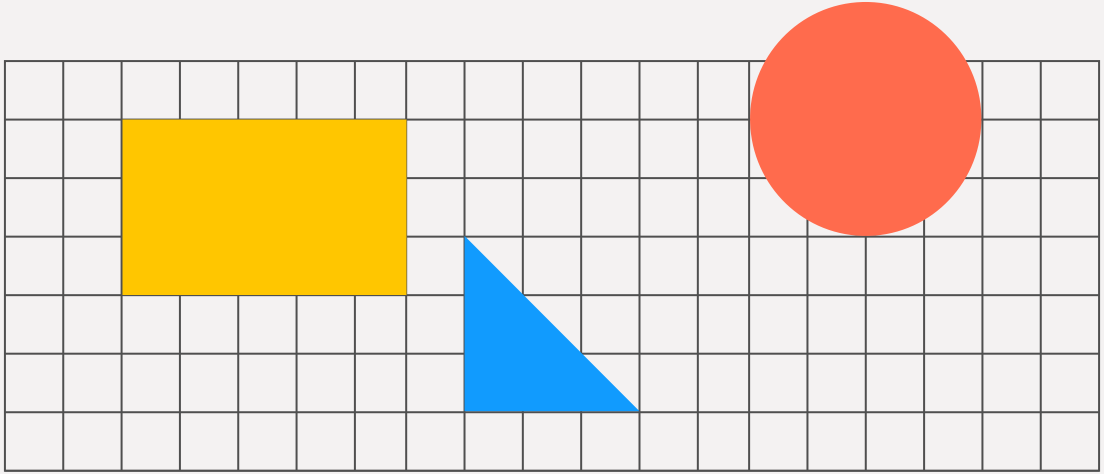
Lee's Maze Router

Aly Elaswad	900225517
Mohamed Alaa	900212213
Omar Ganna	900222646

CSCE 3304
Spring 2025
The American University in Cairo
Presented to: Dr. Mohamed Shalan



Design and Implementation

Test cases

Problems/ Limitations

Contact

Design and Implementation

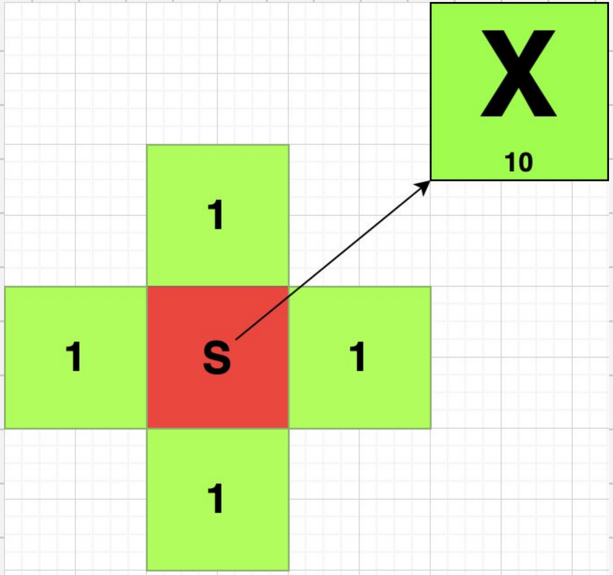
Data Structures

- **3D Grid: vector<vector<vector<int>>> grid represents the routing space**

  .
- **Net Representation: *all_nets, net_names, net_name_grid* to track pins and net identifiers.**
- ***Point* Struct: Encapsulates grid coordinates and routing cost.**

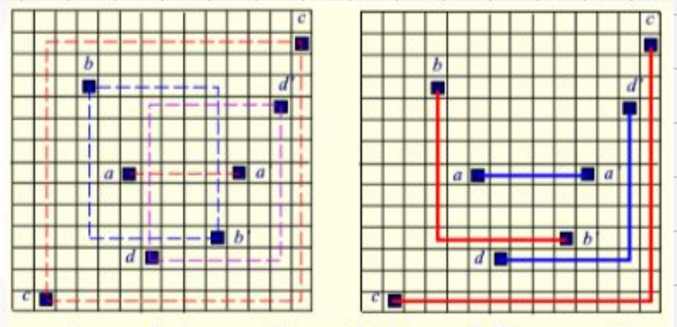
Routing Strategy

- **BFS - based: wave propagation**
- **Movement cost depends on layer (e.g., layer 0 favors horizontal moves).**
- **Considers x/y moves and via (layer change).**
- **Backtrack to mark routed path with net identifiers.**

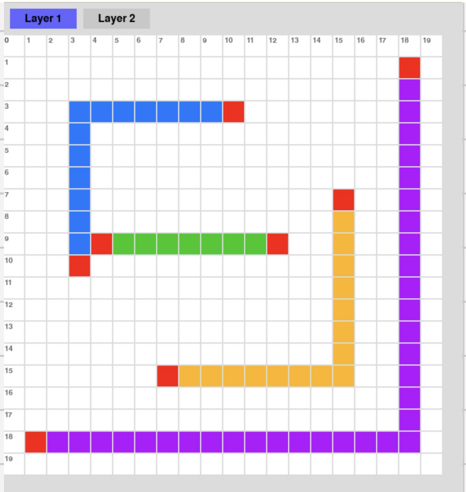


Net Ordering Heuristic (Bonus)

- Implemented a net ordering heuristic to optimize the routing process.
- We prioritized nets based on bounding box heuristic And if they have the same bound box we choose the shortest net first
- This approach helped reduce routing conflicts and improved overall success rate.
- We used an example provided in the slide to confirm correctness



