# Giải thích qua về code tôi viết:

* Chương trình của mình được chia làm 3 tầng:
  + Model: chứa dữ liệu của các đối tượng trong chương trình, ở đây là node, map, và 1 struct đặc biệt là Params chứa các thông số về chương trình
  + ViewModel: chứa các thông tin cần hiển thị ra giao diện, các lệnh mà giao diện thực hiện sẽ được xử lí tại đây
  + View: sẽ chứa giao diện của chương trình, giao diện này độc lập với code logic, được biểu diễn bằng cách binding với viewmodel
* Ngoài ra còn 1 số class khác bổ sung
  + Control: đây là các control bổ sung thêm để hỗ trợ cho giao diện
  + Converter: đây là các class hỗ trợ cho việc chuyển đổi dữ liệu khi binding
  + Helper: là các class hỗ trợ cho code logic và viewmodel
  + Algorithm: là các class về thuật toán, class AStarAlgorithm sẽ là class điều khiển việc tìm đường

# Chi tiết các class:

## Node

Đây là class chứa dữ liệu của 1 nút trong mạng, bao gồm các thuộc tính và phương thức

* X, Y: toạ độ của nút
* Type: loại của nút (normal/start/goal/…)
* Neighbors: các nút kề của nút đó
* Id: id của nút start và goal trong trường hợp sử dụng đa luồng
* Previous: nút liền trước của nút đó trong lúc tìm đường
* Fscore, Gscore: các đại lượng theo thuật toán A\*
* Reset(): khôi phục lại nút về trạng thái mặc định

## Map

Đây là class chứa dữ liệu của cả mạng, bao gồm

* nodes: danh sách các node có trong mạng. Ở đây không thể dùng mảng 2 chiều vì mình dùng toạ độ chéo theo hình lục giác nên các chỉ số các hàng và cột không đồng nhất
* start: các nút bắt đầu (vì mình sẽ phải làm đa luồng nên phải có nhiều nút bắt đầu và kết thúc)
* goal: các nút đích
* this[i,j]: sẽ đưa ra nút có toạ độ (i,j) trên bản đồ, vì không dùng mảng 2 chiều nên phải có hàm này
* GetStart, GetGoal, SetStart, SetGoal
* Clear(): xoá toàn bộ bản đồ
* Clean(): xoá toàn bộ bản đồ trừ các chướng ngại vật và nút bắt đầu/đích. Hàm này được dùng để xoá bản đồ nếu người dùng muốn chạy lại tìm đường trên bản đồ cũ
* RandomGenerate(): tạo ngẫu nhiên bản đồ

## Params

Có 1 vài thông số ở trong struct này

* MapHeight, MapWidth: chiều dài, rộng của bản đồ trên giao diện
* Scale: độ co của từng nút (cái này dùng để chỉnh kích thước của các nút trong mạng)
* Delay: độ trễ khi thực hiện việc tìm đường, tính theo mili giây. Cái này sử dụng để điều chỉnh tốc độ tìm đường
* SQRT3: căn của 3. Cái này dùng nhiều trong việc tính toạ độ nên sẽ đặt làm 1 hằng số. Cách tính toạ độ hiện tại hơi ngu nên chắc sẽ phải sửa lại trong các bản tới

## PriorityQueue

Đây là hàng đợi có ưu tiên

* comparer: hàm so sánh giữa các phần tử