CGS Brief – Unity Networked Game

Modular Networking Engine Wrapper

Complete the four **Networking in Unity3D** tutorials up to **Unity Networking Server-side Objects**. This will give you a small networked shooter that you can then extend upon in order to create a complex game system.

Complex Networked Systems

You can choose at least one (1) of the following to pursue in order to complete this assessment:

1. Networked Inventory

As part of your networking system you should be able to have an inventory system that all players on a sever can connect to, allow them to drag and drop items into the world from their personal inventories and let them or other players pick up items in the world to add to their own. The system should allow the developer to control how if an item can be dropped, how much they can drop at a time, whether the items will drop as a stack or individually, how much of a specific item they can have and allow the server itself to perpetually contain items.

Proof-of-Concept Application

What to think about when creating the application:

- Your application will need to run off a server, either independent or a client host.
- You will need the players to be able to trade items with each other.
- The items will need to be able to be stored in the game world
- Items need to be able to be discarded by a player and picked up by others
- Use games like Minecraft, Raft, 7 Day to Die as inspiration for how your networked inventory system should function.

2. Networked Resource Management

As part of your networking system you should be able to have a working resource system where, items in the server can be either found existing or planted by a player, in both cases the server controls their growth and any player can harvest said items in the shared space. Players can be set by the developer to show certain amounts of a resource.

Proof-of-Concept Application

What to think about when creating the application:

- Your application will need to run off a server, either independent or a client host.
- Players will be able to see each other in a shared space and interaction with other objects.
- Objects in the shared space can be marked for harvesting, for example, cutting down trees to collect wood, mining boulders to collect stone, fishing in lakes generates caught fish, collecting water/food, etc.
- Places in the shared space can be marked for planting and grow harvestable items over time, for example, placing seeds to make trees, feeding fish to improve spawn rates, etc.
- Player can know how much of a specific resource they have at any given time and displayed on the HUD.
- If the player runs out of a resource or a specific resource the player is killed.

3. Simple Networked RPG

As part of your networking system you allow the developer to create Scriptable Object based powers that are known by the server and allow the players to be able to find and trade gameobjects that contain these Scriptable Object based powers, if the power is used then all players in the shared space see the player perform the power at the same time.

Proof-of-Concept Application

What to think about when creating the application:

- Your application will need to run off a server, either independent or a client host.
- Players will be able to see each other in a shared space and interaction with other objects.
- Players will have powers they start with and can find in the world.
- Players use the power to affect the world or other players in a meanful way, for example, set them on fire for damage over time, an instance of damage with a played particle effect, players knocked back by the original power of the user, etc.
- Extension: If you combine this with the Networked Resource Management component you can have the powers remove from the player know resources.