Game AI R&D Project

Implementation Report

Synopsis: This document outlines the Immersive AI final implementation details and contains all source code developed for the system.

Reference: GameAI.Implementation v1.0

Date: 23 October 2006

Author: Gavin Bunney & Tom Romano

Status: Definitive

iAI

immersive ai engine

Document Control

Version History

Every change to this document is logged in the table below.

Ver.	Date	Author	Description
v0.1	2006-10-23	Gavin Bunney	Initial document creation
v0.2	2006-10-23	Gavin Bunney	Added initial C++ class outlines
v0.3	2006-10-23	Gavin Bunney	Completed C++ class outline
v0.4	2006-10-23	Gavin Bunney	Added commenting/version control information
v0.5	2006-10-23	Gavin Bunney	Added initial TorqueScript class outline
v0.6	2006-10-23	Gavin Bunney	Completed TorqueScript class outline
v0.7	2006-10-23	Gavin Bunney	Added C++ source source code
v0.8	2006-10-23	Gavin Bunney	Added TorqueScript source code
v0.9	2006-10-23	Gavin Bunney	Added SVN Log
v1.0	2006-10-23	Gavin Bunney Tom Romano	Definitive Issue

Project Abstract

Implemented game AI, particularly in RPG/MMORPG games, is very script based. The NPC's walk in a set path, speak with set scripts; the mobs have scripted actions. The concept behind the project is to create a more realistic AI, to both provide unpredictability in NPC behaviours, and to immerse a player in the game world.

The project is to research AI techniques, design and implement a goal-based AI system. Goal-based AI is a technique used to create NPC's which act as real players; in that they are given an objective to achieve - e.g. Given a goal of "rake leaves" it is up to the NPC to work out how to achieve their goal, whether through buying a rake, stealing one, or killing another NPC for their rake.

The implemented AI system will be in the form of various classes, created in C++ and Torque Script for the Torque Game Engine, version 1.4. For more information on the Torque Game Engine, visit http://www.garagegames.com.

Contents

List of Figures

1 INTRODUCTION

1.1 Purpose

The purpose of this document is to outline the final implementation details, source code and subversion log of the Immersive AI engine.

1.2 Scope

The scope of this document is limited to the Immersive AI engine which was finalised on 13th October 2006.

2 VERSION CONTROL AND COMMENTING

2.1 Version Control

Subversion (SVN) was used throughout the project to ensure strict version control. It has been instrumental in allowing multiple concurrent users to work on the same source code, without any problems in incremental change merging.

The Subversion log is attached in Appendix A – SVN Log.

2.2 Commenting

The dOxygen commenting system was used in both the C++ and Torque Script files, to allow a common function documenting system and generation of an API document. Due to the vast number of functions and various classes, it has allowed multiple users to lookup functions in an easy-to-use web interface, without having to search through the actual files for the function calls.

3 C++ CLASSES

The Immersive AI source code forms part of the existing Torque Game Engine C++ files. The C++ classes developed enabled 'heavy-lifting' to be performed in fast C++ code, whilst more detailed, easily changeable implementation was in the various Torque Script Files.

The C++ classes were implemented in a separate folder, within the TGE game engine, with the following structure:

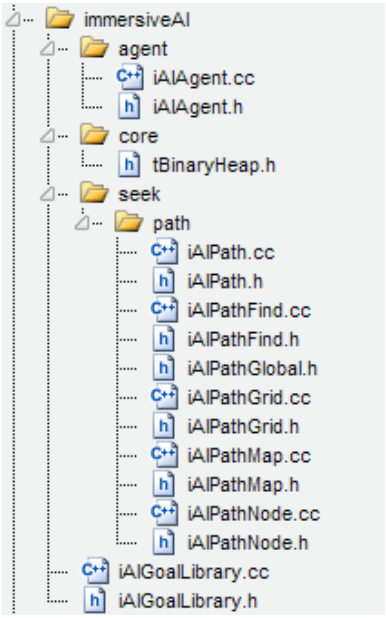

Figure 3-1 C++ Class Folder Structure

3.1 iAIAgent: iAIAgent.h/.cc

The iAIAgent class represents the actual game agent's. It is extended from the core AIPlayer class, bundled with TGE (which handles movement of the player). The iAIAgent class holds the various accessors and mutators for the agent vitals – health, happiness, boredom, fatigue, money and agent type.

The source code listing is attached in Appendix B – IAIAgent.h/.cc source code.

3.2 iAIPath: iAIPath.h/cc, iAPathFind.h/cc, tBinaryHeap.h

The iAIPath class represents an actual path in the game world; an interconnection of iAIPathNode's for an agent to traverse from one point to another.

The iAIPathFind class is a singleton implementation of the A* path finding algorithm (for more information please see the iAI Research document), which utilises the template class, tBinaryHeap, for sorting of the nodes. The use of the tBinaryHeap class has increased the calculation speed, and thus the number of simultaneous path requests, exponentially compared to a simple quick sort.

The source code listing is attached in Appendix D – IAIPath.h/.cc source code, Appendix E – IAIPathFind.h/.cc source code and Appendix F – tBinaryHeap.h source code.

3.3 iAIPathMap: iAIPathMap.h/cc, iAIPathGrid.h/cc, iAIPathNode.h/cc

The iAIPathNode class represents a single point, or node, in the game world. A multitude of nodes are placed throughout the world and interconnected, forming the iAIPathGrid.

The iAIPathGrid attempts to place nodes throughout its given start and end dimensions and interlinks them with their closest neighbours. At the present time, only a single grid is created for the terrain, however, the class has been written to be extended for individual grids to be created for interiors and for instances such as closer density around objects.

The iAIPathGrid's are linked together into a central iAIPathMap, forming the path finding map, utilised in the iAIPathFind class.

The source code listing is attached in Appendix G – IAIPathMap.h/.cc source code, Appendix H – IAIPathGrid.h/.cc source code and Appendix I – IAIPathNode.h/.cc source code.

3.4 iAIGoalLibrary: iAIGoalLibrary.h/cc

The iAIGoalLibrary is used to hold the list of goals/solutions. It provides a structured interface to the Torque Script classes by allowing adding, removing, sorting and requesting lists of goals/solutions for various agent types.

The source code listing is attached in Appendix J – IAIGoalLibrary.h/.cc source code.

4 TORQUE SCRIPT CLASSES

The Immersive AI Torque Script classes provide a flexible implementation of actual game behaviours. They form part of the 'Server' folder, as all agent handling is performed on the Server side. The Torque Script classes interface both to existing Torque Game Engine C++ and the iAI C++ Classes to perform the various logic.

The Torque Script classes were implemented in a separate folder, with the following structure:

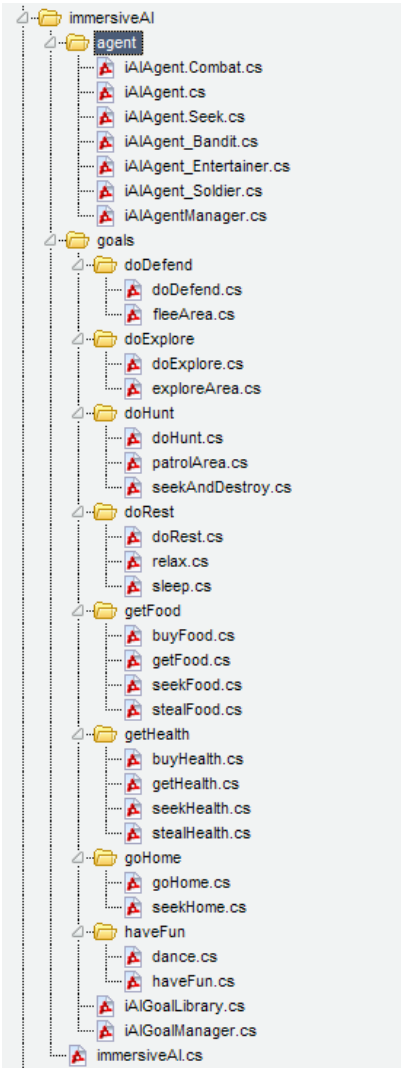

Figure 4-1 Torque Script Class Structure

4.1 Immersive AI

This core file is executed when the game server starts up. It is the central calling point for the Immersive AI system and thus executes each of the other iAI Torque Script classes. It contains all the initialisation calls; the path map is generated, agent manager setup and empty goal library created.

