Project 1 - Potential Fields

Introduction

This write up is the result of a project completed by Edward Ekstrom and Spencer Weight on using Potential Fields to move digital tanks towards a goal. The types of Potential Fields used to guide the digital tanks are Attractive, Repulsive, and Tangential

The parts of the project that Edward has worked on are these:

- Coding Visualization
- Gathering images from the visualization for the report
- Coding the “Tangential Field”
- Finishing the code for “Repulsive Field”
- Writing the descriptions for the tests versus two of the teams we competed with

Edward Worked 20 hours on the project.

The parts of the project that Spencer has worked on are these:

- Coding the “Attractive Field”
- Started the coding for the “Repulsive Field”
- Writing the descriptions for:
    - Potential Field coding
    - The tuning process and alternative ideas for our Potential Fields
    - The tests against our own agents
    - Our competition with the third team we went against

Spencer worked 20 hours on the project.

Coding potential fields - Working Implementations and Alternatives

The first potential field type that we worked on was the attractive field. Our working implementation of the attractive field follows a simple idea. On each time tick, for each tank on the team, find the closest flag to it. Once the closest flag is known, put an attractive potential field around it. The attractive potential field tells the tank to move at a speed of 1.0 outside of the field's sphere of influence. Inside the sphere of influence the tank speed is reduced to a ratio of how far the tank is from the radius of the object at the center. Once the tank picks up the object, in this case, the flag, the attractive potential field is removed from the flag and placed on the home base of the tank. The tank is then attracted back to that base to deposit the flag. When the flag is deposited the tank finds the next closest flag and continues the cycle again.

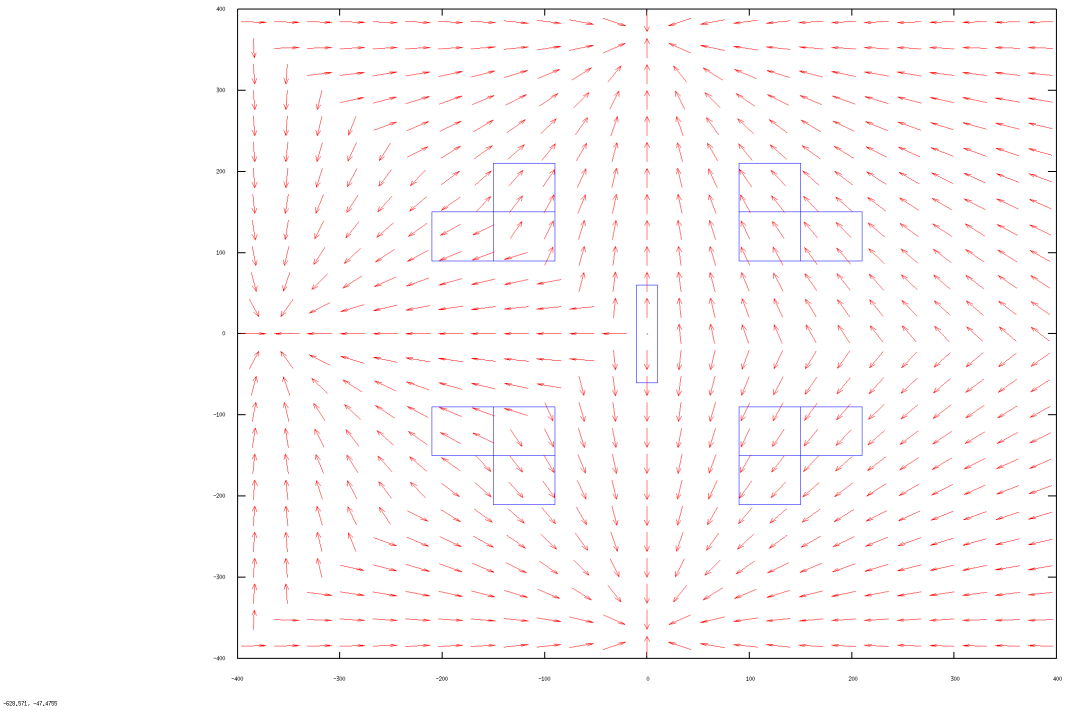

Figure 1. Attractive Fields - Start of Game

We chose to code the field this way because it made sense from the viewpoint of the tank to move to the closest flag. In our initial implementation, we made an attempt to have all flags have potential fields on them, and then add the vectors of attraction so that tanks would be pulled in whatever direction the vectors added up to. This ended up causing the tanks to only attack the base on the opposite side of the field. Calculating the closest flag to each tank allowed the tanks to split up and choose the bases that were closest to them. This maximized the rate of the retrieval of flags from enemies in the sense that the tanks didn't have to travel as far to get back to their home base.

