Andrew Huynh
SID 25607104
PacPack Phase 1

June 18, 2018
Took some time to look over the code briefly and figure out how to make things work. I'm using the features and weights. I thought about q-learning at first but after looking at the code I don't think I can do that effectively. I tried a greedy version that goes after the closest dot. It does pretty well but there are obviously times where this can be improved on. (~20 minutes)

I watched a couple games. Going greedy misses better paths a lot of the time. The biggest "mistake" is when Pacman leaves three or more dots on the left of the game to go clear the right side. After clearing the right side, Pacman wastes a lot of time going back to the left side for those remaining dots. (~15 minutes)

June 19, 2018
In situations where there are two dots to the left of Pacman and three dots to the right, or vice versa, we want to go to the one with three. I tried to implement something for this but Pacman kept going in circles. (~1 hour)

I also tried a more general version for the problem I noticed yesterday. Again, Pacman just went in circles. (~1 hour)

I tried a heuristic for preferring dots on the left. This helped quite a bit in solving the problem. (~5 minutes)

I also tried adding a heuristic for dots on the bottom. (~1 minute)

June 20, 2018
I spent a lot of time just messing with weights and scaling things to see what works and what doesn't work. (~2 hours)

I added a heuristic for the next dot closest to a dot. I decided that this heuristic should be used later in the game as to not hinder exploration. (~30 minutes)

I added a line where if there are only three dots left, then to just go after the closest dot with no heuristics (~1 minute)

Andrew Huynh
SID 25607104
PacPack Phase 2

June 21, 2018
I copied my Phase 1 code to Phase 2. It took me a while to figure out how to find which food the teammate was planning to get. (~30 minutes)

I filtered out the dots that the teammate planned to get. I watched a game and Pacman moves randomly after a while. Fixed this by making Pacman go after the closest dot when there are no more filtered dots left. (~1 hour)

Spent some time re-writing the code so that it works properly with the new information from the teammate. (~1 hour)

I spent a lot of time just messing with weights and scaling things again to see what works and what doesn't work. For some reason, getting rid of the next closest dot heuristic makes it better. (~1 hour)

Andrew Huynh
SID 25607104
PacPack Phase 3

June 21, 2018
I copied my Phase 2 code to Phase 3. I started by writing the code to broadcast my future actions. I'm going to broadcast my future actions every set interval of turns. (~30 minutes)

Watched a game and surprise, there was a ghost. Added a feature to not go next to a ghost. Also updated my broadcasting to broadcast whenever Pacman dies to a ghost. (~5 minutes)

I don't have any ideas about what else to add. I watched a bunch of games where my agent plays with itself. There are a lot of times when Pacman goes into a dead end path and the ghost follows Pacman in. This either wastes time to see if the ghost leaves or the ghost just kills Pacman, also wasting time. Not sure how to tell Pacman not to do this. (~15 minutes)

Added a line where Pacman does not do the reverse action when he runs away from a ghost. This helps Pacman get around ghosts more often. Also rebroadcast future actions when Pacman runs away from a ghost. (~10 minutes)

Tried to make it so that Pacman goes for food farthest away from the teammate. Pacman ended up moving wildly. (~30 minutes)

Tried to make Pacman run away from the teammate every 50 turns to try to explore. Didn't work out very well. (~30 minutes)

I've decided to improve on the running away from the teammate thing and see how that works out. After a certain amount of turns, Pacman will go after the dot that is farthest away from the teammate. As the game progress, the amount of turns needed increases and Pacman will spend less and less time trying to move away from the teammate. After 330 turns, I think, Pacman will stop trying to get away from the teammate and play normally. (~30 minutes)

NEW STUFF

Andrew Huynh
SID 25607104
PacPack Phase 3

June 28, 2018
So a lot of things happened in real life and I didn't have enough time to work on the project until now (yay slip days). Since it's been a while I've decided to start Phase 3 over.

