Potential Fields Lab 1 Writeup

21 Sep 2015

Tyler Monson
Austin Soderquist

Time Spent

- Tyler – 10 hours
- Austin – 8 Hours

Implementation Notes

For our implementation, we had an attractive force on the goal (flags, and home base), a repulsive field for each obstacle, and a tangential field for each obstacle.

For the attractive field, we determined the location of the flag (or the center point of the home base) and made a radius half the size the width of a base around that point. Agents inside the circle created by this point and radius experience a smaller force towards the point as the agent gets closer to the point. If the agent is outside of the circle, it experiences a force towards the goal point that is capped out with a constant value. We did this so agent does not experience tremendous amounts of force the further it is away from the goal.

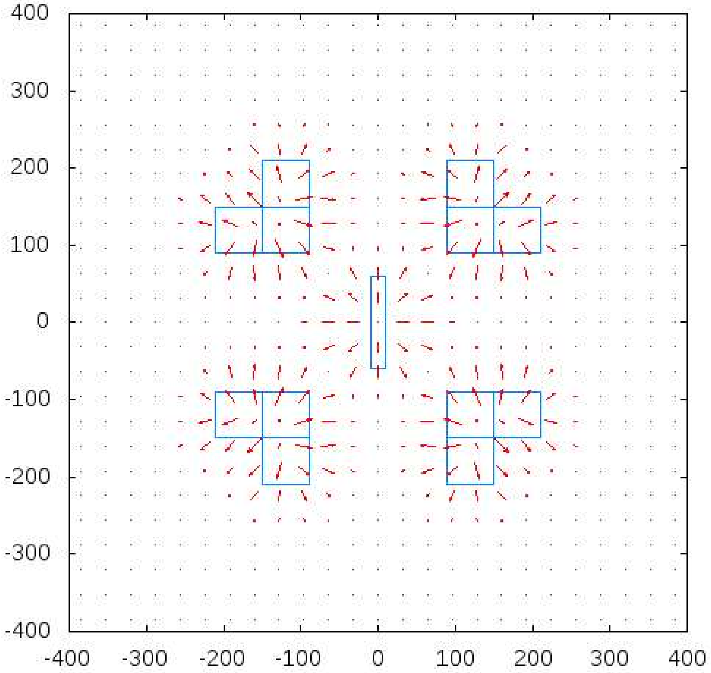
The repulsive fields related to the obstacles are calculated in a similar, but more involved manner. Basically, we drew an "inner circle" around the obstacle. Any agent within this circle experienced a strong and constant repulsive field that simulates a close to infinitely strong force. Any agent outside of this "inner circle" was tested to see if it fell in the "outer circle". Agents experienced a stronger repulsive force as they got closer to the obstacle's "inner circle" and were inside the "outer circle." Agents outside the "outer circle" did not experience any repulsive force from that particular obstacle.

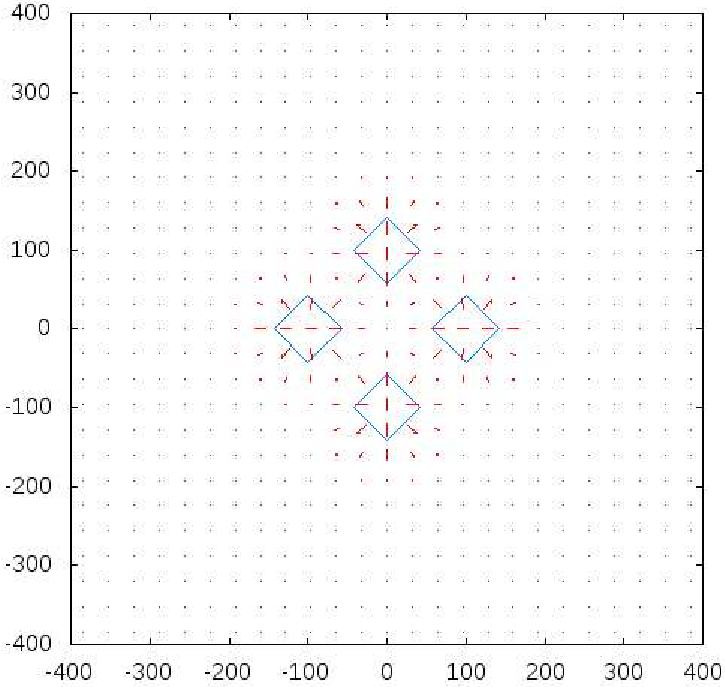
Calculating the tangential field for each obstacle was very similar to the repulsive fields of the obstacles. Our method for this also used an inner and outer circle. First, we calculated the distance between the center of the obstacle and the agent. If the agent was inside the inner circle, it experienced a stronger tangential force. If it was outside the inner circle, but still inside the outer circle, it experienced a proportional tangential force. Any agent outside these circles did not experience tangential force from an obstacle. We used the combination inner circles so the agent could have more power to get around the object if it was too close. We also used the proportional outer circle so the agent could try to move around the obstacle without getting too close.

