REPORT on the test project  
  
I have faced some serious and fine problems when I was building this project.   
I was trying to add up some functions which are speech to speech and also being able to give a textual chat and get an audio back which is text to speech conversion is only needed and to make it real time I used websocket for seamless communication.And also making it async and sync wrapped was necessary. When we send a text An async function (send\_text\_message) for sending text messages via WebSocket. And also A sync wrapper function (send\_text\_message\_sync) for calling it from a synchronous (blocking) context like a game UI or input handler. Managing the AI speech interruption was hard since it was causing some glitches when I try to work it out.  
  
When I try to have a good UX for the user in my controller I managed it by just pressing space to start recording and press it again to send it was working fine but later when I was handling the textual inputs again and try to write something and press space it starts recording and I managed this with ctrl+space instead for the UX conflict I have got even if it is simple to change this it was necessary since we will get this type of conflicts as a developer in the future.   
  
When adding up the textual reponses in the text box I have made an update on my previous code to handle everything without failing the previous code. The challenge was that the previous code was listening for response.text.delta events, but OpenAI's  realtime API doesn't send those. Instead, it sends response.audio\_transcript.delta events. Cause OpenAI's realtime API treats text as a byproduct of audio generation (transcription) rather than a separate text response.   
Text appears incrementally in the text box as the AI speaks, but we can display it instantly just did that on purpose to make it more realistic response.