

Grid.cs

This is a custom made grid with squares as boxes to cover the level.
The grid has a center which is controlled by a custom transform object.
The grid can vary in size, 4x4,8x8 etc.
Each cell can vary in size as well.

The grid is divided into a quad tree. Meaning when trying to find a cell it doesn't loop through all of them but navigates through the tree.

First the root, then one of the four children of the root and so on and so on until a leaf is found.

The bounds can be visualized for debugging purposes.

It is constructed in Awake, so it's before Start.

Each cell has information such as if it is containing an obstacle or not.

Enemy.cs

Inherits from the behavior tree base class.
Contains relevant leaf nodes for the tree.
Has a reference to the player transform.
Has a navigation object.
fills in relevant information to the tree at start.
updates the tree in runtime.
takes action based on the tree.
if navigation is needed it uses that. Only navigates if within range.

Astar.cs

The way to navigate through obstacles finding the shortest path to the goal. Like the player.
Each enemy has this object.

Utility.cs

Random functions and structs to help with.

Behavior Tree

The main AI structure.
Defines different nodes. And how to run the tree.

flock agent

contains flocking behaviour and is inherited by whatever should be flocking.

flock group

instantiates flocking agents and controls them

flocking

contains flocking behaviour functions and etc.