System Improvements:

We also investigated the possibility of, instead of pushing the correct path onto a list, using nested calls of amb to better utilize the call-with-current-continuation. It would have been a better functionally programming style compared to resetting the value of the state variables directly. However, since we generate random mazes each time Maze Solver runs, we do not know the exact number of steps required to solve that given maze. Therefore, not only that would result in nesting many amb calls within each other, but present another logical puzzle that we would need to do more research on.

**Conclusion:**

In conclusion, Maze Solver is just one application of some powerful functional programming paradigms. Maze Solver is not perfect and there are areas where we can improve it. It is intended to be purely educational and experimental and not production. On a different note, learning functional programming and applying to our Maze Solver has been a fun experience. Over the course of this semester, our team developed a love-hate relationship with functional programming, mainly due to the simple fact that it challenged every assumption we had about writing software. Ultimately, we hope this report will inspire programmers get out of their comfortable --Object-Oriented Programming-- zone to explore new programming paradigms.