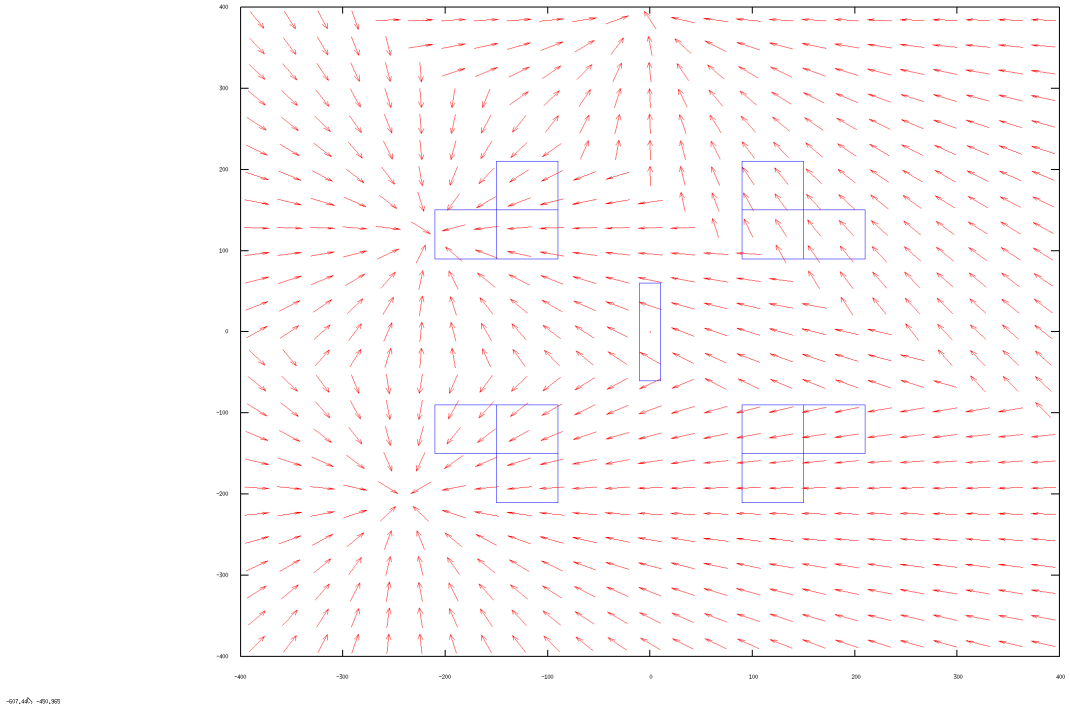

Figure 2. Attractive Fields - Midgame

The second type of potential field that we programmed is the repulsive field. The idea that we implemented was that when a tank enters the sphere of influence of a repulsive field, it would maintain its current speed, but it would change its angle to be completely opposite (0 to pi radians) to avoid the

object. We placed our repulsive field on enemy tanks. If our agent gets close to an enemy tank, it will turn out of the way changing its angle by pi radians until it leaves the enemy's sphere of influence.

