ASSIGNMENT 01: MAZE GENERATION & SOLVING

Brock Davis

GOAL & APPROACH

- Maze Generation
  - Size up to about 20000px x 30000px
  - Recursive Backtracking (Stack) in Java
- Maze Solving
  - Size up to about 13000px x 20000px
  - Right Hand Following in Java
  - Stack to keep track of path
- Custom Point class was made

```
while points not empty:
    int[] possibles = getPossibles()
    if possibles.length > 0:
        int choice = random_choice(possibles)
        makeMove(choice);
        points.push(current.copy())
    else:
        current=points.pop()
```

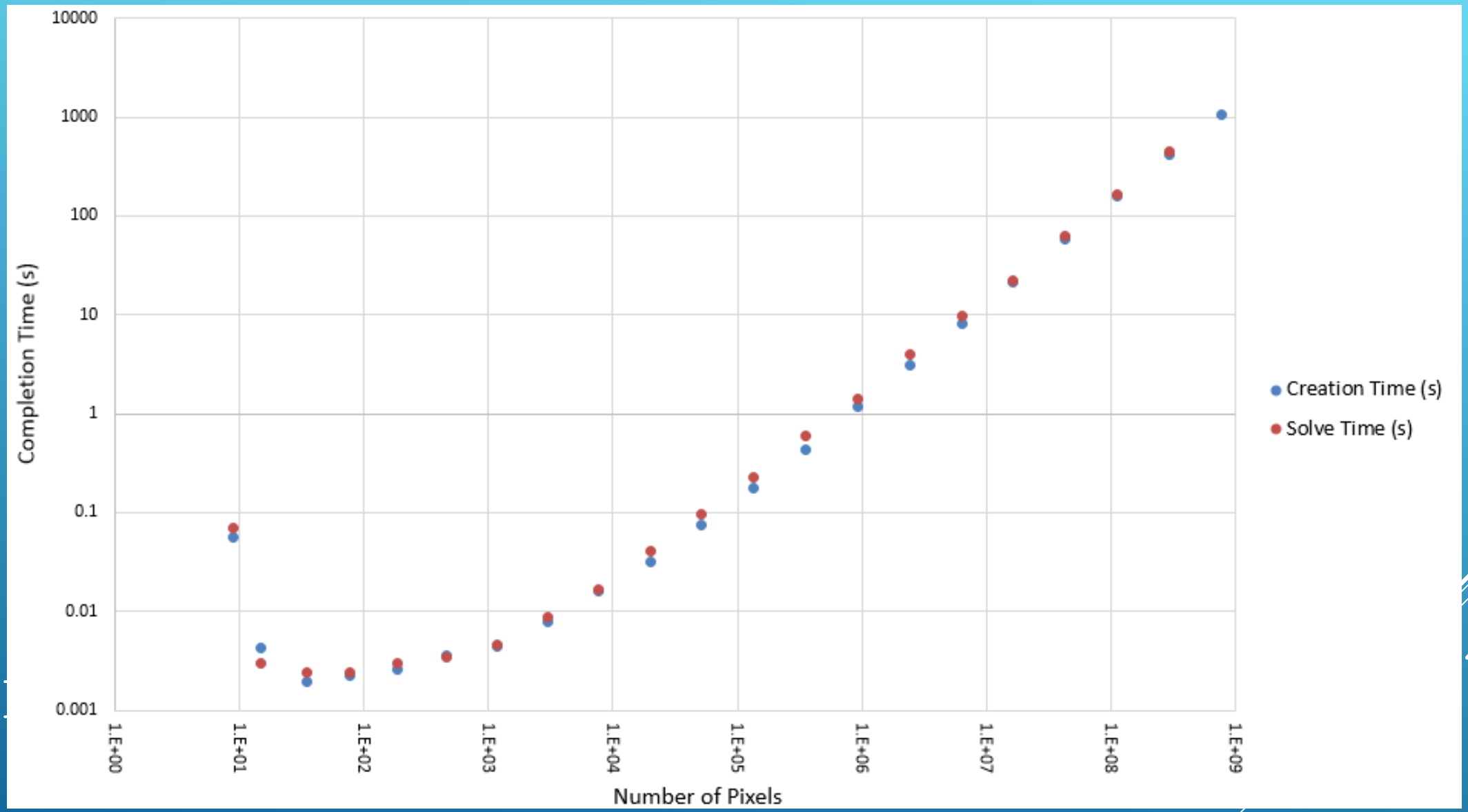
MAZE GENERATION

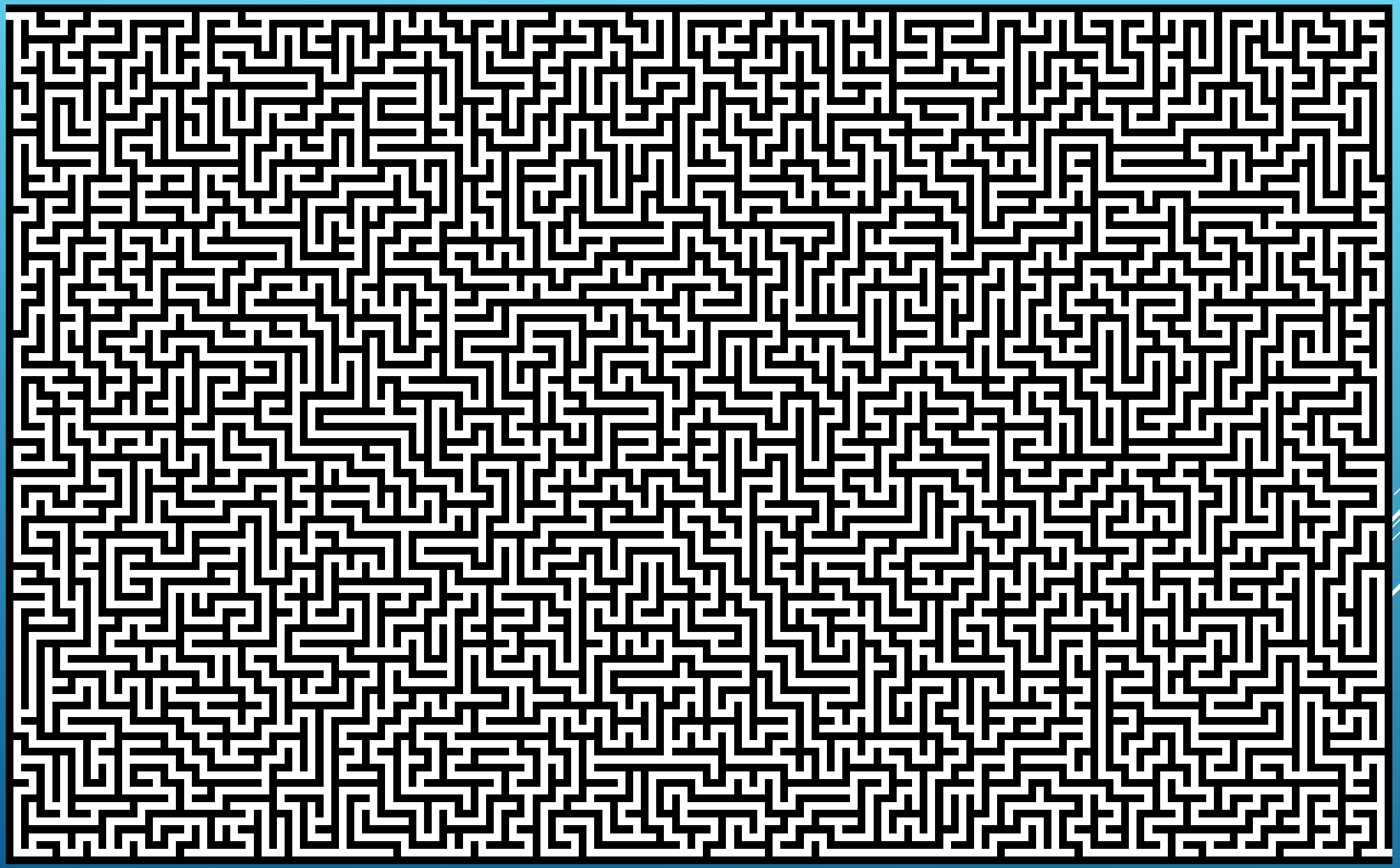
```
while current != final_point:
    boolean[] possibles = getPossibles();
    if (possibles[right of direction])
        direction = right of direction;
    else if (possibles[direction]);
    else if (possibles[left of direction])
        direction = left of direction;
    else if (possibles[backward])
        direction = backward;
    makeMove(direction);
```

