CS 4444 Project 4 Report

Group 2: John Daciuk, Tanmay Chopra, and Griffin Adams

December 7, 2019

Contents

1 Introduction

In Lunch as a Team Sport, family members operate in a fully cooperative environment to eat their respective lunches within a time limit whilst avoiding the greedy onslaught of monkeys and geese who wish to steal their food. Our strategy leverages the cooperative nature of the game by dynamically switching between individualistic behavior (eating our own food) and group enhancing behavior (distracting monkeys by repeatedly flashing food). Our distraction strategy is very effective and we demonstrate empirically that in denser animal environments, this leads to a net positive impact on overall family eating performance.

1.1 Problem Statement

In Lunch, a family seeks to eat its food in limited time. Because the goal is to maximize family eating, Lunch is cooperative. Helping another family member to eat is generally as good as eating oneself. The only obstacles to eating are the m monkeys and g geese that roam the 100 x 100 map, attempting to steal food. As such, all strategy ultimately reduces to outsmarting the animals.[1]

1.2 Challenges

Our distraction algorithm, which we will later demonstrate, renders near optimal play with respect to the monkeys fairly straightforward for a family controlled by a single player. It is sufficient to simply rotate the distractor and eat with the other members after patiently letting the distractor absorb all the monkeys. Thus, *the core challenge of Lunch is the coordination between family members who lack online means of sophisticated communication*. Moreover, while mitigating geese harassment is far less manageable, any successful geese strategy would also have to involve synchronized team work. Set in this context, we enumerate the most salient aspects of the coordination challenge:

1. Because real time communication is limited, we as a class should agree on conventions. Yet we had many competing ideas vying for attention, many of which in direct contradiction. Furthermore, many proposed conventions would actually be deleterious without widespread adoption. For example, if one player is distracting where others are eating, both eating and distracting become more difficult. If a partly adopted convention confuses players about their positions or roles, members may stay in a state of limbo all game.

2. The goal of lunch is to score points by eating; however, eating attracts animals. The proper **balance** may be elusive because it's difficult to know when waiting to eat will skew the distribution of monkeys in one's favor. After all, without food in hand, the monkey distribution is determined by the food of other players.

3. Although distracting large numbers of monkeys is fairly simply, knowing when to distract is far more technical with a host of potential upsides and downsides to weigh. Eventual convention consensus tells us that distractors will be in the bottom right quadrant, but the mere presence of a distractor does not indicate how well the job is being done, nor does a lack a monkeys near the distractor. If a current distractor is ineffective, every other family member's eating is disrupted, and perhaps one should help distract or else assert one's command of the position to dissuade others from trying. Moreover, for all but the shortest games the burden of distraction will have to be spread to more than 1 player else the distractor would not get a chance to eat.

4. Location is often critical to eating efficiently. The corners are prime locations because animals can not access them from all sides; however, with potentially more than 4 family members, not everyone can have their own corner. While eating close to another member gives animals the leverage to harass

[1] For a full problem description see: http://www.cs.columbia.edu/ kar/4444f19/node20.html

two players simultaneously, spreading the family out requires a larger area to be clear of monkeys if members are to eat in peace.

1.3 Solution Summary

First, after recognizing that most conventions were too intricate to garner widespread support, we suggested a simple idea on piazza that captures the essence of what virtually all groups wanted: *distractors shall act in the center.* With such a protocol, players could assume free space in the corners while also being able to make reasonable real-time judgements about the roles of others. To follow up a majority piazza vote, during class we suggested a referendum to formally settle the issue of convention once and for all. Other students responded to the simplicity of declaring a dedicated distraction area, and the vote was to set the bottom right as that territory. While it's easy to imagine further convention improvements, we believe that our contributions helped avert the disaster of no convention at all.

With a convention in hand, the success of our player reduces to three components: eating, distracting and role switching.

1.3.1 Eating

We seek to eat in areas of low player and monkey density. Our first eating location is at (-50,0), in-between the two left-most corners. We noticed that other groups always eat in corners, so we help everyone by finding a niche spot. Dynamically deciding where to initially go is problematic since other players are either on the move or waiting for others to stake out territory first. Every 150 seconds we consider updating our location. To update, we first check if any corners are free of family members, and if so we go to one. Otherwise, we check the far side in-between corner locations and go to them if available. If we *must* eat near other family, we go to the corner of least monkey density, averaged over the past 20 seconds.

Once in location, patient eating begins in an attempt to mitigate monkey harassment. We measure the number of monkeys leaving and entering our sphere of influence: if that ratio is above some threshold, we wait to eat so as to let nearby monkeys wander out of our vicinity and hopefully to the distractor. The threshold itself is a function of the state of the map. As is the case at any point in the game, during eating it is of critical importance to not get food stolen. To avoid being ransacked, we simply put food away if 3 monkeys or 1 goose is close enough to potentially steal on the next time step.

1.3.2 Distracting

To distract we go to the center of the distracting quadrant, avoiding the critical mistake of attempting to distract from the corner itself [2]. Once in place, we work to get food out whenever it is safe to do so. With food in hand, we take a random safe step; that is we step to a location that is far enough from geese and monkeys to guarantee we will not have food stolen. If a safe step is not found, we simply wait. Assuming that the eaters are patient enough to let monkeys leave their orbit, such a simply strategy is remarkably able to sweep up over 300 monkeys, rendering the eating corners completely monkey free.

1.3.3 Changing Roles

Because of the coordination challenges discussed, deciding when to distract and when to stop is one of the most technical aspects of Lunch. There are many edge cases that are important to get right, but for now we leave them for the body of this report and present a broad outline.

First, it would be a grave mistake to spend the game roaming the map in limbo. To avoid such a blunder, we only make a role change if we've been in our current rule for at least 10% of the total time duration.

[2] The reason why the absolute corner is a mistake will be explicated later

Otherwise, we will generally switch from eating to distracting if no one else has taken the job or the current distractor is ineffective. Likewise, if we are the distractor, we stop if someone else has entered the quadrant and distraction is effective. Also, if we manage to eat 99% we trigger distraction mode but stay at our eating outpost. If time is short and we still have eating to do, we always exit the distraction role for good in order to finish our food.

2 Convention Discussion

Theoretically speaking, Lunch as a Team Sport would be optimally solved by a central oracle who controls all family members concurrently. Therefore, perfect, centralized coordination provides a theoretical upper bound on the overall performance of any set of individually operated players.

Since we could not explicitly coordinate through shared code, we had to resort to decentralized, or implicit, coordination, to close the gap between theoretically optimal performance and imperfect coordination. To aid implicit coordination, we as a class came up with convention proposals to define a set of mutually agreed upon expectations of behavior. An effective convention must address the following goals and constraints:

1. Adherence improves overall team performance.
2. It is comprehensive enough to be unambiguous for all possible configurations of the game.
3. It is simple enough to be willingly adopted by as many team members as possible.
4. It does not breakdown in the event that not all members adhere.

With this formulation, it is clear that 1-2 are somewhat at odds with 3-4. The theoretical optimal performance would come from a central oracle who dictates each player's moves. Such obedience would be comprehensive yet impossibly brittle and unattractive for adoption as it would stymie individualistic behaviors.

So, to balance these competing but necessary ingredients for an effective convention, we proposed the following convention:

If you are going to distract, please do so in the center of the park.

As mentioned in the intro, this attracted widespread support on piazza for its brevity and clarity. Although we initially specified the center as the monkey house, as far as we were concerned where to have the distractor is subordinate to the concept of agreed upon distracting turf. Our suggested in-class referendum to agree to a convention based on labeled territory passed, and the final vote narrowly went to the following convention:

If you are going to distract, please do so in the bottom right quadrant of the park.

