

AI-Driven
Destiny 2 Player Coach
CS39440 Project Outline Document
G4RR Computer Science and AI

Declan R A Wadsworth | 200061043 | drw8

Version 1.0 (Release) | 09/02/2025

Supervisor Chuan Lu

Project Description

This project aims to produce a piece of software that assists beginner and intermediary players of the MMORPG game Destiny 2 by leveraging emerging AI chatbot services to optimize their character builds and improve their core game skills.

Destiny 2 is a sci-fi fantasy, MMORPG, first-person shooter game available on Xbox, PlayStation and PC. The game consists of a wide range of activities that can be unique in their content, posing specific challenges to a player to overcome. There exists long-form activities with unique mechanics to progress which require further considerations towards a players build configuration. Each weapon, armour piece, subclass fragment as well as the modifications and perks that come with each, have a large impact on a players overall experience and gives weight to their set of skills within the game and its activities. These aspects have continually made it hard for the newer to intermediary level players to come to grips with, with their being very little digestible guidance from the game itself.

The deliverable software will read a players “vault” and “inventory” to suggest builds for specific activities; maximising survivability and activity modifier coherence, giving a player the best chance of passing an activity. The software will also suggest new weapons and build configurations to become familiar and confident with, increasing a players overall experience with the configuration systems that exist within the game and expanding their skillset. The software should also provide a knowledge base to which the player can refer to, this being a list of all keywords in the game that a chatbot service could possibly explain in a better way than the game itself. The software could also provide live suggestions based on a players current activity. The form of these suggestions should be concise and easily readable, whilst also giving brief detail as to why a suggestion gives a player an advantage. Finally, the software could also suggest new weapon types, build types or activities based off of a players activity history.

The target platform for this software requires further investigation, as to reach as many players as possible in the most accessible way. A web-based service (server-client) should be considered due to similar tools developed by the Destiny 2 community being similarly based, but the possibility of an android based application and the accessibility of smartphone applications should be an aspect in any investigation. Both performance and API limitations such as request limits will also affect any decision as to target platform.

The software will make use of a chatbot service such as “Google Gemini”[1], but alternate implementations (especially in regards to receiving responses in digestible formats) exist, such as through services like “Langchain”[2]. “The Bungie API”[3] will also obviously be used to retrieve player data (vault/inventory contents, activity history, live data), which utilises a REST based approach to retrieving game data. An alternate implementation exists here too through “aiobungie”[4].

The development language and build-system utilised will be given weight to by possible API implementation configurations, as well as determined through investigations into its target platform. Due to the complexity of these interactions, a phase-based waterfall approach is best to orchestrate an optimal and well equipped design.

Proposed Tasks

Over the course of project planning, design and development, several different tasks and investigations will need to take place in order to facilitate a professional, relevant and efficient program. These tasks include, but may not be limited to:

- **Investigation of Chatbots:** Considerations need to be made regarding the accuracy of a proposed chatbot. Although Gemini is a likely contender due to its pricing and easy implementation, a full investigation involving alternative bots must be made to ensure the best quality of content for the proposed software.
- **Investigation of target platform:** The deliverable software needs to be as accessible as possible to players, meaning considerations as to the deployable software's platform need to be made. This could be through a server running the back-end logic (ran on a machine outside the university firewall) and accessed through a website (most likely hosted on public_html). Responsive design can be utilised to be accessible to both desktop and mobile devices. However, alternate targets should be considered such as running as a native application on mobile or desktop.
- **Investigation of API technologies and implementations:** Several different implementations exist for each technology that will be utilized for the project. The best configurations of these for the project must be considered and gives weight to the target platform of the deployable software.
- **Planning:** This task takes place as part of the development methodology (waterfall)
- **Development of a solution:** This can be broken down into individual steps:
 - **Development of Game Data Wrapper:** A wrapper class of some sort will need to be developed in order to provide the chatbot with the necessary player information and in the correct format. This will also involve providing the API with the appropriate player credentials.
 - **Integration of chatbot:** A stage concerning connecting the chatbot or langchain with the players game data in each case of proposed functionality, creating endpoints for integration with specific UI elements.
 - **Development of Front-End UI:** The last stage where an appropriately designed UI (design technology dependent on target platform and development environment) is developed and connected with the solution back-end.
- **Project Meetings and Diary:** Each week, a meeting is conducted with the project supervisor to provide updates on progress and discuss any issues that have presented themselves. A directory will be kept of what was discussed at these meetings in the form of individual text documents. In the same format, personal diaries will be kept to detail any work made on the project by myself as a historical reference.
- **Demonstrations:** Two demonstrations are required for this project. Ensuring the project is at relevant stages in its development for these will be a consideration throughout the project lifecycle. The mid-term demonstration aims to demonstrate the capabilities of the main features discussed, focusing on back-end logic. The final demo will include the front-end and back-end working in tandem, representing a deployable piece of software.

Project Deliverables

The project requires multiple pieces of work to be produced, including different aspects of documentation, as well as the software itself. These include and are limited to:

- **Deployable Application and Source Code:** The application itself having been built into an executable for the target device/platform as well as an accessible git repository to the application source code.
- **Final Report:** A document that details the full project lifecycle, including all third-party libraries, frameworks and tools utilized.
- **Final Demonstration:** A demonstration meeting to showcase the applications full capabilities. It must be considered throughout development as an end-point.
- **Mid-Term Demonstration:** Another demonstration occurring half-way through the project lifecycle, focusing on back-end logic and implementation. Another consideration when beginning and pursuing development.

Annotated Bibliography

[1] Google Gemini, “Gemini API Docs”, Last Update: 2025. [Online]. Available: <https://ai.google.dev/gemini-api/docs>. [Accessed 28/01/2025].

Google Gemini is an AI reasoning model similar to other services such as ChatGPT. The API offers developers the ability to integrate AI reasoning into their applications.

[2] Langchain, “Langchain Homepage”, Last Update: 2025. [Online]. Available: <https://www.langchain.com/>. [Accessed 08/02/2025].

Langchain is an AI technology that allows for structured interactions with AI reasoning models, allowing for less strenuous output parsing and formatting. It is used to a large extent in professional environments.

[3] Bungie API, “Bungie API Docs”, Last Update: 2024. [Online]. Available: <https://bungie-net.github.io/>. [Accessed 31/01/2025].

The Bungie API is used for accessing everything related to Destiny 2 including player information, activities, weapons and confidential player data upon credentials. It is necessary for the software, allowing for tailored responses from the AI for the player.

[4] AIOBungie, “AIOBungie GitHub”, Last Update: 2025. [Online]. Available: <https://github.com/nxtlo/aiobungie>. [Accessed 01/02/2025].

AIOBungie offers a python wrapper by “nxtlo” to the official REST accessible API by Bungie. It is an alternative and easier implementation that would allow for more cohesion between other API technologies used, as most have an API for python, giving weight to what development environment is best for the project.