Generally I have 3 classes. One of them is the main class and the others child classes of the base class. Classes are called LivingThings which is the base class, Enemy and Hero. Common parameters like health, position are stored in LivingThings. I tried to use OOP in my project.

I have an event listener in LivingThing to listen the LivingThing’s health to determine if it is dead or not, and it triggers the HasDied method in Program.cs. These Action parameters are very close to interface implementation in Java and helps to trigger some events or methods when they are needed.

Hero has Move function to reach his target. Hero Class already has an event listener to reaching the target, but it is not used.

Enemy and Hero classes has an enum called LivingThingType, but it is not used.

In Program.cs, there are lots of functions. To sustain functionality and maintenance, I tried to split every work step. Reset function helps to demonstrate initial texts and clears the console for later use cases. GatherInputs helps to fill all essential Hero’s and Enemies’ properties. After getting the inputs, user can start the exploring. Then, StartExploring takes the place, and it checks the inputs and starts the journey.