We now show how this addresses each of the above goals and constraints:

1. Early greedy experimentation revealed that, especially for large amount of monkeys, it is beneficial to have a family member corralling monkeys away from dedicated eaters.
2. The convention can be expressed in one pithy statement that is unambiguous.
3. It is incredibly simple and allows for stylistic differences among groups. It only mandates a specific location for a specific kind of role, but isn't prescriptive beyond that regarding how to perform the role.

4. The only situation in which it breaks down is if a family member goes to the distracting corner and doesn't distract. This, however, seems unlikely and was addressed in class and resolved. Yet even if a family member goes to the distracting corner and doesn't distract, other convention-abiding family members should be able to see this behavior manifest in the monkeys being spread out and can choose to distract in the proper corner.

3 Our Implementation

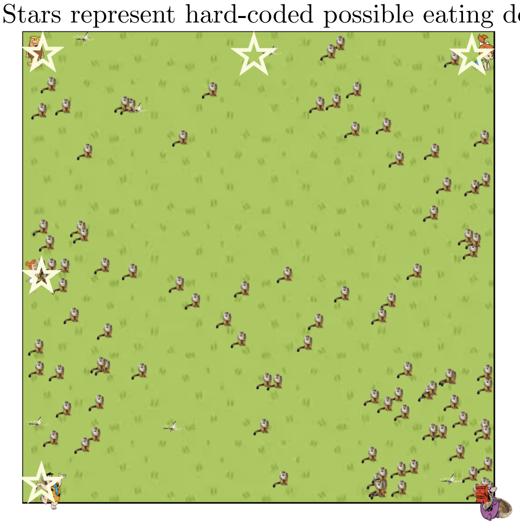
3.1 Eating

3.1.1 Positioning

There are many factors worth considering when deciding where to eat. Clearly we'd like to eat away from the monkeys and geese, but both distributions create moving targets. The monkey distribution in particular heavily depends on the actions of other players, highlighting once again the challenge of coordination. One may make an argument that we should have had a convention about a single specific eating location; however, there's really no advantage to having players all eat at the same location since a simple distraction algorithm can effortlessly clean all the corners of monkeys just as well as it can clean one corner. Moreover, when all family members eat in the same location, the animals have maximum intimidation leverage in that they can threaten all members simultaneously. Indeed, when players do get a chance to eat in the midst of animals, it's often because of a rhythm they have established with their assailants: while keeping food back to manage threats, animals wander off giving a member opportunity to pull food back out again before the animals loop back for another attack. This cycle is clear to observe for the monkeys but also happens for geese because geese too are less likely to be close to a member without the incentive of food. Thus when single members eat in the midst of animals, we observe a rather consistent frequency to the cycle of eating. When multiple family members eat in close proximity, the effect can be much like multiple people playing a musical instrument at the same time. The notes destructively interfere and everyone's eating breaks down into cacophony. Thus, there may have been incentive to create a convention about how to *spread out* the family, but such a convention, with so many moving parts, would suffer frequent break down. With the lack of any such convention, we now turn to addressing how we handled eating position.

Upon player initialization we set 5 global eat points at (50,-50), (0,-50), (-50,-50), (-50,0) and (-50,50) figured below.

Figure 1: Stars represent hard-coded possible eating destinations.



We determine these locations by the following logic: we cannot eat near the distraction quadrant and also want alternative eating locations when there are more than 4 players that are uniformly spread out. In the beginning of the game we head to (-50,0) so as to avoid the initial confusion of symmetry (most other players are heading to corners). Then, every 150 seconds we again consider the 5 options and move if there is a clear advantage. We choose the number 150 to put a hard stop to any attempts at frequent moving to the detriment of eating; nevertheless, our player has strong incentive to not move at all and thus would often stay put even if we considered updating more often.

Firstly, if we have not had a forced put-back in the last 20 seconds or more, we do not move. Next we compute two metrics to help rank the candidate locations: monkey density and member occupancy counts. If there are locations with 0 occupants, we go to one, prioritizing the corners over the mid spots[3] as well as where we currently are. If no spot is empty including where we are (not including ourselves), we neglect occupancy counts and prioritize corners with low monkey counts averaged over the last 200 seconds. The expectation is that some corners have had more patient eaters than others, and even if a corner has multiple members currently occupying it, that's OK *if* they are patient eaters.

3.1.2 Patience Strategy

A naive implementation of the eating role would be to eat whenever possible. In this formulation, the only time food would not be out would be when imminent danger caused it to be held back. Yet, while locally optimal, this policy may not be globally optimal given the limited perceptive field of the animals. To exploit the animals myopic vision sometimes it may be beneficial to wait until a certain number of monkeys have cleared the 40 meter radius (20 meters for geese) before flashing food again. A simple compromise would be to hide food whenever any monkeys or geese are within 40 or 20 meters respectively. Yet this strategy would be overly conservative: when there are a lot of animals, it would be prohibitive to getting any eating done at all. It also ignores the fact that while waiting, other animals could enter our orbit.

Thus, we focus on the relative ratio of animal inflow and outflow when deciding whether it is prudent to first wait before continuing to eat. If waiting leads to many animals leaving and few entering our orbit, it is potentially beneficial. If not, then there is limited upside to waiting and likely more downside in allowing more time to pass.

Our algorithm for generating animal in-flow and out-flow estimates involves computing approximate animal trajectories. To do this, at time t we compute the positional angle for each animal between $position_{t-1}$ and $position_t$ and project a straight line path. For each animal, we estimate the number of time steps it will take for the animal to enter, or leave, our orbit - where orbit is defined by the food-sensing receptive field of the animal. At every time step, we fix the range of time we care about to the time remaining for the last animal within our orbit to leave it. Then, within that time frame, we compute the number of animals that will enter our orbit. If the ratio of currently within orbit animals to the number of animals that would enter our orbit is greater than $flowRatioThreshold$, we issue the wait command if we are not already eating. Otherwise we allow eating to continue until danger is imminent, which we define as 3 monkeys within 6.5 meters of us or 1 goose within 5.5 meters [4].

Clearly it makes less sense to be patient when monkeys are uniformly spread across the map. Typically such a distribution will not improve soon just by waiting. By contrast if even one region of the map is bereft of monkeys, we should have some hope that our region can clear out as well if we give the monkeys time to wander towards the clutches of the distractor. Therefor, we have a function called `areMonkysEverywhere()` which indicates whether any quadrant has less than 10% of the monkeys averaged over the last 75 seconds.

[3] Corners tend to be further away from the monkey orbit of other players.

[4] Clearly if 3 monkeys are within 6 meters of us, on the next time step they could be within 5 meters and steal our food. We choose 6.5 as the cutoff for safety from monkeys to be on the conservative side. From a similar calculation we can get the 5.5 cutoff for geese. Through out all of our play in fact, we adhere to these cutoffs and never risk losing food

This historical distribution information allows us to define two separate $flowRatioThreshold$, making our player more greedy when patience is rendered more futile. Through hyper-parameter tuning, which we scored by measuring the average time it takes to finish lunch, over several family/animal settings, we arrived at thresholds in low variance distribution monkey environments to be 2.5 and high variance monkey environments to be 4.0.

Because of the rapid pace of geese and the difficulty in forecasting their movements, we decided to only implement the patient eating strategy with respect to monkey inflows and outflows. Adding geese to the picture would make the flow ratio estimates far more unstable and lead to sticky and unpredictable behavior.

