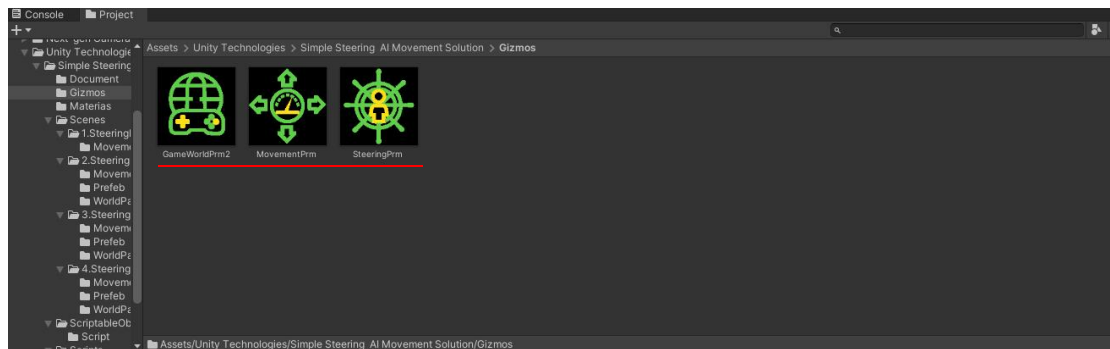
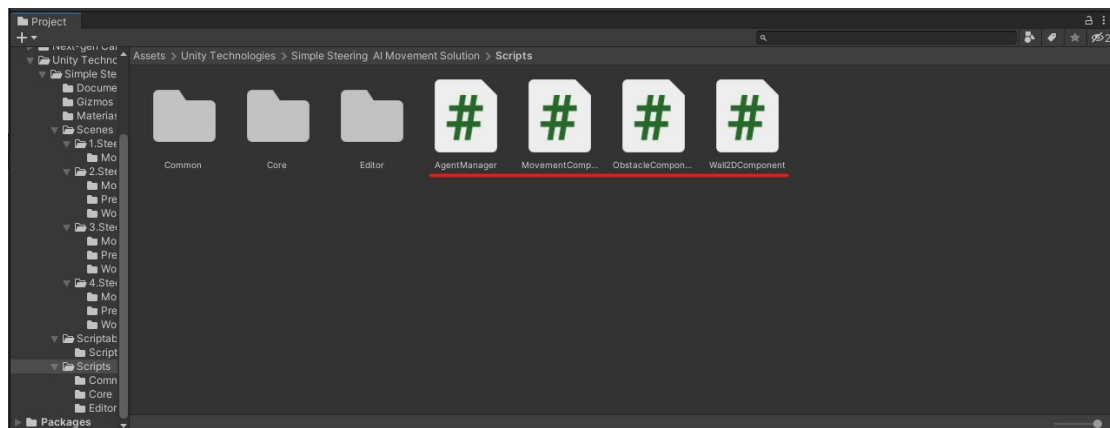
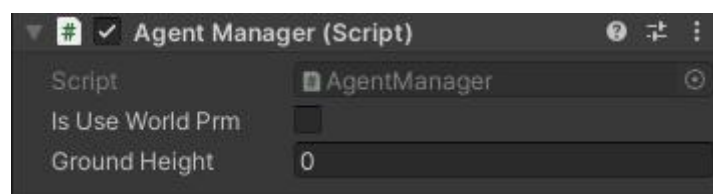