The source code listing is attached in Appendix K – Torque Script: immersive AI Source Code.

4.2 Agent

The Agent collection of classes contains the logic for both the general agent and for each sub-type of agent. The main iAIAgent class contains the logic for all agents, with the iAIAgent_Bandit, iAIAgent_Entertainer and iAIAgent_Soldier classes containing the logic specific for that agent type.

The iAIAgent.Combat and iAIAgent.Seek classes are a subdivision of the iAIAgent class and provide the Combat and Seek functions utilised in the goals; the interfaces to the engine which perform the Combat and Seek module logic.

The iAIAgentManager class is simply a monitor which checks whether to spawn more agents into the world.

The source code listing for these classes is attached in Appendix L – Torque Script: Agent Source Code.

4.3 Goals

The Goals collection of classes contains all the goal/solution implementation details. It also contains the interface class, iAIGoalLibrary to the C++ iAIGoalLibrary.

Each of the goals, in the subfolders, registers themselves with the iAIGoalLibrary and provides a standard interface for evaluation and execution. This self-registration of goals/solutions allows for goals/solutions to be available for a single agent type only and allow the goals/solutions to be implemented however desired; as long as it contains the standard, evaluate, onEnter, execute and onExit methods.

The iAIGoalManager is used by agents to request new goals/solutions. It contains a very flexible interface to request new goals, new goals (but not the current goal), new solution (but not the current solution), and many more.

The source code listing for these classes is attached in Appendix M – Torque Script: Goals source code.

5 CONCLUSION

The core C++ classes developed allowed for various computationally heavy algorithms (such as the A* path finding) to be implemented in a fast and efficient manner. The TorqueScript interfacing to the core engine code has also allowed for a more module design (such as the iAIGoalLibrary).

The use of subversion and the dOxygen commenting system has been instrumental in allowing multiple developers to work on the same system, without any incremental merging problems.

6 GLOSSARY

Item	Description
AI	Artificial Intelligence - The ability of a computer or other machine to perform those activities that are normally thought to require intelligence
Agent	A computer controlled entity within a game environment
Ghosted	The process of simultaneous updating of server-side objects to any connected clients; used for client-server architectures where all objects are created on server side and displayed on clients
Goal	The purpose toward which an endeavor is directed, an objective; describes the AI agent's current overall objective.
RPG	Role-Playing Game - A game in which players assume the roles of characters and act out fantastical adventures, the outcomes of which are partially determined by chance
State	A condition of being in a stage or form; describes the AI agent's current action algorithmic state
SVN	Subversion – Modern replacement for CVS as a comprehensive version control system.

7 REFERENCES

- Bunney, Gavin & Romano, Tom. “iAI Research” 2006 irombu.com, Brisbane.

8 APPENDIX A – SVN LOG

Revision: 191

Author: tom

Date: 1:09:17 PM, Friday, 13 October 2006

Message:

fixed foodItem positions

commented the animation fix thing did previously as was buggy if dead (i think)

Modified : /Game/client/data/missions/stronghold.mis

Modified : /Game/server/immersiveAI/goals/iAIGoalManager.cs

Revision: 190

Author: tom

Date: 12:52:24 PM, Friday, 13 October 2006

Message:

changes to goalManager assignGoal to reset their animations....

Modified : /Game/server/immersiveAI/goals/iAIGoalManager.cs

Revision: 189

Author: gavin

Date: 10:53:18 PM, Thursday, 12 October 2006

Message:

added scheduling of refresh button

Modified : /Game/client/immersiveAI/iAIControlCentre.cs

Revision: 188

Author: tom

Date: 10:22:36 PM, Thursday, 12 October 2006

Message:

updated mission to have new healthVendor

Modified : /Game/client/data/missions/stronghold.mis

Revision: 187

Author: tom

Date: 10:21:55 PM, Thursday, 12 October 2006

Message:

added separation for foodVendor and healthVendor

Added : /Game/client/data/shapes/items/foodVendor

Added : /Game/client/data/shapes/items/foodVendor/ClubSign.dts

Added : /Game/client/data/shapes/items/foodVendor/OrkvaleSign.JPG

Added : /Game/client/data/shapes/items/healthVendor

Added : /Game/client/data/shapes/items/healthVendor/ClubSign.dts

Added : /Game/client/data/shapes/items/healthVendor/OrkvaleSign.JPG

Revision: 186

Author: gavin

Date: 9:54:50 PM, Thursday, 12 October 2006

Message:

added logic to not resume a previous goal if it is the same as the current one!

Modified : /Game/server/immersiveAI/goals/iAIGoalManager.cs

Revision: 185

Author: gavin

Date: 9:38:43 PM, Thursday, 12 October 2006

Message:

+ fixed bug in S8 numbers for agent statistics

+ added statues to housing

+ more work on goal matrix - fine tuning

Modified : /Documentation/Goal & Vital Matrix.xlsx

Modified : /Engine/engine/immersiveAI/agent/iAIAgent.cc

Modified : /Engine/engine/immersiveAI/agent/iAIAgent.h

Modified : /Game/Immersive AI.exe

Modified : /Game/client/data/missions/stronghold.mis

Modified : /Game/server/immersiveAI/goals/doRest/relax.cs

Modified : /Game/server/immersiveAI/goals/doRest/sleep.cs

Modified : /Game/server/immersiveAI/goals/getFood/buyFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/seekFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/stealFood.cs

Modified : /Game/server/immersiveAI/goals/getHealth/buyHealth.cs

Modified : /Game/server/immersiveAI/goals/getHealth/seekHealth.cs

Modified : /Game/server/immersiveAI/goals/getHealth/stealHealth.cs

Modified : /Game/server/immersiveAI/goals/goHome/goHome.cs

Modified : /Game/server/immersiveAI/goals/goHome/seekHome.cs

Modified : /Game/server/immersiveAI/goals/haveFun/dance.cs

Modified : /Game/server/immersiveAI/goals/haveFun/haveFun.cs

Modified : /Game/server/immersiveAI/goals/iAIGoalManager.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 184

Author: tom

Date: 9:29:14 PM, Thursday, 12 October 2006

Message:

Modified : /Game/server/immersiveAI/goals/getHealth/buyHealth.cs

Revision: 183

Author: tom

Date: 9:24:53 PM, Thursday, 12 October 2006

Message:

Modified : /Game/server/immersiveAI/goals/getHealth/stealHealth.cs

Revision: 182

Author: tom

Date: 9:20:26 PM, Thursday, 12 October 2006

Message:

updated buyFood.cs

need changes to the request new goal stuff tho

Modified : /Game/server/immersiveAI/goals/getFood/buyFood.cs

Revision: 181

Author: tom

Date: 9:17:56 PM, Thursday, 12 October 2006

Message:

updated stealFood.cs

need changes to the request new goal stuff tho

Modified : /Game/server/immersiveAI/goals/getFood/stealFood.cs

Revision: 180

Author: tom

Date: 9:06:14 PM, Thursday, 12 October 2006

Message:

changed the foodKit max inventory so it works

Modified : /Game/server/avatars/player.cs

Revision: 179

Author: tom

Date: 9:05:44 PM, Thursday, 12 October 2006

Message:

added FoodKit changes

Modified : /Game/server/items/food.cs

Revision: 178

Author: tom

Date: 8:36:56 PM, Thursday, 12 October 2006

Message:

mission editor changes

Modified : /Game/client/data/missions/stronghold.mis

Revision: 177

Author: gavin

Date: 7:31:22 PM, Thursday, 12 October 2006

Message:

completed updated to Goal & Vital Matrix

+ added onCheckVitals to all solutions

+ added onExit vital updates to those requiring them; still need to be added to food & health ones though

