This project is post poned . Scope was much bigger than initially thought.

When game starts it spawns 2 heroes which attack monsters in zones they are in.

By clicking on unit on map you will activate him / make him active   
UI - Bottom bar:  
Buttons that work:   
Equipment - shows Eq. UI, nothing is there  
BackPack - Shows items currently on heroes backpack  
KillMobs - is used to Set Task, only IDLE and KILLMOBS are working  
Travel - sets area where hero should perform task  
Return home - if hero is not in combat he will travel to base area  
Return home & Unload - same as return home, upon arrival hero unloads his backpack to WareHouse

TopRight Corner  
SpawnHero: spawns heroin base area  
Add Sword - adds item Sword to active heroes Backpack  
AddLootbox - adds LootBox to heroes inventory  
Areas - displays rea list (buttons dont do anything at the moment), its done automatically per Areas in “Areas” GameObject in scene.  
  
Areas, Enemies and Items are defined by ScripptableObjects.- race of enemies, number of objects spawned on map…  
Areas - there is area prefab   
Travelling- if hero has set task and area which he is not inside - if he is not in combat he will go to portal then he will spend time travelling }he will be in white square walking - simulating travel between zones and spawn in destination area on portal  
  
Buttons work as publishers of events to which methods are subscribbed.  
AI of units is implemented by using switches.   
  
  
I am using HIPPOs Pixel Heroes for unit visuals and 5800 Fantasy RPG Icons Pack by CraftPix on icons, rest is from random sites

Tasks: - Idle - hero is staying idle  
Kill Mobs - will kill enemies in area, if his health drops bellow treshold, , when not in combat he will rest to refill his health  
After his invenotry is full he will automatically travel to base area and unload his inventory  
  
I have attempted to use these patterns:  
Command pattern - commands are sent to AI script which is like brain of hero / monster

State Pattern - Heros and monsters behavior is using state pattern, same for HP

Observer - unit movement/travel, Buttons on UI

MVC - interface  
Object pooling for enemies