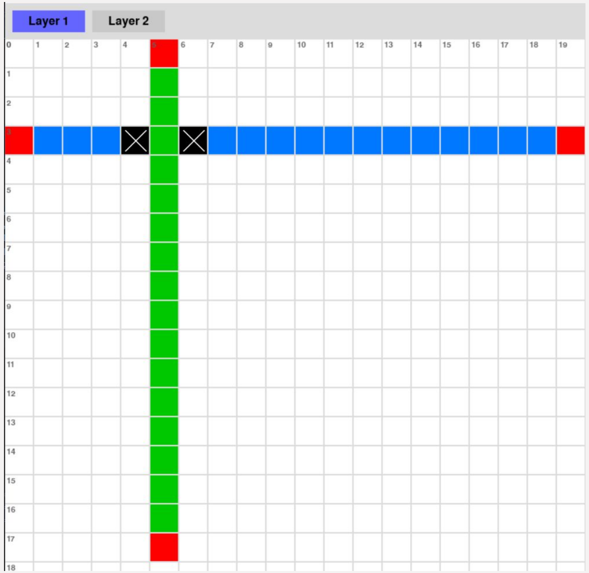
routing ordering: a (0) –> b (1) –> d (2) –> c (6)

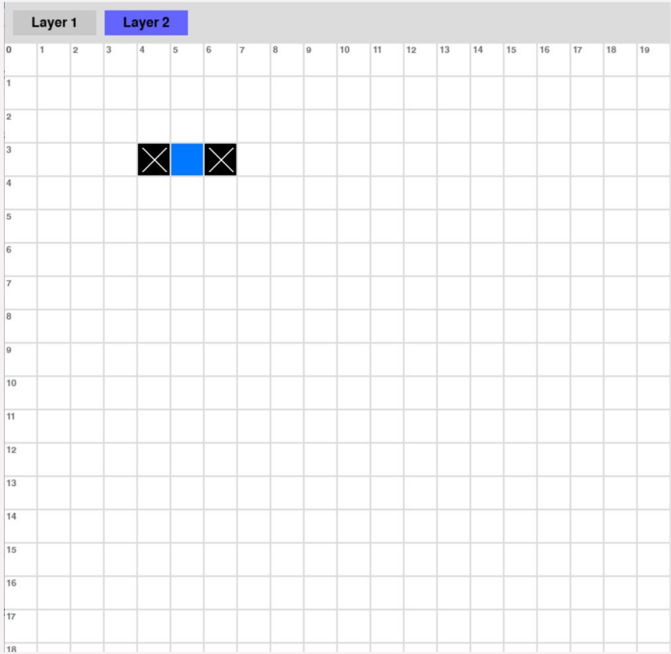


Test Cases

Test Case 1

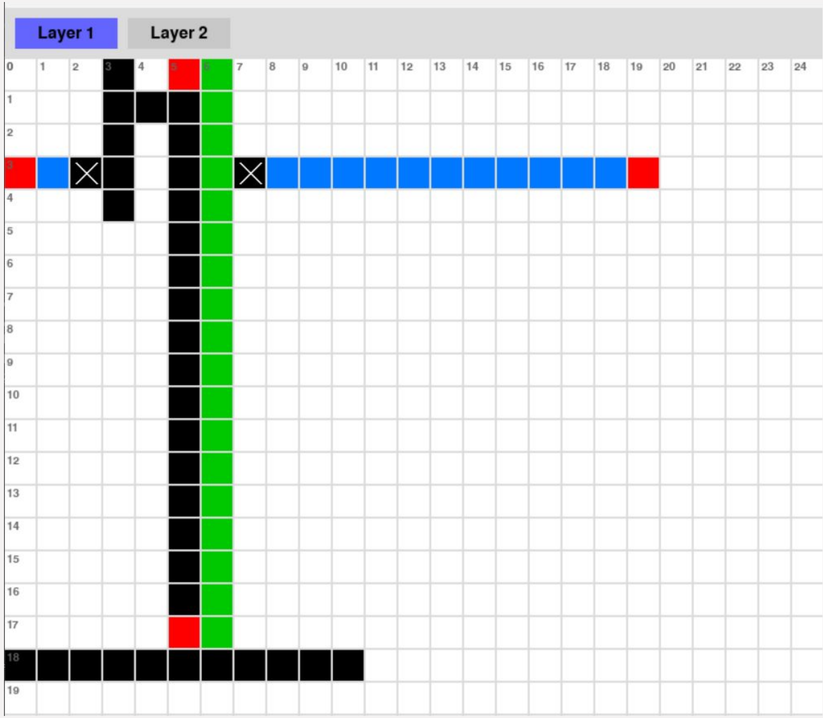
Input:
20x20
net0 (1,0,3)(1,19,3)
net1 (1,5,0)(1,5,17)

