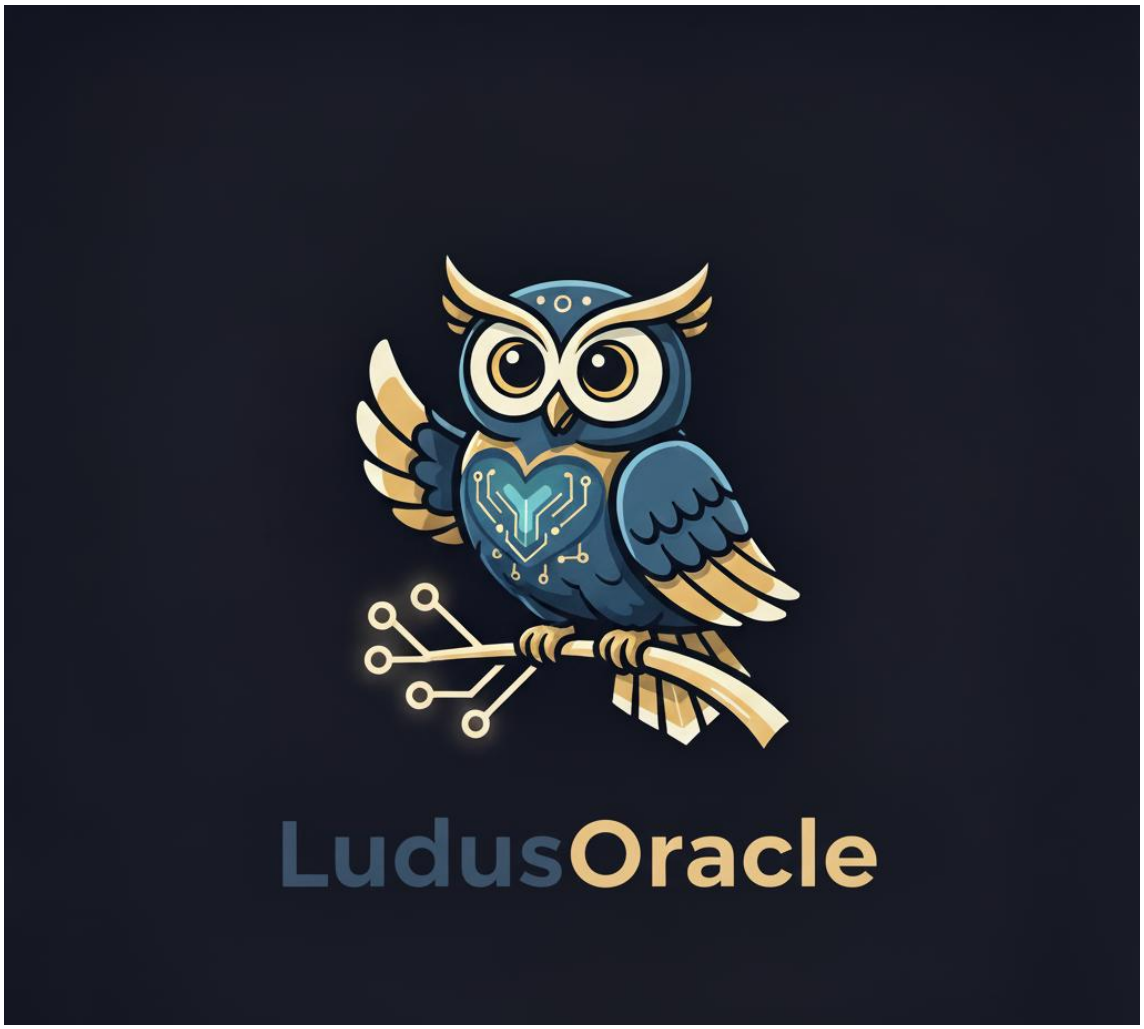
LudusOracle

-

Vision Document



Authored by: Vinícius Ribeiro Peixoto

Brasília

-

2025

SUMARY

1. INTRODUCTION:

This project is about creating an AI-powered Game Assistant, named *'LudusOracle',* designed to help and guide players with anything they need inside a specific game.

The purpose of the project is to gather valuable, game-related information in one place and present it in a clear, well-structured way, so users can easily access the guidance or support they need for any task connected to the game in question.

The assistant is intended for all types of players, from complete beginners to highly experienced experts, offering personalized help based on each user's level, goals, and context.

Players usually need to search through scattered guides, outdated tutorials, and multiple sources to find reliable information. This project aims to unify all that knowledge into a single intelligent assistant that can respond instantly and adapt to the user's needs.