# Simple Steering Tool Document

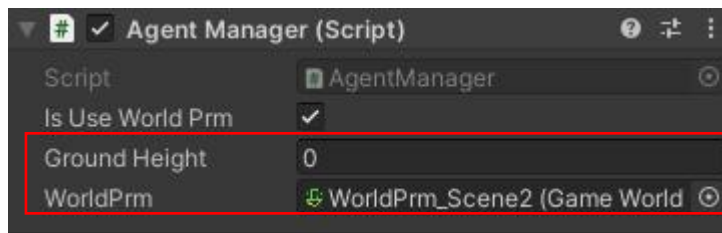
Simple Steering contains 4 core components and 3 custom parameter files, which components are - AgentManager, MovementComponent, ObstacleComponent, and Wall2DComponent and custom files are GameWorldPrm, MovementPrm, and SteeringPrm. To use this tool, Firstly, create a scene, and add 'AgentManager' to a GameObject then, what you need to do is add 'MovementComponent' to the object you want to move; add 'ObstacleComponent' to the object in the scene to create an obstacle in the scene; add 'Wall2DComponent' to create a wall in the scene; Besides, AgentManager can use 'GameWorldPrm' file to Randomly generate preset Movement Agents and Obstacles. Other than above, to move the object with 'MovementComponent', a MovementPrm and SteeringPrm should be set up;



## AgentManager



AgentManager, as the core of this tool, runs the other three components (MovementComponent, ObstacleComponent, and Wall2DComponent). There are three variables in its Inspector(In fact there are only two, the one 'Is Use World Prm' do not have actually effect)



#### Ground Height:

Adjust the Height of wall, obstacle, and moving agent in world space.

#### World Prm (World Parameter):

##### Movement Agent

Prefeb\_Size: the number of kinds of agents

MovementAgent\_Prefab: the ready to be created prefab

Agent Num: the number of agents

Initialized Position Range: the creation position range of agents

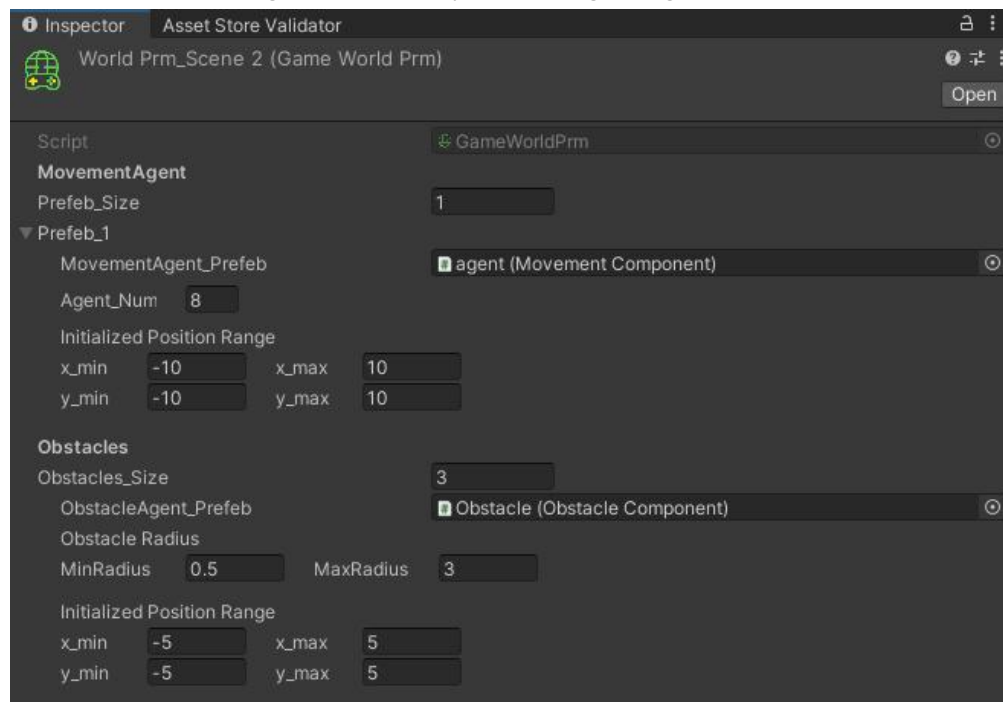
##### Obstacles

Obstacle\_Size: the number of obstacles

ObstacleAgent\_Prefab: the ready to be created prefab

Obstacle Radius: the radius range of the obstacles

Initialized Position Range: the creation position range of agents.