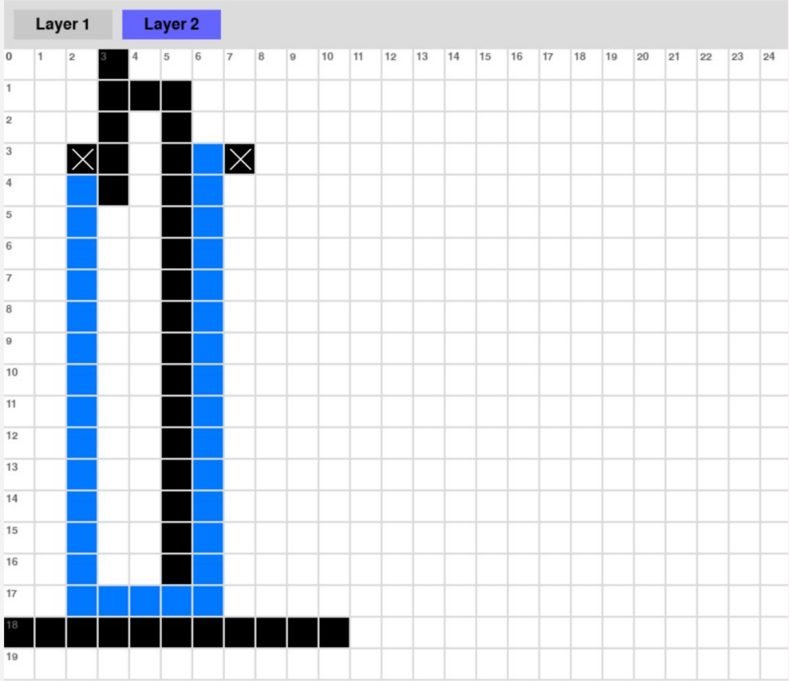




Test Case 2

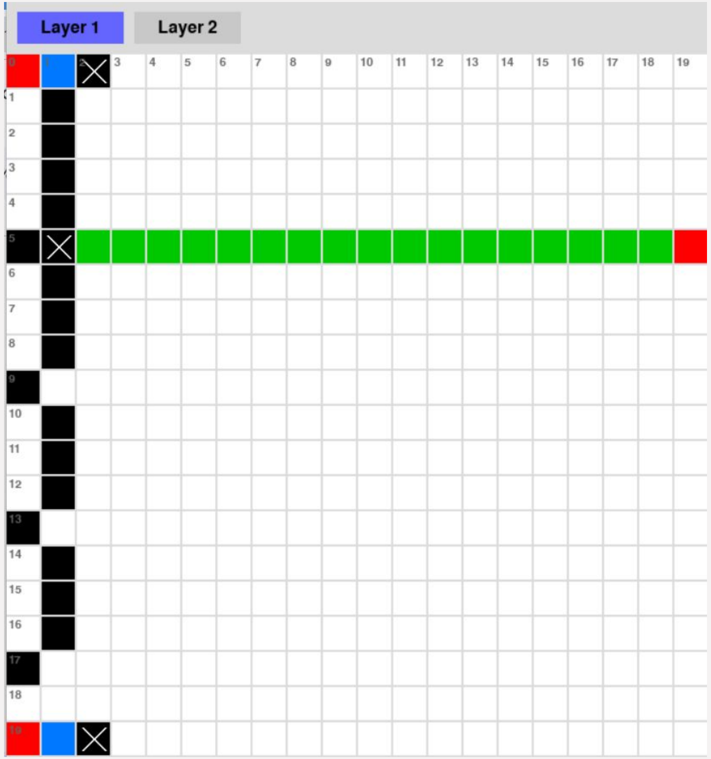
Net Input:
40x40
net0 (1,0,3)(1,19,3)
net1 (1,5,0)(1,5,17)

