本文用来反思整个项目过程中发生的错误与问题。首先是项目目标：一开始的项目目标设置为：Stage1

Connect chatgpt to my game. Train the chatgpt behind different NPCCS separately to differentiate the dialogue and behavior patterns of NPCS

Finish this stage will be a successful project and OK to submit.

Stage2

Explore how much freedom chatgpt can and should gives players by adjusting chatgpt's parameters. Specific exploration direction: intelligent dialogue, quest generation, PCG, and ending orientation.

但是后来项目的目标被修改为了：

Connect chatgpt to unity. Train the chatgpt behind different NPCCS separately to differentiate the dialogue and behavior patterns of NPCS. 在unity中同时进行两个chatgpt的链接。根据gpt的回复来生成实体。简述3.5与4o的区别。

其中的原因有以下几个：

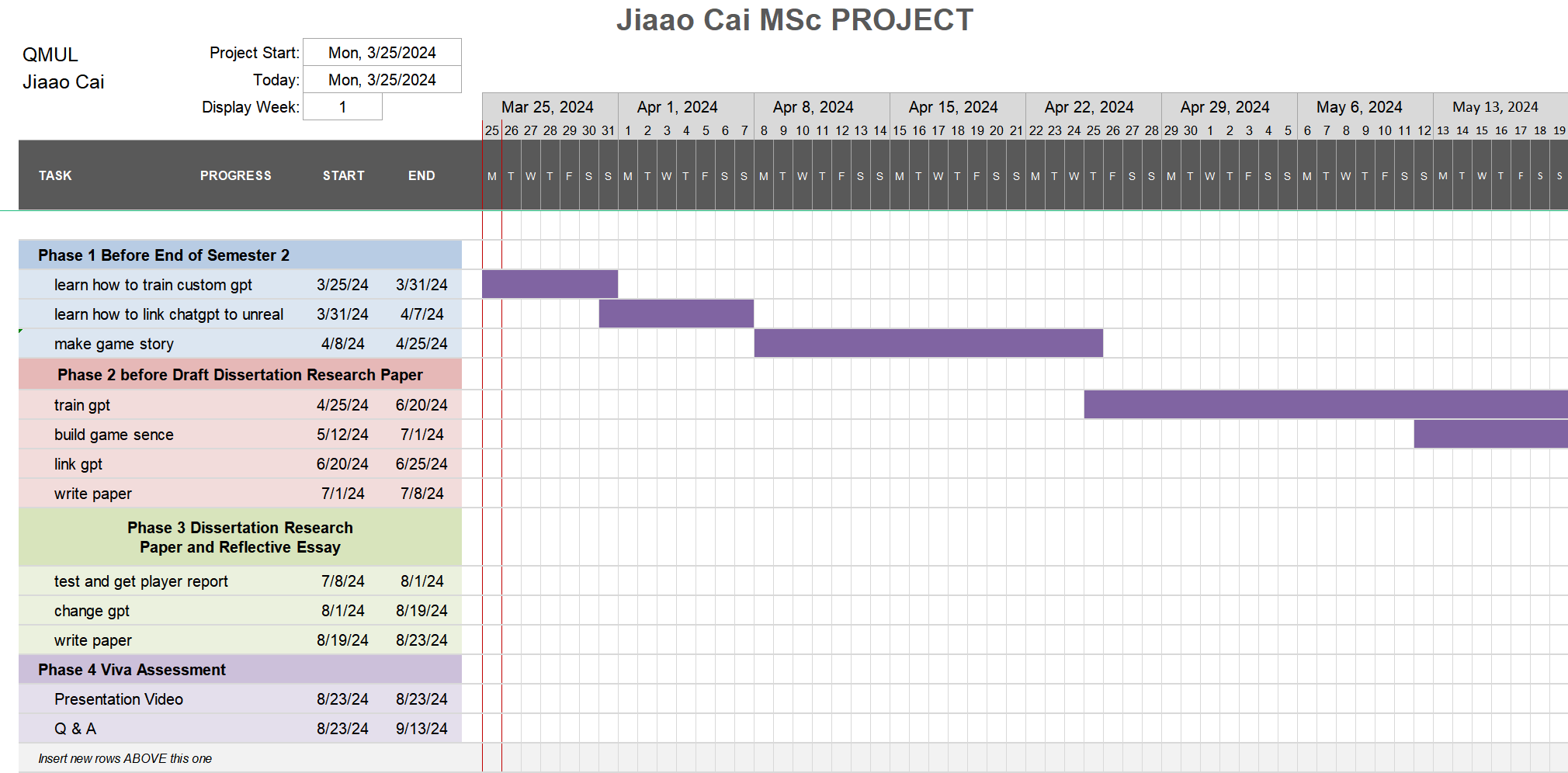
1. 因为个人原因导致有一个月的时间并没有做出任何东西，整个stage2的预定时间被挤占消失了。
2. 错误的估计了自己的能力和任务的难度。认为自己能快速的学会gpt的api应用并将其连接到unity当中。
3. 当时的stage1并没有考虑很多具体的细节的问题，而这些问题在实际操作的时候都可以扩展并尝试探索。在一定程度上，definition 文档中写的过于笼统。

如何在下一次的开发过程中改进这些问题：

1. 给自己留有更多的时间空间。谁也不知道过程中会有怎么样的意外事情发生，因此把自己的工程期限合理的延长是个很好的选择。目前我的情况来看应该按照我预计的工作时长的2倍设定截止日期。
2. 应该在写开题报告前更多的探索之前人的研究。更充分的了解项目难度，项目的核心。甚至最好在开始项目研究之前就写好一份比实际论文需要的更长的前期研究报告。这不仅可以帮助在写论文的时候更快的完成相关工作部分，同时也可以给研究一个清晰的定位。这样方便找到目前研究方向的空白部分防止做无用功。

