Procedural Generation Algorithms and Why Use Them Over Machine Learning

Nicholaus Fleming
California State University, Stanislaus
Email: nfleming@csustan.edu

Introduction:

Procedural generation is a particularly interesting feature that appears in some video games. This concept is a type of algorithm that was designed to allow a computer to generate specific content in a seemingly random manner, such as 3-dimensional and 2-dimensional assets in video games. Although, most players do not have a great understanding of how they work, only that when used in games, the algorithm is directly tied to a line of text and numbers called a seed. The question is, how does it work, and more importantly, why would one use either procedural generation or newer content generating algorithms that have come up in recent years like machine learning when creating digital assets?

Background and Literature Review:

There are two main purposes for procedural generation algorithms, with the first being that it allows for a decrease in workload for creating content by allowing the computer to do so instead. The second purpose being that when used in video games for things such as world or level generation, it allows for a fresh experience each time the game is played as a new world is created each. To get an idea of what content generated for a game world could entail, Freinecht and Effelsberg went over the various aspects of game worlds that could be procedurally generated: landscapes, textures, people, buildings, roads, vegetation, and bodies of water (2017).

The basis of procedural generation is pseudo random number generation that is used to create the lines of letters and numbers called seeds (Watkins, 2016). These seeds are used as parameters for the functions of these algorithms in order to automatically create seemingly random digital content. Watkins goes over a simple application of generating a 2-dimensional game world, where in order to create a seemingly endless world that is procedurally generating, the screen it is thought of as a grid dictated by a data structure, such as an array. As the player moves across the screen, and the screen moves, the objects set into each grid is shifted away in the opposite direction, and in their place on the grid are pseudo randomly chosen objects from a predetermined list (2016).

It is known that machine learning works by creating an algorithm that takes in a large set of data to create an artificial intelligence that improves itself at the task it is assigned by training on the data set as a source of reference for its performance. Wang and his associates explored the idea of using a form of machine learning algorithms for the purpose of creating realistic game content for use in virtual reality (2020). One machine learning algorithm they had showed was one that would create a digital 3-dimensional model of a face by letting it scan an image of a woman so it may attempt to replicate it (Wang et al., 2020).

Methods:

Materials:

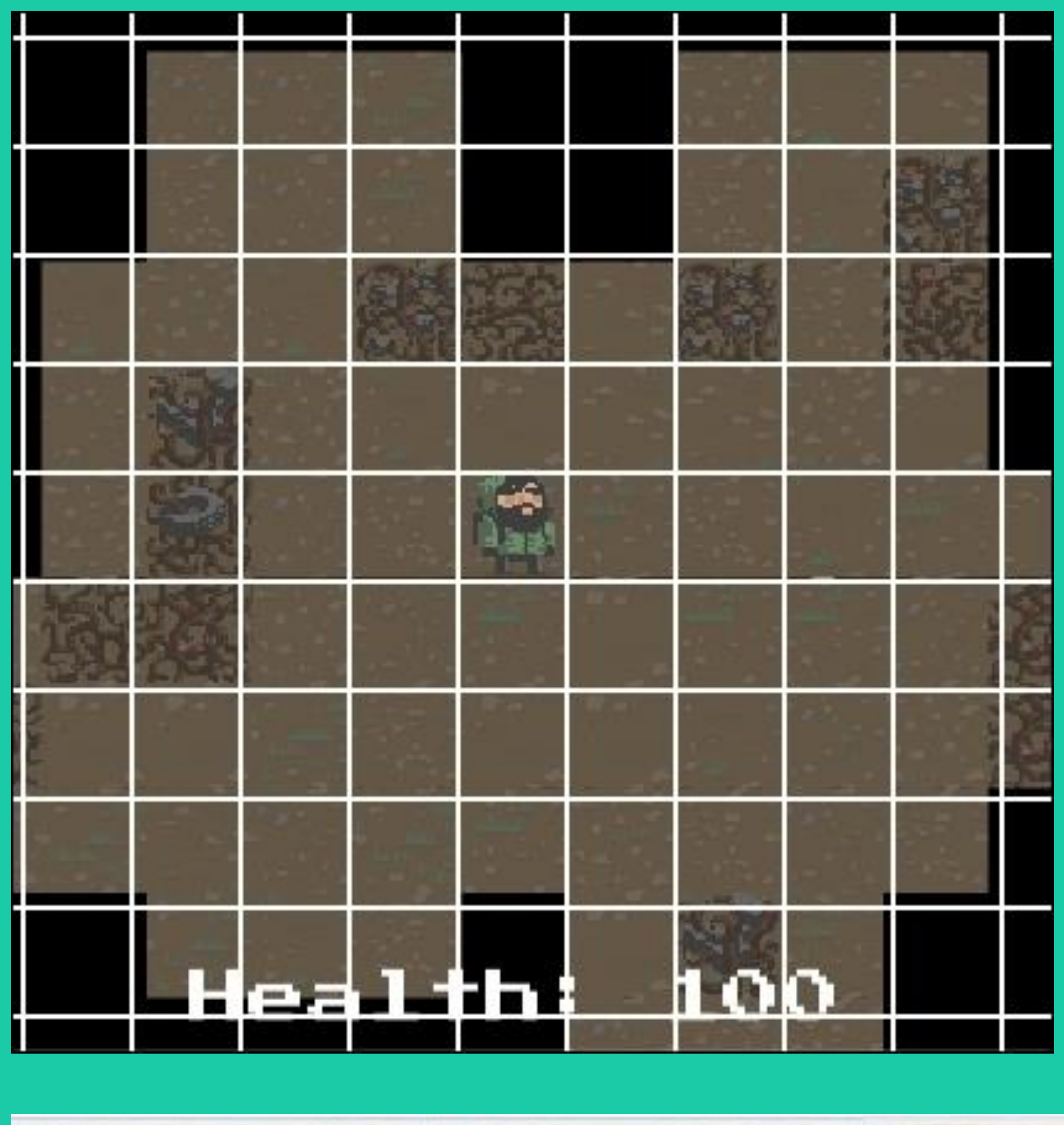
For my research question, I will try and only use primary sources, and I do not plan to use any secondary sources yet. The sources I have so far are all scholarly research and survey articles focused on digital content generation with either procedural generation or machine learning. The main database I have been and will be using is the CSU Stanislaus library onesearch database and any connected ones that it provides access to.

