Ceng 491 Proposal: Develop a Simlink for Simulating Many

Team: Random Soft
Samet CAN,
Okan HARPUT,
Emre AKSAN,
Çağlar GÜNDOĞDU

Tuesday 23rd October, 2012

1 Motivation

A lot of humans, animals, vechiles are interacting with each other on a daily basis and these interactions may vary in a large-scale. Often understanding and estimating the behaviours of these agents is a hard task. To understand, estimate the interactions of these agents in a large crowd or even visualization is a common problem due to some reasons such as cost of real demonstrations, security issues, etc. For these very specific cases where visualization of the behaviors of agents is important simulation technology helps us.

While imitating the real world and gathering realistic responses are possible, simulating many agents can be costly and usually ineffective with common simulation techniques. The need for crowd simulation arises when a scene calls for more characters than can be practically animated using conventional systems. Simulating crowds offer the advantages of being cost effective as well as allow for total control of each simulated character or agent [1].

To produce an unusual result and to benefit from built-in features of some external systems like GIS, game engines, databases or virtual globe systems, simulations can be linked to such systems.

In this project, based on a scenario, movement of a large number of entities will be simulated by linking crowd simulation with an external system.

2 Background

Several external systems can be linked to a simulation according to purpose of the simulation. To store and process big amount of data, simulation may interact with a database system. Or to provide high quality visualization, more realistic environment in the simulation, game engines' features like 3D graphics renders, physics engines or artificial intelligence can be exploited. Or instead of creating a virtual world, map of a location/area can be used as simulation environment [2].

There are several systems that can be linked to simulation. To facilitate high quality visualization Unreal Engine can be used since unreal engine provides a powerful and efficient 3D game simulation and visualization environment [3]. It not only incorporates highly optimised 3D graphics rendering capability but also has level editing tools that can expedite the modelling of agent's test environments [4]. Unreal Engine also uses a rather low-level scripting, also object-oriented, language which seems like C++ and Java, and is flexible to made jobs that includes AI logic.

Google Earth can be another external system that visualizes the real location of the crowd by updating coordinates of the entities on the map over time. Morever, the effort spent to create identical world is prevented.

The High Level Architecture (HLA) provides the infrastructure needed for large-scale distributed simulation and defines rules to support interoperability of different simulators [5]. In HLA terminology, each simulation component is called as federate. Using HLA enables interactions between each participating federates. In our case, unreal game engine and google earth are two different federates that must collaborate.

The communication between the simulation and external systems that send/receive current state information or messages of the corresponding federates is another problem. MAK technologies offers a product namely HLA GAME-LINK that allows for an HLA compliant Unreal Engine and thus would reduce any additional overheads required for making an HLA compliant adapter for Unreal Engine [6].

3 Proposal

In this project, simulation of many agents, based on a scenario will be implemented. For such a crowd simulation to be useful, the system must support the modeling of realistic individual and crowd behaviours of large number of entities. While there are existing tools for work on crowd simulation, we will take advantage of features from external system(s), Unreal Engine or Google Earth or both. A bridge/interface between the simulation and external system(s) will be created to drive the visualization/animation of virtual 3D space.

References

[1] Crowd simulation wikipedia: http://en.wikipedia.org/wiki/crowd_simulation.

[2] 3d driving simulator on google earth: http://geoquake.jp/drivingsimulator/earth/.

[3] Lewis M and Jacobson J. Game engines in scientific research. In *Communications of the ACM*, January 2002.

[4] Qasim Mehdi Pawan S Kumar and Norman Gough. High level architecture for distributed agent simulation in computer games.

[5] Suiping ZHOU Malcolm Yoke Hean LOW, Wentong CAI. A federated agent-based crowd simulation architecture.

[6] Qasim Mehdi Pawan S Kumar and Norman Gough. High level architecture for distributed agent simulation in computer games.