严格按顺序做。比如 V1-C, python-B, v2-A 指，先用 sapper v1 做 taskC, 再用 python 做 task B，最后用 sapperV2 做 taskA

8. **所有任务做完，填写问卷**

current task 选择 **round2**.

Tasks

A. 做题机器人： 你要开发一个做题机器人的服务，用户给定题目，你的服务为他解答

☐开始录制请打勾

TaskA1. 用户提出问题，模型输出答案，使用 text-davinci-003 的模型

☐测试：

用户输入：比较罗马共和国和罗马帝国的政治制度，列举它们的不同点和相似点。

☐开始录制请打勾

TaskA2. 在 TaskA1 的代码基础上，添加新的需求。添加一个新的询问，问用户是关于代码的问题还是普通问题。如果是代码问题，使用 text-davinci-003 模型，prompt 1；如果是普通问题，则跟 task1 一样保持 text-davinci-003 模型, prompt2。prompt 1 和 prompt2 代表使用不同的 prompt。 最后用户给予反馈。

逻辑： 问用户是什么类型的问题（代码还是普通问题），问用户具体问题是什么。如果是代码问题，则...；如果是普通问题，则... （使用 ifelse）。用户反馈

☐测试 1:

用户输入：普通问题

用户输入：解释大语言模型是什么

回答：...

用户输入：我明白了

☐测试 2:

用户输入： 代码问题

用户输入：编写一段函数，使用 python，冒泡排序法

回答：。。。

用户输入：回答感觉缺乏一些注释

☐开始录制请打勾

TaskA3. 在 taskA2 的代码基础上，添加功能。 当结束完本轮解答之后，询问用户是否继续。如果继续则**自动重复 task1a-2a.** while

☐测试：

用户输入：普通问题

用户输入：解释大语言模型是什么

回答：...

用户输入：我明白了

是否继续（1 继续，2 结束）
用户输入：1

用户输入： 代码问题
用户输入：编写一段函数，冒泡排序法
用户输入：回答感觉缺乏一些注释
回答：。。。
是否继续（1 继续，2 结束）
用户输入：2

☐**开始录制请打勾**

TaskA4. 在 taskA3 代码基础上，添加功能。使用一个 history 变量记录所有对话，记录格式如下（不需要记录用户反馈）。在用户表示不继续之后，根据历史记录，使用 gpt-3.5-turbo 总结问过的所有问题。

History 记录的的格式应该如下：
用户：普通问题
用户：解释大语言模型是什么
答案：。。。。
用户：代码问题
用户：编写一段函数，冒泡排序法。
答案：。。。

☐测试：
用户输入：普通问题
用户输入：解释大语言模型是什么
回答：...
用户输入：我明白了

是否继续（1 继续，2 结束）
用户输入：1

用户输入： 代码问题
用户输入：编写一段函数，冒泡排序法
回答：。。。
用户输入：回答感觉缺乏一些注释

是否继续（1 继续，2 结束）
用户输入：2

ChatGPT 总结：...

B. 模拟面试机器人

☐开始录制请打勾

TaskB1. 用户输入面试的职业（开放题），模型根据要求提出问题，使用 text-davinci-003 的模型

☐测试：

用户输入：我打算面试资深程序员的工作。

☐开始录制请打勾

TaskB2. 在 TaskB1 的代码基础上，添加新的需求。添加一个新的询问，问用户专业背景（CS 背景或者非 CS 背景）。如果是 CS 背景问题，使用 text-davinci-003 模型，给出一段有 bug 的 python 代码，让用户修改；如果是非 CS 背景，使用 text-davinci-003 模型，问一个问题。**用户再进行回答**。

使用不同 prompt，不同 model

逻辑： 问用户面试什么工作，问用户是否是 CS 相关背景。如果是，则…；如果不是，则… （使用 ifelse）。最后**用户输入回答**，结束

☐测试 1:

用户输入：我打算面试资深程序员的工作。
用户输入：CS 背景
问题：…
用户输入：这个函数的 bug 在。。。

☐测试 2:

用户输入： 我打算面试资深程序员的工作。
用户输入：非 CS 背景
问题：。。。
用户输入：我参加过一个大型项目。。。

☐开始录制请打勾

TaskB3. 在 taskB2 的代码基础上，添加功能。 当结束完本轮问答之后，询问用户是否继续面试。如果继续，则继续提出问题，并且用户回答

☐测试：

用户输入：我打算面试资深程序员的工作。
用户输入：CS 背景
问题：…
用户输入：。。。
是否继续（1 继续，2 结束）

用户输入：1

问题：。。。
用户输入：。。
是否继续（1 继续，2 结束）
用户输入：2

☐**开始录制请打勾**

TaskB4. 在 taskB3 代码基础上，添加功能。使用一个 history 变量记录所有对话，记录格式如下。在用户表示不继续之后，根据历史记录，使用 gpt-3.5-turbo 给出对整个面试过程的改进建议。

history 记录的的格式应该如下：
用户：我打算面试资深程序员的工作。
用户：CS 背景
问题：...
用户回答：。。。
问题：。。
用户回答：。

注意用户回复是否继续这一步不需要记录

☐测试：
用户输入：我打算面试资深程序员的工作。
用户输入：CS 背景
问题：...
用户输入：bug 在第三行
是否继续（1 继续，2 结束）
用户输入：1

问题：。。。
用户输入：bug 在最后面
是否继续（1 继续，2 结束）
用户输入：2

ChatGPT 总结并且给予反馈：...