时间规划问题：

之前的gantt chart为：



这份表格在后续的工作中再也没有更新过。Gantt chart是一个很好的控制工作进度的工具但是应该持续的更新，追踪。否则一开始做的表格只是一个表面工程实际上根本没有起到应有的作用。我认为在工作期间应该至少每一周更新一次gantt chart并且如果有任何的实质性进展或者在原有的gantt chart中没有展示出来的额外工作内容都应该加入到图中。

沟通问题。

按照老师要求的应该至少每个星期或每两个星期都联系一次老师，和老师对接项目进度并让老师提供帮助与建议。但是实际上在整个项目期间都出现了各种各样的问题导致完全没有做到按照这个频率进行线上会议：

1. 在5月之前因为课程原因，过于繁忙没有做到。
2. 在6月因为身体原因导致整个月基本上什么都没做
3. 在七月的线上会议参加的还算频繁但是也只是做到了每两周参加一次。因为之前拖欠的工作过于多导致每一周的进展并不顺利。有的时候没有什么新的进展导致完全不像和老师交谈。但是实际上应该多参加就算是没有任何进展那也应该让老师知道。老师应该能帮助我给出一些建议这样比自己一个人努力要好得多。

项目中问题

在项目开发的一开始发现了一个很好用的已经被他人集成好的unity package的时候我是十分开心的因为这样可以省去我许多的时间。但是我在使用的时候并没有很好的阅读全部的代码，了解整个package的代码逻辑导致在使用的时候问题频发。甚至一度认为这个包写的并不好需要自己再重新写一个。但是在通过学习openai提供给用户的指导文档并深度阅读了一下所引用的package之后发现其实很多的基础的功能他都提供好了。Package已经完全将使用unitywebquest这个深层的实际上和openai api交互的代码完全封装好并给出了各种各样的接口。我只需要在外部调用这些接口并做适当的扩展与修改就可以了。这样的无用功耗费了我接近2个星期的时间。完全破坏了我的工作安排。这样的事情完全可以避免，在下次使用别人的package之前应该完全阅读他们的指导文档。同时也应该仔细阅读他们的代码以做到完全理解package的运行逻辑。在这之后再进行应用或者开发会省事很多。

在关于gpt模型优化这个方面，我一开始接触的是chatgpt这个webUI提供的my gpt功能。当时在开发的过程中还算顺利。整个功能十分的方便，可以上传多个额外的知识文档，同时也有助手辅助开发正确的prompt。但是在开发结束后我想要将其导入到unity的时候出现了问题。实际上这个gpt只是一个webUI。这个webUI并不提供一个api接口来让我调用。我一开始以为只需要将代码中的网址换成我制作的这个gpt就可以了但实际上会导致程序崩溃。我进行了多方的探索，在询问gpt客服的时候客服给出的答案就是他们不会提供一个api给我就算是我只是为了学术目标。但是他们建议我使用openai 的api方法。我在api训练的时候一开始始终无法找到如何将pdf传入。但实际上是因为我过度依赖于在web上进行上传，应该使用代码在每次通讯前和prompt一起上传到gpt中。但在这个过程中走了太多的弯路。我还一度试图将我开发的两个npc的gpt封装成api然后再接入unity。但是这样首先是不合规的，其次需要再电脑上搭建一个云服务器才能运行。最重要的是那我就得重新开发package来调用我的api。这会给我带来很大的工程量。还好最好我仔细研究了代码并找到了如何从代码中上传文件的方法。

虽然整个对于gpt的开发和连接过程十分的曲折但是也让我收获了许多。首先我认为在网页上开发gpt并最终发现无法使用这个问题在开发流程上并没有什么问题。我认为是openai没有说明清楚导致的。但是同时也给了我论文中的另一个结果：chatgpt在面对非专业人士的时候有着良好的足够的功能性和互动性。如果玩家只需要开发一个助手或者文字游戏（DnD）中的NPC，完全可以不用将gpt接入到unity中，而只是使用webUI即可。而对于后面一个问题，在寻找问题解法的时候积极和官方人员沟通是一个好的主意。单纯的在互联网上搜索解决方法可能会让你走向错误的方向。

突发状况

在最后的时刻我写论文的那台电脑出现了无法开机的问题。而我没有遵循一个良好的备份习惯。我的代码有着三重备份：云端，电脑，移动硬盘。但是因为论文写的太快导致在我没有来得及备份到其他两个地方。没有办法我只好从新开始写论文。这又浪费了我2天时间。从这件事情上得出还是得至少每半天就云保存一次。

可拓展工作

因为时间原因很多很好的工作没有办法完成。在这里列举一下可以开发的方向。

1. 更好的识别gpt的输出并在游戏中生成物品。目前识别gpt的输出使用的是正则表达式判断。这个方法我认为没有问题，是最好用的文本识别方法。但是因为如果不对gpt的输出加以限制gpt输出的文本种类太多。因为时间的关系我没有办法将所有可能加入检测中。但是如果时间允许应该放弃对gpt的限制而让gpt随意在故事范围内生成。只要能将所有的可能性都考虑到就可以保证gpt输出的东西能让我们检测到并成功生成物体。
2. 可以生成更多。目前只检测生成物体。还可以拓展生成任务，生成地图，甚至生成新的NPC。这样我们甚至可以做到用少量的NPC控制整个游戏的进程与发展。