

Postmortem Document

Part A

My Algorithm for the Attacker Behavior (Diagram is located at the bottom)

```
Method: public int update(Game game, long timeDue)
```

(The only method that I used. The other Methods were left blank).

Update Method:

The algorithm that I chose to implement was fairly simple within the update method. I decided that it was best that I risk going after the things that granted me the most points rather than going for the regular pills the majority of the time. So, what I did was in the update method I had the program check whether individual defenders were in their vulnerable states in an if block. Depending on what happens during the last tick we either do the default action which is eat normal pills or we most likely go into the portion where we search for the nearest pill. We achieve this by using a for loop and iterating through the list of Power Pill locations, grab the closest one and send our gator to it from there we then decide whether to eat right away if there are no enemies in the vicinity or wait if they're close enough and sneakily eat it as it tries to get us and then eat them.

Part B

The Good the Bad and the Ugly... of our algorithm

Successes:

It took several tries of switching around if statements and their locations in the code, swapping Nodes in for loops, and statements involving Pac-man movements across nodes in the maze to finally get to the point that I am at right now. As far as performance I would say that overall it does ok as we get well above the goal score. My Pac-man manages to get to power pills and to vulnerable monsters in a somewhat efficient manner by waiting for the defenders to come close enough (sometimes in bulk) to eat the power pill and eliminate them quickly; if they are far away we prioritize gathering power pills first while obtaining the normal pills. Sneaking behind some defender to gather points or having the defender follow the attacker to a power pill, as well as the order of which defender to approach first in the if block is another mild success behind strategy that I had in mind and obtaining above a 6400 score.

Failure(s):

However here is where things would go wrong. We do get a score above 6400 but we don't keep going because my gator dude easily becomes cornered whenever it's just chasing a defender or when he is quite close to the lair. Also, whenever the attacker goes for vulnerable defenders it's susceptible to running into a revived defender. A possible solution could be creating a conditional within my if statement block that would take that into account, however it would negatively affect the score so I thought it would be best to leave it alone. Another failure is that my gator doesn't take efficient enough pathways when eating normal pills which

is what snowballs into the other problems listed above. A possible solution would possibly be tweaking the default movement a bit with maybe with a for loop like done for the power pill but I'm not sure what kind of consequences that would bring as far as the score average.

Part C

Reflection on the Project:

This project gave me some excellent practice with critical and logic/algorithmic thinking. It took a lot of focus, guidance from TA's, and breaking things down into small components to achieve an algorithm that I was somewhat satisfied with. I was also a bit different than the other projects we were working on, so it was overwhelming at first with all the different functions and classes. The project was also beneficial in understanding the hierarchy of classes in project and how they interacted with each other. It gave me greater insight on inheritance and polymorphism. I also gained greater understanding with Array Lists and their functions in Java.

My algorithm in a flow diagram:

