Project Number	TQP105
Project Title	Smart Terrains
Project Description	Smart terrain is a term used to describe a virtual world with objects empowered by AI. The smart objects pass information about themselves to the game agents. With this information, the game agents can make different decisions of actions and behaviors. This enables the game agents to have emergent behaviors. A good advantage of smart terrain is that artist can specify the actions or behaviors of the AI agent on an object. For example, an artist can specify the racing AI to totally avoid huge rocks and purposely specify the racing AI to slightly brush through a wooden box. The aim of this project is to add AI into all intractable objects in the game that helps the AI improve the competitive by getting more information from the objects when the AI is near them. Objectives: • Create an obstacle-racing mode in which the players and AI will play together in the same track which has no obvious way to finish the lap because of superabundance of obstacles. There will be a lot of objects scattered along the track to help the players to pass the obstacles like flying over a ramp, driving through a gate or window. In such scenarios, human player can realize what they can do with each type of the objects, so the creativeness is unlimited. However, AI players need the objects to tell them what to do with the objects. This is achievable using smart terrain as the data is embedded in the objects and the messages will be sent to AI players so they can know how they should act. • When the system contains too many AI objects like this, the processing requirements increase manifold. A scalable dynamics system becomes inevitable to be used here. Dynamics is the computation of anything that is changing like: AI is the dynamics of character's behaviors, physics is dynamics of natural phenomena, keeping track of resources in a world is dynamics. Scalable dynamics allows us to add size and complexity at low cost for massive environments in which we only want to pay for what we see.

	Current techniques include: • Dynamic culling: Do nothing for what will not be seen. • Simulation proxies: Reduce the cost of dynamics for not visible objects. • LOD: For rendering as well as for dynamics like physics and AI With these techniques, the challenge for scalable dynamics is being able to quantify the changes and deciding the metrics for measuring changes.
Hardware/Software/ References	• Scalable Dynamics – Stephen Chenney, University of Wisconsin – Madison. • http://www.cs.wisc.edu/~schenneyHardware/Software/ • XLEngine • AI wisdom 2 page 23. • Microsoft Visual Studio 2005 C/C++ environment