+ added global constants for vital updates to allow for modifiers on vitals

---

Modified : /Documentation/Goal & Vital Matrix.xlsx

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/agent/iAIAgentManager.cs

Modified : /Game/server/immersiveAI/goals/doDefend/fleeArea.cs

Modified : /Game/server/immersiveAI/goals/doExplore/exploreArea.cs

Modified : /Game/server/immersiveAI/goals/doHunt/doHunt.cs

Modified : /Game/server/immersiveAI/goals/doHunt/patrolArea.cs

Modified : /Game/server/immersiveAI/goals/doHunt/seekAndDestroy.cs

Modified : /Game/server/immersiveAI/goals/doRest/doRest.cs

Modified : /Game/server/immersiveAI/goals/doRest/relax.cs

Modified : /Game/server/immersiveAI/goals/doRest/sleep.cs

Modified : /Game/server/immersiveAI/goals/getFood/getFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/seekFood.cs

Modified : /Game/server/immersiveAI/goals/getHealth/seekHealth.cs

Modified : /Game/server/immersiveAI/goals/goHome/goHome.cs

Modified : /Game/server/immersiveAI/goals/goHome/seekHome.cs

Modified : /Game/server/immersiveAI/goals/haveFun/dance.cs

Modified : /Game/server/immersiveAI/goals/haveFun/haveFun.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 176

Author: gavin

Date: 6:45:25 PM, Thursday, 12 October 2006

Message:

adjusted incorrect formula

Modified : /Documentation/Goal & Vital Matrix.xlsx

Revision: 175

Author: gavin

Date: 6:42:47 PM, Thursday, 12 October 2006

Message:

completed evaluate tab

Modified : /Documentation/Goal & Vital Matrix.xlsx

Revision: 174

Author: gavin

Date: 6:36:48 PM, Thursday, 12 October 2006

Message:

Completed formula

Modified : /Documentation/Goal & Vital Matrix.xlsx

Revision: 173

Author: gavin

Date: 6:20:45 PM, Thursday, 12 October 2006

Message:

start of proper Goal & Vital Matrix analysing of transitions

Modified : /Documentation/Goal & Vital Matrix.xlsx

Revision: 172

Author: gavin

Date: 5:27:13 PM, Thursday, 12 October 2006

Message:

+ changed onExit of each solution to call completedSolution() method of iAIGoalManager - allows single place to change logic of completion of goals

+ added timeout-ticks for goals so agents dont stay in the same goal/solution all the time

---

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/agent/iAIAgentManager.cs

Modified : /Game/server/immersiveAI/goals/doDefend/fleeArea.cs

Modified : /Game/server/immersiveAI/goals/doExplore/exploreArea.cs

Modified : /Game/server/immersiveAI/goals/doHunt/patrolArea.cs

Modified : /Game/server/immersiveAI/goals/doHunt/seekAndDestroy.cs

Modified : /Game/server/immersiveAI/goals/doRest/relax.cs

Modified : /Game/server/immersiveAI/goals/doRest/sleep.cs

Modified : /Game/server/immersiveAI/goals/getFood/buyFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/seekFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/stealFood.cs

Modified : /Game/server/immersiveAI/goals/getHealth/buyHealth.cs

Modified : /Game/server/immersiveAI/goals/getHealth/seekHealth.cs

Modified : /Game/server/immersiveAI/goals/getHealth/stealHealth.cs

Modified : /Game/server/immersiveAI/goals/goHome/seekHome.cs

Modified : /Game/server/immersiveAI/goals/haveFun/dance.cs

Modified : /Game/server/immersiveAI/goals/iAIGoalManager.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 171

Author: gavin

Date: 5:03:34 PM, Thursday, 12 October 2006

Message:

added current/previous goal/solution to goals pane in gui

Modified : /Game/client/immersiveAI/iAIControlCentre.cs

Modified : /Game/client/immersiveAI/iAIControlCentre.gui

Revision: 170

Author: gavin

Date: 4:32:14 PM, Thursday, 12 October 2006

Message:

work on goalList treeView

Modified : /Game/client/immersiveAI/iAIControlCentre.cs

Modified : /Game/client/immersiveAI/iAIControlCentre.gui

Revision: 169

Author: gavin

Date: 3:59:08 PM, Thursday, 12 October 2006

Message:

added never ending ammo

Modified : /Game/server/immersiveAI/agent/iAIAgent.Combat.cs

Revision: 168

Author: gavin

Date: 3:55:48 PM, Thursday, 12 October 2006

Message:

+ completed doRest goal/solutions

+ deleted haveTalk goal

Modified : /Game/client/data/shapes/items/vendor

Modified : /Game/server/immersiveAI/goals/doHunt/doHunt.cs

Modified : /Game/server/immersiveAI/goals/doRest/relax.cs

Modified : /Game/server/immersiveAI/goals/doRest/sleep.cs

Deleted : /Game/server/immersiveAI/goals/haveTalk

Modified : /Game/server/immersiveAI/goals/iAIGoalLibrary.cs

Revision: 167

Author: gavin

Date: 3:26:32 PM, Thursday, 12 October 2006

Message:

Completed doDefend

+ removed defendSelf as implicit when seekAndDestroy is active

+ completed fleeArea

+ completed iAIAgent.Combat

+ fixed bug with projectile not saving ID of agent who created it

---

Modified : /Engine/engine/game/projectile.cc

Modified : /Game/Immersive AI.exe

Modified : /Game/server/avatars/player.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent.Combat.cs

Deleted : /Game/server/immersiveAI/goals/doDefend/defendSelf.cs

Modified : /Game/server/immersiveAI/goals/doDefend/doDefend.cs

Modified : /Game/server/immersiveAI/goals/doDefend/fleeArea.cs

Modified : /Game/server/immersiveAI/goals/doHunt/seekAndDestroy.cs

Modified : /Game/server/immersiveAI/goals/iAIGoalLibrary.cs

Modified : /Game/server/immersiveAI/goals/iAIGoalManager.cs

Revision: 166

Author: tom

Date: 3:12:35 PM, Thursday, 12 October 2006

Message:

added vendor and updated mission

Modified : /Documentation/Goal & Vital Matrix.xlsx

Modified : /Game/client/data/missions/stronghold.mis

Added : /Game/client/data/shapes/items/vendor

Added : /Game/client/data/shapes/items/vendor/ClubSign.dts

Added : /Game/client/data/shapes/items/vendor/OrkvaleSign.JPG

Revision: 165

Author: gavin

Date: 12:08:36 PM, Thursday, 12 October 2006

Message:

doHunt solutions now complete

+ completed detecting of objects whilst on paths

+ added resuming of previous goals/solutions if interrupted (see patrolArea for example on how to interrupt a solution)

Modified : /Game/client/immersiveAI/iAIControlCentre.gui

Modified : /Game/server/immersiveAI/agent/iAIAgent.Combat.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent.Seek.cs

Modified : /Game/server/immersiveAI/agent/iAIAgentManager.cs

Modified : /Game/server/immersiveAI/goals/doHunt/doHunt.cs

Modified : /Game/server/immersiveAI/goals/doHunt/patrolArea.cs

Modified : /Game/server/immersiveAI/goals/doHunt/seekAndDestroy.cs

Modified : /Game/server/immersiveAI/goals/iAIGoalManager.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 164

Author: gavin

Date: 11:16:51 AM, Thursday, 12 October 2006

Message:

fixed bug with DRL and viewport.

changed default sweep to 0.

