Seminar Sheet 7

Dr Aaron Bostrom

January 21, 2020

We are considering data structures and how we can efficiently store information.

This seminar sheet is aimed at exploring and expanding on your current knowledge. Especially putting to use Maps, Sets and Arrays.

In previous worksheets we looked at items, and inventories and recipes and crafting.

The aim of this worksheet is to get you to design your own algorithm for a procedural system within a game.

Research a simple procedural system from some popular games. This may include, but is not limited to, (simple) maze generation, procedural level generation, procedural spawning of items or weapons or enemies, or more generally procedural placing of assets within a scene.

You can choose to implement this seminar sheet in either blueprint, C++ or both.

Procedural systems can be very simple or very complex. Consider your design carefully, and be careful not to over scope what you can create in the time frame. Think about the potential pitfalls, and make sure to thoroughly test your system. Think about how designers might be able to tweak and play with your procedural system. Can they add additional meshes to an array to change the types of objects that can spawn at different points.

If you're placing meshes or assets consider how you might want to add random variation when placed in the scene? Can you generate rotations and offsets to give the spawned items variation, how can a designer play with these parameters through the details panel?

Problem 1. Research and design a simple procedural game system. Draw upon ideas from games you have played or enjoyed. Consider how the system operates, and what the code requirements might be.

Problem 2. Design the UML and consider the information you will need to store to represent the parameters and options your system may have. What are you generating or spawning. How can it be extended by designers or react to sets of assets provided

Problem 3. Implement your UML diagram in either blueprint, C++ or both and create a procedural system