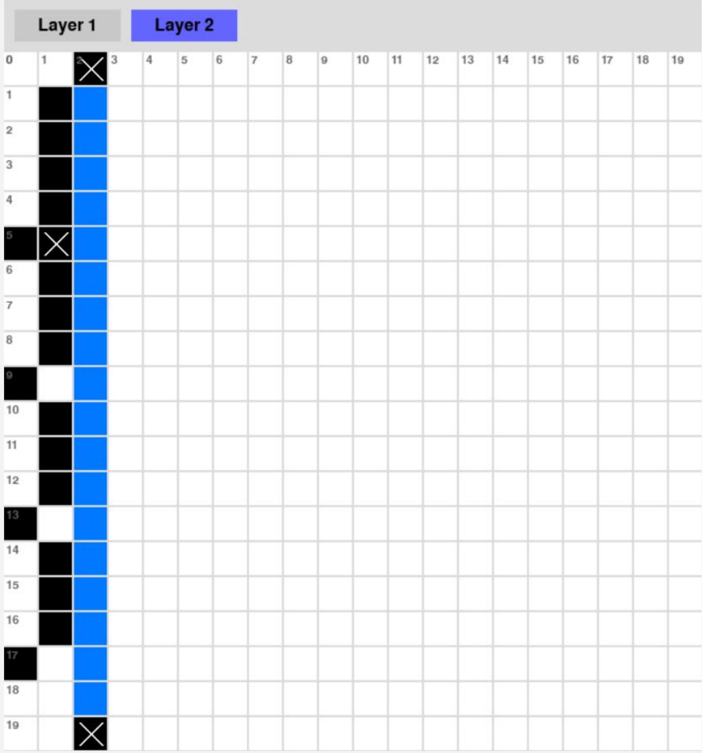




Test Case 3

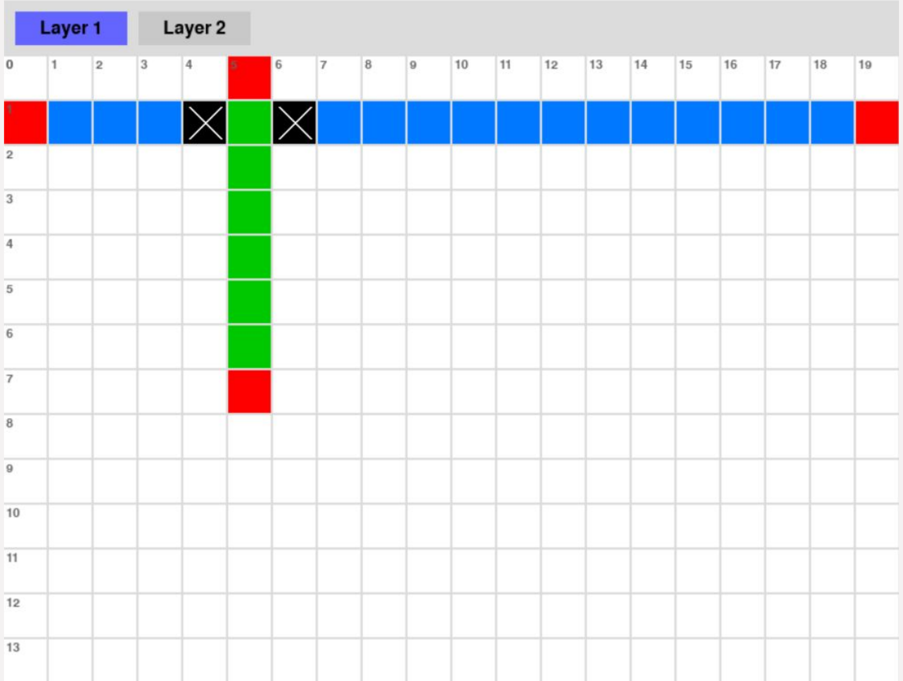
Net Input:
20x20
netA(1,0,0)(1,0,19)
netB(1,1,5)(1,19,5)(2,1,5)

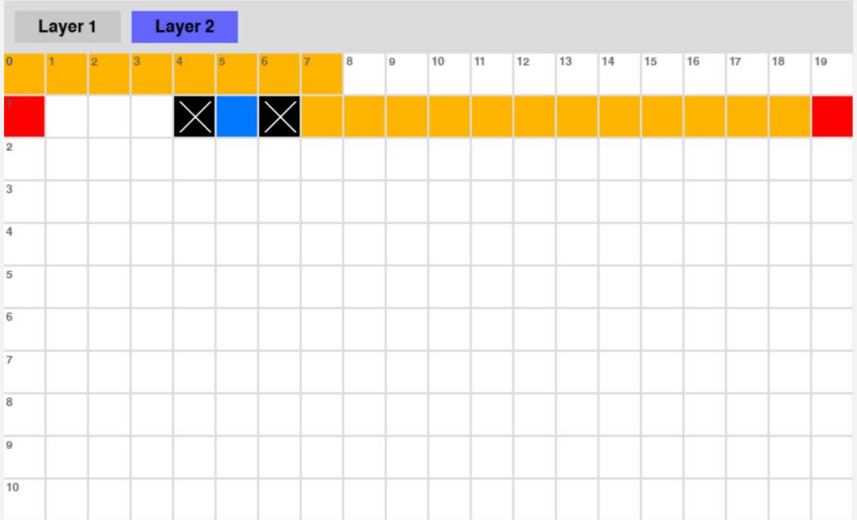




Test Case 4

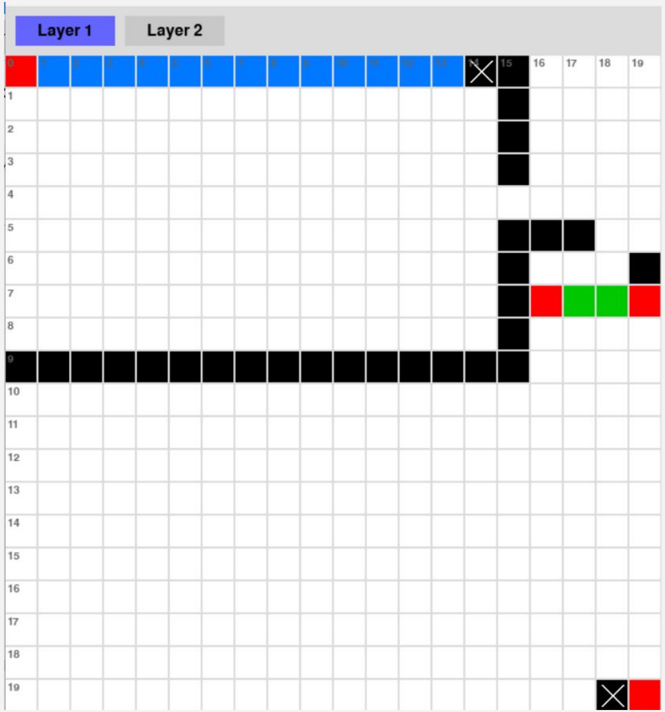
Net Input:
20x20
netA(1,0,0)(1,19,0)
netB(2,0,0)(2,19,0)
netC(1,5,0)(1,5,7)