3.1.3 Eating Hierarchy

One of the important decisions each family member makes is the choice of food item they eat. In cases when there is not enough time to eat all the food, the order in which food items are chosen becomes a quite important factor in the total score. The first step we took was to calculate the point value per second for each food item. That is, a numerical representation for the best return on investment of eating time. To do so, we simply calculated the ratio of the points acquired by completely eating the food to the time needed to finish the food. The values were as follows:

$$Cookie: \frac{4}{1} = 4points/minute \approx 0.067points/second \tag{1}$$

$$Fruits: \frac{2}{2} = 1point/minute \approx 0.017points/second \tag{2}$$

$$Eggs: \frac{2}{2} = 1point/minute \approx 0.017points/second \tag{3}$$

$$Sandwiches = \frac{3}{3} = 1point/minute \approx 0.017points/second \tag{4}$$

Clearly, there is an outsized advantage to investing time in finishing the cookies, over the other foods, although all the other foods have the same return on investment in terms of time - even though the sandwich does take longer, which was considered if there was not as much time left. Another factor we considered is that geese only steal sandwiches, which meant that in high geese densities, eating sandwiches was a luxury. We also took into account the fact that pulling out and putting back food was incredibly costly, since each operation took 10 seconds.

As such, our flow actually first prioritized eating what was already in hand, if something was already out of the bag in our hand, we would eat it if we could. We did so because this act of switching foods was far more costly than any minor gains we might have made since even 10 such switches across the game would be the equivalent of eating a whole fruit or egg. If there was nothing in hand, we fell back to our return on investment calculations, prioritizing the cookie, followed by the fruit and eggs and lastly the sandwiches. We placed the fruit and eggs above the sandwiches due to the time consideration - we were always more likely to finish the fruit and eggs than the sandwiches which would take longer. Sandwiches also had to be put back more often since both monkeys and geese could steal them, while the other two could only be stolen by monkeys.

The only exception to this flow was in high geese density scenarios. In cases where getting to eat sandwiches was a luxury and we truly believed we could finish all our food if optimized correctly (based on time left), we actually prioritized the sandwich over everything if there were no geese in sight, since we wanted to take full advantage of those opportune moments. This allowed us to marginally increase our likelihood of finishing the sandwiches, since the rest could be eaten any time but sandwiches had very short time spans of eatability.

3.2 Distracting

3.2.1 Free Space

The goal of any distractor is quite simple - to create free space on the board where other members can eat without being disturbed, ensuring that n-1 members are able to eat effectively while 1 member sacrifices eating efficiency for the greater good. This approach clearly has increasing returns to scale; more members can be eating while one member distracts and hence the total points gained per minute increase with an increase in family members. As geese are too challenging to effectively distract, we limited our notion of free space to monkeys. Hence, free space is essentially the aggregation of all the positions in the park where there are less than 3 monkeys at any given point in time, allowing the member present there to eat free of monkey danger.

3.2.2 Positioning Distractor

There was a lot of debate in class regarding the position of the distractor - with camps advocating for a centrally positioned distractor and a corner distractor. Mathematically, both have their benefits.

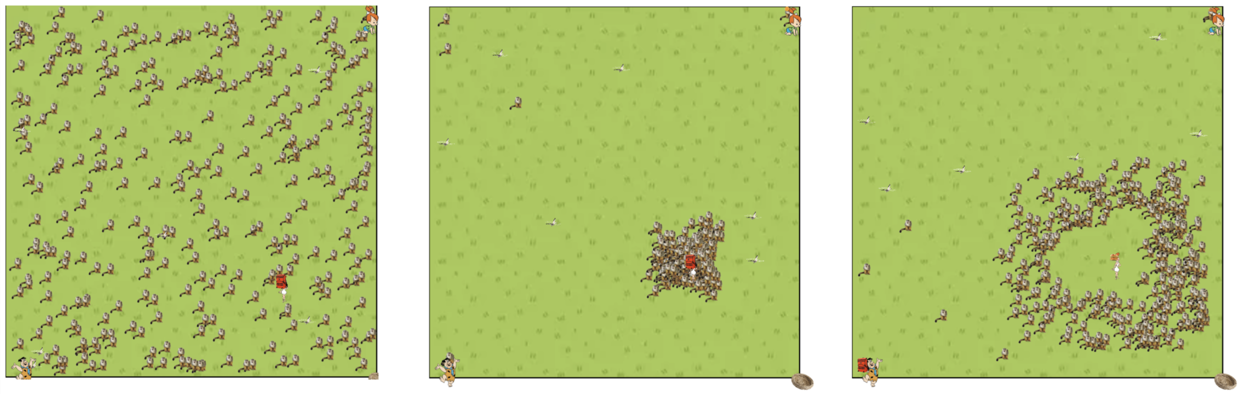
A centrally positioned distractor provides a large catchment area of 1600π or 5026 square metres out of a total of 10000 square meters - accounting for roughly 50% of the total playing field. By catchment area, we mean the area around the distractor where monkeys will be able to see food taken and get attracted. Note that the goal of the distractor is the opposite of all the other players - while the eaters want to minimize the monkeys around them by minimizing the number of monkeys that see them with food, the distractor wants to maximize the number of monkeys nearby and so attract as many as possible. Hence, being in the center gives the distractor the largest catchment area. Conversely, being in the corner maximally reduces this catchment area, since if the distractor was in the absolute corner, it would have a catchment area of just 1/4th of the previous value $\approx 12.5\%$, being bounded by the borders of the rectangular park. At the same time though, it would keep monkeys within its realm the furthest away from the rest of the eaters, since they would be at the other corners of the park ideally. Distracting in the center had this lacking - at the extreme edge of the catchment area, the monkeys could still see the other eaters in the corners and so would occasionally leave the distractor and harass the eaters. As we know, the diagonal distance is $50\sqrt{2} = 70.71$ and the monkey visibility is 40 meters. Hence monkeys that are only 31m away from the distractor could still get attracted to the eaters in the corners and leave the realm of distractions. Since monkeys move at 1 m/s and it takes 20 seconds to put in and take back out, this only leaves a 10 second buffer in case of any errant monkeys that might not follow the in flow and outflow process of the batch - since our above hypothesis relies on all the monkeys converging and diverging from the distractor together, which is not always the case. Essentially, the center is great at catching monkeys but the edges are the best at keeping them away from other eaters. Furthermore, the fact that animals randomly bounce off the walls also causes greater unpredictability in monkey motion and creates trouble - making the absolute corner undesirable as we'll discuss in more detail.

To avoid this, we position our monkey at the center of the lower right quadrant, which we found to be a successful trade-off between increasing catchment size and maximising eater and distractor distances.

3.2.3 Algorithm

Our distracting algorithm is simple yet effective. First, we navigate to (25,25), directly in the middle of the 4th quadrant. *It is essential that we give ourselves breath on all sides* because as we will soon discuss, our distraction strategy hinges on the assumption that monkeys will be moving unobstructed by boundaries, typically radially outward or inward. Then, as always we keep food back whenever at risk; otherwise, we pull food out and take a random walk with the food in hand. At worst we get a ring of monkeys around us, oscillating inward and outward. At best, we can hope to become the Pied Piper with the monkeys narrowly clumped in tow. Our distraction algorithm can envelop all of the monkeys for any m in the configuration space. To demonstrate, we plot several screenshots from four family runs.

Figure 2: We effectively entertain ≈ 300 monkeys.



We can explain the effectiveness through three simple quantitative facts.

1. The field of influence of the distractor - or equivalently, the radial perceptiveness of the monkeys, spans 40 meters.
2. Each time food is flashed within our receptive field, monkeys start to converge toward the same point.
3. When food is not present, monkeys wander off in random directions.