2. SCOPE:

The Game Assistant AI aims to provide intelligent, dynamic, and context-aware support to players across a wide variety of games. The system will deliver guidance, explanations, strategies, and relevant information through natural language, image understanding, and structured knowledge sources.

Its scope includes the ability to analyze user requests, understand the game context, and provide accurate, actionable insights. This may include gameplay tips, item explanations, build suggestions, step-by-step assistance, or direct analysis of screenshots. Furthermore, UI elements will be an important part of the project. The goal is to build an interactive and thematic interface that the user can adjust to their needs. Widgets are

also intended to be implemented, allowing users to quickly access the information they need.

The assistant will be designed to expand over time, supporting multiple games, various knowledge bases, and new interaction modalities. It will also be capable of integrating external data sources, evolving prompts, advanced reasoning capabilities, and future UI layers such as web and mobile interfaces.

Overall, the scope of the project encompasses the creation of a flexible, extensible AI-powered assistant capable of delivering high-quality support to players of all skill levels, from beginners to experts, regardless of the game being played.

3. REQUIREMENTS AND FEATURES:

3.1. Functional Requirements:

- **FR001**: The system must be able to receive natural-language requests from the user.

- **FR002**: The system must interpret and understand user intent based on the game context.

- **FR003**: The system must generate helpful responses related to gameplay, strategies, and mechanics.

- **FR004**: The system must support requests involving explanations, tips, builds, item details, or step-by-step guidance.

- **FR005**: The system must be able to analyze images or screenshots related to the game.

- **FR006**: The system must adapt its responses to the player's skill level when possible.

- **FR007**: The system must allow expansion to support multiple games.

- **FR008**: The system must provide structured information when necessary (lists, steps, tables, etc.).

- **FR009**: The system must integrate with external data sources or knowledge bases in future versions.

- **FR010**: The system must present information through a thematic, interactive UI.

- **FR011**: The system must support a widget exhibition feature.

3.2. Non-Functional Requirements:

- **NFR001**: The system must be powered by a Large Language Model (LLM).

- **NFR002**: The system must be designed in modular and extensible architecture.

- **NFR003**: The system must support future integration with RAG and vector databases.

- **NFR004**: The system must be able to handle ambiguous or incomplete user inputs gracefully.

- **NFR005**: The system must ensure fast response time under normal operating conditions.

- **NFR006**: The system must be built with scalability in mind to support new games and features.

- **NFR007**: The UI must be responsive and adaptable across devices.

- **NFR008**: The UI must be thematic based on the game that is in question.

- **NFR009**: The system must maintain understandable logs for debugging and improvement.

- **NFR010**: The system must follow data privacy and safety policies appropriate to the platform.

- **NFR011**: The behavior and style of the assistant must remain consistent across interactions.

4. Target Audience:

The system targets every type of player, from complete beginners to experienced experts. This is because the Game Assistant will be capable of answering questions about beginner tutorials, speedrun tactics, ways to accomplish specific goals, and much more. Since each player's needs can be very different, the system should adapt itself accordingly and adjust how it approaches each user.

5. Use Cases:

- **UC01 - Searching for Game Information:** The user searches for structured information such as item stats, NPC locations, crafting materials, or quest requirements. The system retrieves the relevant data from available sources and presents the information in a clear and well-organized format.

- **UC02 - Receiving Recommendations or Optimizations:** The user asks the assistant for recommendations such as the best build, best route, better gear, or optimal strategies based on their current situation or goals. The system analyzes the request and suggests an optimized approach.

- **UC03 - Explaining Game Mechanics:** The user asks the system to explain a mechanic, feature, or concept from the game. The assistant delivers a clear explanation tailored to the user's experience level.

- **UC04 - Providing Step-by-Step Guides for Tasks:** The user asks how to complete a task or achieve a specific goal. The system generates a step-by-step guide, adapted to the user's skill level and game progress.

- **UC05 - Uploading a Screenshot for Analysis:** The user uploads a screenshot (PNG, JPG, or PDF) up to 15MB. The assistant analyzes the image and returns appropriate feedback, insights, or explanations based on the visual content.

- **UC06 - Switching to a Different Game Context:** The user wants to discuss a different game and switches the chat to another supported title by activating the appropriate feature. The system updates its context accordingly, and the UI adjusts to the new game's theme and style.

6. System Overview:

The system is organized into a set of high-level components designed to work together to support text-based queries, screenshot analysis, structured information retrieval, and multi-game interaction. At its core, the system relies on a Large Language Model (LLM), with optional layers that extend its capabilities through retrieval, image processing, and user interface adaptation.

