Background:

*Transport Tycoon Deluxe* (TTD) is a business and construction simulation game released by MicroProse in 1995. In TTD, players attempt to earn profits but transporting passengers and goods using roads, rails, airplanes, and boats. Players are given a random map populated with cities, industries, natural resources, and obstacles such as trees and water. They must create a transportation network connecting the sources of passengers (cities) and goods (industries and natural resources) in the most efficient and profitable way possible. In addition, they must maintain this network and deal with in game events such as vehicle aging and breakdown, technology upgrades, offered subsidies, and actions by competitors operating on the same map.

*OpenTTD* is an open source clone of TTD that provides an artificial intelligence framework called *NoAI* that allows the development of AIs that play TTD. AIs are programmed in Squirrel, and an API is provided that allows the AIs to access and play the game in all the same ways that a human player would.

Goals:

Our goal is to develop an AI for OpenTTD that creates an efficient transportation network that allows for maximum profitability. This AI will attempt to implement an algorithm to plan a theoretically efficient network while working within and adapting to the constraints of the TTD environment (ex. construction costs, terrain features, subsidies, and competitor actions).

Methods:

TTD allows for the creation of a wide variety of transportation systems - roads, railroads, airports, and seaports. These networks can connect a variety of destinations. Passenger and mail traffic can be carried between towns and cities. In addition, resources can be carried between complementary industries and between industries and towns. For example, steel from a steel mill may be transported to a factory whose goods may then be transported to a city for consumption.

Due to the constraints of working with a new programming language to build an AI program from the ground up, we will limit ourselves in the types of transportation
methods we use and routes we construct. We will start with an AI that builds roads and railroads to connect towns with other towns. This will allow us to save time in the implementation of basic functionality so that we can focus more on the interesting aspects of the project – creating an efficient and innovative AI. Despite this initial limitation, we will write code that can easily be generalized to expand our approach – for example to allow the construction of routes between industries or the construction of airports. Should we then choose to alter our methods or focus, adaptation of our AI will be easy.

Our AI will consist of two major levels of operation. The ‘lower’ level of operation will perform tasks that allow for the construction of transportation routes between destinations. Given two towns, this level of the AI will build a route connecting the towns. It will determine the placement and construction of new transportation infrastructure (such as stations, roads, and railroads), integrate the new route into existing infrastructure, create vehicles to travel along the route, and take into account various constraints including natural obstructions, previously existing structures, and previously built routes. The level of the AI will be the primary focus and responsibility of Charlie.

The second level of the AI will determine which routes to construct. It will design an efficient network to connect the towns on the map and will create a plan for the order in which routes in this network should be constructed. It will also take into account current information such as budget constraints, the offering of subsidies, and the actions of competitors to update and alter its plans. Abstractly, this level of AI will create and maintain an ordered list of routes to construct – where each route is defined by two towns to be connected. This list will be ‘executed’ by the first level of the AI described above. The second part of the AI will be the primary focus and responsibility of Cameron.

General Network Model:

Although a large part of our (and especially Cameron’s) early work on the project will be researching and evaluating methods of network and routing optimization, our general model will be as follows:

Our network will consist of three distinct types of routes described below:

*Intra-city routes:* In large towns (cities) these short routes will be used to capture passenger traffic throughout the city. These routes will all feed into a primary station, which will act as a departure and arrival point for all passenger flow into and out of the city. Intra-city routes will largely be roads serviced by busses.

*Feeder routes:* These routes will connect the primary stations of cities to larger transfer stations. Transfer stations will be located using an algorithm that minimizes cost of transport to nearby cities (ex. a center of mass or network centrality calculation may be used). Each station or small group of stations will correspond to a region of the map and construction of feeder routes to a transfer station will occur independent of construction
of other transfer stations and feeder routes. Transfer stations may or may not be located within towns. The locations of transfer stations will be predetermined to allow for an organized network construction. However, they may change with changing information such as competitor actions and the offering of subsidies. Once a transfer station is established it will be treated as a town in the sense that the second level of the AI, described in the previous section, will be able to order the construction of routes between the transfer station and nearby towns or other transfer stations. Feeder routes will largely be railways, which have higher capacities than roads.

*Inter-region routes:* These high-capacity routes will connect transfer stations to each other and will largely be determined by the placement of feeder routes and transfer stations. Once two transfer stations are well developed and serviced by sufficient feeder routes they will be connected by an inter-region route. These routes will be high-capacity railways, perhaps consisting of multiple tracks and trains.

**Deliverables:**

1. The source code of our final AI. This source code will also be available on the Internet as part of the open source nature of the OpenTTD and NoAI projects. Especially due to the easily adapted two tier structure of our AI, we hope that it will be a valuable resource for future developers of OpenTTD (and other) AIs.

2. An individual final paper for both students discussing our specific roles and focuses in the design of our AI, the process and theory behind the development of this design, and an analysis of the strengths, weaknesses, and possible expansion points of the design. As the project unfolds throughout the semester it is sure to develop and change course and our papers will chronicle the development process.

**Resources:**

OpenTTD Homepage:
http://www.openttd.org/en/

OpenTTD Manual and Wiki:
http://wiki.openttd.org/Main_Page

NoAI Homepage:
http://noai.openttd.org/

NoAI Manual and Wiki:
http://wiki.openttd.org/AI:Main_Page

NoAI API Documents:
http://noai.openttd.org/docs/trunk/