First, if we are forced to put food away, it takes us 20 seconds to get it back out again at best. The only monkeys walking away from us will be the ones waiting closest to us when we first put the food away; they will have wandered off in random directions. The other monkeys will keep their momentum and still be walking towards us. Therefor we have the full 40 seconds it will take the monkeys walking away from us to leave our center of influence. After spending 10 seconds to put the food we had out away, we have 30 seconds left to get it back out again, giving us 2-3 tries to do so. Because while trying to get food out monkeys walk right past us it turns out to be very unlikely that three monkeys will be close enough to us on each take out attempt to prevent us from flashing food again within the critical time frame.

It's interesting to consider why all the monkeys tend to converge to the same radius from us. We attribute radial convergence to the fact that when food is flashed all the monkeys tend to the same point and otherwise move randomly. Over time their randomness evens out. In the limit, provided sufficient player movement, monkeys will eventually converge to the same radius. Note again that our position with space on all sides is important.

Another way to look at the ring of monkey phenomenon in the third screen shot is the following: we get the ring behavior because no monkeys are close to us while all monkeys are contained within a 40 meter radius. We've discussed why no monkeys are beyond 40 meters, but why are no monkeys close to us? The answer is simply because we have no flashed food in some amount of time, thus all monkeys are moving randomly away from us.

In the intro summary we mentioned that trying to distract from the absolute bottom right corner is a mistake. To address why, consider that a key fact from our explanation above is that monkeys all move away from us when we don't have food out. This point breaks down if we are in a corner because monkeys moving away from us quickly hit a corner and have a solid chance to randomly ricochet towards us again! Thus bombarded by monkeys we would not be able to consistently get food out again in time.

3.3 Role switching

When eating, we question at every time step whether or not we should take on the distractor role. Let's begin by enumerating the reasons why we would decline the distraction role:

1. There is already an *effective* distractor present in the bottom right quadrant.
2. We have changed roles in the recent past, defined as within 10% of the total game time.
3. The time we need to optimistically finish our food is more than half the time remaining. That is, the minimum amount of time needed to complete all the food(if we ate every second) is more than half the total remaining time.
4. We are the only family member in the game.
5. Less than 400 seconds remains.
6. We've been distracting too often.
7. There are less than 3 monkeys in the park.

If we would not decline to distract, then we assume the responsibility of the role. We also always distract if we've eaten 99% of our food, but in that case we distract from our last eating location. There is generally some benefit to having distractors elsewhere on the board when eaters are impatient, and when we are almost out of food, others likely are as well and will be heading to distract. Therefore, by staying in place we also cause less confusion.

Now, we have still left open the critical points of how we define "effective distraction" and "distracting too often".

3.3.1 Definition of Effective Distraction

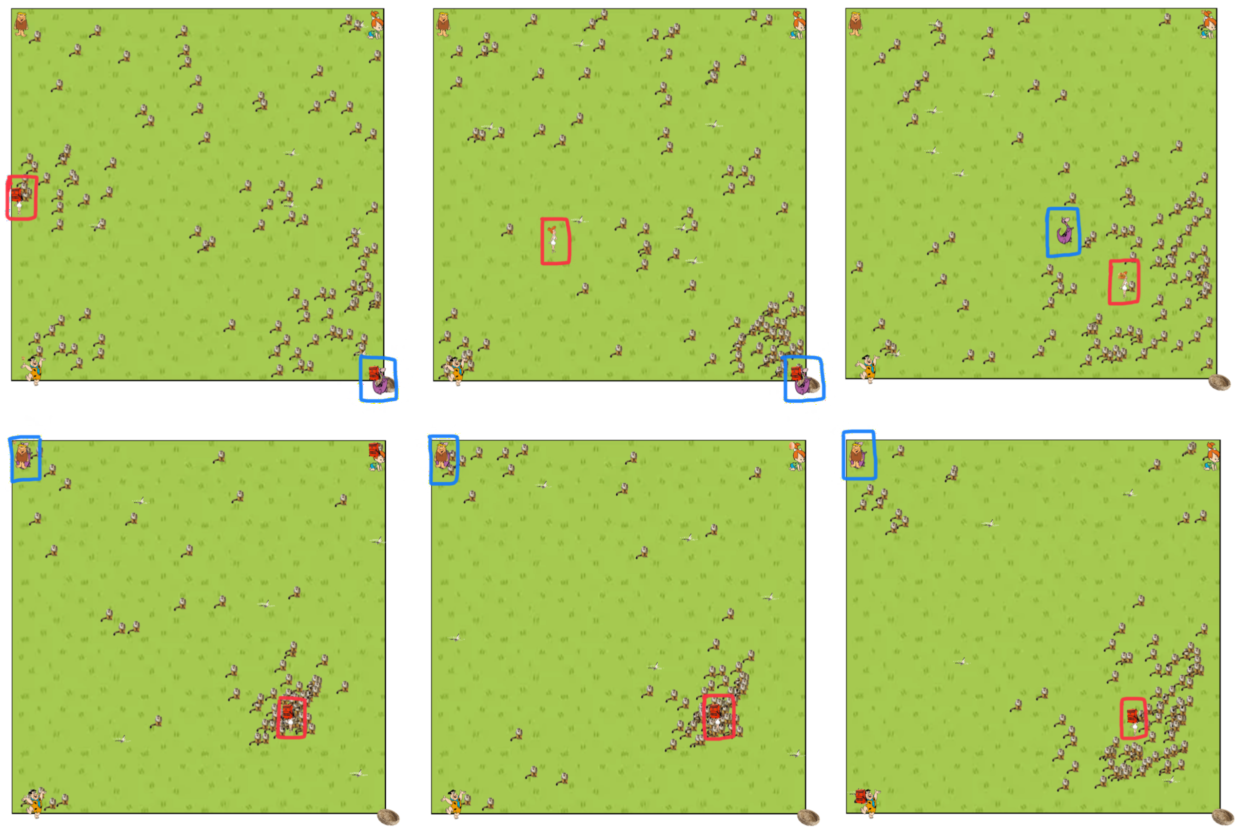
As family size grows, having an effective distractor becomes ever more important. One member's sacrifice is a boon to the rest of the family, allowing them to avoid many costly put backs due to monkey harassment. Therefor, if a distractor is not effective we should intervene, confident that we are adept at the task. Since the goal of distraction is to skew the distribution of monkeys to bottom right, we measure effectiveness by considering monkey distribution over time. Every time step we create a hashmap in which keys pertain to the 4 regions of the map and values are set to the number of monkeys in each region. This record of monkey counts it then stored in a global array, giving us access at anytime to the entire history of monkey distributions. Now, since distractors may momentarily and intentionally let monkeys out of the proper quadrant, we must not intervene everytime we see monkeys elsewhere. Three precautions help us to only intervene when it is very likely necessary:

1. We average monkey counts over the last 20 seconds of history.
2. We enlarge the distraction quadrant from (-25,-25) to (50,50) when taking counts.
3. We only intervene if less than 65% of monkeys are in the distraction sphere.

Ideally, when we intervene the current distractor will sense our presence and be triggered to go back to eating without offense. As an example consider the following screen shots from a game with g1, g2, g3, g6 and g7 as of submission night.

We sense that the distractor is ineffective because 65% of monkeys are not restrained to the bottom right. Note the distractor is making the blunder of attempting to operate from the absolute corner. Once the distractor detects our presence, it gracefully goes back to eating and we go on to clean up the vast majority of monkeys. At this point all an eater would have to do is to wait 40 seconds and they would be free from

Figure 3: Us (red) intervening to help an ineffective corner distractor (blue).



monkey harassment so long as we keep distracting. Note how much more pleasant the situation has become for patient Fred Flinstone located bottom left.