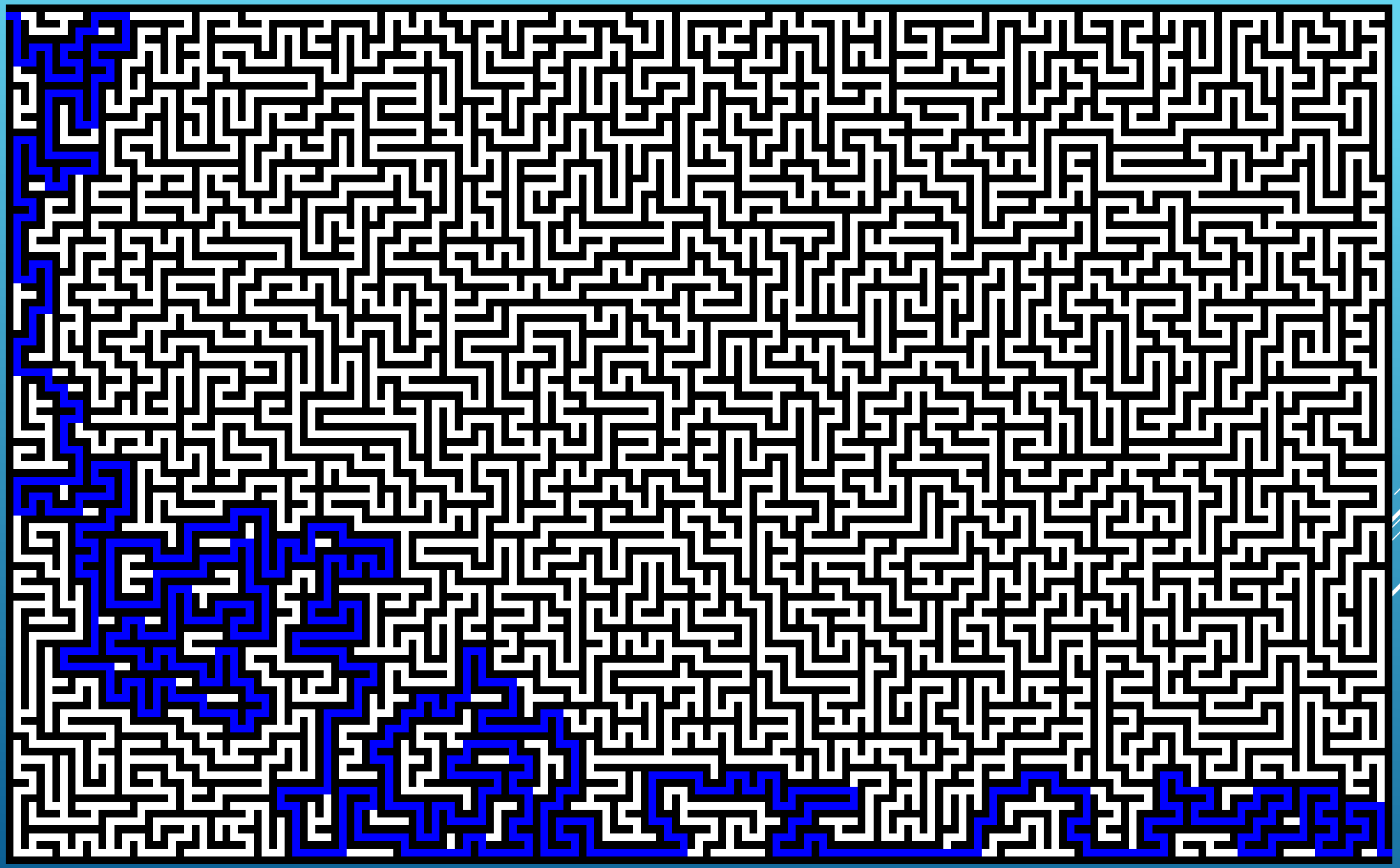
MAZE SOLVING

```
void makeMove(int direction):
    change x, y based on direction
    if current in points:
        remove last point from points
    else:
        add current to points
```

MAZE SOLVING (CONT.)


10000
1000
100
10
1
0.1
0.01
0.001
Completion Time (s)
1.E+00
1.E+01
1.E+02
1.E+03
1.E+04
1.E+05
1.E+06
1.E+07
1.E+08
1.E+09
Number of Pixels
Creation Time (s)
Solve Time (s)





RESULTS

- The goal was accomplished
  - Mazes up to 20000px x 30000px were created
  - Mazes up to 13000px x 20000px were solved
- Maze creation was faster than solving
  - Mazes took about 10% longer to make
  - Creation also took more memory
    - maze20 could not be solved because of insufficient memory
  - Much of time spent was i/o of images

RESOURCES

- Maze Creation

https://youtu.be/WDONIYY8ljY

- Maze Solving

https://youtu.be/9LDbFS1Qh1I