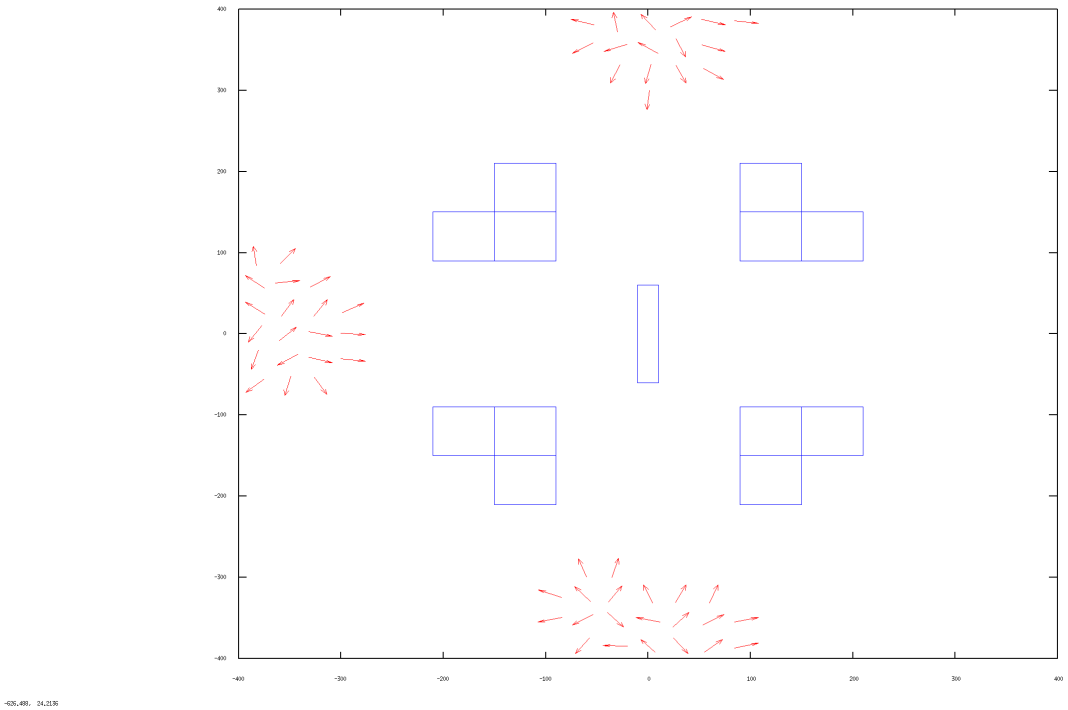

Figure 3. Repulsive Fields - Start of Game

Our original implementation of this field was similar. If an enemy tank was oncoming, then the agent would just randomly choose left or right and then turn pi/8 radians to avoid it. This ended up not working because the agent wouldn't veer out of the way at enough of an angle to dodge the enemy tank. Earlier implementations of our repulsive field would just change the speed of the tank to 0.0 if it go too close. This failed because then the tank became a stationary target more often than it would move out of the way.