3.3.2 Alternative Distracting Effectiveness Metric

While the above metric worked for most cases, it actually failed to adequately detect the performance of certain groups' distractors since they would often stay in the lower right quadrant themselves but drag the monkeys behind them, which resulted in the monkeys moving out of the lower right quadrant, which would make them appear unsuccessful to us, even when they were actually successfully restraining the monkeys. This actually happened quite often when we tested our player alongside G5, who had a great distractor, but would lead the monkeys out of that region occasionally and trigger our 'ineffective distractor flag'.

To overcome this, we actually flipped the meaning of effectiveness. Instead of trying to track monkey positions, we started keeping track of the times that we were forced to respond to monkey danger. We realized that the effectiveness we truly cared about was whether we were able to eat in peace, and so this made perfect sense. By tracking whether we had more than a given threshold of time to eat (strictly over 20 seconds because that's the time taken to put in food and take it back out) of roughly 30 seconds between interruptions, we were able to use this as a short circuit for our earlier distraction function. If this metric showed effectiveness, we wouldn't deem the distractor ineffective regardless of where the monkeys were. This allowed us to supplement our earlier metric. We didn't use this alone because we didn't want to focus on just ourselves and actually ensure everyone was able to eat - which we couldn't guarantee with this metric but could with the other. This metric is far more tolerant to different distraction approaches while the previous is far more robust and provides greater guarantees of free space. Hence, combining the two allowed us to craft a robust yet tolerant metric for distractor success.

3.3.3 Definition of Distracting Too Often

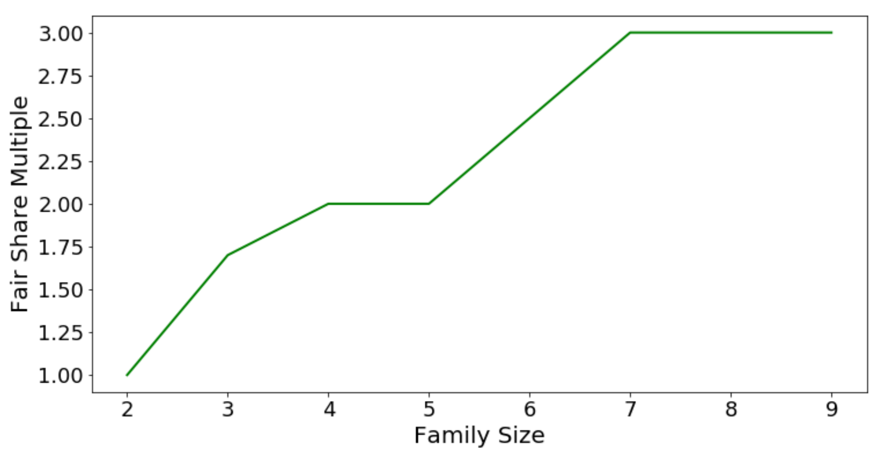
If only one dexterous distractor could assume the position for the entire game, we would circumvent the principle challenge of Lunch: *coordinating roles*. Alas, this sole distractor would never have time then to eat much. Now, with hindsight, this would actually not be a problem for games such that the total duration does not allow the rest of the family to finish eating anyway. If we can be sure that all other members still have food that's at risk of not being finished, then we may just as well not finish our food to help them finish theirs. Although we could have and should have programmed our player to sense the edge case of very short games, it's generally difficult to know how much food other players have eaten because we don't have explicit access to the information. Furthermore, most games are not so short. Consequently, we circle back to the question of how to know when we've done our fair share of distracting and should terminate permanently so as to eat.

A naive solution would be to never distract more than our fair share of the total time. For example, if there are 4 family members, we may sum the time of all instances we've spent distracting and never let that number exceed total time / 4. This is not a bad start, but recall two important factors:

1. As of submission time, there was only one other distractor capable of entertaining any number of monkeys, G5.
2. We already have a number of other measures in place to avoid distracting unless it's necessary.

Consequently, the cap we put on our distraction efforts should be more liberal than just our fair share. We hard coded multiples of fair share distracting time we'd be willing to tolerate as a function of the number of family members, using reasoning and experiment to justify.

Figure 4: Share multiple function describing how to put an upper bound on the total time we're willing to distract



To summarize the idea, if we're in a family of size 2 we don't do more than our fair share of distracting. At a family size of 9, we are willing to do up to 3 times our fair share, but note that this is still only 1/3 of the total game duration. The more family members, the higher the multiple we're willing to take because more members stand to reap the benefits of our better than average distraction[5].

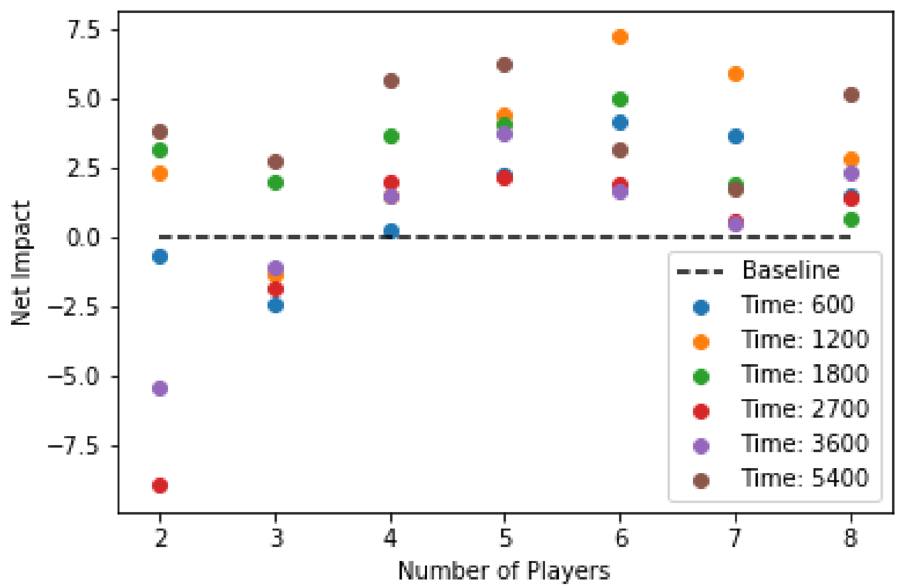
[5]As measured at time of submission

4 Results

4.1 Do We Help?

The first question we wanted to answer is whether we make a positive or negative contribution to our family. For this, we compared the averages for the games that we were part of to the games that we were not part of. We then normalized these differences by number of players in the game to ensure that all differences had the same scales. Finally, we added up these differences across all configurations for each family size.

Figure 5: In-Out Comparison



In the In-Out comparison graph, the dotted black line signifies a 0.0 net impact, that is, we simply didn't matter - games with us and without us had the same average player scores. Consequently, anything above signifies a positive impact while anything below signifies a negative impact. As can be seen from the graph, we had a strictly positive net contribution for all families with 4 or more members. We seem to flourish in families that are about 5 or 6 members large, where our contributions are far above the baseline for all time spans across all cases.

Equally interestingly though, we are actually harmful to families with 2 or 3 members, having a net negative impact more often than not. It is possible that this is due to our propensity to distract and help weak distractors. With small families, the payoff from distracting is far lower, since the total amount of points acquired by the family are directly proportional to the number of family members during this distraction time. This can actually be understood quite well through an analogy to economics. Distraction has a fixed cost of one member not eating, but variable gains since all the other members can eat in peace if it is done well. The gains are just a sum total of the points gained by eating done during that time. Hence, for smaller families, it seems that these fixed costs were greater than the variable gains since only one or two members would be eating while we invested precious time in distracting.