## MovementComponent



Is Player Control: tag on if it is the player avatar

Tag Filter: agent with the same tag can be easily accessed in the code

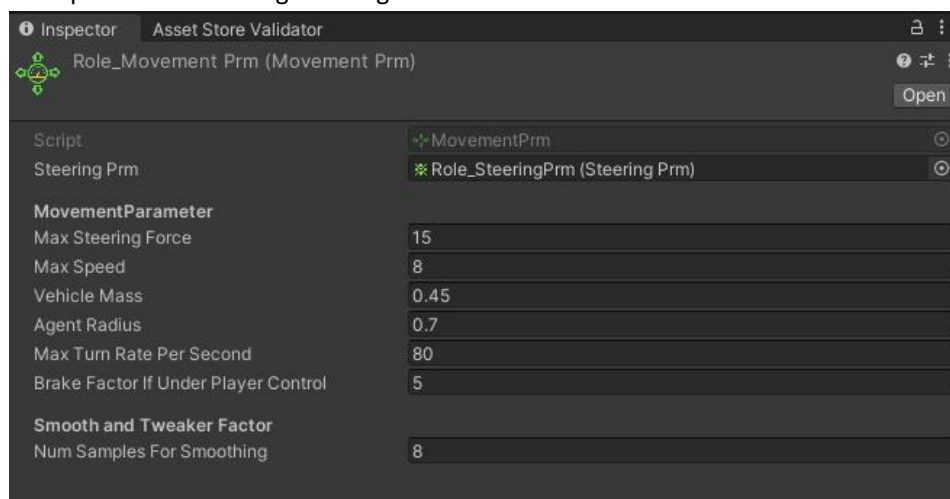
```
{
    foreach(var vehicle in AgentManager.Instance.Vehicles())
    {
        if(vehicle.GetEntityType() == (int)EntityType.AIVehicle)
        {
            vehicle.Steering().WallAvoidanceOn();
            vehicle.Steering().WanderOn();
            //vehicle.Steering().FlockingOn();
        }
    }
}
```

MovementParameter:

The parameter in this file is in relatively straight meaning, except 'Break Down Factor If Under Player Control' and 'Num Samples For Smoothing'.

Brake Factor If Under Player Control: the larger the factor the faster to stop(when agent with no input)

Num Samples For Smoothing: the larger the number the smoother the turn



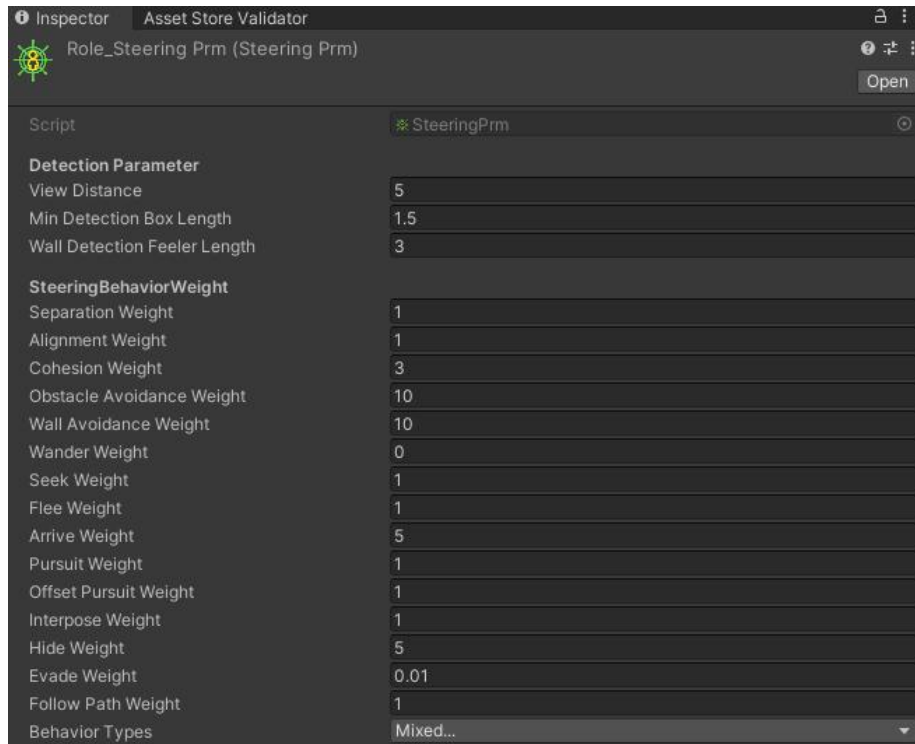
SteeringParameter: as a variable for 'Movement Parameter'