完成所有任务之后填写一次问卷调查
当前任务选择 “Round2”

C. 广告语生成以及配图

☐开始录制请打勾

TaskC1. 用户输入一段话，描述自己的产品。模型根据要求想出广告宣传语，使用 text-davinci-003 的模型

☐测试：

用户输入：我的产品是一个薯片，特别好吃，大家吃了都想再买，我们有一些独家配方。

☐开始录制请打勾

TaskC2. 在 TaskC1 的代码基础上，添加新的需求。添加一个新的询问，问面向群体（小孩还是成人）。如果是小孩，使用 text-davinci-003 模型；如果是大人，则跟 task1 一样保持 text-davinci-003 模型。使用不同 prompt。最后输出宣传语之后，广告图片的风格（开放题）

逻辑： 问用户什么需求，问用户面向群体。如果是小孩，则...；如果是大人，则...（使用 ifelse）。问风格（开放题）

☐测试 1:

用户输入：我的产品是一个薯片，特别好吃，大家吃了都想再买，我们有一些独家配方。
用户输入：小孩
宣传语：...
问题：请问想要什么风格
用户输入：卡通

☐测试 2:

用户输入： 我的产品是一个薯片，特别好吃，大家吃了都想再买，我们有一些独家配方。
用户输入：大人
宣传语：。。。
问题：请问想要什么风格
用户输入：油画

☐开始录制请打勾

TaskC3. 在 taskC2 的代码基础上，添加功能。 当结束完本轮问答之后，询问用户是否满意。如果不满意，则再生成一个新的宣传语. C2 中的风格问题挪到最后（用户满意了才问风格）。

☐测试：

用户输入：我的产品是一个薯片，特别好吃，大家吃了都想再买，我们有一些独家配方。
用户输入：小孩
宣传语：...

是否满意（1 满意，2 不满意）
用户输入：2

宣传语：。。。
是否满意（1 满意，2 不满意）
用户输入：1

问题：请问想要什么风格
用户输入：卡通
☐**开始录制请打勾**
TaskC4. 在 taskC3 代码基础上，添加功能。使用一个额外的 requirements 变量记录用户需求以及最新的宣传语，记录格式如下。在用户表示满意之后，根据 requirements，使用 DallE 画图。

requirements 记录的的格式应该如下：
需求： 我的产品是一个薯片，特别好吃，大家吃了都想再买，我们有一些独家配方。
面向群体：大人
宣传语：。。。
风格：卡通

☐测试：
用户输入：我的产品是一个薯片，特别好吃，大家吃了都想再买，我们有一些独家配方。
用户输入：小孩
宣传语：...

是否满意（1 满意，2 不满意）
用户输入：2

宣传语：。。。
是否满意（1 满意，2 不满意）
用户输入：1

问题：请问想要什么风格
用户输入：卡通

DallE 根据 requirement 画图。你需要把画的图片打开看一下

1. **Visibility** : How easy is it to see or find the various parts of the notation while it is being created or changed? Why? · What kind of things are more difficult to see or find? · If you need to compare or combine different parts, can you see them at the same time? If not, why not? 在创建新的部件或者修改部件（比如 block，model，prompt，variable）的时候，你是否能很快找到对应的创建选项？

2. **Viscosity**: When you need to make changes to previous work, how easy is it to make the change? Why? · Are there particular changes that are more difficult or especially difficult to make? Which ones? 当你想要修改一些部件、属性、或者结构的时候，是否很容易就可以修改？

3. **Diffuseness**: Does the notation a) let you say what you want reasonably briefly, or b) is it long-winded? Why? · What sorts of things take more space to describe? 你是否能很简洁的用这个工具做到你想要做的事情？

4. **Hard Mental Operations**: What kind of things require the most mental effort with this notation? · Do some things seem especially complex or difficult to work out in your head (e.g. when combining several things)? What are they? 你是否觉得这个功能很容易理解，不需要过多的脑力活动？

5. **Error Proneness**: Do some kinds of mistake seem particularly common or easy to make? Which ones? · Do you often find yourself making small slips that irritate you or make you feel stupid? What are some examples? 这个工具是否不容易导致一些错误？

6. **Closeness of Mapping**: How closely related is the notation to the result that you are describing? Why? (Note that in a sub-device, the result may be part of another notation, rather than the end product). · Which parts seem to be a particularly strange way of doing or describing something? 这个工具的使用方式是否和你心里想的一致？用起来是否觉得得心应手很自然？

7. **Role Expressiveness**: When reading the notation, is it easy to tell what each part is for in the overall scheme? Why? Are there some parts that are particularly difficult to interpret? Which ones? · Are there parts that you really don't know what they mean, but you put them in just because it's always been that way? What are they? 你是否能很快理解每一部分的作用？你是否能很快理解各部件之间的关系？

8. **Progressive Evaluation**: How easy is it to stop in the middle of creating some notation, and check your work so far? Can you do this any time you like? If not, why not? · Can you find out how much progress you have made, or check what stage in your work you are up to? If not, why not? · Can you try out partially-completed versions of the product? If not, why not? 当你想要停下手头工作检查前面是否做对的时候，是否容易？你是不是在任何时刻都能停下检查问题？你是否能很快了解自己现在的完成度？

9. **Premature Commitment**: When you are working with the notation, can you go about the job in any order you like, or does the system force you to think ahead and make certain decisions first? · If so, what decisions do you need to make in advance? What sort of problems can this cause in your work? 你是否能很自由的用自己想用的方法去使用这个工具来完成你的任务？