altered selection of agent to use ID instead of name

Modified : /Engine/engine/gui/game/fxGuiSnooper.cc

Modified : /Game/Immersive AI.exe

Modified : /Game/client/immersiveAI/iAIControlCentre.cs

Modified : /Game/client/immersiveAI/iAIControlCentre.gui

Modified : /Game/server/immersiveAI/agent/iAIAgent_Bandit.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Entertainer.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Soldier.cs

Revision: 163

Author: tom

Date: 12:15:00 AM, Thursday, 12 October 2006

Message:

updated control centre to have the GUI built in and fixed code to use it

still buggy

needs looking at for the rotation stuff

Modified : /Game/client/immersiveAI/iAIControlCentre.cs

Modified : /Game/client/immersiveAI/iAIControlCentre.gui

Revision: 162

Author: tom

Date: 12:14:16 AM, Thursday, 12 October 2006

Message:

added shapeName to be the agent's Name as well for use with the fxGuiSnooper

---

Modified : /Game/server/immersiveAI/agent/iAIAgent_Bandit.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Entertainer.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Soldier.cs

Revision: 161

Author: gavin

Date: 10:01:39 PM, Wednesday, 11 October 2006

Message:

bug fixes on side stepping

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/agent/iAIAgentManager.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Entertainer.cs

Revision: 160

Author: gavin

Date: 7:15:16 PM, Wednesday, 11 October 2006

Message:

implemented start of combat think method. Added sidestepping

Added : /Game/server/immersiveAI/agent/iAIAgent.Combat.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/agent/iAIAgentManager.cs

Modified : /Game/server/immersiveAI/goals/doHunt/seekAndDestroy.cs

Revision: 159

Author: gavin

Date: 6:48:59 PM, Wednesday, 11 October 2006

Message:

fixed bug with aim not being set correctly

Modified : /Engine/engine/game/aiPlayer.cc

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/agent/iAIAgentManager.cs

Modified : /Game/server/immersiveAI/goals/doHunt/seekAndDestroy.cs

Revision: 158

Author: gavin

Date: 9:47:02 PM, Tuesday, 10 October 2006

Message:

fixed bug where never going out of execute in seekAndDestroy

Modified : /Game/server/immersiveAI/agent/iAIAgent.Seek.cs

Modified : /Game/server/immersiveAI/goals/doHunt/seekAndDestroy.cs

Revision: 157

Author: gavin

Date: 8:58:20 PM, Tuesday, 10 October 2006

Message:

split seek code out into separate file and some logic conundrums

---

Added : /Game/server/immersiveAI/agent/iAIAgent.Seek.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Bandit.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Soldier.cs

Modified : /Game/server/immersiveAI/goals/doHunt/seekAndDestroy.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 156

Author: tom

Date: 6:04:53 PM, Tuesday, 10 October 2006

Message:

removed the checkVitals loop requestNewGoal problem...

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Revision: 155

Author: tom

Date: 5:49:08 PM, Tuesday, 10 October 2006

Message:

changed repair amount of "HealthKits" to 20 from 50

Modified : /Game/server/items/health.cs

Revision: 154

Author: tom

Date: 5:48:18 PM, Tuesday, 10 October 2006

Message:

adds call to request new goal during every agents on check vitals loop

bit buggy mite need to be removed...

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Revision: 153

Author: tom

Date: 5:47:19 PM, Tuesday, 10 October 2006

Message:

changed check vital loop time to 3sec

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 152

Author: tom

Date: 5:42:26 PM, Tuesday, 10 October 2006

Message:

buy stuff changes

Modified : /Game/server/immersiveAI/goals/getHealth/buyHealth.cs

Revision: 151

Author: gavin

Date: 12:33:14 PM, Tuesday, 10 October 2006

Message:

work on combat situations - still needs some fine tuning about warping inside players when being close enough

fixed bug on Death

Deleted : /Game/server/avatars/aiPlayer.cs

Modified : /Game/server/avatars/player.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Bandit.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Soldier.cs

Modified : /Game/server/immersiveAI/goals/doHunt/doHunt.cs

Modified : /Game/server/immersiveAI/goals/doHunt/seekAndDestroy.cs

Modified : /Game/server/immersiveAI/goals/getHealth/getHealth.cs

Modified : /Game/server/immersiveAI/goals/getHealth/seekHealth.cs

Revision: 150

Author: gavin

Date: 7:26:45 PM, Monday, 9 October 2006

Message:

split out think into module think loops

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Revision: 149

Author: gavin

Date: 7:22:31 PM, Monday, 9 October 2006

Message:

+ moved spawn point of agents to the camp fire, until find fix for falling through terrain

+ cleaned up debug messages in think() loop

+ adjusted density of nodes in path map to 0.4 to ensure proper entrance navigation

---

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathGlobal.h

Modified : /Game/Immersive AI.exe

Modified : /Game/client/data/shapes/items/food

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/agent/iAIAgentManager.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 148

Author: gavin

Date: 6:56:54 PM, Monday, 9 October 2006

Message:

fixed crash bug due to not initializing the mCurrentPath in iAIAgent.cc

---

Modified : /Engine/engine/immersiveAI/agent/iAIAgent.cc

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/agent/iAIAgentManager.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 147

Author: tom

Date: 6:22:37 PM, Monday, 9 October 2006

Message:

updated foodItem to have nice new picture n changed mission to have a few more foodItems around

---

Added : /Game/client/data/shapes/items/food

Added : /Game/client/data/shapes/items/food/foodPatch.dts

Added : /Game/client/data/shapes/items/food/healthKit.jpg

Added : /Game/client/data/shapes/items/food/healthKitaOLD.jpg

Modified : /Game/server/items/food.cs

Revision: 146

Author: gavin

Date: 6:02:27 PM, Monday, 9 October 2006

Message:

+ fixed bug in iAIAgent that was crashing when had now path

+ moved the spawn points a bit to help with world falling through bug

+ more work on think() loop for seeking to be less crash-like :D

Modified : /Engine/engine/immersiveAI/agent/iAIAgent.cc

Modified : /Game/Immersive AI.exe

Modified : /Game/client/data/missions/stronghold.mis

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/agent/iAIAgentManager.cs

Modified : /Game/server/immersiveAI/goals/getHealth/getHealth.cs

Modified : /Game/server/immersiveAI/goals/getHealth/seekHealth.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 145

Author: tom

Date: 5:58:59 PM, Monday, 9 October 2006

Message:

added relax and sleep solutions to the doRest goal... should be completed and working with modifiers

---

Modified : /Game/server/immersiveAI/goals/doRest/relax.cs

Modified : /Game/server/immersiveAI/goals/doRest/sleep.cs

Revision: 144

Author: gavin

Date: 4:20:08 PM, Monday, 9 October 2006

Message:

completed seeking logic within the think loop!

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathGlobal.h

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/agent/iAIAgentManager.cs

Modified : /Game/server/immersiveAI/goals/doExplore/exploreArea.cs

Modified : /Game/server/immersiveAI/goals/doHunt/doHunt.cs

Modified : /Game/server/immersiveAI/goals/doHunt/patrolArea.cs

Modified : /Game/server/immersiveAI/goals/getFood/seekFood.cs

Modified : /Game/server/immersiveAI/goals/getHealth/buyHealth.cs

Modified : /Game/server/immersiveAI/goals/getHealth/getHealth.cs

Modified : /Game/server/immersiveAI/goals/getHealth/seekHealth.cs

Modified : /Game/server/immersiveAI/goals/goHome/seekHome.cs

Modified : /Game/server/immersiveAI/goals/haveFun/dance.cs

Modified : /Game/server/items/food.cs

Modified : /Game/server/items/health.cs

Revision: 143

Author: gavin

Date: 2:18:13 PM, Monday, 9 October 2006

Message:

bug fixes

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 142

Author: gavin

Date: 2:05:02 PM, Monday, 9 October 2006

Message:

added more think() code

+ now makes new paths if you move :D

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Revision: 141

Author: gavin

Date: 12:28:04 PM, Monday, 9 October 2006

Message:

added start of logic for think();