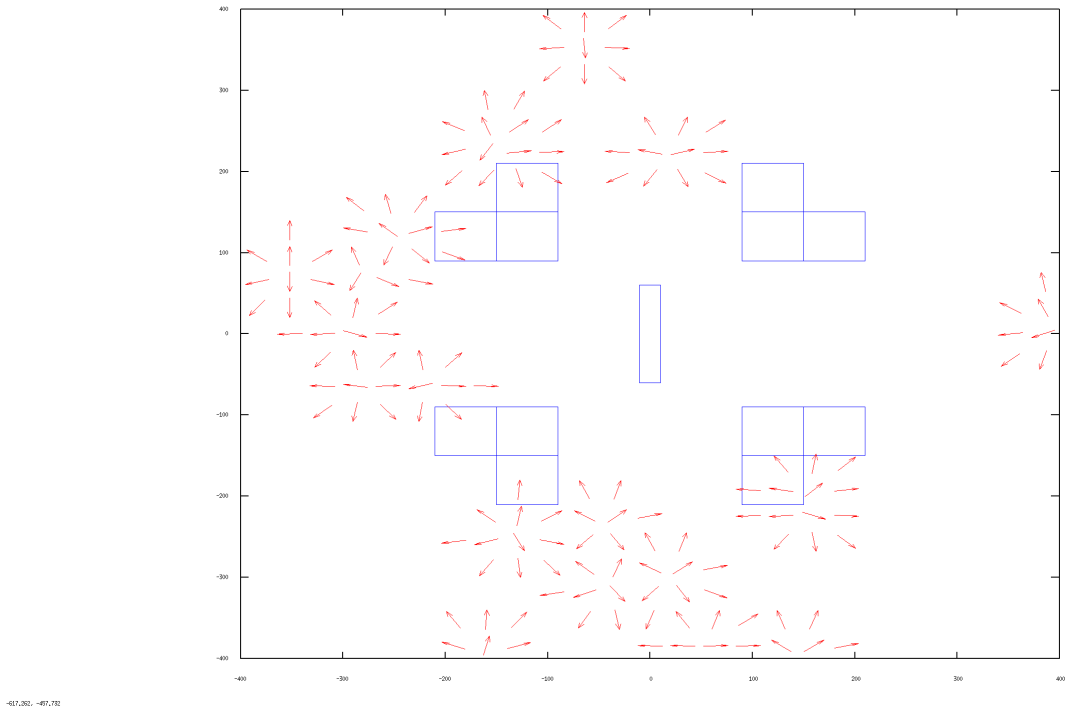

Figure 4. Repulsive Fields - Midgame

The last field that we worked on is the tangential field. This field took a bit of work, but our final working implementation works well enough to guide the agent around an obstacle. The way that we managed to get this working was bt calculating the center of the obstacle block based on the corner locations returned from the get_obstacles function in the bzrc.py file. When the center was known we put a tangential field at the center of the obstacle block, and made the sphere of influence large enough to cover the entire block. The calculation for this was dependent on the furthest corner from the center to make sure that the entire block was in the tangential field. Once the field is set in place and the agent has entered into the sphere of influence, the agent will calculate the angle for a path that is tangent to the sphere of influence. The agent then continues on that path until it is out of the sphere of influence.

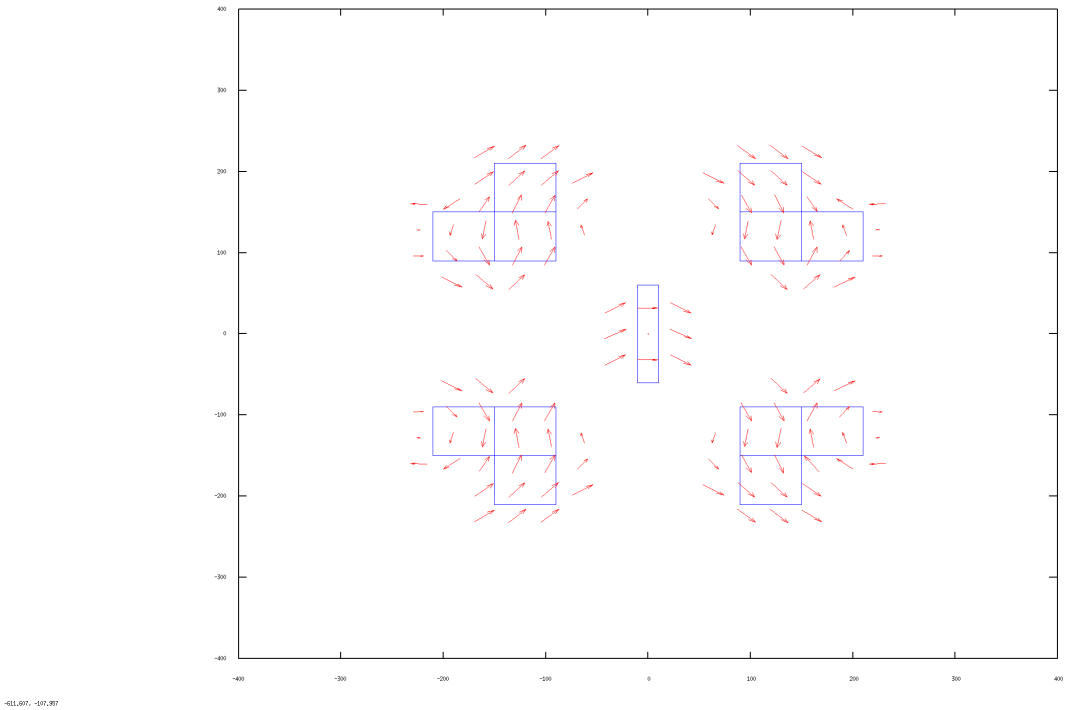

Figure 5. Tangential Fields

Our initial implementation of this field would add an angle that was slightly more than pi/2 radians. This was supposed to turn the agent away from its current path enough to have the agent continue on a path parallel to the edge of an obstacle. Adding a constant angle didn't work out as well as we hoped, so we changed the tangential field to use cos and -sin to figure out the appropriate angle to get the agent to move in the correct direction. This benefited the agent by not steering the tank into the wall if the agent naturally found a path that was parallel to the wall.\