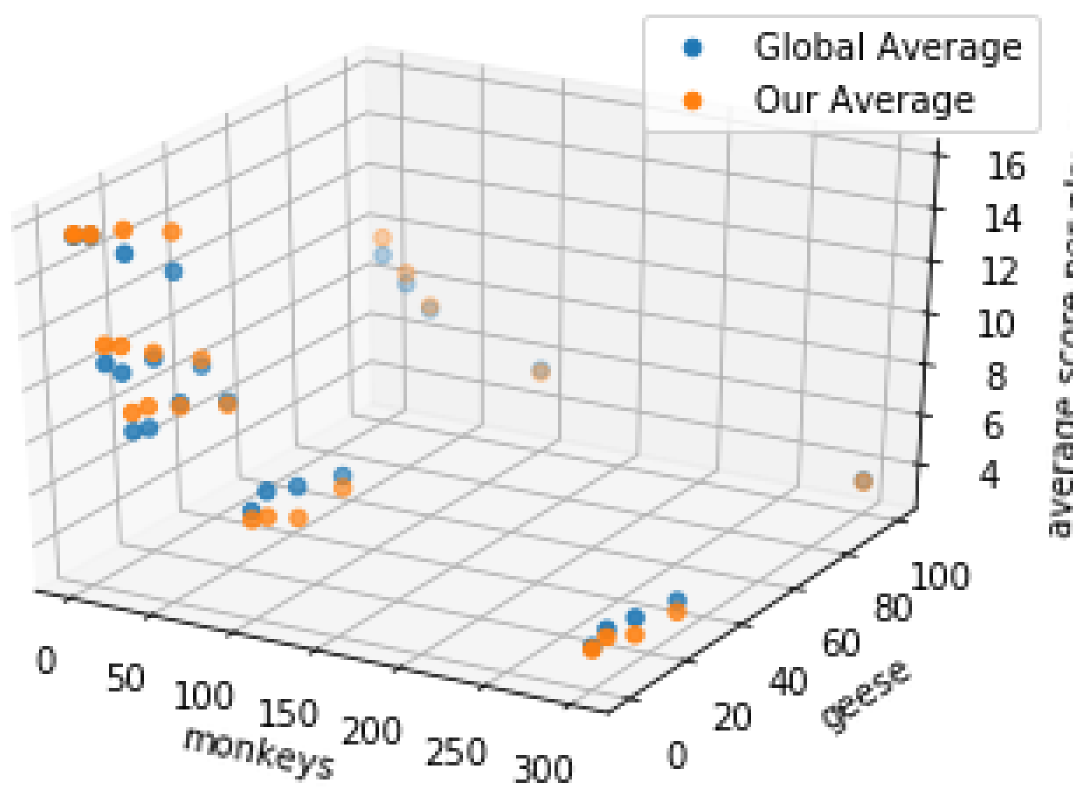
One family size that is not present here is 1 family member. That is because we actually did incredibly well for this case, with our impact being +17. This completely destroyed the y axis scale and distorted the resolution for the rest of our metric, which is why we excluded it from the graph. Obviously, since there was only one member involved in this case, the metric was incredibly sensitive to our individual performance, as opposed to other games, where performances depended both on individual player abilities and also coordination abilities. The reason this was not impacted by the issue afflicting other small family sizes is that we actually specifically accounted for this case by turning off the distraction role when there was no other family member. That is, we simply would never switch to the distraction role and stick to eating. This is actually a really good indicator of the 'greedy' performance of our player, isolating the eating side of our strategy by eliminating any scope for distraction.

Furthermore, from a time standpoint, we seem to be very effective contributors for shorter time spans. In the same In-Out graph, you can see that the orange, blue and green dots-which correspond to 600, 1200, and 1800 seconds - are continuously towards the top for almost all family sizes. This could be due to two reasons. First, for longer time spans, most families have almost all members finish all their food in the given time, which means that the difference between games we played in and didn't play in wouldn't be too major, since both would be quite close to 16 (all food eaten) regardless.

4.2 Investigating Deeper

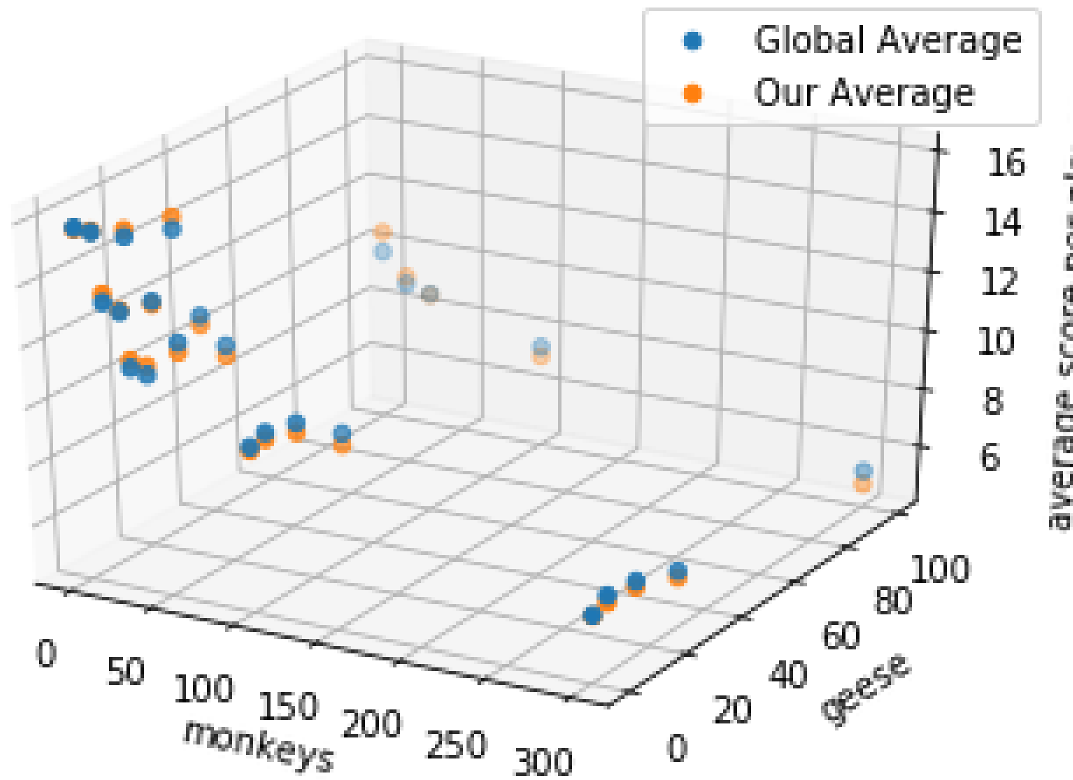
Although the previous section clearly established a general understanding of where we actually benefited the team and where we hurt it, we wanted to go deeper and see where we flourished and where we failed, in terms of different configurations of animals. This time around, we used the global average as a baseline and compared it to our personal average since the goal was to understand where we deviated, rather than by exactly how much.

