COMP110: Principles of Computing

Test Driven Development

Learning outcomes

By the end of this session, you will...

- asdf

FALMOUTH
UNIVERSITY

Pathfinding

The problem

- We have a **graph**
  - **Nodes** (points)
  - **Edges** (lines between points, each with a **length**)
- E.g. a road map
  - Nodes = addresses
  - Edges = roads
- E.g. a tile-based 2D game
  - Nodes = grid squares
  - Edges = connections between adjacent squares
- Given two nodes *A* and *B*, find the **shortest path** from *A* to *B*
  - "Shortest" in terms of edge lengths — could be distance, time, fuel cost, ...

Applications of pathfinding

- Journey planning (sat-nav, Google Maps, TheTrainLine, ...)
- Routing (utility pipes, circuit design, ...)
- Many applications in game AI
  - Non-player character AI
  - Mouse-based movement (e.g. strategy games)
  - Maze navigation
  - Puzzle solving

A* search

Idea:

- Expand out from the start node
- Let $g(n)$ be the length of the path currently found between the start and node n
- Let $h(n)$ be an **estimate** of the distance from n to the goal
- Prioritise nodes for which $g(n) + h(n)$ is small

Properties of A* search

- A* is **guaranteed** to find the shortest path if the distance estimate is **admissible**
- Essentially, **admissible** means it must be an **underestimate**
  - E.g. straight line Euclidean distance is clearly an underestimate for actual travel distance
- A* is an example of **heuristic** search
  - In AI, a heuristic is an estimate based on human intuition
  - Heuristics are often used to prioritise search, i.e. explore the most promising options first

Implementing A*

- Worksheet 5 on LearningSpace
- Deadline: **9th March at 6pm** (i.e. two weeks from today)