Christopher Williams - s0955088 - SDP 7

Performance Review 3 - March 7, 2012

Introduction: I have been working on the Planning and Strategy of the robot. This includes working on the A* path finding and developing the ”vehicle” that creates and carries a plan to the Control Interface of the Robot. I have also been keeping the UML diagram up to date

Strategy & Planning: I have been working on the structure of Strategy and implementing the code and therefore classes behind the UML diagrams which I designed in the previous weeks. I have also been a component in keeping my colleagues' code strictly to the design model.

Aim: To build the framework that I discussed in the previous performance report and test the planning structure

Methodology: Firstly a simple structure was created to test out the threading, whereby a few classes were implemented that would start a thread and create a few simple objects. This was to ensure that it was sensible and feasible to use a Thread to make Plans in the first instance. This worked successfully, proven in a literal sense by use.

Secondly the real classes were implemented, a Thread would create an object called Plan, which in turn would take all of the needed variables from the worldState, and pass them through a Grid class, which in turn uses Astar to create a path. this original design had many flaws. One of the most obvious was the unnecessary Grid class, which was a class which did not make sense in the model as it ended up with some of the decision structure inside of it. A new UML diagram was built, whereby all the information required to make a plan was put into two separate objects, AllMovingObjects and AllStaticObjects, and the conversion into the node points that Astar requires now happens inside these classes, meaning that all of the decision making (AI) could now happen in a much easier to read TargetDecision class, whilst at the same time not overloading the class with Astar conversion methods. Essentially a considerable portion of time was spent on creating classes, and discovering that they were not human friendly, or sensibly named or placed with in the structure as a whole. This was proven by following through the lines of communication between the objects themselves and seeing what the result could be. After a few iterations in design a new Model was ”organically” created.

At the same time the algorithms and mathematics behind the AI were also developed (and moved to TargetDecision), these original designs also had some flaws: for example initially we were expecting the robot to drive to the ball, and then turn to the best angle to shoot and kick. This idea was flawed because the robot could end up being either ”north” or ”south” of the ball, and thus turning to empty space to kick. This was solved by modifying our Path, so that the Astar is given a navigation point to drive through, this navigation point is always on a line drawn from the goal centre, through the ball, so that if the Robot were to get to this nav point, and then approach the ball it will always be on target. TargetDecision also had a period of constant redesign, until finally settling down to the current structure, still yet to be finished.

Using a visual overlay provided by a fellow student we were able to map the path and obstacles onto the GUI and prove that the planning system was working; although the Targeting System and Astar path finding were not complete (the Targeting system had a naive goal angle system) or working as successfully as we would like (in the case of Astar there was an issue with having a ball inside an obstacle area) the framework, and Plan delivery system were working. Working can be defined in terms of consistency, and accuracy. The plans were consistent and not switching navigation / target point without sensible reason, and were accurate in that they found the ball, the goal, and the start point for the robot.

As stated I have also been upgrading the Astar Path finding algorithm that we originally started with, this was due to the nature of our Astar only being able to have binary obstacles, so that for example if the ball was too close to the opposition (a drawn obstacle) then the ball would appear inside an obstacle, causing the Astar to fail. The answer to this problem is to make the Astar deal with cost, instead of obstacles, so that there is a high cost to enter into an obstacle, it will only go through a high cost area if it has to.

Conclusion: There are still many small areas that need addressing, however the main structure is in place and is successful.