**Descriptions of Scripts**

**Fong Zhi Zhong**

**Pathfinding Scripts**

**DijkstraAlgorithm.cs** – Uses Dijkstra Algorithm to Calculate Path

* Parameters (GameObject[] Graph, GameObject Source, GameObject Target)
* How
  1. Pass in Array of Game Objects as Graph.
  2. Pass in Start Point in Source.
  3. Pass in End Point in Target.
  4. Returns a Stack of GameObjects for movement.
* Used in
  1. AIPathfindingScript.cs
  2. Public Static Function. Use globally, in another class.

**AStarAlgorithm.cs** – Uses Astar Algorithm to Calculate Path

* Parameters (GameObject[] Graph, GameObject Source, GameObject Target)
* How
  1. Pass in Array of Game Objects as Graph.
  2. Pass in Start Point in Source.
  3. Pass in End Point in Target.
  4. Returns a Stack of GameObjects for movement.
* Used in
  1. AIAStarScript.cs
  2. Public Static Function. Use globally, in another class.

**Node Control Scripts**

**NodeScript.cs** – Optimizes Calculation of all Pathfinding Algorithms.

* How
  1. Attach to GameObject in Scene.
  2. Environment Linking for Blank Spaces
* Used In
  1. Empty.prefab
  2. PlayerSpawn.prefab
  3. EnemySpawn.prefab
  4. Bits.prefab
  5. Big Bits.prefab
* Why
  1. By reducing the amount of Nodes, by linking it up to the neighbors on Creation, I can reduce the amount of Nodes the Path find needs to calculate, and Increase Performance.

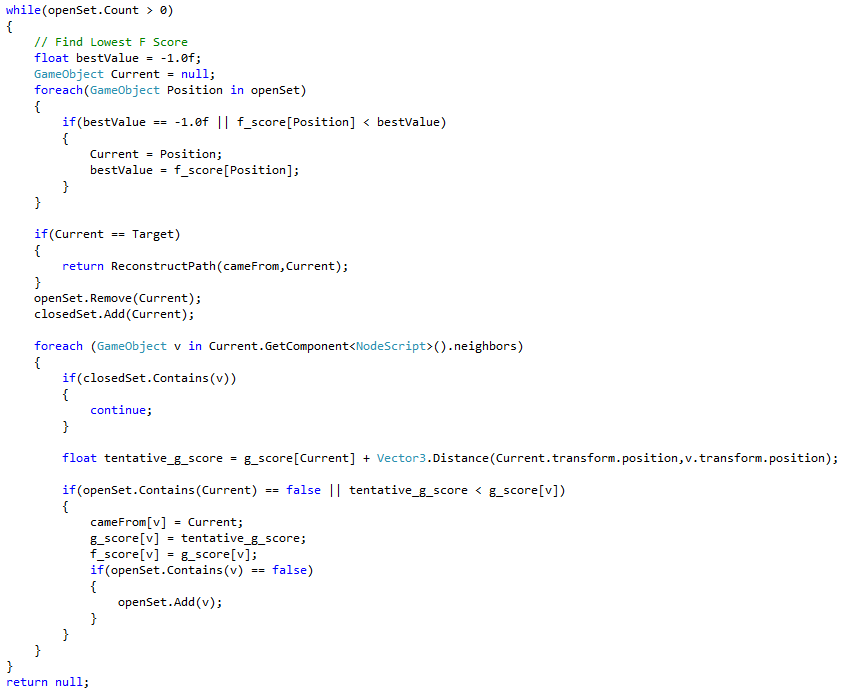
**CurrentNodeScript.cs** – Attach to Player/AI

* How
  1. Attach to Player/AI in Scene.
  2. Meant as a checking between Nodes.
  3. For Movement of AI.
* Used in
  1. Player.prefab
  2. AStarAI.prefab
  3. DijkstraAI.prefab

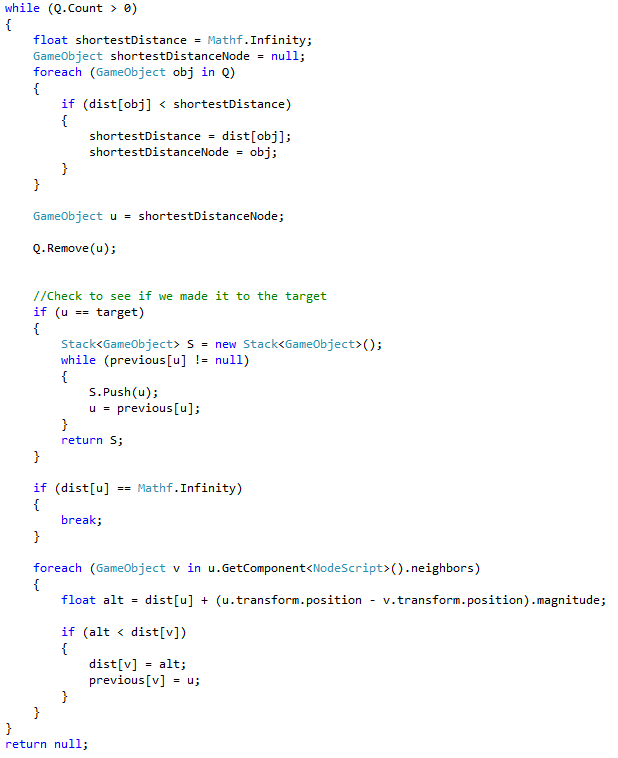
**Level Creation and Handling**

**LevelGenerator.cs** – Game Manager Instance

* How
  1. Loads all the levels, that have been created by the level editor, at Run Time
  2. Switches what is shown to the player, during the game, after a level.
  3. Handles changing from level to level.
  4. Creates environment at runtime.
  5. Optimizes performance
* Used in
  1. LevelGenerator.prefab
  2. Holds Map Generator List of Objects, Created at Run Time
  3. Holds AI List Of Objects, Created at Run Time
* Why
  1. Our Game manager handles the changing of levels.
  2. When the level is over, it changes to the next level through LevelGenerator.cs
  3. Procedural generation of Levels and AI
  4. Optimizes performance of the game by creating nodes all at once.

**A\* Algorithm Artificial Intelligence**

The above is A\* Algorithm

**Dijkstra Algorithm Artificial Intelligence**

The above is Dijkstra’s Algorithm

**C:\Users\KBI13XZhiZF\Desktop\HowToUse2.pngC:\Users\KBI13XZhiZF\Desktop\HowToUse1.pngEase of Use**

The above are examples in AIAStarScript.cs, and AIPathfindingScript.cs.

I believe that in coding, there must be consistency. I have implemented it in a way such that it is very easy to change the Algorithm around, just by changing the function name.

As they both give you the same result, it is easier to understand.

They both return you a stack, for you to manipulate the data after that.