COMP3000 Project Proposal

Title: Integration of a Local Large Language Model Framework for Real-Time NPC Dialogue Generation and Behavioural Simulation in Interactive Game Worlds
Student: Mehmet Isitmen
Programme: BSc (Hons) Computer Science (Artificial Intelligence)
Module: COMP3000
Supervisor: Vasilios Kelefouras
Date: 22 October 2025

1. Abstract

This proposed project aims to research, design, and implement an LLM system that can dynamically produce contextually appropriate, meaningful, and story-consistent dialogue for NPCs in a video game environment. Specifically, it aims to incorporate an optimised, game-engine-based LLM into video games, making it possible for NPCs to adapt in real time during conversation without requiring cloud API connectivity.

By integrating the inference of a quantised model, parallel processing on a GPU, and an adaptive prompt conditioning and memory system, it is planned to show that it is possible to run advanced language models on high-end consumer devices while preserving story complexity, character consistency, and low latency.

The final project outcome would be a research-based dissertation accompanied by an operational prototype in which players would be able to engage with NPCs supported by LLMs displaying memory, personality, and autonomous decision-making capabilities. It is an attempt to combine the fields of artificial intelligence research, narrative design, and parallel computing to present a feasible, private, and economical solution for the existing dialogue-based systems.

2. Background and Rationale

Historically, video games have been dependent on scripted dialogue trees with finite behavioural tree scripting in order to convincingly simulate character dialogue. As useful for video game applications as these constructs were, they were inherently limited in scope and predictability; each possible conversation could only be authored manually, with no authentic memory between characters.

Recent advancements in transformer-based language models (Vaswani et al., 2017) have radically expanded the potential for procedural storytelling. Models such as GPT-4, Llama 3, and Mistral can generate human-like dialogue and reasoning. However, most implementations are cloud-based, introducing three critical limitations:

(1) Latency and reliance on internet connectivity
(2) Ongoing operational costs
(3) Loss of creative/data sovereignty

Recent progress in quantisation (Dettmers et al., 2022) and the development of local inference infrastructures (in particular, llama.cpp, Ollama, and LM Studio) make it possible to operate multi-billion parameter models on GPUs for consumers. This technology shift raises a new research question.

Can a fine-tuned, quantised LLM be incorporated into a game engine to produce real-time contextual NPCs with acceptable latency and coherence without any internet connectivity? By

engaging with the problem, the project is at the edge of using AI in entertainment-related applications. The project combines topics in parallel computing (CUDA programming, token streaming) with using artificial intelligence in narrative creation, considering the impact of localisation-based intelligence on immersion and development.
This is not only a technical problem but a philosophical one: to create self-contained artificial minds within virtual worlds—NPCs that remember, reason, and evolve autonomously, unshackled from external servers or prewritten scripts.

3. Research Aim and Objectives

The underlying objective for the project is the design, optimisation, and integration of a local LLM dialog system for supporting emergent memory-driven NPC behaviour in a video game setup.
By doing so, the project intends to attain the following objectives:

- Examine Localisation Models' Feasibility:
  Compare open-weight models (Llama 3 8B, Mistral 7B, Phi-3 Mini) for the feasibility of quantization formats (GGUF, GPTQ) in terms of inference performance on RTX 4090 hardware.

- Design and Implement Middleware for the Integration:
  Create an interface layer connecting the LLM backend with the game engine (Unity 6 and Unreal Engine 5) using C++/C# bindings or local REST sockets. The layer would control the contextual delivery, personality conditioning, and streaming of responses from the generator.

- Establish a Persistent Memory and Context System:
  Develop an efficient database (SQLite or LiteDB) for storing summary conversation experience stories and world state metadata. Establish a Retrieval-Augmented Generation (RAG) system to allow the LLM to dynamically retrieve information for better continuity between sessions.

- Fine-Tune for Narrative Coherence:
  Use parameter efficient fine-tune (QLoRA/PEFT) on a dataset constructed from original game lore sources and autogenerated dialog scripts. Assess the impact of the fine-tune on stylistic control, factual knowledge, and narrative coherence.

- Optimise Inference Performance:
  Monitor latency, token rate, and GPU usage. Use caching, batching, and context length optimisations to attain near real-time interactive performance (< 500 ms first token latency).

- Empirical And Qualitative Assessment:
  Use metrics that range from computational criteria to human evaluations for dialogue quality in comparison between LLM-based NPCs and traditional scripted NPCs.

4. Methodology

The research method is organised into four phases with an integration of investigation and development in each phase.

Phase 1 – Literature Review and System Benchmarking
The literature review would involve an examination of existing research on transformers, quantisation approaches, and dialogue generation models. System benchmarks for shortlisted models would help in understanding their performance in terms of VRAM usage, speed, and the effects of quantisation.

Phase 2 – Architecture Design and Engine Integration
An architecture will be designed comprising three cooperating subsystems:

(1) a local inference engine for hosting the LLM;
(2) a middleware bridge for handling input/output operations;
(3) a game engine front-end for handling NPC embedding;

This phase would incorporate steps for building the communication pipeline, designing the prompt template, and implementing dialogue streaming with personality parameters.