+ generates path to closest object

+ still needs checking if object has moved

+ needs checking of LOS too

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.cc

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/agent/iAIAgentManager.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 140

Author: gavin

Date: 12:43:35 PM, Sunday, 8 October 2006

Message:

+ added C++ interface for getHealth and setHealth; updated goals/solutions accordingly

+ added basis for think algorithm

+ removed doDefend goal from library as only interrupt driven, not selectable :D

Modified : /Engine/engine/immersiveAI/agent/iAIAgent.cc

Modified : /Engine/engine/immersiveAI/agent/iAIAgent.h

Modified : /Game/Immersive AI.exe

Modified : /Game/Immersive AI.torsion.exports

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/agent/iAIAgentManager.cs

Modified : /Game/server/immersiveAI/goals/doDefend/defendSelf.cs

Modified : /Game/server/immersiveAI/goals/doDefend/doDefend.cs

Modified : /Game/server/immersiveAI/goals/doDefend/fleeArea.cs

Modified : /Game/server/immersiveAI/goals/doExplore/exploreArea.cs

Modified : /Game/server/immersiveAI/goals/doHunt/doHunt.cs

Modified : /Game/server/immersiveAI/goals/doHunt/patrolArea.cs

Modified : /Game/server/immersiveAI/goals/doRest/doRest.cs

Modified : /Game/server/immersiveAI/goals/getFood/getFood.cs

Modified : /Game/server/immersiveAI/goals/getHealth/getHealth.cs

Modified : /Game/server/immersiveAI/goals/goHome/goHome.cs

Modified : /Game/server/immersiveAI/goals/goHome/seekHome.cs

Modified : /Game/server/immersiveAI/goals/iAIGoalLibrary.cs

Revision: 139

Author: tom

Date: 2:12:02 AM, Saturday, 7 October 2006

Message:

updated mission file to have new placement of foodItems and scaled

changed Food.cs to show other skin (need to fix draw distance problem, as still there)

completed seekFood goal

completed buyHealth goal - just goes to a previously decided location (the club, like dance) and deducts 60 and inc health

upped range of sight from 50 to 150 in immersiveAI.cs

Modified : /Game/client/data/missions/stronghold.mis

Modified : /Game/server/immersiveAI/goals/getFood/seekFood.cs

Modified : /Game/server/immersiveAI/goals/getHealth/buyHealth.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Game/server/items/food.cs

Revision: 138

Author: gavin

Date: 6:57:02 PM, Tuesday, 3 October 2006

Message:

+ completed all the _evaulate methods

+ added _onReachDestination and _onCheckVitals callback function stubs (still need to be completed where necessary

---

Modified : /Documentation/Goal & Vital Matrix.xlsx

Modified : /Game/server/immersiveAI/goals/doDefend/defendSelf.cs

Modified : /Game/server/immersiveAI/goals/doDefend/doDefend.cs

Modified : /Game/server/immersiveAI/goals/doDefend/fleeArea.cs

Modified : /Game/server/immersiveAI/goals/doExplore/doExplore.cs

Modified : /Game/server/immersiveAI/goals/doExplore/exploreArea.cs

Modified : /Game/server/immersiveAI/goals/doHunt/doHunt.cs

Modified : /Game/server/immersiveAI/goals/doHunt/patrolArea.cs

Modified : /Game/server/immersiveAI/goals/doHunt/seekAndDestroy.cs

Modified : /Game/server/immersiveAI/goals/doRest/doRest.cs

Modified : /Game/server/immersiveAI/goals/doRest/relax.cs

Modified : /Game/server/immersiveAI/goals/doRest/sleep.cs

Modified : /Game/server/immersiveAI/goals/getFood/buyFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/getFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/seekFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/stealFood.cs

Modified : /Game/server/immersiveAI/goals/getHealth/buyHealth.cs

Modified : /Game/server/immersiveAI/goals/getHealth/seekHealth.cs

Modified : /Game/server/immersiveAI/goals/getHealth/stealHealth.cs

Modified : /Game/server/immersiveAI/goals/haveTalk/haveTalk.cs

Modified : /Game/server/immersiveAI/goals/haveTalk/talkToAnother.cs

Modified : /Game/server/immersiveAI/goals/haveTalk/talkToSelf.cs

Revision: 137

Author: gavin

Date: 5:59:42 PM, Tuesday, 3 October 2006

Message:

+ fixed bug where wrong HealthPatch::onCollision was being executed

+ added commenting and code clean-up

Modified : /Game/client/data/missions/stronghold.mis

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/goals/getHealth/getHealth.cs

Modified : /Game/server/immersiveAI/goals/getHealth/seekHealth.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Game/server/items/food.cs

Modified : /Game/server/items/health.cs

Revision: 136

Author: gavin

Date: 12:34:55 PM, Tuesday, 3 October 2006

Message:

fixed spawn point under the terrain

Deleted : /Game/client/data/missions/simplePathTest.mis

Deleted : /Game/client/data/missions/simplePathTest.ter

Modified : /Game/client/data/missions/stronghold.mis

Deleted : /Game/client/data/missions/strongholdNight.mis

Revision: 135

Author: tom

Date: 12:31:56 PM, Tuesday, 3 October 2006

Message:

mission n shiza

Modified : /Game/client/data/missions/stronghold.mis

Revision: 134

Author: tom

Date: 12:14:59 PM, Tuesday, 3 October 2006

Message:

added evaluates for doRest

Modified : /Game/server/immersiveAI/goals/doRest/doRest.cs

Modified : /Game/server/immersiveAI/goals/doRest/relax.cs

Modified : /Game/server/immersiveAI/goals/doRest/sleep.cs

Modified : /Game/server/items/food.cs

Revision: 133

Author: gavin

Date: 12:12:44 PM, Tuesday, 3 October 2006

Message:

Fixed heaps of seekHealth bugs

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/goals/getHealth/seekHealth.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Game/server/items/health.cs

Revision: 132

Author: tom

Date: 11:22:32 AM, Tuesday, 3 October 2006

Message:

added the new food.cs FoodItem item...

Added : /Game/server/items/food.cs

Revision: 131

Author: tom

Date: 11:21:57 AM, Tuesday, 3 October 2006

Message:

added the new FoodItem to the game.cs

Modified : /Game/server/core/game.cs

Revision: 130

Author: gavin

Date: 11:17:39 AM, Tuesday, 3 October 2006

Message:

+ more work on seekHealth

+ increased density of nodes in world, to more accuratley have interior navigation

+ fixed bug where control centre still have names from previous game

---

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathGlobal.h

Modified : /Game/Immersive AI.exe

Modified : /Game/client/data/missions/stronghold.mis

Modified : /Game/client/game/game.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/goals/getHealth/seekHealth.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Game/server/items/health.cs

Revision: 129

Author: gavin

Date: 10:29:30 AM, Tuesday, 3 October 2006

Message:

Added seekObject function

Modified : /Game/client/data/missions/stronghold.mis

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/goals/getHealth/getHealth.cs

Modified : /Game/server/immersiveAI/goals/getHealth/seekHealth.cs

Modified : /Game/server/immersiveAI/goals/goHome/seekHome.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 128

Author: tom

Date: 2:20:16 AM, Tuesday, 3 October 2006

Message:

testing get health and seekHealth.cs trying to make it work as desired

changed goal return values too while testing (remove return in evaluate) to use other goals...

