

# Project Report: Better NPC Combat Responses

Arman Rezaei (9723034)

May 2022

## Abstract

Games are perhaps one of the first fields in which AI was used. NPCs—Non-Player Characters—were one of the first types of “agent” in AI. Through the years, AAA (Triple-A) gaming companies have developed different algorithms and methods for their NPCs so that they will feel more *alive*—most of which are proprietary[1]. This paper aims to provide methods of equipping NPCs with the right artificial intelligence so that they will feel more proficient and competent in real-time RPG<sup>1</sup> combat systems.

## High-Level Overview

The project is structured as a Python package. On the highest level, in order to run a simulation, first you need to run a `pip install` command so that the necessary packages are installed. In your relevant operating system shell, type the following commands—make sure to run your relevant shell command for accessing the virtual environment.

```
> python -m venv venv # Create virtual environment
> source ./venv/bin/activate # Unix-like systems
> & ./venv/Scripts/Activate.ps1 # PowerShell on Windows
> python -m pip install -r requirements.txt
```

Now you may run the simulation simply by typing

```
> python main.py
```

The game will keep running until it is stopped via a keyboard interrupt (`Ctrl+C` in most operating systems).

---

<sup>1</sup>Role Playing Game

## Modules

The file `params.json` provides runtime parameters for the game which can be used to tweak the behavior of the game, e.g. changing the relevant scores for attacking or defending, thereby providing the programmer the ability to observe different behaviors of its agent given different circumstances.

Diving a little deeper into the package folder `ai_game`, there are a couple of files which organize the projects modules. The most high-level module is `game.py` which provides the `run` function used for running the game. Other packages provide different classes used in simulating the game—the most important of which being `Agent` and `KnowledgeBase`.

## Game Mechanics

The agent and player will take turns playing the game, but that is rarely the case in a real-time combat. As such, I've tweaked the “turn-based” behavior of the game so that turns are selected randomly. This way, an agent might be able to attack the player multiple times in a row, if luck is on its side.

At each turn, the *actor* (either the player or the agent, depending on the turn) will have a chance to either *attack*, *dodge* or *block*. Attacking is a highly valued attribute and the agent will be rewarded some amount of points just for this; additional points will be awarded if it manages to deal damage. Receiving damage is a negative point for the agent and will take away some of its hard-earned scores, therefore, it should try to avoid it as much as possible. After each action is taken by the actor, the opponent will have a chance to respond, regardless of whether it is its turn or not.

Upon each action taken and results specified, the new information is added to the agent's knowledge base, allowing it to take more informed decisions in the future

## Algorithm

The algorithm used for the agent's decision-making is hill-climbing search. It is an efficient searching algorithm which provides the agent with a “good-enough” result in a relatively short time. Other algorithms have been postponed due to the project's tight deadline.

## Quirks and Pitfalls

This project is still far, far away from what I've had initially envisioned for it. Due to the project's tight deadline and its massive scope, I was able to only partially implement the planned features and algorithms. As an example, I was initially planning to implement a fully autonomous playground where the agent plays against itself, opening the door for many interesting behaviors. I also wanted to investigate a number of different algorithms, including the ones based on logic, and measure which of them would provide a better result.

The precise values for scores awarded to an agent to allow a more fair game are also in much need for tweaking. Designing a game is an elegant task, one which cannot be done in a matter of few days.

## References

- [1] Steven Woodcock. *Game AI: The State of the Industry*. Nov. 20, 1998. URL: <https://www.gamedeveloper.com/design/game-ai-the-state-of-the-industry>.