Procedural Generation

Evan Barry – T00202376

Procedural Generation has existed in the gaming industry since the 1980’s and is becoming increasingly important for creating game worlds, backstory, and characters across many genres. The most high-profile games that have incorporated procedural generation in their development over the last number of years have been Minecraft(2011) and No Man’s Sky(2016) (de Kegel and Haahr, 2020). In terms of the areas where procedural generation is used, the most popular area is world building, which both of the previous mentioned games are best-known for. With respect to Minecraft(2011), the game world is generated when the user, or player, first starts playing the game. Minecraft(2011) uses blocks of different material, i.e. dirt, stone, sand, water, etc., to build the world. This relatively low detail in world building allows the game to procedurally generate a world virtually infinite and implausible that a player would explore all of it. Meanwhile, No Man’s Sky(2016) has a much higher level of detail to what is procedurally generated that the content is generated when the players enters a new area, in this case a galaxy as No Man’s Sky(2016) is a space and planet exploration game.

Procedural Generation has become particularly popular with developers. This is due to the fact that the production of game content has grown to the point of becoming a bottleneck in companies’ schedules and budgets(Soares De Lima, Feijo and Furtado, 2019). To help the developers with this problem, Procedural Generation techniques are used to reduce their workload. Algorithms are used to generate content for games with little to no human interaction and can be fine tuned to fit a certain style or layout. The type of content that is procedurally generated to reduce a developer’s workload are generally items that fill out a world such as trees, grass and other vegetation, which can been seen in games such as Horizon Zero Dawn(2017), Far Cry 5(2018) and Forza Horizon 4(2018)(Soares De Lima, Feijo and Furtado, 2019).

Another main area where Procedural Generation is used is in level design. There is a general pipeline for procedural level generation. There are stages in the pipeline as follows:- generation of the base graph, weighting and computation of the minimum span tree, construction of the level graph and merging, and conversion to geometric representation(von Rymon Lipinski et al., 2019). The first stage, generation of the base graph, describes the first abstract structure of an indoor level. It does not represent the final appearance of the level. The second stage, weighting and computation of the minimum span tree, produces a level skeleton where each point in a graph can be reached via exactly one unique path. The third stage, construction of the level graph and merging, results in a level graph which can be either used to generate the final geometric representation or it can be connected to other previously generated level graphs to allow for the generation of more complex indoor systems. The final stage, conversion to geometric representation, converts the level graph to a geometric representation. This process requires a set of pre-modeled 3d components, rooms, dead-ends, corridors, door, etc. Each model is labelled and mapped to a vertex on the level graph for the model to be placed. This level generation pipeline appears to be versatile and basically suitable for practical use.

## References

de Kegel, B. and Haahr, M., 2020. *Procedural puzzle generation: A survey*. *IEEE Transactions on Games*, .

von Rymon Lipinski, B., Seibt, S., Roth, J. and Abé, D., 2019. *2019 IEEE Conference on Games (CoG).* *2019 IEEE Conference on Games (CoG), London, United Kingdom, 2019, pp. 1-7, doi: 10.1109/CIG.2019.8847956.* IEEE.

Soares De Lima, E., Feijo, B. and Furtado, A.L., 2019. Procedural Generation of Quests for Games Using Genetic Algorithms and Automated Planning. In: *Brazilian Symposium on Games and Digital Entertainment, SBGAMES*. IEEE Computer Society.pp.144–153.