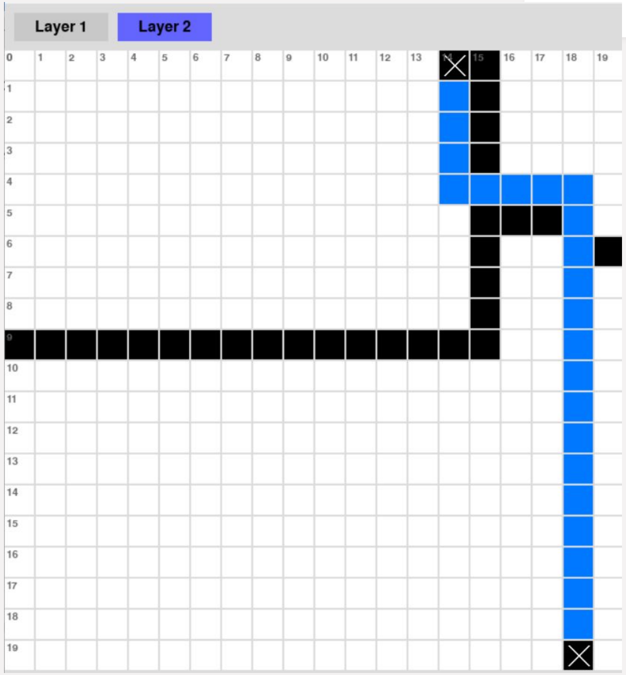




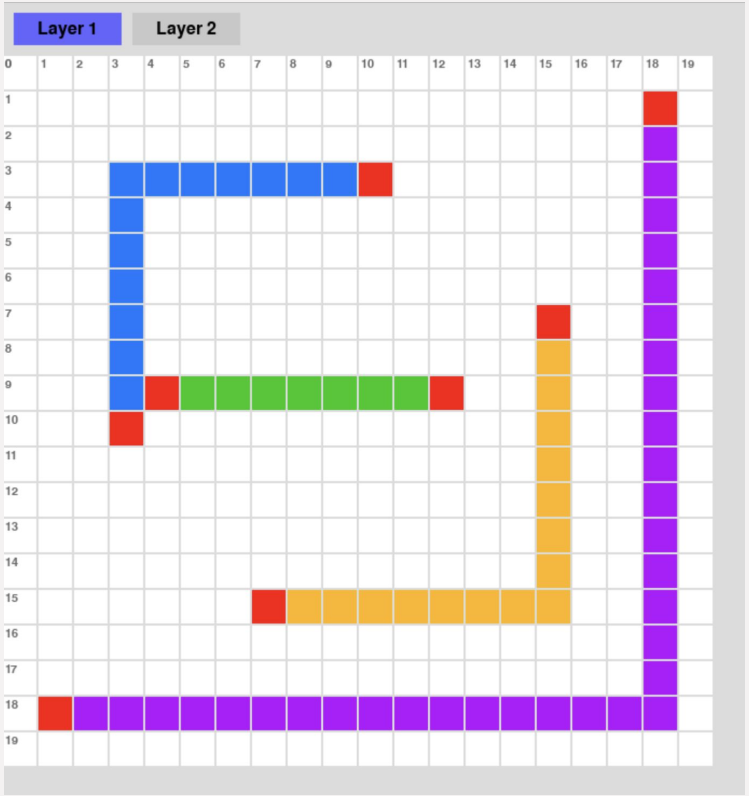
Test Case 5

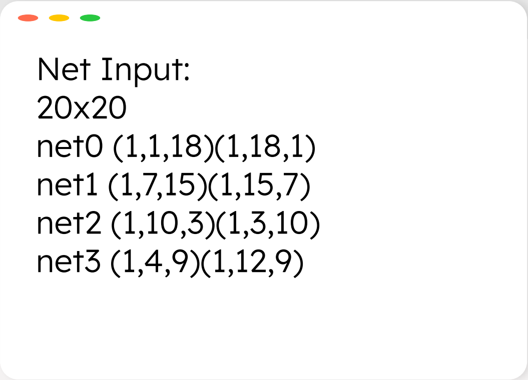
Net Input:
20x20
netA(1,0,0)(1,19,19)
netB(1,16,7)(1,19,7)