To create the final force acting on the agent, we took the vectors from all of the above mentioned fields and added them together. From this, combined with the use of PD controllers, we were able to determine the agent speed and angle velocity.

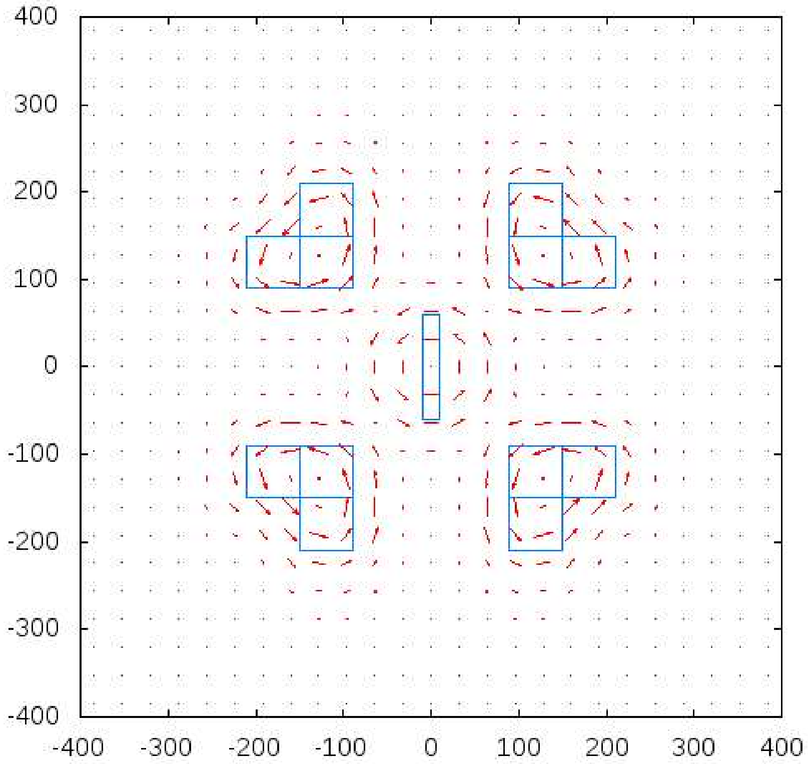
Plots

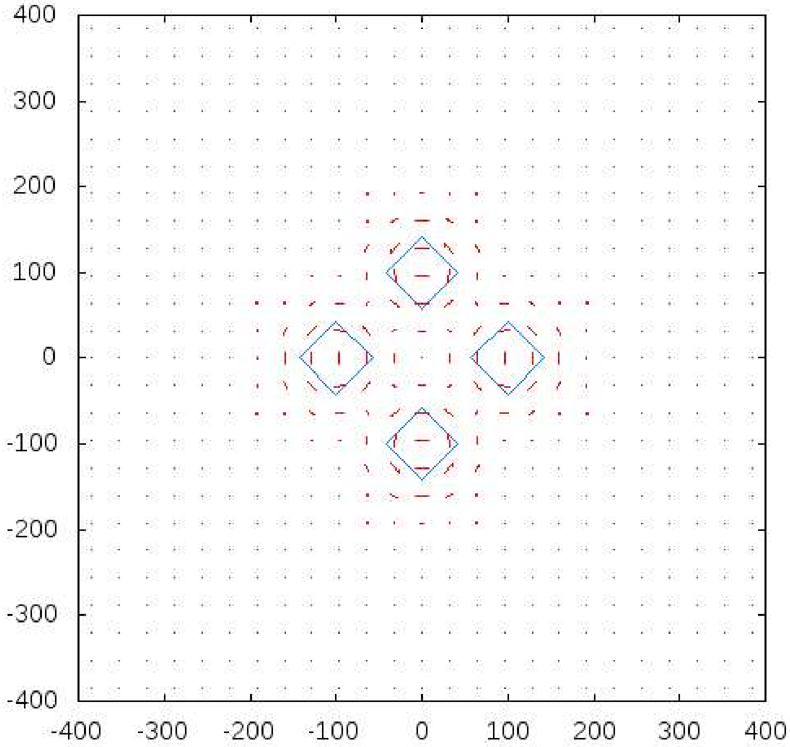
Repulsive Fields



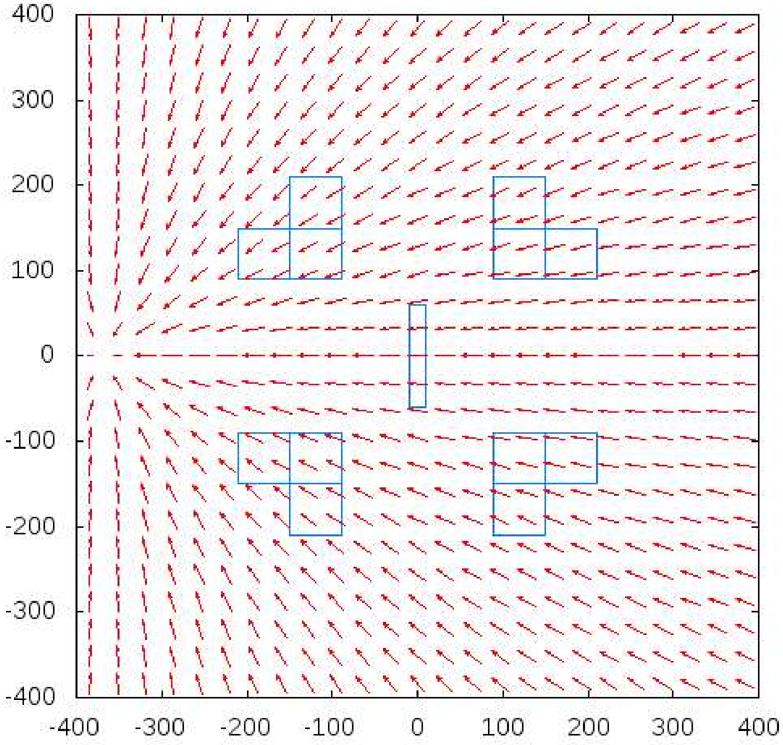


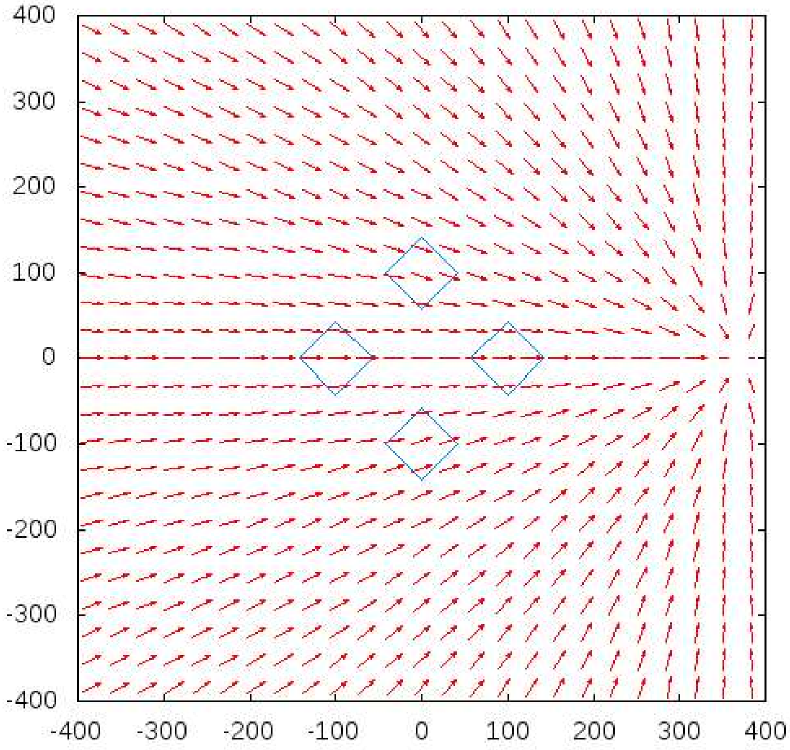
Tangential Fields



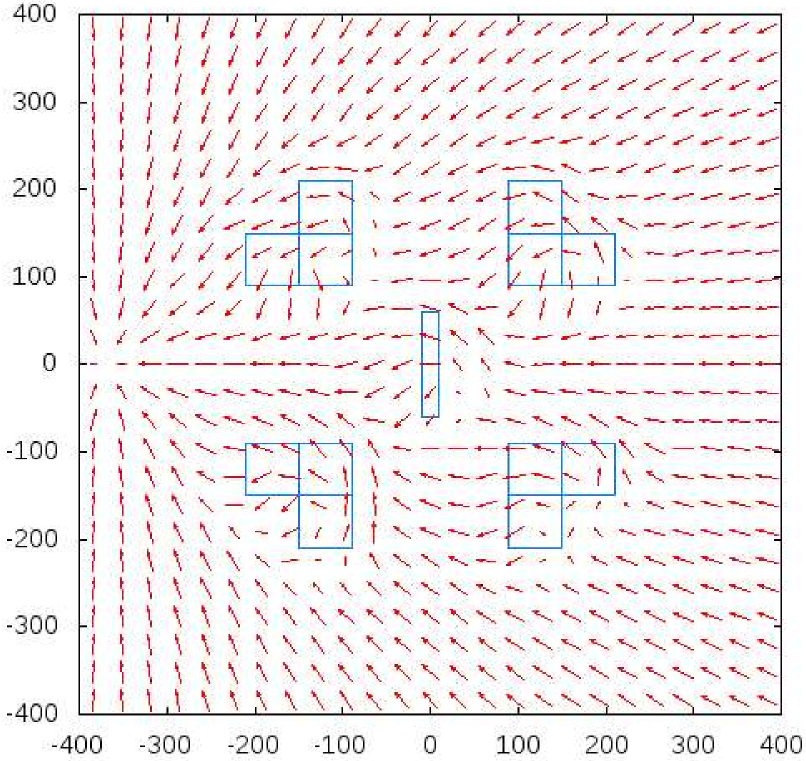


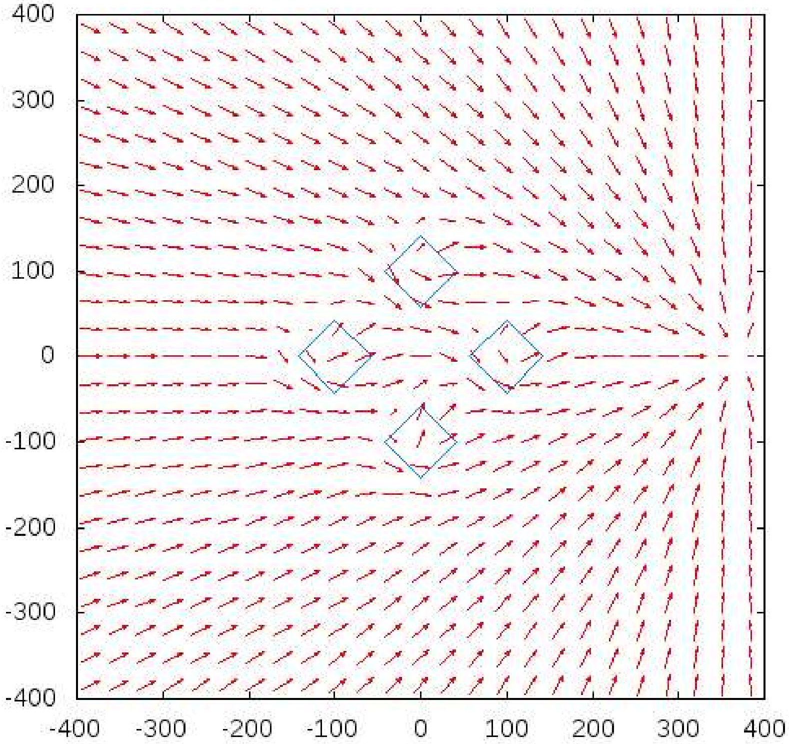
Attractive Fields





Combined Fields





Tuning Process

Our tuning process involved adding and modifying force constants relative to each field. For instance had force constants for the goal force, inner and outer tangential forces, as well as inner and outer repulsive forces. These variables gave us a lot of opportunities to tweak our fields. To tweak the fields, we would watch our agent move through the world and make changes based on its reactions. For example, if we saw the agent getting too close to an obstacle, we would tweak the tangential