Task 6   Spike: Navigation with Graphs

Context:

Moving agents that intelligently plan and navigate environments, both static and dynamic, is a common and useful problem to solve. The solutions are relevant in both a game context, as well as real-world navigation and routing problems. In this spike you will create a simulation where agents plan paths using heuristic search algorithms, and then navigate the environment using steering-based movement. The result is a solid basis for a tower-defence game, and can be used for a custom project that uses a combination of techniques.

Knowledge/Skill Gap:

Developers need to be able to create and navigation graphs for dynamic environments so that moving agents can move in an environment.

Goals/Deliverables:

Create a navigation graph simulation that demonstrates the following
- A game world that is divided into navigation tiles, and that supports a navigation graph structure.
- A path-planning system that can create paths for agents, based on the current dynamic environment, using cost-based heuristic algorithms.
- Demonstrate multiple independent moving agent characters (at least four), of different types (at least two), that are able to each follow their own independent paths.

Planning Notes:

- Suggest not using force-based movement. Just use simple constant speed movement from point to point.
- Reuse the box-world and path-planning code as much as you want.
- The different types of agents might be fast / slow, hunter / prey, ground / air-borne, etc.

Extensions:

- Have some agents destroy others when on the same tile – then update the 'cost' of the tiles for different agent types so they start to avoid places where their compatriots die a lot
- Try a different tile shape – hexes, triangles, or even cubes. Or for a **real** challenge: try an irregular tessellation with multiple tile shapes.