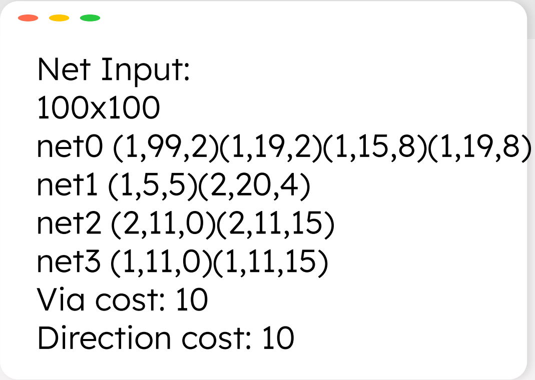
Test Case 6



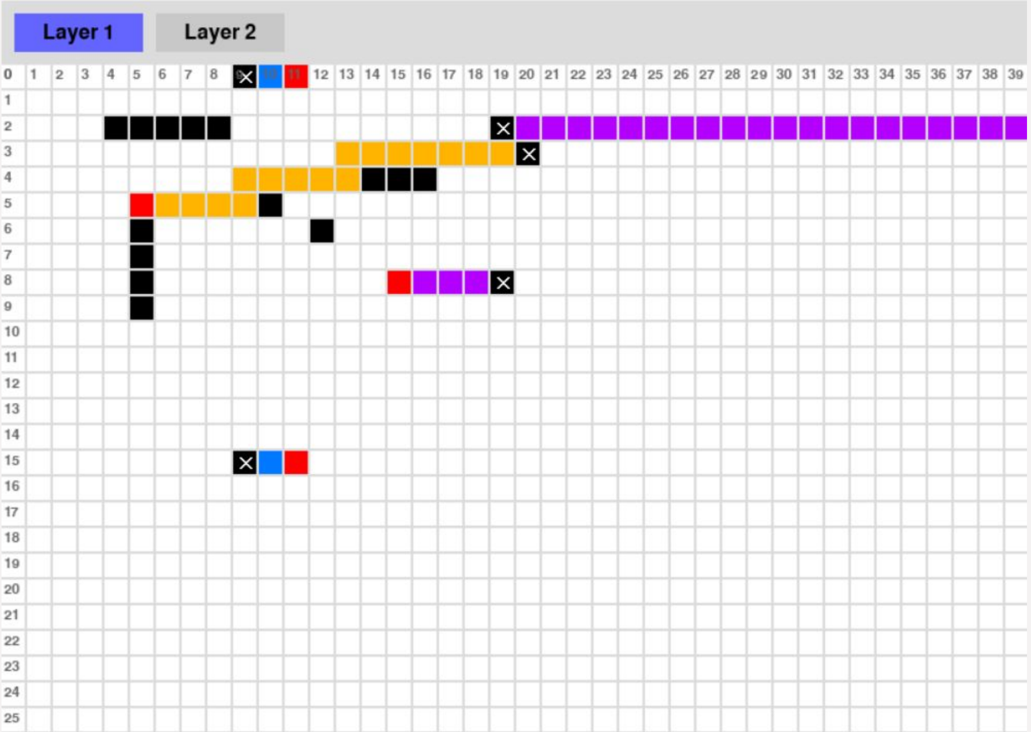


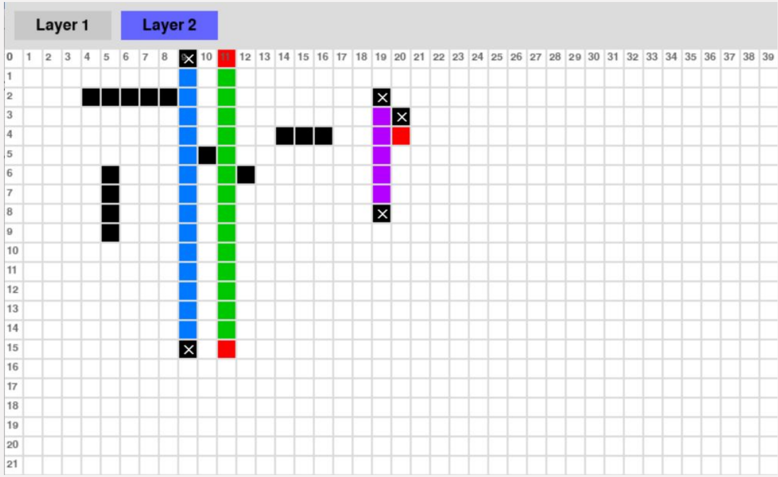
Net Input:
20x20
net0 (1,1,18)(1,18,1)
net1 (1,7,15)(1,15,7)
net2 (1,10,3)(1,3,10)
net3 (1,4,9)(1,12,9)

Test Case 7



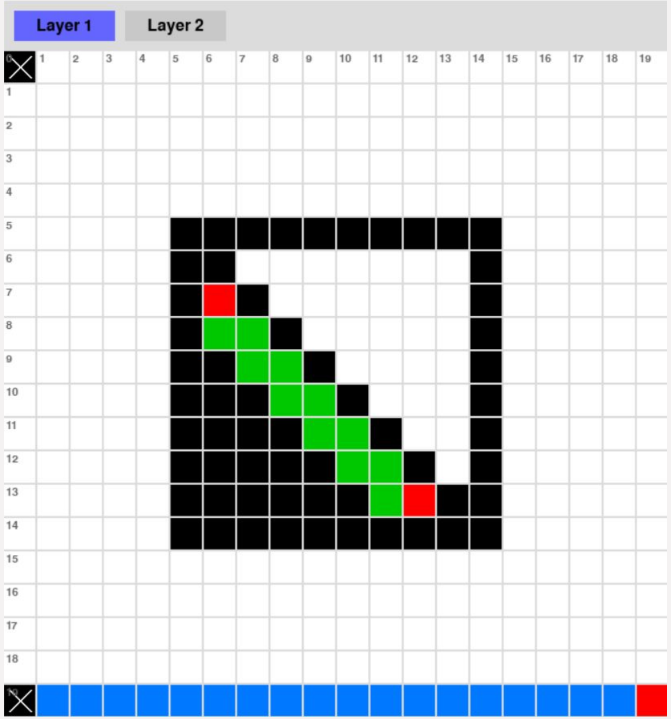
Net Input:
100x100
net0 (1,99,2)(1,19,2)(1,15,8)(1,19,8)
net1 (1,5,5)(2,20,4)
net2 (2,11,0)(2,11,15)
net3 (1,11,0)(1,11,15)
Via cost: 10
Direction cost: 10