Modified : /Game/server/immersiveAI/goals/getHealth/getHealth.cs

Modified : /Game/server/immersiveAI/goals/getHealth/seekHealth.cs

Revision: 127

Author: tom

Date: 10:54:36 PM, Monday, 2 October 2006

Message:

toms changes/experimenting

Modified : /Game/server/immersiveAI/goals/getHealth/seekHealth.cs

Revision: 126

Author: gavin

Date: 6:08:55 PM, Monday, 2 October 2006

Message:

Added housing and club signs

Modified : /Game/client/data/missions/stronghold.mis

Added : /Game/client/data/shapes/signs

Added : /Game/client/data/shapes/signs/baseSign.jpg

Added : /Game/client/data/shapes/signs/club

Added : /Game/client/data/shapes/signs/club/ClubSign.dts

Added : /Game/client/data/shapes/signs/club/OrkvaleSign.JPG

Added : /Game/client/data/shapes/signs/housing

Added : /Game/client/data/shapes/signs/housing/HousingSign.dts

Added : /Game/client/data/shapes/signs/housing/OrkvaleSign.JPG

Revision: 125

Author: gavin

Date: 3:28:56 PM, Monday, 2 October 2006

Message:

+ housing building put in!

+ added some funky Structures

Modified : /Game/client/data/missions/stronghold.mis

Modified : /Game/client/data/missions/stronghold.ter

Added : /Game/client/data/shapes/Structures

Modified : /Game/server/immersiveAI/goals/goHome/seekHome.cs

Revision: 124

Author: gavin

Date: 3:01:16 PM, Monday, 2 October 2006

Message:

goHome completed - still needs a house in the world, but evaluate methods and actual traversing is now complete

---

Modified : /Game/server/immersiveAI/goals/getFood/getFood.cs

Modified : /Game/server/immersiveAI/goals/getHealth/getHealth.cs

Modified : /Game/server/immersiveAI/goals/goHome/goHome.cs

Modified : /Game/server/immersiveAI/goals/goHome/seekHome.cs

Revision: 123

Author: gavin

Date: 2:41:07 PM, Monday, 2 October 2006

Message:

Completed haveFun goal; removed sing

Modified : /Game/server/immersiveAI/goals/haveFun/dance.cs

Modified : /Game/server/immersiveAI/goals/haveFun/haveFun.cs

Deleted : /Game/server/immersiveAI/goals/haveFun/sing.cs

Revision: 122

Author: gavin

Date: 1:50:45 PM, Monday, 2 October 2006

Message:

Updated with latest goal/solution list

Modified : /Documentation/Goal & Vital Matrix.xlsx

Revision: 121

Author: gavin

Date: 1:44:45 PM, Monday, 2 October 2006

Message:

+ removed doPatrol (as merged into doHunt)

+ updated iAIGoalLibrary to include doHunt

Deleted : /Game/server/immersiveAI/goals/doPatrol

Modified : /Game/server/immersiveAI/goals/iAIGoalLibrary.cs

Revision: 120

Author: gavin

Date: 8:01:26 PM, Tuesday, 26 September 2006

Message:

did the work for the getHealth goals and solutions and made a few changes to the other getFood solutions

---

Modified : /Game/server/immersiveAI/goals/getFood/buyFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/seekFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/stealFood.cs

Modified : /Game/server/immersiveAI/goals/getHealth/buyHealth.cs

Modified : /Game/server/immersiveAI/goals/getHealth/getHealth.cs

Modified : /Game/server/immersiveAI/goals/getHealth/seekHealth.cs

Modified : /Game/server/immersiveAI/goals/getHealth/stealHealth.cs

Revision: 119

Author: tom

Date: 2:11:58 PM, Tuesday, 26 September 2006

Message:

immersiveAI.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 118

Author: tom

Date: 2:11:09 PM, Tuesday, 26 September 2006

Message:

haveFun.cs

Modified : /Game/server/immersiveAI/goals/haveFun/haveFun.cs

Revision: 117

Author: tom

Date: 2:10:35 PM, Tuesday, 26 September 2006

Message:

buyFood and seekFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/buyFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/seekFood.cs

Revision: 116

Author: gavin

Date: 1:43:52 PM, Tuesday, 26 September 2006

Message:

+ added checking of max & min to set agent properties

Modified : /Engine/engine/immersiveAI/agent/iAIAgent.cc

Modified : /Engine/engine/immersiveAI/agent/iAIAgent.h

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/goals/haveFun/dance.cs

Revision: 115

Author: tom

Date: 12:27:36 PM, Tuesday, 26 September 2006

Message:

Modified : /Game/server/immersiveAI/goals/haveFun/dance.cs

Modified : /Game/server/immersiveAI/goals/haveFun/haveFun.cs

Revision: 114

Author: gavin

Date: 12:25:36 PM, Tuesday, 26 September 2006

Message:

dropped big building below the terrain the make the club!

Modified : /Game/client/data/missions/stronghold.mis

Modified : /Game/server/immersiveAI/goals/haveFun/dance.cs

Revision: 113

Author: tom

Date: 12:22:11 PM, Tuesday, 26 September 2006

Message:

delete doAttack stuff

Deleted : /Game/server/immersiveAI/goals/doHunt/doAttack.cs

Deleted : /Game/server/immersiveAI/goals/doHunt/killOpponent.cs

Deleted : /Game/server/immersiveAI/goals/doHunt/woundOpponent.cs

Revision: 112

Author: tom

Date: 12:21:56 PM, Tuesday, 26 September 2006

Message:

delete doAttack stuff

Deleted : /Game/server/immersiveAI/goals/doAttack

Revision: 111

Author: tom

Date: 12:15:54 PM, Tuesday, 26 September 2006

Message:

new doHunt instead of doAttack

Added : /Game/server/immersiveAI/goals/doHunt(Copy from path: /Game/server/immersiveAI/goals/doAttack, Revision, 108

Added : /Game/server/immersiveAI/goals/doHunt/doHunt.cs

Added : /Game/server/immersiveAI/goals/doHunt/patrolArea.cs

Added : /Game/server/immersiveAI/goals/doHunt/seekAndDestroy.cs

Revision: 110

Author: tom

Date: 12:13:48 PM, Tuesday, 26 September 2006

Message:

changes to getFood and that and uncommented temporarily dance states

---

Modified : /Game/server/immersiveAI/goals/getFood/buyFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/getFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/seekFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/stealFood.cs

Revision: 109

Author: gavin

Date: 12:07:11 PM, Tuesday, 26 September 2006

Message:

Added checkVitals loop and callback to current solution's _onCheckVitals function

---

Deleted : /Documentation/GFX/Thumbs.db

Modified : /Engine/engine/immersiveAI/agent/iAIAgent.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.cc

Modified : /Game/Immersive AI.exe

Modified : /Game/Immersive AI.torsion.exports

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/agent/iAIAgentManager.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Bandit.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Entertainer.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Soldier.cs

Modified : /Game/server/immersiveAI/goals/haveFun/dance.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 108

Author: gavin

Date: 10:13:58 AM, Monday, 25 September 2006

Message:

more work on onMoveStuck

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.h

Modified : /Game/Immersive AI.exe

Modified : /Game/Immersive AI.torsion.exports

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Revision: 107

Author: gavin

Date: 7:15:03 PM, Sunday, 24 September 2006

Message:

+ added iAIMessage function; just call iAIMessagef("some message") and will go into the console in a nice blue colour :)

+ changed all messages to use iAIMessage

Modified : /Engine/engine/console/console.cc

Modified : /Engine/engine/console/console.h

Modified : /Engine/engine/console/consoleFunctions.cc

Modified : /Engine/engine/gui/controls/guiConsole.cc

Modified : /Engine/engine/immersiveAI/iAIGoalLibrary.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Game/Immersive AI.exe

Modified : /Game/client/elements/defaultProfiles.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/agent/iAIAgentManager.cs

Modified : /Game/server/immersiveAI/goals/doAttack/killOpponent.cs

Modified : /Game/server/immersiveAI/goals/doDefend/defendSelf.cs

Modified : /Game/server/immersiveAI/goals/doDefend/fleeArea.cs

Modified : /Game/server/immersiveAI/goals/doExplore/exploreArea.cs

Modified : /Game/server/immersiveAI/goals/doPatrol/patrolArea.cs

Modified : /Game/server/immersiveAI/goals/doRest/relax.cs

Modified : /Game/server/immersiveAI/goals/doRest/sleep.cs

Modified : /Game/server/immersiveAI/goals/getFood/buyFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/seekFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/stealFood.cs

