Research Assessment #4

Name: Nigel Tatem

Date: September 18, 2023

Subject: Artificial Intelligence

Source:

Lara-Cabrera, Raúl, Carlos Cotta, and Antonio J. Fernández-Leiva. "Procedural map generation for a RTS game." *13th International GAME-ON Conference on Intelligent Games and Simulation*. Malaga (Spain): Eurosis, 2012.

Assessment:

After further research into my topic of Computer Science I determined that my project would need to be narrowed down to some small concepts of the broad field. These smaller concepts include artificial intelligence, computer programming, and software development. Each of these I have high interest in as our world shifts towards technology as its future, and my past memories being flooded with nostalgia of the technologies that came before such as Nintendo's DS. The article selected on procedural map generation for RTS games holds significant relevance in the context of my upcoming project involving artificial intelligence (AI). It serves as a foundation for learning about a generative process that is commonly used in the most popular software such as Minecraft.

What is most important to take away from this article is what terms and processes are used in this relatively 'simple' algorithm for procedural map generation. Parameters in coding can be defined as instructions that must be followed, something I have known for years. These parameters are essential to procedural generation and in order to best optimize a program with parameters the following must be considered: fitness function, dynamic creation, diversity creation, feedback loops, elitism, and stochastic process. Firstly the term elitism must be defined, it describes the behavior of the computer in procedural generation of this experiment. Elitism is when the computer only allows the best solution to survive. After finding the best solution the AI continues the stochastic process, in simple terms using random variables to create more solutions to be compared to the fitness of the top solution. This all makes good sense, but something important to know from this article is, how do you determine the fitness function?

The fitness function is a term often used in experiments with procedural generation, it is an equation similar to y = mx + b except y is the "fitness" defined by the sum of its parameters mx+b. In order to receive better results from the generator every factor that goes into creating something such as a balanced map must be considered. In the case of a balanced map, if your fitness factors parameters only consist of, "20 planets and 100 fleets on the map each constrained to this area" you will get only this. On the other hand if you added more specificity to what makes a map balanced such as including resource distribution, diversity creation, and terrain features the fitness function's sum will be higher and produce more accurate maps. In order to have more randomness with your maps while still maintaining its balance, a new objective can be introduced to the stochastic process like creating more “strategically interesting” layouts. This can be applied in a broader sense essentially to create everything affecting gameplay besides NPCs (non-playable characters) which require their own AI.

In conclusion this source provides an explanation of a foundational piece of artificial intelligence in applications used in popular apps like Minecraft, Terraria and with uses for things such as generating practice problems for SAT and much more that are completely original/unique. In finding a better understanding of this concept I can better incorporate it into my final project which will likely involve procedural generation in some form. When contacting a mentor they will be satisfied that I am caught up on where to create procedural generation (in places like java console), how to optimize the system, and what its function and truly endless capabilities are.