

AI GAME OPPONENTS

DESCRIPTION

The project is designed to create a game with several different kinds of AI opponents within a physics engine. Each AI opponent would be trained using different systems such as in needing to move between two points on a map, or in needing to survive a fight with a different neural network taking the form of a creature in the same engine. To provide for more interesting gameplay, it is important to have the AI systems have access to output neurons that can control the movement of the creature's joints. The fights between AI-controlled creatures would be decided using a hit point system similar to either Dungeons and Dragons or Dark Souls (though with hits being determined based on whether the attack was physically in the right location (rather than being dependent on a random number generator), and having the damage be based on the inverse of the distance from where the attack hit to where the center of mass of the creature is (with a minimum of 1 point of damage (plus relevant modifiers chosen at character creation time depending on the extent to which character creation is found to fit player expectations))). In order to allow for human-level reaction times to be useful, the AI agents will be limited in their actions in real time such that the acceleration of any joint of the creature model can't exceed some function of the creature's dexterity score (if using a character creation process), and each computation of the AI's neural network in passing information across each connection once only happens once before the physics engine refreshes with all current object locations and camera views of all creatures. The actual mechanics of gameplay can be further developed as the base physics engine gets tested to determine how a human player can operate. If time allows in the development of the project over the semester, then the project shall also include a consistent magic system that all creatures can use (with a cooldown time on spells determined based off how a creature scores in a given attribute, using something akin to a Dungeons and Dragons fifth edition Sorcerer's "Sorcery points" (though with changes like having regeneration of those points per minute of simulated time being based on the equivalent of a "charisma" score for the creatures of this game). Previous court cases have determined that game mechanics are not copyrightable, and significant changes like those planned for how magic would work in this world (and the lack of names of Dungeons and Dragons' mechanics appearing in the user interface) prevents any sort of infringement on anything Wizards of the Coast claims copyright over. The plan for the magic system is to have it require a series of actions taken in a particular order with a particular margin of error in order to cause a spell to go into effect, with no description appearing anywhere in the game as to how to perform the spell (except user generated notes, and users observing AI agents that have randomly discovered certain magical secrets); an example of how this could work would be to have the spell "conjure chair" be cast by raising one's hand, moving forward the equivalent of 1 space, backward one space, left one space, and right one space, then dropping the hand to have the chair appear 2 spaces in front (disappearing after 10 minutes).

MOTIVATION

The purpose of this project is to create a neural network-based opponent inside of a game similar to the Dark Souls franchise. To do this, this project proposes using multiple, unique styles of neural network design within the frameworks of a genetic algorithm and others trained on a reinforcement learning algorithm. The different designs of neural networks will be trained by having them randomly tested by either walking between two objectives in a hostile area, or in trying to fight a different creature running either the same design of neural network, or in trying to fight a different creature with

a unique neural network design. Some work will also be done in adapting open source world-building projects to create a physics engine (if other engines are not suitable for our design) with a variable environment that will be interesting to observe the creatures walking through. The other big motivation is to determine what kind of illusion drawing tools work well for both an AI creature and a human player, and how would they compare to human players simply uploading 3D models of what they want to have in the game or the neural network activations of an AI agent simply being mapped onto points and colors around a sphere to create a mesh of triangles. Creating a full and consistent magic system that the players can discover through trial and error may provide greater satisfaction to players than what current AAA games (excluding Minecraft (which does not natively support as arbitrary of illusions as can be created here, nor as intelligent of enemies)) have provided recently.

OBJECTIVES/GOALS

Our first objective is to establish a base set of requirements for what the AI networks can expand to, what the user interface should look like, and the spells that will be supported, and how those spells can be uniquely represented by natural-looking actions within the game environment.

The second objective is to figure out what game engine will allow us to have the necessary control over the AI networks (specifically in making sure the AI networks don't think and move faster than a human player when a human player is involved), and to make sure that that game engine will allow for many different AI models to satisfy different strategies to take over the "gene pool" with respect to a genetic algorithm to determine the traits of some of the AIs.

The third objective is to build the user interface for the game so that we can play the game as a means by which we can debug any problems created with the AIs and the magic system.

The fourth objective is to build the magic system for the game starting with 16 generic spells that can be used to express most of what players would think about doing in some way, such as Creation, which creates an object that fits in a 5 ft by 5 ft by 5 ft cube, and would allow a player to input a file of a 3D model of something that now exists in the game.

The fifth objective is to design the neural networks used by the AI opponents, and design fast ways to test the quality of neural networks by having them either compete to complete tasks or fight each other, presumably using an evolutionary algorithm to replicate the most "fit" parts of the best neural networks to more networks.

The sixth objective is to have other students at UT Dallas play against these AI opponents and carefully track what they are saying is fun or not fun about playing. These tests can be used to determine if there can be a more convenient way to design the game for players, and for making sure the feel of the game to players is consistent with edits to the code.

(Stretch goal) The seventh objective is to design a way that multiple players can be in the game (theoretically playing together via the equivalent of a LAN, since it is unlikely that the feel of the game will lend itself well to being an MMO), and to design more of a story into the game (dependent upon how interesting a "natural" AI can be without necessarily being hostile to the players).

(stretch goal) The eighth objective is to build a consistent terrain with functioning AIs that can complete a story of sorts for most players.

(stretch goal) The ninth objective is to allow the players of the game to have mounted combat-type mechanics for the creatures they summon or make with the magic system.

ROLES AND ESTIMATED EFFORT

Roles Needed

Team Leader: There needs to be someone who can come up with an optimistic schedule and make the other members of the group stick to deadlines on parts needed, and get the group to meet to work

as efficiently as possible on each part. The team leader will also need to work on code for the program to run and know how to divide up parts of the code.