Modified : /Game/server/immersiveAI/goals/getHealth/buyHealth.cs

Modified : /Game/server/immersiveAI/goals/getHealth/seekHealth.cs

Modified : /Game/server/immersiveAI/goals/getHealth/stealHealth.cs

Modified : /Game/server/immersiveAI/goals/goHome/seekHome.cs

Modified : /Game/server/immersiveAI/goals/haveFun/dance.cs

Modified : /Game/server/immersiveAI/goals/haveFun/sing.cs

Modified : /Game/server/immersiveAI/goals/haveTalk/talkToAnother.cs

Modified : /Game/server/immersiveAI/goals/haveTalk/talkToSelf.cs

Revision: 106

Author: gavin

Date: 3:18:03 PM, Sunday, 24 September 2006

Message:

+ fixed bug where is a path is requested from the same position as the destination, not valid path is found

+ added dance solution; to demonstrate how to work it

+ added call back to the solution when agent reaches a destination; so when onReachDestination is called, the solutions _onReachDestination is also called :) see dance.cs for example usage

---

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.cc

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/agent/iAIAgentManager.cs

Modified : /Game/server/immersiveAI/goals/haveFun/dance.cs

Modified : /Game/server/immersiveAI/goals/haveFun/haveFun.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 105

Author: gavin

Date: 1:58:19 PM, Sunday, 24 September 2006

Message:

Added Immersive AI.torsion.opt to ignore list

Modified : /

Revision: 104

Author: gavin

Date: 1:57:53 PM, Sunday, 24 September 2006

Message:

removed torsion opt file from repo

Deleted : /Immersive AI.torsion.opt

Revision: 103

Author: gavin

Date: 1:56:04 PM, Sunday, 24 September 2006

Message:

+ moved iAIGoalManager to goals directory

+ cleaned up a lot of comments & code structure

+ added function commenting to missing functions

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/agent/iAIAgentManager.cs

Added : /Game/server/immersiveAI/goals/iAIGoalManager.cs

Deleted : /Game/server/immersiveAI/iAIGoalManager.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 102

Author: gavin

Date: 1:16:46 PM, Sunday, 24 September 2006

Message:

+ fixed some deconstructors

+ added better comments/formatting to iAIAgent types

+ renamed iAIManager to iAIAgentManager; moved into agent folder

---

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathGrid.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathGrid.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.h

Modified : /Engine/engine/platform/platformMemory.cc

Modified : /Game/Immersive AI.exe

Added : /Game/server/immersiveAI/agent/iAIAgentManager.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Bandit.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Entertainer.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Soldier.cs

Deleted : /Game/server/immersiveAI/iAIManager.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 101

Author: gavin

Date: 8:07:24 PM, Wednesday, 20 September 2006

Message:

+ added params to specify to avoid current goal/solution

+ added function to request just a new solution (leave goal intact)

+ fixed bug in evaluateList function

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/iAIGoalManager.cs

Modified : /Game/server/immersiveAI/iAIManager.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 100

Author: gavin

Date: 7:07:10 PM, Wednesday, 20 September 2006

Message:

+ moved default path settings to iAIManager, instead of within the spawn functions of each agent.

+ corrected Terrain problem where agents where getting stuck

Modified : /Game/client/data/missions/stronghold.ter

Modified : /Game/server/immersiveAI/agent/iAIAgent_Bandit.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Entertainer.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Soldier.cs

Modified : /Game/server/immersiveAI/iAIManager.cs

Modified : /Immersive AI.torsion.opt

Revision: 99

Author: gavin

Date: 7:21:05 PM, Monday, 18 September 2006

Message:

Added onMoveStuck fixity

Modified : /Engine/engine/game/aiPlayer.cc

Modified : /Game/Immersive AI.exe

Modified : /Game/Immersive AI.torsion.exports

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/goals/iAIGoalLibrary.cs

Modified : /Game/server/immersiveAI/iAIManager.cs

Modified : /Immersive AI.torsion.opt

Revision: 98

Author: gavin

Date: 11:52:03 AM, Monday, 18 September 2006

Message:

Fixy ControlCentre

Modified : /Game/client/immersiveAI/iAIControlCentre.cs

Revision: 97

Author: gavin

Date: 11:15:53 AM, Monday, 18 September 2006

Message:

Fixed the crashing! Changed current/previous goal/solution to just a TS variable

---

Modified : /Engine/engine/game/player.cc

Modified : /Engine/engine/immersiveAI/agent/iAIAgent.cc

Modified : /Engine/engine/immersiveAI/agent/iAIAgent.h

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Bandit.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Entertainer.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Soldier.cs

Modified : /Game/server/immersiveAI/iAIGoalManager.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 96

Author: gavin

Date: 10:55:41 AM, Saturday, 16 September 2006

Message:

+ work on goal manager.cs; implemented as TS file instead of C++

+ still some bugs, but seems more stable when no pathfinding... sometimes crashes though!!1!!

Modified : /Engine/VS2005/Immersive AI.vcproj

Added : /Engine/engine/console/arrayobject.cc

Modified : /Engine/engine/immersiveAI/agent/iAIAgent.cc

Modified : /Engine/engine/immersiveAI/agent/iAIAgent.h

Modified : /Engine/engine/immersiveAI/iAIGoalLibrary.cc

Modified : /Game/Immersive AI.exe

Modified : /Game/glu2d3d.dll

Modified : /Game/opengl2d3d.dll

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Bandit.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Entertainer.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Soldier.cs

Modified : /Game/server/immersiveAI/iAIGoalManager.cs

Modified : /Game/server/immersiveAI/iAIManager.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 95

Author: tom

Date: 5:17:51 PM, Wednesday, 13 September 2006

Message:

changed getFood.cs still not working but i think

Modified : /Game/server/immersiveAI/goals/getFood/buyFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/getFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/seekFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/stealFood.cs

Revision: 94

Author: tom

Date: 1:18:00 PM, Wednesday, 13 September 2006

Message:

added the view status of agent ability and stuff... gav put in that chase camera thing!

Modified : /Game/client/immersiveAI/iAIControlCentre.cs

Modified : /Game/client/immersiveAI/iAIControlCentre.gui

Revision: 93

Author: tom

Date: 10:16:02 PM, Tuesday, 12 September 2006

Message:

Modified : /Game/server/immersiveAI/goals/getFood/buyFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/getFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/seekFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/stealFood.cs

Revision: 92

Author: tom

Date: 6:40:20 PM, Tuesday, 12 September 2006

Message:

getFood goals evaluate methods completed

Modified : /Game/server/immersiveAI/goals/getFood/buyFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/seekFood.cs

Modified : /Game/server/immersiveAI/goals/getFood/stealFood.cs

Revision: 91

Author: gavin

Date: 6:25:16 PM, Tuesday, 12 September 2006

Message:

+ added random name generator

+ removed thread of goalManager - now just a normal SimObject, which needs a schedule to process the batch using processRequestBatch

+ corrected BinaryHeap deconstructor to ensure deletion of objects

+ fixed buffer in gameConnectionEvents.cc that was set too low

---

Modified : /Engine/VS2005/Immersive AI.vcproj

Modified : /Engine/engine/game/gameConnectionEvents.cc

Modified : /Engine/engine/game/main.cc

Added : /Engine/engine/game/nameList.cc

Added : /Engine/engine/game/nameList.h

Modified : /Engine/engine/immersiveAI/core/tBinaryHeap.h

Modified : /Engine/engine/immersiveAI/iAIGoalLibrary.cc

Modified : /Engine/engine/immersiveAI/iAIGoalManager.cc

Modified : /Engine/engine/immersiveAI/iAIGoalManager.h

Modified : /Game/Immersive AI.exe

Added : /Game/client/data/names.nam

Modified : /Game/server/core/game.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/goals/getFood/getFood.cs