Design:

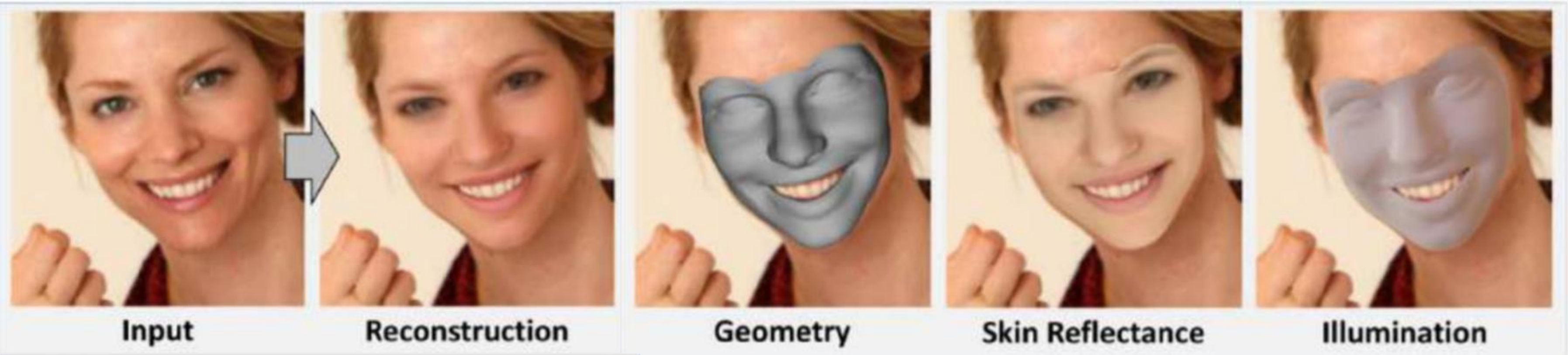
I have decided that I will be taking the qualitative design approach of a case study and apply it to my research question. For this question, I am required to compare two algorithms, and to effectively do so I must analyze cases of their usage. In analyzing these cases of use that I will be providing from my primary sources, I will be able to focus in on my research question and be able to see why they were used for each case that I will provide.