Comments on how fields work together

In our implementation of the three potential field types, we got them to work together by only allowing one field to affect the agent at a time. The logic that the agent follows directs the agent to default to moving towards the nearest flag. If on the way to a flag, the agent enters the sphere of

influence on an enemy, then the agent reacts to the repulsive field only and ignores the attractive flag field until it leaves the sphere of influence around the enemy tank. If the agent encounters an obstacle, the tangential field around the obstacle takes precedence over a repulsive enemy potential field and the attractive flag potential field.

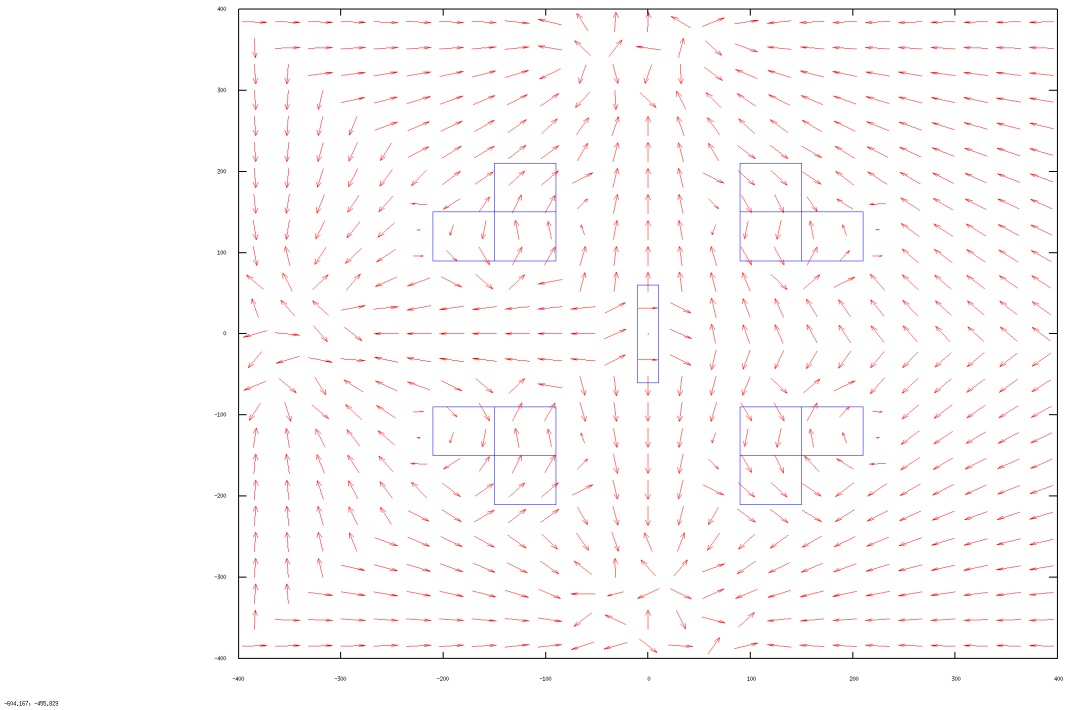

Figure 6. All Fields - Start of Game

Choosing to have only one field affect the agent at a time proved to be easier to code. The original method of combining the fields was to put the fields in a list and then have the agent iterate through the list and add up the vectors from each potential field. This proved to be unwieldy due to there being too many fields influencing the agent in the wrong direction from its goal which was to capture a flag.