Figure 6: Spread for three family members with a 1800 second time cap



We actually plotted this distribution for each family size from 1-8 for each time cap - for a total of 48 graphs, but obviously that was way too many to include in the report. While narrowing down on the most

Figure 7: Spread for three family members with a 2700 second time cap



influential ones, we found that there was actually a replication of trends across time caps. Although the average scores per player obviously increased as the time cap increased, the trends remained quite similar across the timeline. We initially suspected that this could be due to the lack of resolution in our graphs - since the x and y axes are of the order of 10^2 but since the actual score axis is quite granular, this was not the case. It seemed that our performance relative to the average (our baseline) was actually relatively consistent across timelines, when controlled for specific configurations. We finally realized that this actually reinforced our previous graph - we did consistently better for all players values in a relatively similar manner for all players greater than 3, which meant that these distributions were also similar. So, we decided to focus on the graphs where 1) we'd acted as negative influences on the mean score and 2) where there were clear distinctions between our values and the global means.

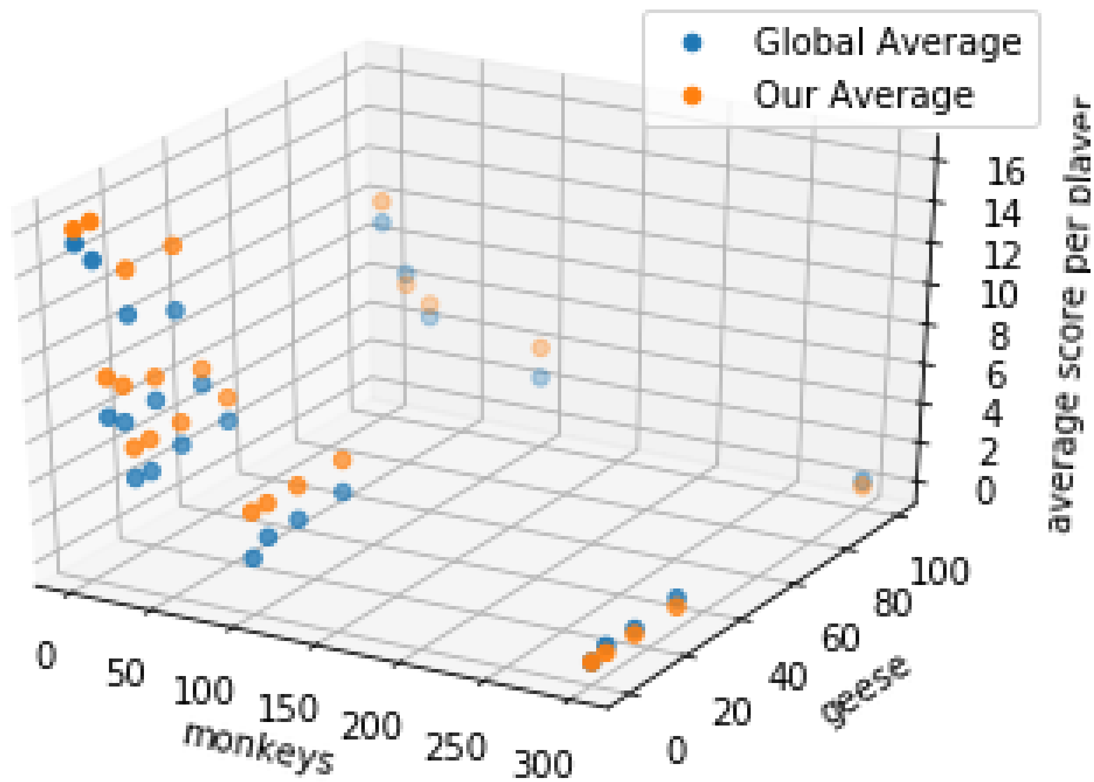
As one can predict from the previous graph, this was the case for n=1, n=2 and n=3. In the latter two, we'd done better than baseline and worse, so our points were usually distinct from the average in different ways.

In the two member family, we can see that that there is a clear difference between our player and the global average for low animals numbers (low monkey and geese numbers). Our player seems to do quite well in relatively greedy scenarios, which was true across the board, regardless of family size - which makes sense since family size usually comes in with higher animal numbers, where coordination becomes a factor. Interestingly enough, we did distinguishably poorly for middling level of geese and monkeys. This seems to be an interesting outlier, since if you look further behind (saturation represents depth) we've actually done quite well for larger number of geese, with our faint orange dot slightly above the faint blue dot. Similarly, in the bottom corner, we seem to be very close to the global average, but slightly below it, for cases of a few geese and a lot of monkeys.

In 3 member family, we can immediately see that there is far more overlap. The points are much closer together. We seem to have covered up both our deficiencies as well as lost quite a bit of our advantage. At the same time though, we seem to have gained a minor advantage with the high monkey middle geese values, which shows the increasing returns of scale on our distraction strategy. Graphs for higher member families showed even more gains at the front. We have quite a bit of success with high monkey numbers, where our player's ability to distract provides a strong value add to the team.

For the rest of the family sizes, the dots were almost exactly on top of each other, not providing for as robust an analysis as these. We made minor gains across the board of just a few percentage points, which accumulate to the results shown in the first figure. But in those cases, the gains are quite minor and uniform across almost all configurations.

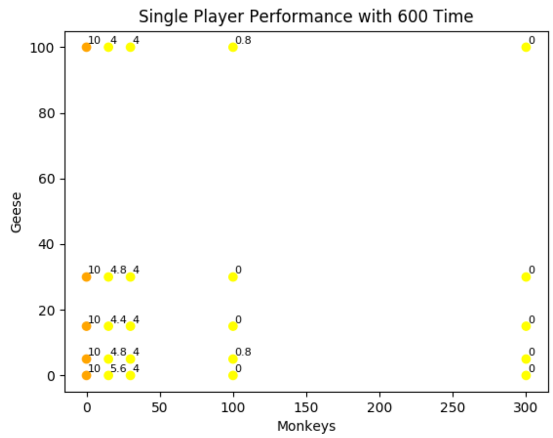
Figure 8: Spread for one family member

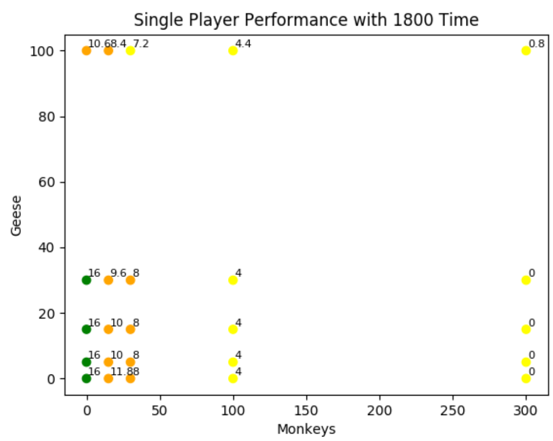


The one other figure that was incredibly interesting to analyze was the 1 family member case. As mentioned before, the gains were substantial enough that they skewed the scales in the previous graph to the extent that we had to exclude it to maintain decent resolution for the rest. This is actually quite evident from the graph, where we are almost always better than the average, by quite a substantial amount. There could be two possible reasons for this. The first is that the other teams didn't account for this edge case - and are wasting time distracting even when there is only one player. The other is that our eating optimizations have actually made a distinguishable differences. We do believe it is the latter, but that could also just be wishful thinking.

To show just our scores in a 2D setting, we plot (See Figures 9, 10, 11) our average scores for each monkey-goose configuration separately for each total time configuration and demonstrate that, even in hurried settings, we do a solid job of eating.

Figure 9: Average Scores for one family member. Low time settings.