To make starting easy, I'm focusing on my moves first without teammate broadcasts. I'm going to the closest dot with a heuristic of the two next closest dots after collected in succession. After watching a couple games, Pacman will often ignore two close by dots and instead go to a tight cluster of dots. (~35 minutes)

To make sure this doesn't happen, I instead am now going to make the heuristic just the next closest dot after. After watching some more games, Pacman will often ignore a dot right next to him and instead go for two dots that are right next to each other but far away from Pacman. It turns out that both plans have the same cost plus heuristic. I weighed the initial distance cost to the closest dot by two to make it so that Pacman will want to go to closer dots. (~40 minutes)

I copied over my previous future actions function. The game crashed occasionally due to timing out. I found that when making the future actions list, if there is a ghost in the way, Pacman cannot path through the ghost. Instead, the action that brings Pacman closest to his next destination is "Stop" and will be stuck in a loop. Trying to filter the actions to not include "Stop," Pacman instead loops between "East"-"West" or "North"-"South." I added a clause where if the next step is a "Stop" action to not add the "Stop" action to the future actions list and to just return the future actions list so far. (~1 hour)

However, doing this brought a new problem. What if Pacman is at a dead end with a ghost in front of him. There's also the case of if Pacman needs to make a new future actions plan that has a ghost right next to him in the way of the path he wants to take. The only sensible action then is either "Stop" or move around the ghost. I added a clause to deal with this. (~20 minutes)

I put in broadcasting and filtering out food that the teammate plans to eat. Just copied code from my previous Phase 3 file. (~2 minutes)

To try to save computation time, I saved my future actions as an instance variable and just went through it to get the next action. However, this didn't interact nicely with the rest of the code.

Namelty, whenever the next action from this runs into a ghost, I have to try to get around it and I'm no longer on my previously planned path. I tried a lot of stuff like making new future actions whenever this happened. Unfortunately, occasionally my games would crash with illegal actions performed. Instead, I'm computing the next action at every step while leaving my future actions plan alone, even if I diverge from it a little. I only update it when important things happen such as Pacman dies to a ghost, Pacman east all his planned dots, or the teammate eats one of Pacman's planned dots. (~1 hour 20 minutes)

My score on the autograder is about 860 with one game not finishing. Pacman and the teammate keep running into the ghost which is blocking the way. Also two of the sanity check games time out.

June 29, 2018

I noticed that the ghost moves away from Pacman sometimes when Pacman goes towards it. Instead of trying to avoid the ghost, I'm just ignoring it unless my next move immediately kills me. The hope is that the ghost will move out of the way often enough for this to be a good idea. Turns out this works pretty well. Dying also isn't so bad. This brings Pacman back to the start and Pacman can go after those dots closer to to home that got ignored in the beginning.

I thought back to my trying to get away from the teammate thing I tried a while ago. This time I'm using boolean flags to help me keep track of when to move away from the teammate instead of trying to use set intervals of time to start moving away. After lots of rewriting code to work with this and numerous game crashes, I ended with the idea of initially getting as far from the teammate as possible and keep this up until half of the food is gone. Now, only go after the farthest food from the teammate whenever Pacman dies and after go for the next closest dot using the heuristic from earlier, the dot closest after that one. Towards the end of the game, Pacman will stop going for the farthest dot from the teammate and just go for close dots to end the game. (~2 hours 30 minutes)

The idea here is to split and conquer. There's two of us and only one ghost. If we split up, the ghost can only effectively go after one of us. This leaves the other one completely free to go after their dots. In an even better case, If the ghost is in the middle constantly switching from both Pacmen to go after, this leaves both of us free to collect all the dots we want.

The autograder score now is around 745 with all games completing. I watched some of the games and saw that as Pacman went for the farthest dot that Pacman walks by so many dots that would be very easy to pick up. So I added a clause where if Pacman is currently next to a dot, to go get that dot and continue on as normal. This brings the autograder score down to 660 and proves to be a great optimization.

THOUGHTS

This project was overall fun, a lot more so towards the end. I'm more of a single problem solver so an open question project like this one with tons of ideas at once was hard for me to start. At first, I was pretty uncomfortable trying complicated steps and just defaulted to using the features and weights from phase 1. It worked for phase 1 and phase 2 well but when it went to phase 3 the scores became really bad. At that point, I had little idea what I could try to get Pacman to path better and smarter. Some general guided suggestions might help people in getting started and more comfortable with the project. Having the Piazza collaboration was somewhat helpful in helping me move forward with the project and introducing me to things that I could do.