CMPUT 350 Assignment 3

Due: Wednesday Oct. 23, 22:00

Important: Read this text completely and start working on the assignment as soon as possible. Only then will you get an idea about how long it will take you to complete it.

If you worked on the assignment on your personal computer, copy your files to the undergraduate lab machine (e.g. uf13.cs.ualberta.ca) with scp and then test your code there as well before submitting it. Like the labs, we will be using this environment to test your code.

When done, submit your solution files on eClass:

```
README.txt GridPriv.h GridInclude.h Grid.cpp Problem2.txt/pdf
```

Your submissions will be marked considering their correctness (syntax and semantics), documentation, efficiency, and consistent coding style. Add assert statements to check important pre/post-conditions.

Finally, make sure that you follow the course collaboration policy which is stated on the course webpage. File `README.txt` must be completed!

---

1. [24 marks]
The objective of this assignment part is to produce a class that can store a map-like structure and perform pathfinding and connectivity operations on it.

Class `Grid` represents a rectangular tile map with octile topology, i.e. 8 compass directions using Euclidean distance. There are blocked, ground, or water tiles. Moving objects are not represented in the grid.

The moving object for pathfinding has a location (x, y) and size s. It is considered to occupy the tiles from x to $x + s$ in the x-axis, and from y to $y + s$ in the y-axis. Thus, an object with size 0 occupies one tile, and object with size 1 occupies a 2x2 square, and an object with size 2 occupies a 3x3 square. You are only required to support objects of sizes 0, 1, and 2.

Moves are permitted if and only if all tiles passed through during the move have the same tile type. For instance, if a 3x3 object is on ground tiles, it may only move diagonally if the 4x4 region it passes over is composed solely of ground tiles.

Similarly, a 2x2 object on water tiles may move to the east if the 3x2 region it passes over is composed entirely of water tiles.

Example:

```
  01234
0 wwggg   A 3x3 object centered in this 5x5 map may move west, south, or
1 ggggg   southwest, because it will pass only over ground tiles. However,
2 ggggg   it may NOT move northeast, even though it would wind up on a 3x3
3 ggggw   patch of ground tiles, because it would clip through some water
4 ggggw   tiles when moving
```

Note that in the above example, the object's initial location would be $(1, 1)$ and its size would be 2, meaning it occupies from 1 to 3 on both the x and y axises.

To test your implementation issue make which creates `testGrid`. See `TestGrid.cpp` for documentation. `exampleGrid` is our solution that should run on lab machines.

Do not include a `main()` function in your submitted code. If you want to test your function separately, define your `main()` function in a separate `.cpp` file and link with the corresponding `.o` file.

Also, private class members you might want to add to class `Grid` must be defined in file `GridPriv.h`, which you will submit together with `Grid.cpp`.

You may use any C/C++ standard libraries you want, such as STL, and Boost. Feel free to use C++17 features that `g++` on the lab computers support. We will compile your code using `-std=c++17`.

---

2. PRA* Map Abstraction [16 marks]

   (a) Apply the clique-based abstraction used in PRA* (**ai-part2, page 12**) to the graph below. For this, scan the nodes from **top-left** to **bottom-right** - row by row, identifying new maximum size complete subgraphs of size **up to 4** that can be abstracted into new nodes in the next abstraction level - starting from the current node and scanning its unvisited neighbourhood. Continue this process until only one node remains, showing the resulting abstraction levels and indicating which nodes got abstracted by what node in the process. **Use uppercase letters A,B,C,...** to name new abstract nodes and explain all steps.

```
a-b-c-d-e-f-g
|   |   |   |
h-i j-k l-m-n
|   |       |
o-p-q-r-s-t-u
```

   (b) Assuming the original graph size is $N = 4^n$ and clique-based abstraction being able to **always group 4 nodes** to form a new abstract node in each step:

      (i) how many nodes does the k-th abstraction level consist of ($k = 0$: original graph with 4^n nodes, $k = 1$: first abstraction level with ? nodes, etc.)?

      (ii) which abstraction level contains exactly one node? Explain

      (iii) assuming we reach one node at abstraction level $n = 2u$ for an integer u, how many nodes does abstraction level u consist of in relation to the original number $N = 4^n$? Simplify your answer and explain your steps

(iv) Prove that the clique abstraction used in PRA* preserves connectivity, i.e., given A,A' and B,B', where A' is the abstract node for A and B' is the abstract node for B, there is a path from A to B in the original map if and only if there is a path from A' to B' in the abstract map. **Note**: your proof needs to show both implication directions.