and repulsive fields constants. Other times, if we noticed the agent was not very attracted to the goal, we would increase the constant related to goal attraction.

Alternative Options

Alternatives that we have considered include adding a random field, just to help with some sticky spots in both maps. We also considered using an inner square field for the obstacles, instead of an inner circle. This might have worked better for obstacles like the long skinny one in the four ls world.

Test Results

Own Code

One dumb agent

When we ran our smart agent against our dumb agent, we consistently stole the flag from the dumb agent. Because the opposing agents and their shots didn't have any sort of field applied, our agents would occasionally crash into them or get shot.

Two dumb agents

Similarly to the test against one agent, our smart agents would occasionally get shot at or crash into a dumb agent. In addition, the way we determined the flag to capture, our smart agents would try to capture the flag from one of the dumb agents more often than the other agent. Regardless of the increase in dumb agents, our smart agents were still able to capture the flags.

Against copy of “smart” agent

Due to the way we had coded our potential field agent, when we pitted it against itself in a four team world, One team did much better than the other. This is probably because we didn’t include code that would cause the agents to go towards the nearest flag. Instead, it aimed for the next flag in the order they came. In general, this meant that the agents would aim for the red flag first, then the blue, green and purple flags.

Opposing group

We tested with Sam McDonald and Bryant Chandler.

Dumb agent

The other team's dumb agents acted differently from ours. Their dumb agents shot more randomly than ours, but they were also clumped closer together. Regardless of the change in challenge, our smart agents were still able to capture the flag frequently.

“Smart” agent

When we pitted our smart agent against the other team's smart agent, we were playing on a four team map with our team on one side of the map and their team on the other. We lost poorly, because their smart agent would go towards the nearest flag, while ours always targeted the same flag until it was captured by one of our tanks. Then the other tanks would target other flags.

We also (for fun) had two teams be our agents while their agents were the other two teams. Both of our teams lost very badly, while one of their teams won by a land slide. This is because both our teams were targeting only one of their teams for the majority of the match. This allowed the team that won by a land slide to gather flags from which ever base they wanted with very little counter fire.
lost badly.

During some of these games, we noticed that some of the flags were dropped near obstacles when the holder was shot. When we saw this, we realized that our smart agent would probably have a hard time picking up the flag. This guess is based on the fact that the strong repulsive and tangential forces near the obstacle would make it hard for the agent to get close to it in some cases. We think it would be fun and interesting to continue frobbing so we can get our agents close to the obstacles when we want to.