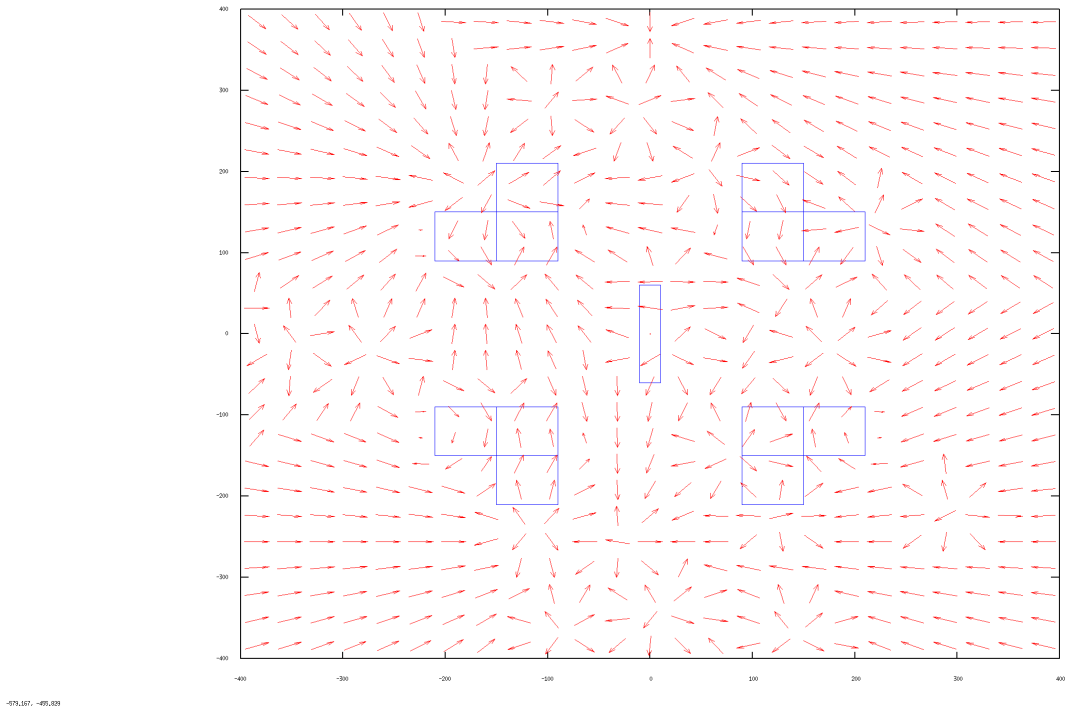

Figure 8. All Fields - Midgame

Tuning Process

The tuning process for our potential fields took some time to get to a state where we agreed that it was good enough. The fields each have a sphere of influence that we tuned to be just large enough to cause the necessary reactions from our agent. The sphere of influence around the enemy tanks was reduced to 100 units in every direction to prevent the agent from always being influenced by an enemy repulsive field instead of a flag attractive field.

Tuning the tangential fields was by far the most difficult. Whenever we turned on the tangential fields our tanks started to oscillate no matter where they were. We thought that it must be because our tangential fields are influencing them no matter where they are on the board. We would turn off the tangential fields and the other two fields would work perfectly. After hours trying to tweak how the

tangential fields worked to no avail, we decided it was not a problem with the tangential fields themselves. We examined our code closely and found that the function we were using to get the closest obstacle was querying the server for all of the obstacles on the map every time! This was unnecessary seeing as obstacles never move. After we stopped that silliness, the tanks stopped oscillating.

Now that the tangential fields were working, we tried to determine their radius by trial and error. We would start and stop our server while tweaking the obstacles' sphere of influence a little bit each time. Finally, we decided that that was much too tedious and came up with a brilliant idea. The idea was to take the distance from the center of the obstacle to its furthest corner and make that the radius for the sphere of influence. After we applied that logic, the tanks worked perfectly how we wanted them to.

The last small change we made was with the repulsive fields. We decided we wanted the tanks to turn the exact opposite direction of the enemy tank when they entered the enemy tank's sphere of influence. We also wanted them to maintain their current speed as they turned so they could get away faster.

