

Introduction

This project is a modular, generative AI system that allows designers to generate AI NPCs with random or preset personalities that will carry out their given tasks based on their predispositions according to their personality types.

Included Features

Designers can create their own preset personality types that will generate AI behaviour based on their parameters while still maintaining a degree of randomness. Designers can also choose to randomly generate AI behaviour and leave the action weightings entirely up to the algorithm.

Literature Review

This project is based largely off a technique called Dynamic Scripting. "Dynamic scripting maintains several rulebases, one for each agent class in the game. Every time a new instance of an agent is generated, the rulebases are used to create a new script that controls the agent's behaviour. The rules that comprise a script controlling a particular agent are extracted from the rulebase associated with the agent's class. The probability that a rule is selected for a script is influenced by a weight value that is attached to each rule." [2].

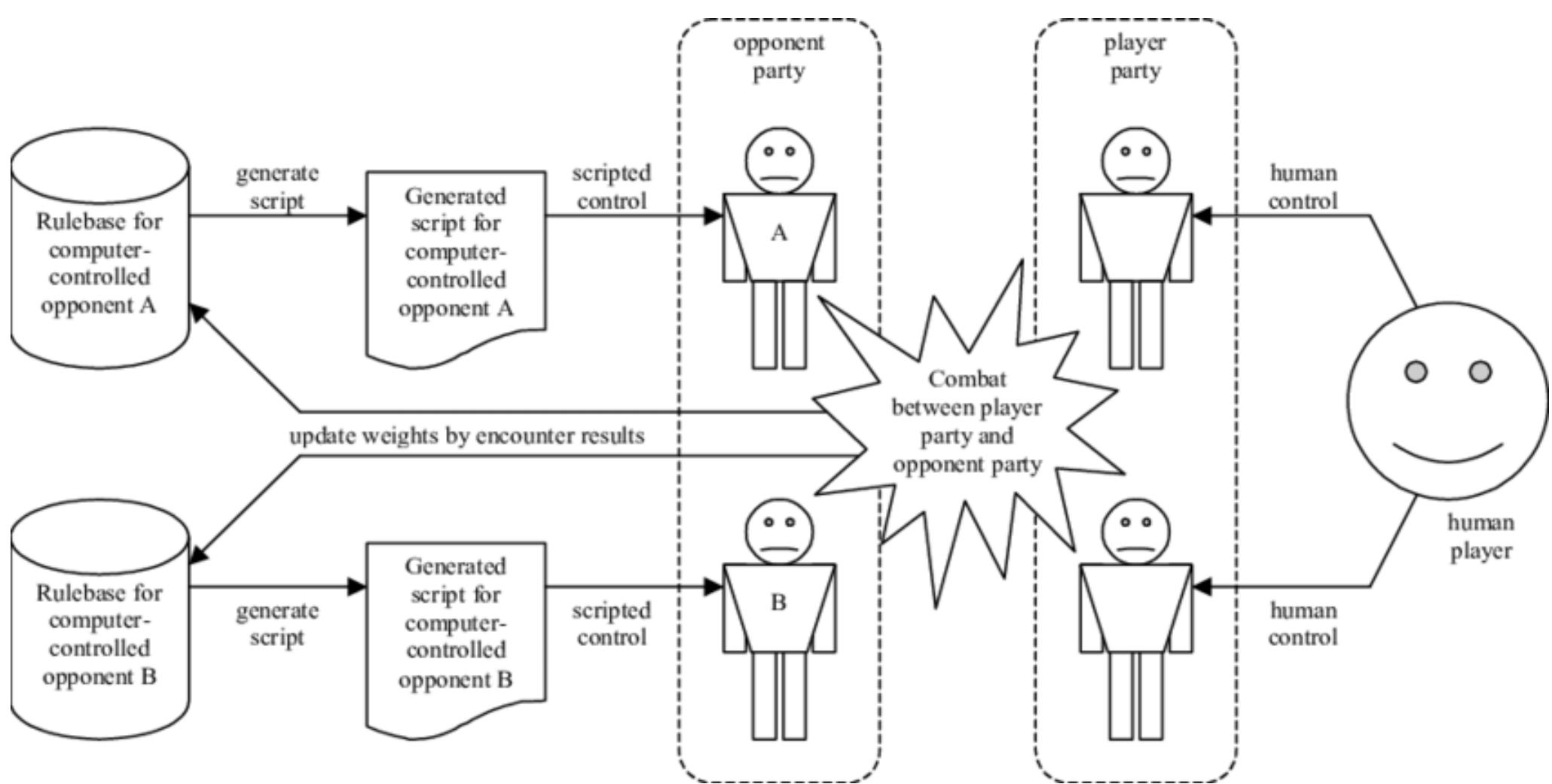


Figure 1. An example of dynamic scripting involving combat. [3]

