Assignment A1: Project Landscape

Markson Chen (1009002598)
Barry Jiang (1008774097)
Kevin Liu (1008495237)
Yibin Cui (1008826512)

Part One: Interest Statements

Our team aims to build an interactive virtual character system driven by multimodal foundation models, focusing on real-time interactability that combines speech, text, and motion. As a team, we are excited to explore how conversational AI, vision understanding, and motion generation can be combined into a cohesive, low-latency system that feels engaging in virtual spaces.

Kevin I have a background in training and fine-tuning LLMs in modalities such as natural language, symbolic music and chemistry. Therefore, I want to fine-tune VLMs for virtual-idol-centered interactions and work on the memory retrieval system, with an end goal of reviving my favourite VTubers Xiangwan and Jiale.

Barry My background is in embodied AI, and I have aspired to build an AI VTuber like Neuro-sama that can play games while streaming. In this project, I hope to work on vision-language navigation and high-level actions.

Yibin I am particularly interested in multimodal AI and human-AI interaction, and I am ready to apply my experience in sequence modelling and memorization in ML toward integrating LLMs with speech and real-time interactive pipelines.

Markson As the proposer of this project, I wish to not only recreate and interact with my favourite AI streamer, Neuro-sama, but also use this as an opportunity to expand my tech stack in embodied intelligence, as my field of research primarily involves modelling embodied intelligence in computational neuroscience.

Part Two: Landscape Analysis

Item	**Description**	**Commentary**
Project: Open-LLM-VTuber	An open-source project for AI VTubers, providing an end-to-end pipeline for LLM with speech recognition, text-to-speech, voice conversion, and avatar control.	It is a practical reference for multimodal VTuber systems, but has limited support for low-latency tightly integrated perception-action loops for embodied interaction with the user in 3D.
Project: AIAvatarKit	An open-source projects for LLM-driven 3D avatars that integrated conversation with avatar controls.	It is another practical reference for us, but its avatar animation system is lacking. It generates emotion but not bodily motion.

Paper: DartControl	A autoregressive diffusion model that generates real-time full-body motion conditioned on high-level language instructions and low-dimensional goals.	Useful for our project as it provides goal-directed motion generation and has low latency. But it cannot model semantically rich scene interactions, requiring upstream perception and planning modules.
Paper: DualVLN	A dual-system VLN foundation model that integrates Qwen2.5 7B VL for high-level reasoning, generates pixel goals, and converts to low-level action execution.	Traditional SLAM-based navigation produces static maps and cannot handle dynamic scenes, but VLM can understand conversational context as well as used for navigational for our project.
Product: Grok companion & Project AVA	Grok provides voice-to-voice AI companions with 3D avatars, and is used in Razer Project AVA.	These two products feature 3D avatars standing in void, and cannot interact with the scene or the player.
Product: Whispers From the Star	It is a AI-powered interactive story game that generates conversation along with full animation with facial expressions in real time.	Its animation generation is impressive, but the avatar is not an embodied agent that mapping multimodal perception of the scene and the player to its motion.
Product: Pipecat	An open-source low-latency streaming framework for speech → text → LLM → text → speech loops, integrating various services.	We have considered building on top of it, but our system requires not just a pipeline, but an orchestration of many asynchronous modules (speech, vision, motion, etc).
Product: Qwen3-Omni	A multimodal foundation model for text, image, audio, and video with real-time streaming responses.	It is to be tested whether its "3B active parameter" trick sufficiently reduce memory and latency usage in practice.
Paper: Robotic Collaborative Tasks Through Contextual Human Motion Prediction	They proposed a deep learning architecture that jointly predicts future 3D human motion and human intention in for human–robot handover and collaborative harvesting scenarios.	A real product of an interactive companion should understand the social nuances of the player's motions (which would really differentiate the products), albeit it should be much more generalizable compared to this paper's classification-based approach.
Product: Neuro-sama	Neuro-sama is an AI-driven virtual streamer that has every feature we intend to build: conversation with personality, 3D interaction with the user, visual understanding, and many more like gameplaying.	Being the top Twitch streamer, Neuro-sama demonstrates the market size of interactive avatars. However, its models for spatial interaction and motion generation are very limited (being just released 2025/11), and it is closed-sourced.

