

# COMPSCI 130 Project 2: Report

Feras Albaroudi, falb418, 606316306

## Summary

All tasks were attempted. This project taught me how programming can be done using a limited set of instructions, and it demonstrated how a simple programming language can be designed. The most challenging aspect of this project was writing the code for the Roomber creature as only a limited set of instructions were available. Given more time, I would have liked to experiment more with creating creatures.

## Creatures

### Looper

This creature's behaviour is fairly straightforward - if there's an enemy ahead of it it will infect them, if there's an ally or a wall it will twist, and if there's nothing ahead of it it will hop.

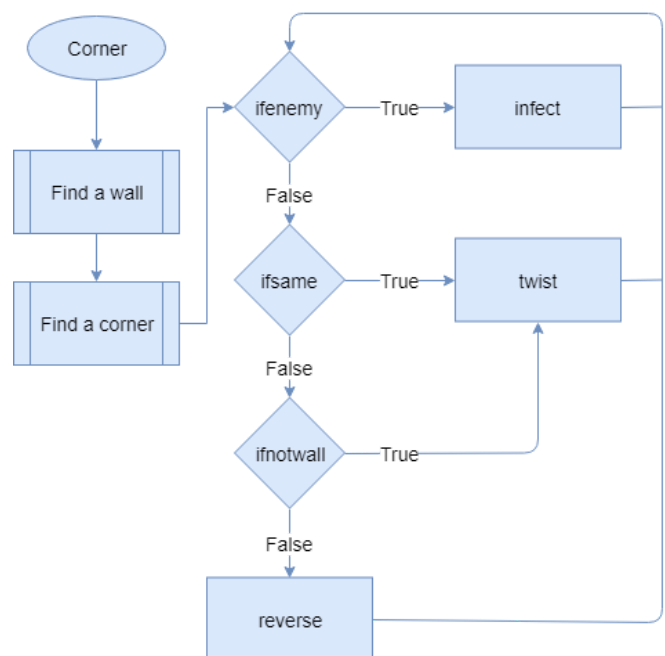
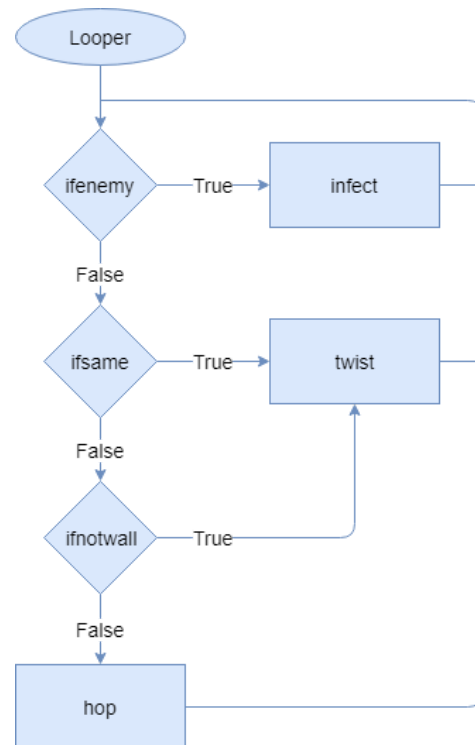
This creature's simple behaviour makes it very efficient at infecting other creatures as after every action it checks if there's anything to infect.

### Corner

The logic of this creature is more complex than that of the Looper creature. It will attempt to find a corner of the map, and, after doing so, it will stay there and continuously twist, infecting anything in front of it.

The strategy behind this is that this creature should only be vulnerable from 2 sides as its "back" will be covered by a corner or an ally.

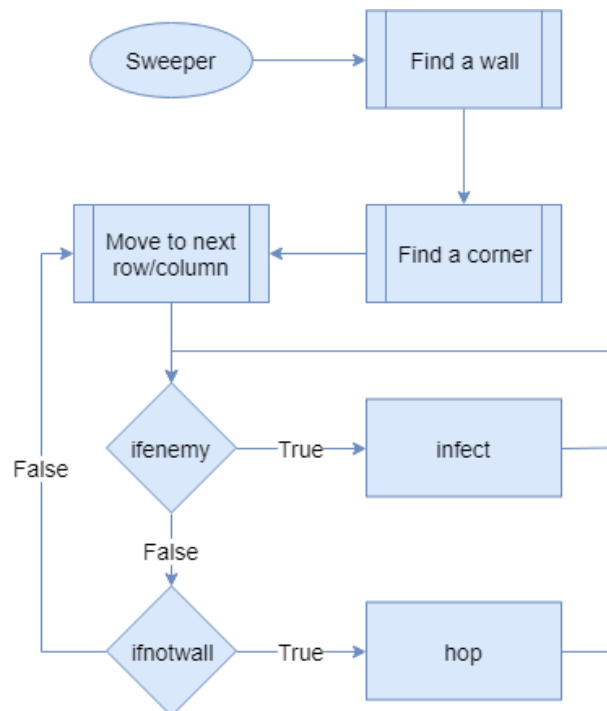
In reality, due to the limitations of the instructions available, there is no way to allow this creature to directly turn left-to-right, so it ends up facing a wall half the time, leaving itself vulnerable; thus, it is not very effective at infecting other creatures.



## Sweeper

This creature takes the idea behind the Roomber creature which traverses the whole world and adds the ability to infect anything ahead of it.

It's main weakness is the number of actions it takes to turn around when moving onto the next row/column in the world as this leaves it vulnerable to creatures like the Looper which prioritizes infecting enemies near the world's walls.

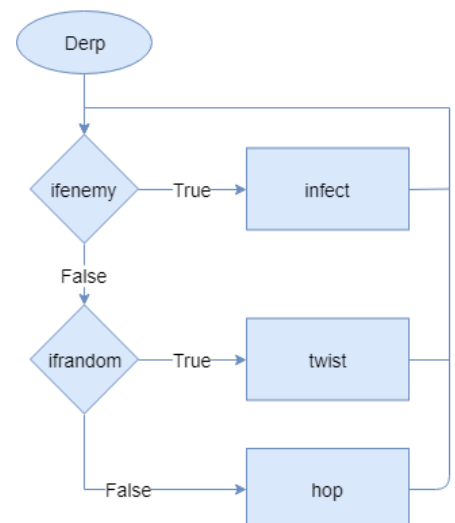


## Mimic

This creature takes the logic of the Looper, Sweeper and Corner creature and picks one at random on it's first move. Thus, in a group, this creature is able to utilise the strengths of the creatures it mimics to compensate for it's weaknesses.

## Derp

This creature randomly decides between a hop or a twist each turn. It is not effective at infecting other creatures, but it is useful as a dummy target for testing other creatures.



## Conclusion:

Creatures which prioritize a single behaviour appear to perform best as it is difficult to construct creatures with more complex behaviour efficiently using the limited instruction set available. However, from experimenting it was also found that creatures which are higher in the turn order tend to perform better.