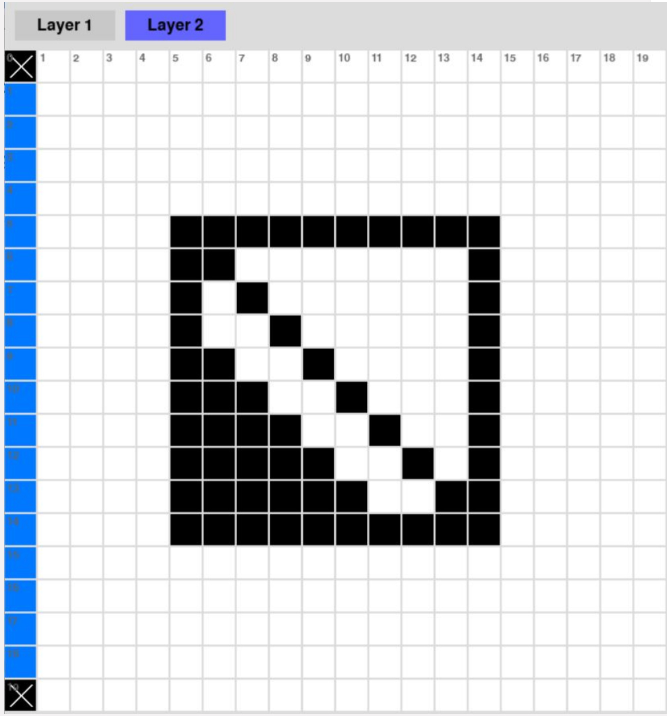




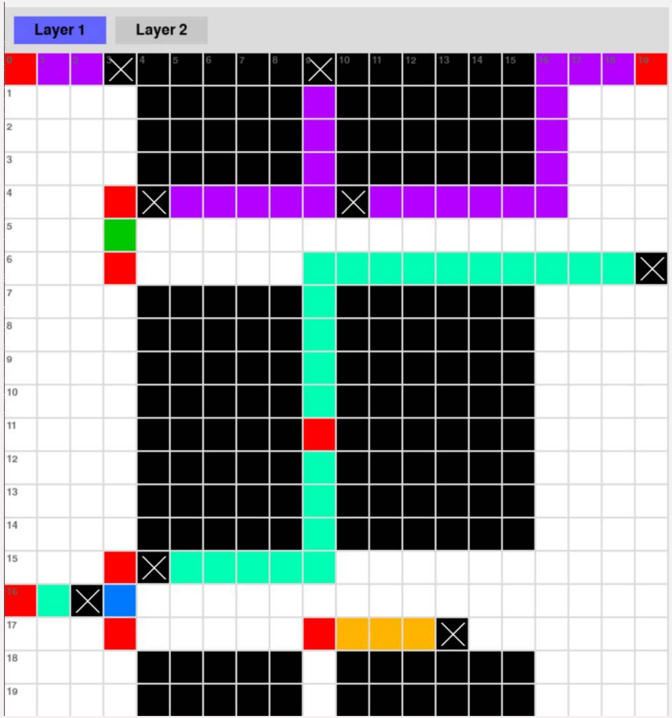
Test Case 8

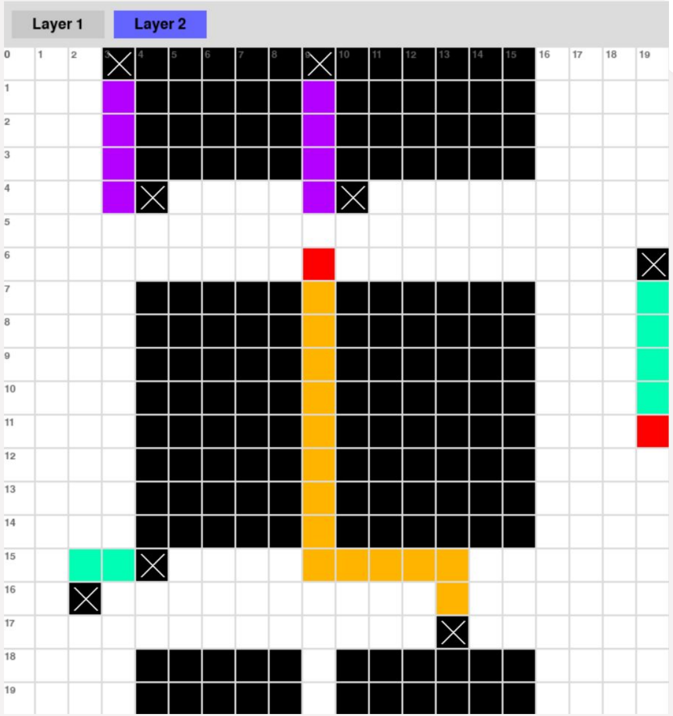
Net Input:
100x100
net0(1,0,0)(1,19,19)
net1(1,6,7)(1,12,13)
Via cost: 10
Direction cost:10