Part Three: Project Outline

Problem statement

A solution for interactive virtual characters in 3D has yet to emerge as of today, and we believe the core engineering challenge is designing a cohesive, low-latency, multimodal architecture that can (1) support modular SOTA conversational models end-to-end, (2) maintain continuous visual grounding, (3) convert binocular visual input into reliable navigation and motion plans, (4) preserve and retrieve memory across sessions, and (5) orchestrate these asynchronous processes into a consistent, engaging, real-time agent experience.

Proposed solution

Our AI virtual avatar system will consist of 4 modules:

1. A speech-to-speech pipeline that includes STT (Speech-To-Text), LLM inferencing, TTS (Text-To-Speech), that supports most of the SOTA local/cloud-based models, and can integrate with the current LLM roleplay ecosystem and voice-changing ecosystem.
2. A vision understanding module that continuously feeds the textual description of the current frame for LLM inferencing. If a suitable model exists, it can be replaced by a video understanding module that feeds the textual description of the recent video vision for LLM inferencing.
3. A motor module that integrates a vision-language navigation model, a 3D understanding model, and a goal-directed motion generation model to achieve motor planning and control from only binocular video input
4. A memory module, implemented using existing solutions like Mem0 (or Letta).

Project Milestones: *See detailed weekly milestones in the Appendix*

Unknowns to investigate

1. Among all the possible options, how should we use OpenXR/OpenVR to bidirectionally connect Unity and VRChat with our model inputs and outputs, aiming for ease of development and a low-latency system?
2. Is it more cost-efficient to start from open-source projects (like ALXR and Monado) that might save us time yet introduce undesired complexity for our project, or should we leverage coding agents to build everything ourselves?
3. Is it actually technically feasible to integrate the vision-language navigation model, 3D understanding model, and goal-directed motion generation model into a coherent motion generation system (nobody has ever done it before)?
4. What is the best system architecture for our entire system (not overly-complicated, integratable with existing ecosystems, and has low-latency)?

Part Four: Project Press Release

Introducing *OpenNeuro*: Interactive Characters in Virtual Worlds - Powered by Artificial Neural Networks

"OpenNeuro brings lifelike AI characters to life, letting users meet, talk with, and interact with vivid virtual characters that move, react, and share space with them in real time."

For decades, virtual characters have lived behind the screens. They are mesmerizing yet can only stay as memories. OpenNeuro changes that experience. Released by the company *Project NEURIA* last week, OpenNeuro is an interactive AI-driven virtual character system built for real-time multimodal interaction. It combines conversational AI, speech processing, visual perception, and motion generation into a single, responsive pipeline.

OpenNeuro is designed for VR users, gaming companies, VTuber creators, and researchers exploring the edge of technology. Being an MIT-licensed open-source project that can be flexibly deployed across VR platforms, desktop applications, and livestream settings, OpenNeuro opens new possibilities for creativity, interaction, and experimentation inside shared digital spaces for content creators and developers.

Meanwhile, individual users can put on a VR headset or open a desktop application and meet any fictional character face-to-face. Users can ask a character to step closer, react to a gesture, or focus on an object nearby and see that intent reflected immediately through lifelike behaviour.

What The Users Are Saying

"It feels less like using an AI tool and more like being in the same space as a character that actually reacts to me."
— Early VR User

"OpenNeuro is revolutionary technology and years ahead of the field."
— Embodied AI Researcher

"This project shows how AI characters can move beyond scripted behavior and become almost… ALIVE."
— Project Stakeholder

"I was not emotionally prepared for how good this project is."
— Vedal

Appendix

Part 3: Iteration 1

We initially described OpenNeuro as an AI virtual idol focused on entertainment and live-streaming.

Target Audience: Otakus

Key Features: An AI virtual idol powered by LLM with a 3D avatar. The idol is accessible as 1. a local running AI companion (mostly for debug) 2. A player in VRChat 3. A channel on a streaming platform like Twitch. The target audience can interact with the model through text, voice and movement through 1. streaming platforms such as Bilibili, Twitch, etc. 2. VRChat 3. A locally running application.

Fictional quotes:

Vedal: “This novel, open-source, end-to-end AI virtual idol is too good! Neuro-sama cannot compete against it😭”

Part 3: Iteration 2

In this iteration, we sought to improve perceived realism, with a focus on rapid system responsiveness, facial expression control, and natural movement.

Introducing OpenNeuro - A Real-Time Embodied AI Companion for Virtual Worlds

We are excited to introduce OpenNeuro, an embodied AI companion designed to interact with users naturally in virtual environments. Unlike traditional chatbots or static AI avatars, OpenNeuro combines conversation, perception, and motion into a real-time system that allows it to exist and respond within shared virtual spaces.

