Seminar Sheet 6

Dr Aaron Bostrom

January 3, 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.

The aim of this worksheet is to combine and build on the work you undertook in previous worksheets. Utilise the inventory and item systems you built in the past. You may need to evaluate the systems you built before to see if they are fit for purpose with a crafting system.

Research and design a simple crafting system. Consider how crafting systems operate within a variety of games and settle upon a design.

Implement a simple crafting system, especially considering how you might represent recipes in your crafting system, and how you might enable designers to easily add new recipes without having to add additional code.

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

Crafting 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 you might setup a crafting test level, with lots of resources so that you can make all of the items in your inventory etc.

Problem 1. Research and design a crafting 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 recipes / crafting materials. How will items be crafted, and how will the requirements be checked against your current items, however they are stored.

Consider the interconnected systems of items, inventory, recipe, and final produced crafted item. Think about the API and where the pitfalls and edge cases may exist.

How are the Inventory, Crafting System, and Recipes going to communicate with each other. How does the player store current crafting recipes, and how can the crafting system resolve a recipe to craft an item.

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

Problem 4. Implement your UML diagram in either blueprint, C++ or both and create a crafting system system that allows you to craft items based on recipes you have currently available.

Problem 5 (Extension). You may want to create a simple UI to show the recipes you have available. Create a system that shows missing components or materials. You could also create a multi-craft option.

Expand your system. Does your current system support multiple recipes for the same item.