6.1. High-Level Architecture:

The architecture is centered around a core LLM module responsible for understanding user requests and generating natural-language responses. Surrounding this core are supporting layers that enhance functionality: a retrieval layer for accessing structured game data, an image analysis pipeline for processing screenshots, a context manager for handling multiple games, and a user interface layer that adapts visually and functionally depending on the active game. Logging and monitoring components operate across all layers to support system evaluation and continuous improvement.

6.2. Main Components:

- **LLM Core** - Interprets user queries, generates responses, and adapts explanations to the player's skill level.

- **Retrieval Layer** - Provides structured game information through vector databases, APIs, or curated knowledge sources.

- **Image Analysis Pipeline** - Processes uploaded screenshots to extract relevant visual information and provide appropriate insights.

- **Context Manager** - Maintains the active game context and allows users to switch between supported games dynamically.

- **User Interface Layer** - Delivers an interactive, thematic interface that adjusts to the current game and presents information clearly.

- **Logging & Monitoring** - Records system events, interactions, and performance metrics to support refinement and quality assessment.

6.3. Data Flow Overview:

The user interacts with the system through text or image input. Text queries are processed directly by the LLLM Core, while image inputs pass through the Image Analysis Pipeline before being interpreted. When relevant, the Retrieval Layer enriches the response with structured knowledge from external or internal sources. The processed result is formatted and returned to the user through the interface, which adapts according to the active game context.

6.4. Interaction Modes:

- Text-based assistance for explanations, tips, strategies, and structured information.
- Image-based analysis for interpreting screenshots and visual game elements.
- Context switching to change the active game and adjust the system accordingly.
- Structured data queries for retrieving specific information such as items, statistics, locations, or crafting requirements.

7. Constraints and Assumptions:

7.1. Constraints:

- **C001 - LLM-Based Architecture:**
  The system must rely on a Large Language Model as the core mechanism for interpretation and response generation.

- **C002 - Multimodal Support:**
  The system must support text-based inputs and image-based inputs such as screenshots (PNG, JPG, PDF).

- **C003 - Web Service Operation:**
  The system must run as an online service and require an internet connection for all major functionalities.

- **C004 - Expandability for Multiple Games:**
  The system must be designed to support multiple games and allow adding new titles without major architectural changes.

- **C005 - Thematic UI Adaptation:**
  The user interface must be adaptable to the game context, adjusting visuals and layout based on the selected game.

- **C006 - Logging Requirement:**
  System interactions and errors must be logged to support monitoring, debugging, and future improvements.

- **C007 - Data Source Compatibility:**
  The system must be capable of integrating structured external game data sources (APIs, wikis, databases) when available.

- **C008 - Privacy & Safety:**
  The system must follow standard privacy and safety guidelines, ensuring that user data is not exposed or misused.

- **C009 - Screenshot Limit:**
  The system must enforce a maximum file size limit for uploaded images to preserve performance and stability.

7.2. Assumptions:

- **A001 - Internet Access:**
  Users are assumed to have stable internet access to use the system.

- **A002 - Clear Screenshots:**
  Uploaded screenshots are assumed to be clear enough for analysis and not excessively compressed or visually corrupted.

- **A003 - Structured Game Data Availability:**
  It is assumed that game information (items, stats, locations, etc.) can be sourced or created in structured formats.

- **A004 - Supported Games Have Consistent Mechanics:**
  The system assumes that each supported game has stable mechanics and rules that can be documented and retrieved.

- **A005 - Users Provide Sufficient Context:**
  The system assumes that users will provide enough context in their prompts to allow the assistant to generate helpful responses.

- **A006 - Device Compatibility:**
  Users' devices (desktop or mobile) are assumed to support standard web interactions and image uploads.

- **A007 - Reasonable Request Complexity:**
  The system assumes that users will not request actions that require real-time gameplay access, control, or live in-game monitoring.

8. **Future Vision and Expansion:**

The system is designed to grow far beyond its initial capabilities, evolving into a comprehensive multi-game assistant with advanced features, multimodal understanding, and highly specialized support for each individual game.

In future versions, the platform will expand to include support for multiple titles, each with its own tailored features, data structures, and interaction models. The assistant should be capable of adapting its behavior depending on the game, providing unique tools and interfaces such as visual farm planners, build optimizers, crafting assistants, exploration routes, or NPC interaction guides.

As the system evolves, additional functionalities may include a fully integrated retrieval pipeline for structured knowledge, deeper image analysis, advanced UI customization, cross-game comparison tools, and a modular architecture that allows entire game modules to be added, updated, or removed independently. Long-term goals also include the development of an interactive experience that combines text, images, and structured data in a unified interface, delivering a seamless and intelligent gameplay companion.

Ultimately, the project aims to become a flexible and expandable AI ecosystem capable of supporting a wide range of games, each with dedicated features that enhance the player's experience and provide insights tailored to the uniqueness of each title.