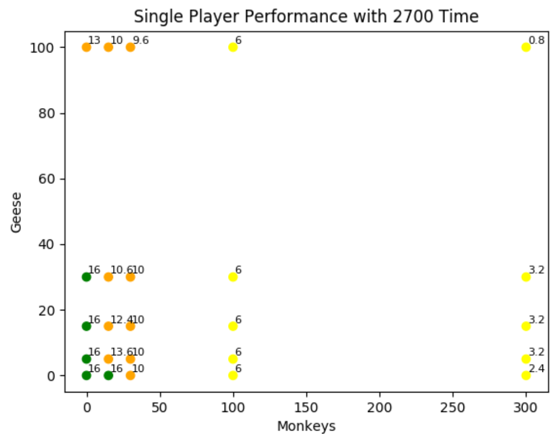


Clearly, we are not finishing the two sandwiches due to our hierarchy, which positions them last. We could have added 2 points by prioritizing sandwiches specifically for the (0,0) case and not eaten the 2 fruits (would've put us under 10 minutes but given 2 more points.)

As can be seen in the figures 10 and 11, given enough time, we are able to eat even in the worst scenarios (300,100). Our player seemingly finishes a cookie regardless of geese density - which makes sense since the geese don't steal cookies.

We attribute this to our ability to switch roles quickly, identification of quieter places to eat, and patient eating.

Figure 10: Average Scores for one family member. Medium time settings.



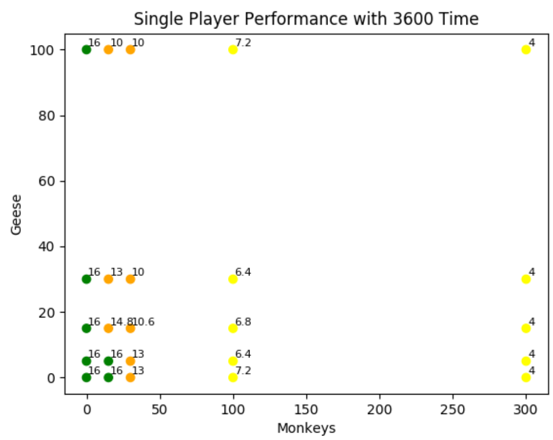
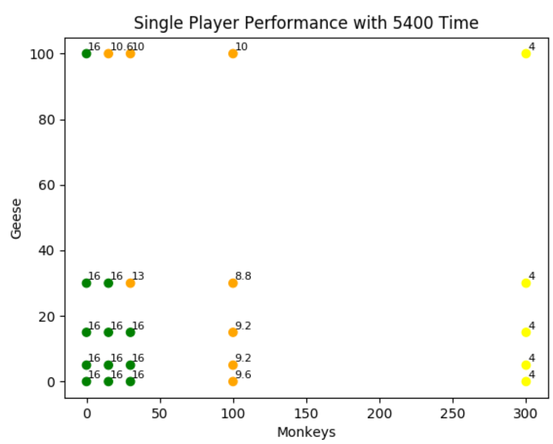


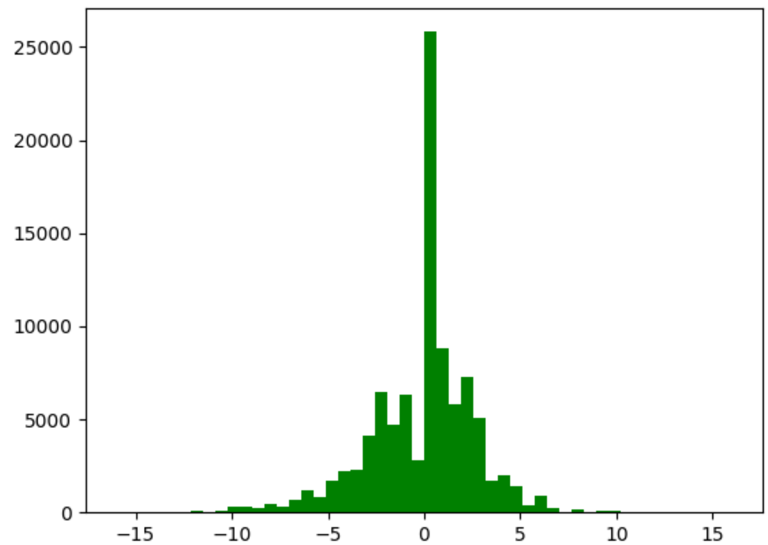
Figure 11: Average Scores for one family member. High time settings.



4.3 Individual Contribution

Even though this is a cooperative game and it is difficult to tease out individual contribution to team score because of the role of distraction, we thought it would be interesting to see how our individual scores compare to others in the games we participate in.

Figure 12: Individual Performance Centered on Mean for Runs in which We Participate. I.e. Individually, we are holding our weight.



To measure this, we plot a histogram (See 4.3) of $ourScore - totalScore/(teamSize - 1)$. 0 represents that our score was perfectly representative vis-a-vis the rest of the family members. Positive means we individually outperformed, and vice versa. The global mean of these figures across all runs in which we participated was -0.17. It appears as though we evenly balance eating and distracting. Furthermore, we can clearly see that our score was very reflective of the average score per person of the team, which indicates that we were a strogly average player - as was evident from our previous graphs as well, where our score overlapped with the average score.

5 Future Work

Future work involves better modeling and adaptation to geese. We largely left geese out of our distraction and patient eating due to the fact that geese have a smaller food-sensing receptive field and are much faster. This makes it harder to dynamically adapt to their movements, but not impossible. Also, since many eaters do not patiently let monkeys to the distractor, we could have devised another more assertive distraction mode where we circle the corners with food out to actively draw monkeys from eaters. To do this successfully, we would need a safe path finding algorithm since running around with food out is quite precarious.

We also hypothesize that a more robust convention, which more clearly delineates expected roles, could be very beneficial. For instance, it would be nice to be able to specify who is distracting at what time so that transitions between eating and distracting are more seamless.

There's also the chance that a reinforcement learning solution could uncover optimal strategies that are more difficult for humans to ascertain. Or at least, illuminate certain spatial configurations that are less obvious and lead to more effective spacing between family members. Yet, there's a good chance that even a well-tuned RL policy would not be able to design distraction policies as effective as the ones we've seen in class, which follow very clear deterministic planning and would fail if too much randomness was introduced.

6 Conclusion

We devised a player able to effectively cooperate with convention while also being robust to convention break down and the many uncertainties for which we had no convention to guide us. For example, we are able to sense an ineffective distractor and respond with aid. Also we are able to continually assess eating locations and move fluidly to respond to various map states. In doing so we not only help ourselves to eat, but also are careful to not disturb other player's eating. Finally, our patient eating strategy is able to respond to qualitatively different monkey distributions to decide an appropriate level of greed and ultimately let monkeys go to the distractor. The tournament results show our robustness to myriad parameters, our ability to handle edge cases and the effectiveness of our distraction algorithm. On the whole, we are strongly average player - as can be seen from our analysis. There are cases where we disadvantage the team, but equally proportional cases where we benefit it. When compared to the global average, we sit right on top of it, although the in-out averages show greater distinction. We do adopt a relatively effective single player strategy though, that does manage to eat something even in the harshest of animal conditions, given enough time.

7 Contributions

John implemented initial player scaffolding, global awareness of animal densities over time, and the state machine to decide how to update player role. John also worked with global positioning, distraction, integrating parts and functions to ensure we never get food stolen. John wrote sections 1 and 6 of the report and the corresponding implementation sections.

Tanmay worked on the distraction strategy, the control flow for choosing food items, the distractor efficiency metrics and the tournament and player analysis.

Griffin worked on hyper-parameter tuning, patient eating, devising convention proposals and appropriate criteria for evaluating them, parsing tournament results, conducting error analysis, descriptive player state formulation, and writing Conventions, Patient Eating, portions of Results / Analysis sections, and Future Work section.

8 Acknowledgement

Patrick - for his tournament summaries.