Unity / Unreal Engine / Other Game Engine Master: There needs to be someone who can learn how to use the game engine for the project like a master, who can then delegate backend coding to others to work on formulas while the game engine master puts the right API calls into place to make the visuals that users see.

Gameplay Tester / Gameplay Designer: There needs to be someone who understands what kind of feel to the gameplay doesn't seem right, and who can spot bugs in the gameplay and have an idea of what someone should look for to fix bugs. This person will also have to write code in the project.

Neural Network Savant: There needs to be a person willing to look deeply into the workings of the AI systems to see what can be changed to make the system work better. This person is mainly responsible for coding the AI agents, but will also need to write some other code as well.

Tasks

1. Write the base requirements of the system.
2. Write the design of the magic system.
3. Write the design of the AI agents.
4. Code a prototype world to make sure that human playability is achieved and can be edited as needed.
5. Code up the interface the human players use to make the movements needed to cast spells, and make sure the in-game sound works for spells that require sounds, and make sure those movements can be properly recorded for the purpose of checking if they correspond to the trigger of a magical effect.
6. Code up the effects of creating each spell on the game world.
7. Test that the correct spell goes into effect when a human player takes the actions needed to activate it.
8. Code up a training environment for AI agents and check to see what inputs and what outputs are practical for such an agent to consider.
9. Code up the AI agents and train their networks with the training environments.
10. Test these AI networks in the real game environment against human opponents.
11. See if any sort of direct edits to the code of the AI agents allows them to pick up on the strategies for using spells.
12. Develop the world and design of the creatures in it.
13. Have other students play the game and see what edits they are looking for can be added to the game to make it more expansive without harming the purpose of the game. (For this

purpose, don't tell these test players about the spells available in the game (other than that there are some things they can potentially discover while playing to alter the game)). Ideas these test players may bring forward could be things like having a hunger and thirst mechanic and an exposure and sleep mechanic to make it important for the character (and AI enemies and NPCs) to find food and build up a culture of sorts, which can allow for more organic world-building.

14. Either develop the game some more to make it deeper for the individual, or widen it by providing a way for friends of a human player to join them or compete with them.

Amount of Hours

For the purpose of determining the time to complete tasks, "1 week" represents 9 hours per person approximately.

- 1 The first 4 tasks can likely be completed on the first week of the semester.
- 1 Task 5 will likely take another week as there will be a need to have to iterate coding and testing to determine how a PC player ought to control all of the character's possible actions ergonomically, and what the signal-to-noise ratio needs to be on what constitutes a "musical note" for the purpose of casting a spell.
- 3 Task 6 will likely take 3 weeks, the first week for developing on the easier 5 of the 16 spells (the first spell will take the longest as all 4 of us would need to meet to make sure each of us understands the process of making a spell, so we could not parallelize that; the next 4 spells can be designed in parallel, with the code sent to each other to make sure each person can run the code on their own device with all 5 spells working), and testing them to make sure the effects look good and can be triggered. The second of those 2 weeks is for developing the rest of the spells except for the hardest three (so 2 spells created by each programmer and tested). The third week (of the 3 weeks for task 6) would be for the hardest spells (which would have been started in the previous weeks), specifically for making sure that the illusions can be efficiently created by a human character without having the game screen look too much like that of a game (i.e. not creating by just giving a menu of objects that can be put in place as options, part of maintaining the proper feel of the game (where a couple intermediate spells could involve a menu or map-like interface for choosing options (like teleport allowing you to choose a place on the map to go to))).
- .025 Task 7 should be accomplished with task 6, and should otherwise only take an hour to verify.
- .25 Task 8 should be accomplished in about 9 hours distributed in the weeks after the first week (once there is something a human can play with and edit, then an AI testing environment can be created). It should then take only 1 hour after task 7 to make the AI testing environments incorporate the spells.
- 3 Task 9 Should be completed in the week after task 6 is completed, particularly since the code for writing different kinds of AIs has already been developed. However, it is likely that there will have been bugs in the design of the test environments, so the code for that will likely need to be rewritten (taking up another week), then the AI systems should only take an hour to test (of human time, since the AIs would be playing against themselves very quickly overnight from Friday to Monday), but could take another week to debug.
- 3 Task 10 will be completed concurrently with tasks 11, 12, and 13 as each member of the group works separately, with one developing the game world and storyline, while two others try editing

the code of AIs directly to see what kinds of things the AIs can be made to do with their semblance of intelligence, and another further develops the models which can be used in the game to make it more full, as well as test the game from the updates given by the other members of the group. The outside testers of the game are also working concurrently to develop the game.

- 2 Task 14 will take both 2 weeks (to make something marketable), and potentially indefinitely as the members of the group work to make the game interesting in new ways with their own copies of the code.

Thus, the total number of weeks (before stretch goals) is approximately 11.275, or about 406 hours over the course of the semester. This is an optimistic timeline based on having 13 weeks that people can be expected to work on the project (assuming starting the week of January 14th and ending the week of May 7th, and excluding spring break, a week's worth of midterms, and classes that have tests or projects due in the week of May 7th that are similar to finals. The timings on everything provides some leeway so we have 3 hours per week per person (of the 12 hours the project can be expected to take per week as per the MS Teams meeting before Thanksgiving 2021), which are definitely going to be used by things taking longer than expected. Setting the deadlines on each of these tasks early also has the psychological effect of having those inclined to procrastinate working much harder the day well before something is due, then getting the feeling the next day so that, as the deadline moves forward, much more of the project is completed than could otherwise be expected to be completed. If the optimistic timeline can be satisfied, then the stretch goal of having something a wider audience would like to buy can be satisfied.