Phase 3 – Fine-Tuning, Optimisation, and Memory Implementation
The tailored corpus with 10 000-50 000 pairs of dialogue responses would be created using synthesised responses and game scripts written for optimisation. The base model would be fine-tuned using QLoRA for adapting to linguistic style according to the tone provided in the story.

After this, the optimisation experiments would investigate token streaming, pruning the context, multi-threaded inferencing using CUDA, and memory persistence for storage of previous conversations to evolve character identity over time. Phase 4 –Evaluation, Analysis, and Reporting Data measurements concerning latency, token throughput, and memory usage will be collected for varying model settings.

Subsequently, a user test will evaluate narrative consistency, personality simulation, and realism. The results would then be statistically interpreted for comparison with theoretical projections, leading to the final dissertation on computational feasibility as well as design implications. During, it would undergo agile iteration with weekly documentation of progress, challenges, and reflections to demonstrate independent research practice.

5. Technical Resources

The project will leverage an HP Omen 17 (RTX 4090, Intel i9-13900HX, 32 GB DDR5) for local inference.

Core software components include:

- Development Environments: Unity 6 or Unreal Engine 5, Visual Studio 2022, CUDA 12, CMake.
- AI Frameworks: llama.cpp, Ollama, Transformers, PEFT, bitsandbytes.
- Languages: C++, C#, Python (for backend and fine-tuning).
- Database Systems: SQLite3 / LiteDB for memory persistence.
- Version Control: GitHub repository with continuous documentation.

This setup ensures compatibility between high-level AI experimentation and low-level engine integration, satisfying both the parallel computing and AI application dimensions of the module.

6. Expected Outcomes

By completion, the project will produce a functional, real-time NPC dialogue system powered by a local LLM. NPCs will demonstrate personality-conditioned, memory-aware conversation that adapts to player behaviour and world events.

Empirically, the project will yield detailed benchmarking data describing the computational viability of local LLM inference under different model sizes and quantisation strategies. Qualitatively, it will generate transcripts and user evaluations evidencing improved immersion and dialogue authenticity compared with scripted baselines.

Academically, the dissertation will contribute to emerging discourse on embedded artificial intelligence, providing one of the first systematic case studies on local LLM deployment in interactive entertainment. Practically, it will establish a reusable framework that can be extended into future titles, laying the foundation for self-contained, intelligent characters in commercial games.

7. Evaluation Criteria and Success Indicators

The project will be assessed according to four dimensions corresponding to Plymouth's COMP3000 rubric:

1. Technical Mastery: Successful integration of the local LLM into the game engine, demonstrating low latency, stable inference, and efficient GPU utilisation.
2. Research Rigour: Depth of theoretical grounding, quality of literature synthesis, and evidence of analytical reflection.
3. Innovation and Originality: Novelty of concept and execution; clear articulation of contribution to current AI/game-dev knowledge.
4. Documentation and Presentation: Professional-quality report, code documentation, diagrams, and viva demonstration evidencing mastery.

Success will be defined not only by a working system but by demonstrable understanding—quantitative metrics, critical evaluation, and theoretical interpretation of why and how the solution performs as it does.

8. Anticipated Challenges and Mitigation Strategies

Running a multi-billion-parameter model locally presents several inherent challenges.
Latency and VRAM constraints may impede real-time responsiveness. These risks will be mitigated through aggressive quantisation, prompt-length optimisation, and incremental streaming of tokens to simulate continuous speech.

Fine-tuning instability, including overfitting and hallucination, will be countered using low-rank adaptation and mixed data sources balancing creativity with factual grounding.
Integration complexity between Python inference back-end and C++/C# front-end will be managed via modular design, robust exception handling, and staged testing.

Finally, evaluation subjectivity will be addressed through mixed-method assessment combining measurable performance metrics with structured user feedback instruments.

9. Project Timeline

The work will proceed across twenty weeks as follows (~November-March/April):

- **Weeks 1–4:** Literature review, environment configuration, baseline model benchmarking.
- **Weeks 5–9:** Middleware design, engine integration, initial prototype.
- **Weeks 10–14:** Fine-tuning, optimisation, memory implementation, internal testing.
- **Weeks 15–20:** Empirical evaluation, user study, final analysis, dissertation compilation, and presentation preparation.
-

Weekly logs and GitHub commits will document progress, ensuring transparency and traceability for assessment.

10. Conclusion

This project proposes an ambitious but feasible exploration of how localised artificial intelligence can redefine interactivity in digital environments. By embedding a fine-tuned, quantised LLM within a live game engine, it aims to demonstrate that conversational intelligence, once accessible only via cloud APIs, can now be executed autonomously, efficiently, and creatively on personal hardware.

The implications extend beyond the immediate prototype. A successful outcome would validate a new paradigm of on-device narrative AI, empowering future developers to craft worlds populated not by scripted automata but by self-contained artificial minds capable of memory, personality, and growth.

Through rigorous research, meticulous engineering, and reflective analysis, this work intends to stand as a pioneering demonstration of AI-driven storytelling through local computation, a synthesis of technological precision and artistic aspiration that fully embodies the objectives of the COMP3000 module.