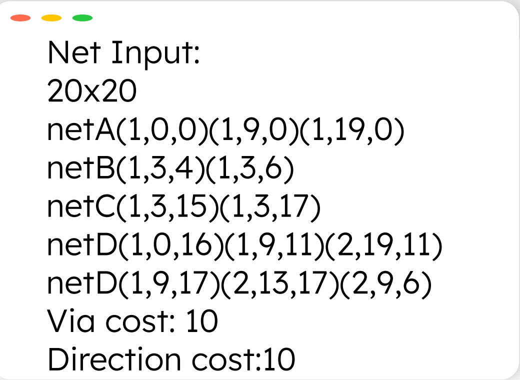




Test Case 9



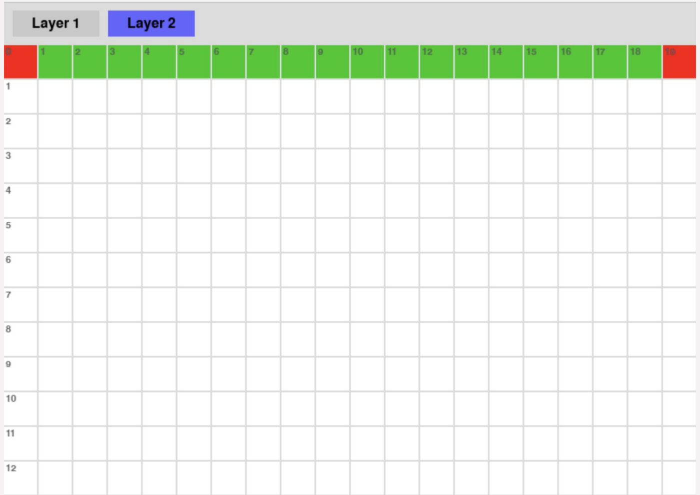




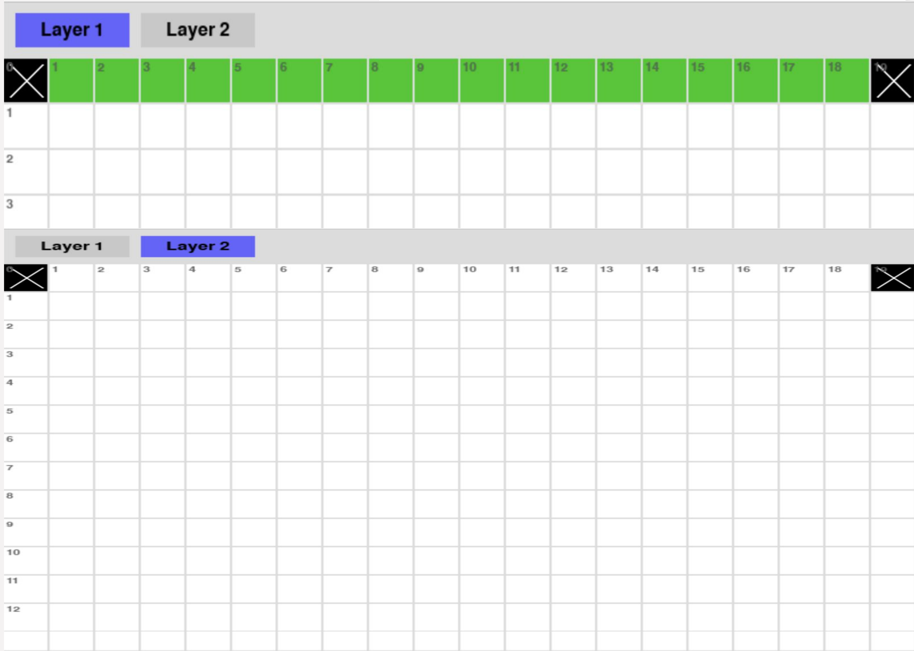
Net Input:
20x20
netA(1,0,0)(1,9,0)(1,19,0)
netB(1,3,4)(1,3,6)
netC(1,3,15)(1,3,17)
netD(1,0,16)(1,9,11)(2,19,11)
netD(1,9,17)(2,13,17)(2,9,6)
Via cost: 10
Direction cost:10

Test Case 10

Net Input:
20x20
net(2,0,0)(2,19,0)

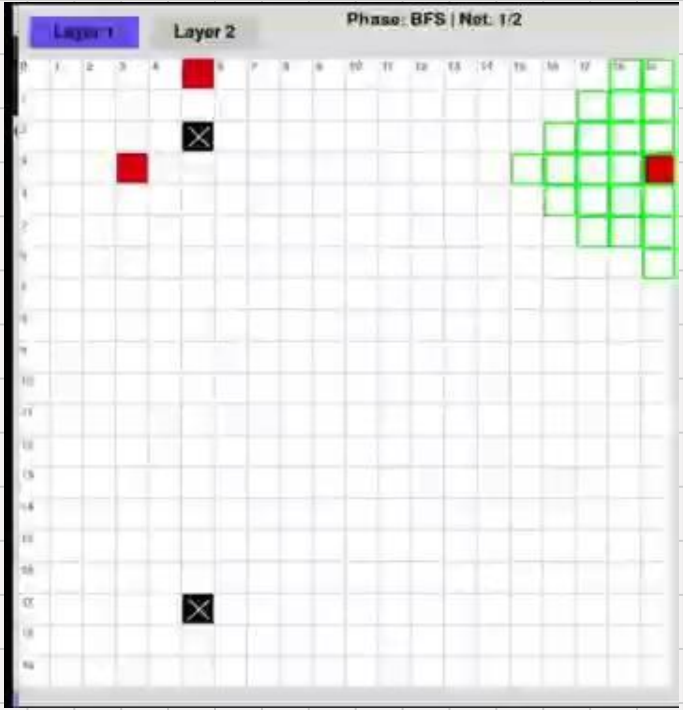


Non-preferred Direction cost: 20
Via cost: 500



Non-preferred Direction cost: 500
Via cost: 20

Dynamic Visualization



Problems & Limitations

- **Dynamic visualization is not merged with the final implementation**
- **When the grid is 1000x1000 the output is printed in the output.txt but is not shown correctly in the GUI**
- **Did not add rip-up and reorder, instead we try to solve routing problems using net ordering heuristics**