OpenNeuro is built for platforms such as VRChat, desktop applications, and livestream environments, where interaction goes beyond text and voice. By responding to speech, gestures, and visual context with coordinated voice, gaze, and body motion, ### creates the experience of a virtual character that feels socially present.

The Customer Problem

As conversational AI improves, users increasingly expect AI characters to behave like social entities rather than disembodied interfaces. However, most AI companions today lack awareness of physical space and user actions. Even AI-powered virtual idols are often limited to scripted animations and delayed responses, breaking immersion in interactive environments.

For users in VR and virtual social platforms, this disconnect makes natural interaction difficult. Simple actions such as asking AI to move closer, react to a gesture, or share attention toward an object are either unsupported or feel unnatural.

Our solution

OpenNeuro addresses this gap by integrating language understanding, visual perception, and motion generation into a unified, low-latency pipeline. Instead of treating conversation and movement separately, OpenNeuro continuously reasons over what users say, what it sees, and how it should act.

Users can speak to OpenNeuro, gesture in the environment, or give spatial instructions, and see those intentions translated into expressive behavior. OpenNeuro understands visual context, plans goal-directed movement, and responds with coordinated speech and motion, allowing it to participate naturally in shared virtual space.

Target Audience

OpenNeuro is designed for VR users, VTuber creators and fans, researchers exploring embodied and multimodal AI, and developers interested in open, interactive AI systems.

Key Benefits

- Embodied interaction: Responds to visual context
- Real-time responsiveness: Act with low-latency
- Expressive presence: Emotion and body motion reinforce social realism
- Flexible deployment: Works as a VRChat avatar, desktop companion, or livestream co-host
- Open and extensible: Designed for research, experimentation, and creative use

What Users Are Saying

“It feels less like talking to a chatbot and more like interacting with a character that’s actually there with me.”
— Early VR User

“OpenNeuro *shows how language, perception, and action can come together in an embodied AI system.”*
— Embodied AI Researcher

“OpenNeuro *demonstrates how conversational AI can move beyond text and voice to become something users can interact with in shared space. That shift opens up entirely new possibilities for virtual platforms.”*
— Project Stakeholder

“This novel, open-source, end-to-end AI virtual idol is too good! Neuro-sama cannot compete against it”
— Vedal

Part 3: Iteration 3 (See Part 3)

Weekly Milestones (For Part 2)

Week 1 (Jan 18 - Jan 24):

- Implement OpenXR driver interface to send body, head, and hand motion into SteamVR and get binocular video back.
- Implement a demo of message queue-based chatbot backend and (perhaps) integrate with Pipecat. Run the full speech→text→LLM→text→speech pipeline.
- Implement API calls to send input into VRChat, like character emotion, text, and avatar height
- Find and run a small Vision Language Model for vision understanding
- Explore the feasibility of semantic mapping from vision for navigation

Week 2 (Jan 25 - Jan 31)

- Finalize the speech-to-speech backend (if not finished or harder than expected)
- Integrate the VLM as real-time visual descriptions into the pipeline
- Integrate a navigation-focused VLM for generating 2D goal coordinates
    - Generate RGB-D from binocular video, and use it to map 2D goal coordinates into 3D global coordinates (as an input into the motion generation model, DART)
- Try DART for avatar motion generation (while navigation is simultaneously being developed)
    - Implement a module that continuously plans for low-level motion for DART

Week 3 (Feb 1 to Feb 7)

- Integrate our current system with the LLM roleplay ecosystem based on character cards
- Finalize the work on navigation if not finished
- Tune the LLM to output an appropriate emotion label
- Exploration of letting the VLM to control the head (camera) movement and output the gaze location

Week 4 (Feb 8 to Feb 14)

- Implement agent memory using Mem0
- Finalize the entire motion module from low-level control to high level planning
- Explore motion stylization approaches
- Implement voice changing methods (RVC, GPT-SoVITS, and the better ones)

Week 5 (Feb 15 to Feb 21)

- Optimize the whole system for latency. Explore quantization, distillation, or other methods.
- If time allows, explore the possibility of replacing single-frame vision understanding with low-latency video understanding
- Design and code the application GUI

Week 6 (Feb 22 to Feb 28)

- Write an entire wiki site for people to use our project
- Test multiplatform support

Week 7, 8, 9

- Buffer time in case we overestimate the complexity of our project.
- If we have finished everything on time, we can explore additional features.