View Distance: the distance agent to find out its neighbour

Min Detection Box Length: detection box is for avoid obstacles.

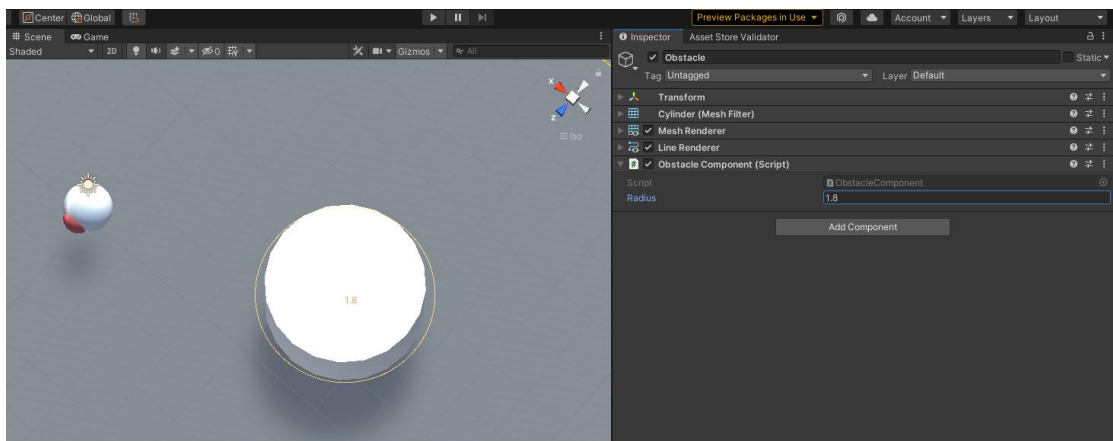
Wall Detection Feelers Length: for detect and react to air wall if the 'use wall force' is tagged

The Weight of each Steering Behaviors:



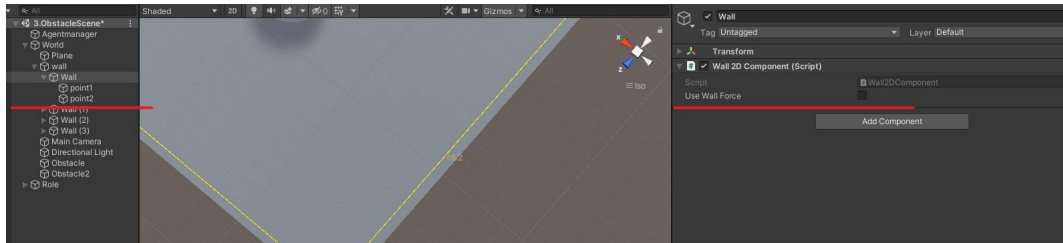
## ObstacleComponent

ObstacleComponent, for now, can only generate a circle shape. Add the component to object you want to set up as obstacle then all down.(That's it)



## Wall2DComponent

To create a wall in world space, add Wall2DComponent to a object and make sure the wall object has two child named 'point1' and 'point2'.



## DrawDebug

There is a debug component can help you to check out if there is anything wrong. Add it to a Object and make sure assign the 'AgentManager' to it.