While dynamic scripting is primarily used as a form of reinforcement learning to teach AI to achieve certain goals, this computer artefact has modified it to instead focus on emulating the flaws of human personalities. "Fun can be maximized when mistakes made by computer opponents are intentional. By finely tuning opponents' mistakes to be intentional but plausible, programmers can prevent computer opponents from looking unintelligent, while ensuring that the player is still capable of winning." [1]

Research Gaps

The largest gap in research for this project is alternative methods. While the use of dynamic scripting that the algorithm is based around works quite well, there is a strong possibility of increasing functionality through more advanced algorithms and techniques.

Research Objectives

- **Objective 1:** Create an AI generation script that allows for fully randomized AI personalities.
- **Objective 2:** Create an AI generation script that allows designers to input parameters to build more targeted, yet still randomized AI personality archetypes.
- **Objective 3:** Build an AI rules-based dynamic script that allows the AI to weigh its choices and determine what it should do with a randomization factor to account for realistic error margins.

Study Methodology

Dynamic scripting typically involves adding or subtracting to a reward function based on outcome. In this artefact, the outcome is based entirely on preset functions involving needs of the NPC. Therefore, the weighting logic remains the same, but the reward logic is static insofar that it is not teaching the AI to behave a certain way, it is allowing the AI to determine what is best based off a preset personality. This allows for nuances in personality programming that is not meant to be optimized. For instance, if one were to want a lazy NPC, it would not make sense to implement a reinforcement algorithm that teaches the AI to accomplish anything other than being lazy. Without the reinforcement aspect, designers will be able to create generative AI that will be purposefully "flawed" in terms of optimizing output, but allow for more realistic determination actions.

```
/// <summary>
/// Method <>GeneratePersonality</> randomly generates weights for each Need.
/// </summary>
void GeneratePersonality ()
{
    // This sets the min and max weight for each need.
    minWeight = 0.1f;
    maxWeight = 1f;

    // This generates a random weight for each need within the set parameters.
    bladderWeight = Random.Range(minWeight, maxWeight);
    boredomWeight = Random.Range(minWeight, maxWeight);
    energyWeight = Random.Range(minWeight, maxWeight);
    hungerWeight = Random.Range(minWeight, maxWeight);
    thirstWeight = Random.Range(minWeight, maxWeight);

    // Normalizes all of the weights to add up to 1. This is to ensure all of the probabilities add up to 100%.
    totalWeight = bladderWeight+ boredomWeight + energyWeight + hungerWeight + thirstWeight;
    bladderWeight /= totalWeight;
    boredomWeight /= totalWeight;
    energyWeight /= totalWeight;
    hungerWeight /= totalWeight;
    thirstWeight /= totalWeight;
}
```

Figure 2. The random generation function.

Results and Discussion

The AI random generation function, based on thorough user testing, is capable of generating any number of randomized AIs that will prioritize different sets of actions. In this example, there were AI that would watch TV all day and refuse to work and AI that would work all day and refuse to eat.

One issue with the results, is that without designer-inputted weight ranges, the AI generation function is capable of producing AI that are incapable of looking after themselves. While this isn't a major issue, this artefact included a death function, where an AI would die if one of it's needs bars reached zero. Disabling this fixes the issue, but raises some concerns around potential limitations.

