Complex System Project Brief

Unity Modular Utility AI

* the purpose of the system,
  + To provide an easy-to-use Unity package that can be imported into any Unity project and used without the developer needing much knowledge of the AI systems behind it.
  + The system will run almost completed on delegates and unity events, which allows developers to add any call back, evaluation, or condition methods through the inspector or code.
  + The goal is to give the developer control over how behaviours are evaluated by allowing them to provide methods that calculate values for their game situations; the conditions that interrupt behaviours; or what happens when a behaviour starts, is active, or has ended. They will not need to know how behaviours are stored and checked or how the subsequent methods are called.
* any libraries it relies on,
  + The system only relies on the in-built Unity systems.
* the mathematical operations to be used,
* the advanced algorithms to be implemented,
* how it will be made modular, and
  + The system will be made modular using the Unity inspector to dynamically generate Utility AI classes.
  + The developer will be provided with the tools to generate as many behaviours as they’d like all from the inspector.
  + The generation systems behind the scenes will generate a brand-new C# script with the Utility AI systems, as well as new serialised variables for each custom behaviour chosen by the developer. The developer can then use a script that inherits from this newly generated script to access the behaviours and alter their functionality.
* how to integrate your system with a new or existing application.
  1. Import the Modular Utility AI package into your Unity project.
  2. Create a new script which inherits from the *UtilityAI* class
  3. Return to the Unity editor and attach your newly created script to an object such as an enemy.
  4. In the inspector, your script will have a behaviour list and a generate button. Fill the list with any number of behaviours you’d like. (Behaviours can be added, updated, or removed at any point in the future)
  5. Once behaviours are chosen, press the Generate AI Instance button. This will create a new Utility AI behind the scenes. Your script should now inherit from this newly generated script, and include an interface which provides AI Awake, Start, and Update methods.
  6. You will now see each of your behaviours as categories in the script inspector,