Tests against our own agents

The first test was putting one of our potential field driven agents up against a copy of one of our dumb agents. During this test I learned that our agent is suicidal and since we had decided to allow friendly fire in all of our tests, I noticed that several of the tanks being controlled by the agent would compete for the same flag by shooting the tank that was carrying it. This act in and of itself caused the time it took for the agent to retrieve a flag to be rather long. Also, the agent spent a lot of time going for the bases guarded by the teams that were not being controlled. This also increased the retrieval time for

a flag due to the time spent heading towards the non-controlled team's flag, and then circling the stationary enemies at the base.

The second test was putting our PF agent up against two copies of our dumb agent. Oddly enough this decreased the amount of time our agent took to retrieve a flag. With two teams of tanks out patrolling, this left two bases somewhat less guarded, and the agent had an easier time retrieving the flag from each base. The dumb agents fired often enough to sometimes make it hard for the agent to get the flag, but in this test, and the previous test, it seems that the biggest hinderance to the agent's retrieval rate is the agent itself.

The third test was to put two copies of our PF agent against each other. This was a more exciting match to observe. Both agent controlled teams made considerable effort to go out and retrieve flags. Both teams attacked the bases of non-playing teams and stole their flags. Also, the agents fired as often as the reload speed of the tank would let them, so each team was able to keep the enemy at bay through mostly lucky shots.

I saw several unique behaviors that are of note. None of these behaviors were programmed by us, but they seem as if they could prove as useful tactics in the future. The first behavior of note was that I saw a dumb agent push a potential field agent out of the way of getting the flag. This struck me as interesting because I never considered ramming another tank as a viable option of protecting the flag. Another interesting behavior that I saw was that when friendly fire was on, at the beginning of the match our potential field agents would all fire a few shots and destroy half their own team before leaving the base. Due to the suicidal nature of the tanks this caused there to be an exceeding amount of respawns at the home base. This seemed like a useful tactic to me because it is as if the tanks are in two places. A few tanks destroy the enemy tanks and are then destroyed. When they respawn they are able to

quickly defend the base from any attackers. This strategy prevented the potential field agent's flag from being taken on multiple occasions.

Tests against other teams' agents

Before we had our tangential fields working we did some testing against a few other teams from our class.

First we tested against Dustin Maxfield and Matt Hess. Against one of their dumb agents, our agent had no problem. It would go back and forth between their flag and our base scoring at will. If any of our tanks got within the their tanks' repulsive field spheres of influence, our tanks would turn out of the way and continue on their path to get the enemy flags.

Next we tested against two of their dumb agents. Once again, our agent would score at will. We would get hit more often by their dumb agent tanks, but it was still no problem for us. This time, since we didn't have tangential fields working yet, one of our tanks got stuck against a corner of one of the Ls for a minute.

Next we tested against their potential fields agent. Their agent did not send all of its tanks toward the enemy flags so ours scored a lot more points since we did send all of our tanks toward enemy flags. Thier tanks also seemed to get confused when they were close to our base and would drive around aimlessly it seemed.

Second we tested against Matt Nielson and Alex Lemon. Their dumb agent would only move one of its tanks around so this was a much easier agent to beat than our own dumb agent (our dumb agent sends all of its tanks on the circular shooting track). Our agent easily beat both one and two of their dumb agents, hardly ever getting hit by their bullets.

Next we tested against their potential fields agent. Their agent would only send one of its tanks to get the enemy flag and they hardcoded in which flag it would go after. Our tanks would shoot that tank inadvertently as they drove toward the flag of their agent. All of their other tanks besides the one that would go after our flag, would hang back to defend. They would hardly ever hit our tanks as we drove past them, stole their flag, and returned it to our base. We won very soundly.

After we got our tangential fields working and made the final tweaks our repulsive fields, we tested against Trevor Buckner and Samuel Gubler. Their agent was similar to ours except their tangential fields needed some more tuning. They had programmed their agent to be more likely to attack enemy tanks that were carrying flags. A behavior that they said they planned to implement that was different than our agent was that if the team flag was dropped off of the team base, then the agent controlling that team would go and pick it up to return it. Our agent one because of our programming all the tanks to leave the base and go seek out flags and thus destroying anything in their path.