# System Requirements Specifications

# Dynamic & Adaptive NPC Personalities in a 2D Game through Integrated AI/ML

CPRO 2901 F.

Yaksh Sharma, Aaryan Garg, Harmanpreet Singh and Ramandeep Kaur

## 1. Functional Requirements

The project must provide these following features:

**a) Player interaction**

- The player/user must be able to move around and initiate interaction with the NPC.

**b) NPC Personality**

- Each NPC will have a genetic/basic personality trait (can also be defined as fixed archetypes such as friendly, aggressive etc.).
- NPCs may change behaviour depending on the player interactions.

**c) Persistence of NPC state**

- All NPC traits and selective player-NPC interactions must be stored in the database.
- NPC states must be retrieved correctly on subsequent play sessions.

**d) Backend front-end integration**

- The unity must communicate with the backend using RESTful API calls.
- The backend will update NPCs' state, process inputs and return the results to the front-end client.

**e) AI/ML**

- The backend must include ML based models that update the NPC traits and dialogue choices.
- The system should allow integration of trained models (ML.NET/ONNX).

---

## 2. Non-functional requirements

a) **Performance**

- Game interactions (movement, dialogue initiation) must respond within 200ms for smooth gameplay.
- Backend API calls should complete within 500ms under normal conditions.

b) **Reliability**

- NPC states must not be lost between sessions.
- System must tolerate network interruptions by retrying or caching data until reconnection.

c) **Scalability**

- Designed to scale from single-player prototype to multiplayer in future versions.

- Backend architecture must support modular addition of new AI/ML models.

d) **Usability**

- Interface must be intuitive, with simple key/mouse interactions for movement and dialogue.
- Players should not require external instructions to interact with NPCs.

e) **Maintainability**

- Source code will be version-controlled via GitHub.
- Documentation must be provided for API endpoints, database schema, and personality models.

f) **Portability**

- The Unity game must run on Windows PC.
- Backend must run on Windows/Linux environments with .NET Core support.

## 3. Hardware requirements

**Minimum (Prototype Testing):**

- Processor: Intel i5 / AMD equivalent
- RAM: 8 GB
- GPU: Integrated graphics (for 2D rendering)
- Storage: 1GB free
- OS: Windows 10 or Linux

**Recommended (Development & Future Expansion):**

- Processor: Intel i7 / AMD Ryzen 7
- RAM: 16 GB
- GPU: Dedicated GPU (e.g., NVIDIA GTX 1060 or higher)
- Storage: 2 GB free
- OS: Windows 10/11

## 4. Software requirements

- **Frontend (Game Engine):** Unity (C#).
- **Backend Framework:** ASP.NET Core Web API (C#).
- **Database:** PostgreSQL.
- **AI/ML Libraries:** ML.NET, ONNX Runtime, (optionally Scikit-learn for training, exporting to ONNX).
- **Version Control:** Git / GitHub, unity version control.
- **Development Tools:** Visual Studio, Unity Hub, unity game development.

**NOTE**: As this project involves multiple integrated components (Unity, AI/ML backend, database systems, and frontend–backend communication), there may be additional software tools, libraries, or frameworks required during development. Any such requirements will be justified as they arise, ensuring transparency and alignment with project objectives.