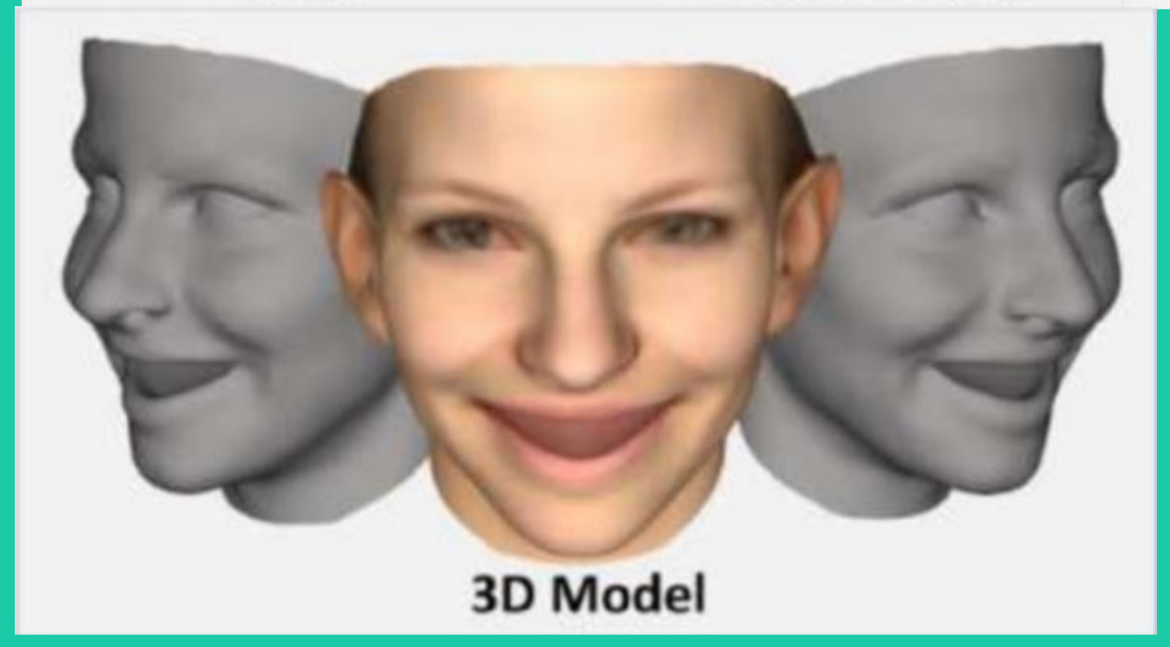
Procedure:

I will be conducting an analysis of my sources to compare procedural generation and machine learning. The goal will be to find key differences in the usages of each, and their drawbacks. With that, I will be able understand the reasoning of when and why to use one over the other and explain why one algorithm is not favored over the other when it comes to digital content creation. I believe that I will have to focus on the drawbacks or issues of their usage, more so than just the cases in which they can and have been used, that way I can accurately depict the reasoning.



Visualization of game world split into grid for procedural generation (Watkins, 2016).





Demonstration of 3D Model constructed using machine learning (Wang et al., 2020).

Expected Results:

Taking a qualitative approach to my research, I expect to find issues the authors of my sources explain for either algorithm. I can use any issues, drawbacks, or even reasonings they state for using their algorithms to reach a conclusion of when and why to use either type of algorithm. I do expect that choosing between two algorithms that can the same thing may come down to the skill and preference of the programmer implementing it.

Significance:

This research is particularly important for programmers in the entertainment industries that create digital content due to the importance of properly understanding which algorithm a programmer should implement for any given task. However, it may also hinder progression of research for either procedural generation or machine learning algorithms as programmers may use the more suitable one for a given task, while the less suitable one gets left out on the chance of being improved if it was attempted to be implemented for that given task.

References:

Freiknecht, J., & Effelsberg, W. (2017, October 30). A survey on the procedural generation of virtual worlds. Retrieved February 28, 2021, from https://www.mdpi.com/2414-4088/1/4/27/htm

Wang, M., Lyu, X.-Q., Li, Y.-J., & Zhang, F.-L. (2020, March 23). *VR content creation and exploration with deep learning: A survey*. Computational Visual Media. https://link.springer.com/article/10.1007/s41095-020-0162-z.

Watkins, R. (2016). *Procedural content generation for Unity game development: harness the power of procedural content generation to design unique games with Unity*. Packt Pub.