Hand-in 5.

You are supposed to implement the A^*-algorithm for this assignment. Look at the Wikipedia page linked to from Fronter and any other links for more information.

We will be working on a 32 × 32 grid of tiles. There are two types of tiles: traversable and non-traversable. Details on implementation is really completely up to you.

However here is a suggestion for data structures:

```
class Tile
{
public:
   type Type;   // 0 = traversable, 1 = untraversable
   bool open = false;
   bool close = false;
   vector2i parent(-1,-1);  // Tile* parent = NULL is also a possibility
   float cost = 10;
   float f,g;   // h can be calculated

   // other stuff
};
```

You can store the tiles in a timemap class

```
class Tilemap
{
public:
  //  tiles[0][x], tiles[x][0], tiles[32][x], tiles[x][32]
  // should all be initialised as untraversable
  Tile tiles[32+2][32+2];
  vector2i start(1,1)
  vector2i goal(31,31);

};
```

Note that a tile doesn't know its own position. It is the position of the tile in the array which determines where it is put on the screen.

a) Choose data structures for your tilemap (or choose mine above) and implement functions for drawing the tilemap to the screen. Implement functionality so that you can hover over a tile with the mouse pointer and change its type by clicking. Choose a colour scheme so that open, closed and untraversable can be seen on the drawing of the tilemap.

b) Choose a movement scheme between tiles. My suggestion is that you are only allowed to move horizontally or vertically, but you can also choose an 8-way movement if you like.

Implement the A* algorithm and let it run on your tilemap. Let the algorithm run once per frame, so that it automatically updates the path as you add or remove obstacles on the tilemap.

c) Discuss how you could make your implementation more efficient. Which data structures would you choose in a more efficient implementation?