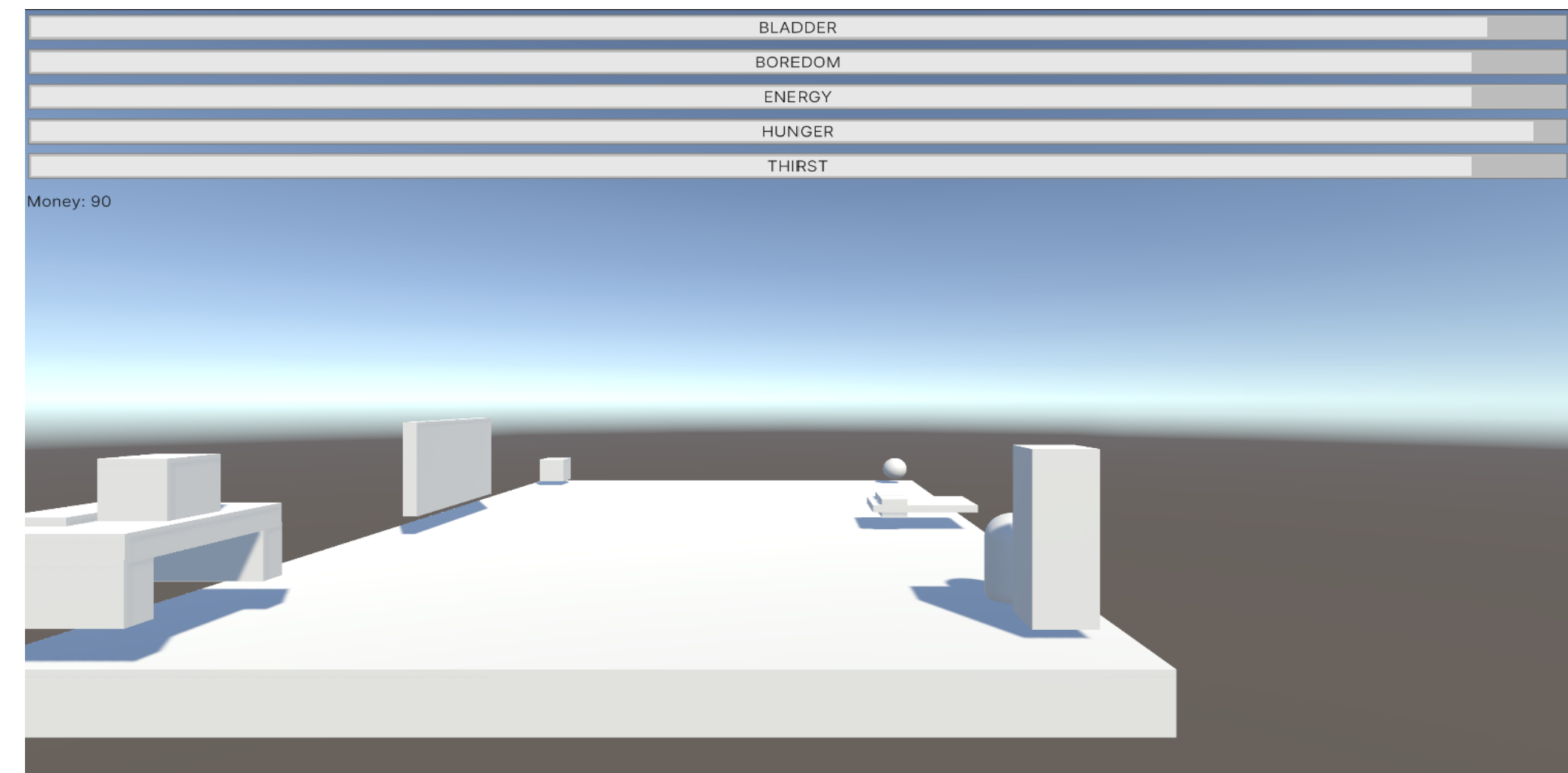


Figure 3. The example AI feeding itself. Notice the needs bars on the top of the screen.

What is already known about this subject?

- **Imperfect AI** are beneficial when it comes to making realistic games.
- **Imperfect AI** are also beneficial when it comes to making AI combatants challenging, yet beatable.
- **Dynamic scripting** is a relevant method for attaining reward-based AI behaviour.

What does this study add?

- **Dynamic scripting** can be adjusted to allow for imperfect AI training via preset, designer-adjusted weighting templates.
- **Emergent AI** can be effectively generated using a modified version of dynamic scripting.
- **This artefact**, provides a modular, easily expandable and exportable AI generation and functionality script that can be used in a myriad of different Unity projects.

References

- [1] L. Lidén.
Artificial stupidity: The art of making intentional mistakes.
AI Game Programming Wisdom, 2:41–38.
- [2] P Spronck, M.J.V. Ponsen, I.G Sprinkhuizen-Kuyper, and E.O Postma.
Adaptive game ai with dynamic scripting.
Machine learning, 63(3):217–248, 2006.
- [3] Pieter Spronck, Ida G. Sprinkhuizen-Kuyper, and Eric O. Postma.
On-line adaptation of game opponent ai in simulation and in practice.
In *GAME-ON*, 2003.