Celica Lysik
Professor Tolstenko
GPR-340
15 December 2023

Blog Post: Procedural Generation and SDL

Procedural Generation is a popular technique to randomly generate content based on a predefined set of rules. While procedural generation has uses in a wide variety of applications, it is most commonly used in Video Games to generate content such as landscapes, character designs, animations, etc. For this project, the use of procedural generation is utilized specifically to generate levels such as the ones seen in games like Enter the Gungeon, The Binding of Isaac or Nuclear Throne.

To create this project, the cross-platform development library, Simple DirectMedia Layer (SDL) was used. This library was created in 1998 by Sam Lantinga while working for Loki Software. While the library was initially used to port software between different hardware, it soon became capable of much more, including providing a general framework that can be used to develop games. SDL provides low level access to keyboard, video playback, audio, joystick inputs as well as graphics through OpenGL or Vulkan. Due to the nature of SDL being cross-platform, it made it a good choice for this project, which aimed to provide a boilerplate that could provide procedural generation algorithms that could easily work on any platform for a wide range of games. For the sake of this assignment though, the project was developed and tested on Windows computers running Windows 11.

As described in the book, *Procedural Content Generation for C++ Game Development*, the author, Dale Green, describes procedural generation and randomness as follows “Procedural generation is the process of content creation using an algorithm. This in itself has no element of randomness — Randomness is induced when we give these algorithms different inputs or alter their expressions. This variance is what creates the variety of the output.” (Green 9). He goes on to describe how these two pieces together form what is more commonly referred to as “procedural generation”. With this in mind, the algorithms that are used to create this project are fed random inputs which fits into this definition of procedural generation.

To avoid feeling too similar to past assignments, two algorithms were looked to for inspiration during this project. In particular, the article “Dungeon Generation in Binding of Isaac” by BorisTheBrave and “Random Level Generation in Nuclear Throne” were used as inspirations for the procedural level generation algorithm during this project. In “Dungeon Generation in Binding of Issac” the author lays out a few specific rules that guide the generation of rooms. Using a queue, the neighbor cell is determined by adding +10/-10/+1/-1 to the current cell. The numbers here represent up,down,left and right respectively. Since each tile is 32x32 in this project, this was modified to account for the specific tile and board size. From there, if the neighbor cell is already occupied, has more than one filled neighbor, or if the algorithm determines that enough rooms have been generated, end the program. Otherwise, mark the neighbor as having a room in it and add it to the queue. This is the general idea of the algorithm outlined in this article, and is similar to the breath first search algorithms that we utilized for maze generation in earlier assignments. A similar algorithm is employed in “Random Level Generation in Nuclear Throne”. The project utilizes a structure similar to random walk called a “FloorMaker” and this will choose a random direction to move in and place a 1x1 floor tile there. Based on what area the FloorMaker is in, it has a higher chance of generating 2x2 floor tile instead, which will create a room. In theory, utilizing these two concepts together would generate a somewhat unique solution to procedurally generated levels in our SDL project.

The project is available at https://github.com/Sodaguts/Game_AI-Final

Sources

Boris. “Dungeon Generation in Binding of Isaac.” BorisTheBrave.Com, November 12, 2021. https://www.boristhebrave.com/2020/09/12/dungeon-generation-in-binding-of-isaac/.

Green, Dale. *Procedural Content Generation for C++ Game Development*. Birmingham: Packt Publishing, 2016.

Highway. “ Random Level Generation in Nuclear Throne.” Indienova, December 24, 2016. https://indienova.com/u/root/blogread/1766.