Risk Management Plan
for
NPC Dialogue System with LLM Integration for Unity

Conor Ryan

02/11/2025

Table of Contents

Background

The NPC Dialogue System with LLM Integration project aims to deliver a reusable Unity package which allows non-player characters (NPCs) to generate context-aware dialogue using a Large Language Model (LLM).

Traditionally dialogue systems in games use prewritten text, which affects player immersion due to repetition and requires a significant amount of development time. The goal of this project is to address these limitations by allowing NPCs to response in a dynamic and varied manner using an LLM, enhancing replayability.

The project will use existing Unity systems to facilitate communication with an LLM API for text generation and integrate ElvenLabs text-to-speech for audio playback. It will also feature an offline fallback mode to ensure usability without an internet connection.

Rationale

The main rationale for this project is to improve player immersion and reduce developer workload in narrative driven games.

Many modern games rely on dynamic storytelling, but most studios have yet to develop tools to implement AI-based dialogue systems.

In developing a reusable and modular Unity package, this project will deliver a scalable framework which other developers can adapt. It will serve as proof of concept for the use of generative AI tools in game development.

The inclusion of ElevenLabs text-to-speech will provide emotional delivery of dialogue and close the gap between static text-based dialogue and voice acting.

Business Objectives

- Improve player immersion by providing minor NPCs with dynamically generated dialogue.
- Reduce developer workload in manually scripting and managing large amounts of prewritten text.
- Deliver a reusable Unity package which can be adapted and used in various projects.

Technical Objectives

- Integrate Unity with a Large Language Model API.
- Develop a dialogue manager to handle conversation state.
- Create a structured prompt builder to generate LLM input based on game context and NPC traits.
- Integrate ElvenLabs text-to-speech for voice generation.
- Implement offline fallack functionality with prewritten text.
- Include Unity Editor tools to allow customization of dialogue templates and NPC personality.

High Level Project Schedule

The project will follow an iterative development process, where each deliverable is represented as a GitHub issue. Progress will be managed using a GitHub project board, which will track tasks in four stages, backlog, in progress, testing, and completed.

GitHub Project Board: GitHub Project Board - Deliverable Schedule

Risk Assessment

Risk ID: R1	**Classification:** Technical	**Report Date:** NA
Description: The LLM may return off-topic or out of context responses, breaking the narrative and reducing player immersion.		
Probability: 0.7	**Impact:** 8	**Risk Exposure:** 5.6
First Indicator: Generated NPC responses would begin to contradict scene or NPC traits when testing.		
Mitigation Approaches: Use structured prompt templates which include the correct context. Test prompts across different models. Limit randomness in API parameters.		
Date Started: NA	**Date to Complete:** In Progress	**Owner:** Conor Ryan
Current Status: Active		
Contingency Plan: Fallback to prewritten dialogue templates if the scene is affected.		
Trigger for Contingency Plan: Three or more invalid or inconsistent LLM responses during testing.		

Risk ID: R2	**Classification:** Operational	**Report Date:** NA
Description: Loss of internet connection or API downtime which prevents the LLM from returning dialogue during gameplay.		
Probability: 0.6	**Impact:** 7	**Risk Exposure:** 4.2
First Indicator: Failed or delayed API responses.		
Mitigation Approaches: Implement an offline fallback mode using prewritten dialogue. Add retry logic and timeout handling in the API connector. Test behavior by simulating network loss.		
Date Started: NA	**Date to Complete:** In Progress	**Owner:** Conor Ryan
Current Status: Active		
Contingency Plan: Activate an offline fallback mode, implementation planned for iteration 4.		
Trigger for Contingency Plan: Failure to receive LLM response within 5 seconds of initial API call during testing.		

Risk ID: R3	**Classification:** Performance	**Report Date:** NA
Description: Integration of ElvenLabs TTS API causes frame rate spikes or hitching, affecting gameplay performance negatively.		
Probability: 0.4	**Impact:** 7	**Risk Exposure:** 2.8
First Indicator: Noticeable hitching or frame rate spikes when playing TTS mp3 buffers during testing.		
Mitigation Approaches: Preload and stream audio in an asynchronous way. Stress test TTS system on different devices. Allow toggling of TTS via settings.		
Date Started: NA	**Date to Complete:** In Progress	**Owner:** Conor Ryan
Current Status: Active		
Contingency Plan: Disable TTS temporarily if frame rate drops below 30 FPS, or latency goes above 200ms.		
Trigger for Contingency Plan: Continuous frame drops when playing TTS mp3 buffer.		

Risk ID: R4	**Classification:** UX	**Report Date:** NA
Description: LLM generated responses may include nonsensical dialogue which affects user experience.		
Probability: 0.3	**Impact:** 9	**Risk Exposure:** 2.7
First Indicator: AI generated output includes nonsense or inappropriate dialogue during testing.		
Mitigation Approaches: Use prompt filtering and text validation before displaying dialogue to the player. Monitor outputs during testing.		
Date Started: NA	**Date to Complete:** In Progress	**Owner:** Conor Ryan
Current Status: Active		
Contingency Plan: Develop prompt filtering methods and a text validation layer. Replace invalid responses with fallback responses.		
Trigger for Contingency Plan: Detection of filtered text, or nonsensical text during testing.		

Risk ID: R5	**Classification:** Schedule	**Report Date:** NA
Description: Implementation of complex APIs causes schedule delays which impact testing and documentation timelines.		
Probability: 0.6	**Impact:** 8	**Risk Exposure:** 4.8
First Indicator: Missed internally set deadlines or unresolved issues with integration.		
Mitigation Approaches: Focus on core deliverables before attempting stretch goals. Use weekly progress tracking.		
Date Started: NA	**Date to Complete:** In Progress	**Owner:** Conor Ryan
Current Status: Active		
Contingency Plan: Defer TTS integration and possibly offline mode if time constraints increase.		
Trigger for Contingency Plan: Any missed timeline which pushes the final iteration passed due date.		

Risk ID: R6	**Classification:** Time Management	**Report Date:** NA
Description: Coursework and other deadlines may reduce available development time, leading to missed iteration goals or incomplete deliverables.		
Probability: 0.7	**Impact:** 7	**Risk Exposure:** 4.9
First Indicator: Reduced commit frequency or backlog items not being started past deadlines.		
Mitigation Approaches: Create a weekly schedule to spend a consistent number of hours on the project. Use GitHub issues to prioritize high priority deliverables first. Inform supervisor if progress is delayed.		
Date Started: NA	**Date to Complete:** In Progress	**Owner:** Conor Ryan
Current Status: Active		
Contingency Plan: Reduce scope to essential features (Core Dialogue System, LLM Integration, Offline fallback mode only) to submit on time.		
Trigger for Contingency Plan: Any backlog item unstarted past its iteration deadline.		

Risk Priority

Find below the list of identified risks and their priority based on risk exposure:

1. **R1:** Inconsistent LLM output.
2. **R6:** Coursework time constraints.
3. **R5:** Schedule Overrun.
4. **R2:** API latency or downtime.
5. **R3:** TTS performance.
6. **R4:** Nonsensical output.