Modified : /Game/server/immersiveAI/iAIGoalManager.cs

Modified : /Game/server/immersiveAI/iAIManager.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 90

Author: gavin

Date: 10:07:56 PM, Monday, 11 September 2006

Message:

+ fixed problem where mutex wasn't being locked on requesting a goal

+ changed agent spawn points to normal spawn sphere's

+ corrected library deconstructor to delete the library correctly

---

Modified : /Engine/engine

Modified : /Engine/engine/immersiveAI/iAIGoalLibrary.cc

Modified : /Engine/engine/immersiveAI/iAIGoalManager.cc

Modified : /Engine/engine/immersiveAI/iAIGoalManager.h

Modified : /Engine/lib

Modified : /Game

Modified : /Game/Immersive AI.exe

Modified : /Game/client/data/missions/stronghold.mis

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/iAIManager.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 89

Author: gavin

Date: 7:04:31 PM, Sunday, 10 September 2006

Message:

updated some values

Modified : /Documentation/Goal & Vital Matrix.xlsx

Revision: 88

Author: tom

Date: 3:51:07 PM, Sunday, 10 September 2006

Message:

made changes

gav to check

Modified : /Documentation/Goal & Vital Matrix.xlsx

Revision: 87

Author: gavin

Date: 12:14:56 PM, Sunday, 10 September 2006

Message:

bugfix on threads & onStuck

Modified : /Engine/engine/immersiveAI/iAIGoalManager.cc

Modified : /Engine/engine/immersiveAI/iAIGoalManager.h

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Immersive AI.torsion.opt

Revision: 86

Author: gavin

Date: 11:03:33 AM, Sunday, 10 September 2006

Message:

Modified : /Documentation/Goal & Vital Matrix.xlsx

Revision: 85

Author: gavin

Date: 10:50:54 AM, Sunday, 10 September 2006

Message:

Added weightings for all agents / goals / solutions

Modified : /Documentation/Goal & Vital Matrix.xlsx

Revision: 84

Author: gavin

Date: 10:28:03 AM, Sunday, 10 September 2006

Message:

added getFood details; some layout changes

Modified : /Documentation/Goal & Vital Matrix.xlsx

Revision: 83

Author: gavin

Date: 10:06:50 AM, Sunday, 10 September 2006

Message:

Added Goal & Vital Matrix... just basic layout, needs to be completed

---

Added : /Documentation/Goal & Vital Matrix.xlsx

Revision: 82

Author: gavin

Date: 5:18:51 PM, Wednesday, 6 September 2006

Message:

+ applied fix for slow moving ai characters

+ added ignore for *.dso in goal directories

Modified : /Engine/engine/game/player.cc

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/goals/doAttack

Modified : /Game/server/immersiveAI/goals/doDefend

Modified : /Game/server/immersiveAI/goals/doExplore

Modified : /Game/server/immersiveAI/goals/doPatrol

Modified : /Game/server/immersiveAI/goals/doRest

Modified : /Game/server/immersiveAI/goals/getFood

Modified : /Game/server/immersiveAI/goals/getHealth

Modified : /Game/server/immersiveAI/goals/goHome

Modified : /Game/server/immersiveAI/goals/haveFun

Modified : /Game/server/immersiveAI/goals/haveTalk

Modified : /Immersive AI.torsion.opt

Revision: 81

Author: gavin

Date: 11:23:04 AM, Wednesday, 6 September 2006

Message:

goal library enhancements

+ removed condition function, always executing <goal>/<solution>_evaluate now, instead of having to always specify 'evaluate'

+ rearranged files into the goals directory, split goals into sub directories and put solutions with them

+ each goal file now registers itself against the goal library

+ reconstructed all the files to have the proper template to allow goal execution/evaluation

---

Modified : /Engine/engine/immersiveAI/iAIGoalLibrary.cc

Modified : /Engine/engine/immersiveAI/iAIGoalLibrary.h

Modified : /Engine/engine/immersiveAI/iAIGoalManager.cc

Modified : /Engine/engine/immersiveAI/iAIGoalManager.h

Modified : /Game/Immersive AI.exe

Added : /Game/server/immersiveAI/goals/doAttack

Added : /Game/server/immersiveAI/goals/doAttack/doAttack.cs

Added : /Game/server/immersiveAI/goals/doAttack/killOpponent.cs

Added : /Game/server/immersiveAI/goals/doAttack/woundOpponent.cs

Deleted : /Game/server/immersiveAI/goals/doAttack.cs

Added : /Game/server/immersiveAI/goals/doDefend

Added : /Game/server/immersiveAI/goals/doDefend/defendSelf.cs

Added : /Game/server/immersiveAI/goals/doDefend/doDefend.cs

Added : /Game/server/immersiveAI/goals/doDefend/fleeArea.cs

Deleted : /Game/server/immersiveAI/goals/doDefend.cs

Added : /Game/server/immersiveAI/goals/doExplore

Added : /Game/server/immersiveAI/goals/doExplore/doExplore.cs

Added : /Game/server/immersiveAI/goals/doExplore/exploreArea.cs

Deleted : /Game/server/immersiveAI/goals/doExplore.cs

Added : /Game/server/immersiveAI/goals/doPatrol

Added : /Game/server/immersiveAI/goals/doPatrol/doPatrol.cs

Added : /Game/server/immersiveAI/goals/doPatrol/patrolArea.cs

Deleted : /Game/server/immersiveAI/goals/doPatrol.cs

Added : /Game/server/immersiveAI/goals/doRest

Added : /Game/server/immersiveAI/goals/doRest/doRest.cs

Added : /Game/server/immersiveAI/goals/doRest/relax.cs

Added : /Game/server/immersiveAI/goals/doRest/sleep.cs

Deleted : /Game/server/immersiveAI/goals/doRest.cs

Added : /Game/server/immersiveAI/goals/getFood

Added : /Game/server/immersiveAI/goals/getFood/buyFood.cs

Added : /Game/server/immersiveAI/goals/getFood/getFood.cs

Added : /Game/server/immersiveAI/goals/getFood/seekFood.cs

Added : /Game/server/immersiveAI/goals/getFood/stealFood.cs

Deleted : /Game/server/immersiveAI/goals/getFood.cs

Added : /Game/server/immersiveAI/goals/getHealth

Added : /Game/server/immersiveAI/goals/getHealth/buyHealth.cs

Added : /Game/server/immersiveAI/goals/getHealth/getHealth.cs

Added : /Game/server/immersiveAI/goals/getHealth/seekHealth.cs

Added : /Game/server/immersiveAI/goals/getHealth/stealHealth.cs

Deleted : /Game/server/immersiveAI/goals/getHealth.cs

Added : /Game/server/immersiveAI/goals/goHome

Added : /Game/server/immersiveAI/goals/goHome/goHome.cs

Added : /Game/server/immersiveAI/goals/goHome/seekHome.cs

Deleted : /Game/server/immersiveAI/goals/goHome.cs

Added : /Game/server/immersiveAI/goals/haveFun

Added : /Game/server/immersiveAI/goals/haveFun/dance.cs

Added : /Game/server/immersiveAI/goals/haveFun/haveFun.cs

Added : /Game/server/immersiveAI/goals/haveFun/sing.cs

Deleted : /Game/server/immersiveAI/goals/haveFun.cs

Added : /Game/server/immersiveAI/goals/haveTalk

Added : /Game/server/immersiveAI/goals/haveTalk/haveTalk.cs

Added : /Game/server/immersiveAI/goals/haveTalk/talkToAnother.cs

Added : /Game/server/immersiveAI/goals/haveTalk/talkToSelf.cs

Deleted : /Game/server/immersiveAI/goals/haveTalk.cs

Added : /Game/server/immersiveAI/goals/iAIGoalLibrary.cs

Deleted : /Game/server/immersiveAI/iAIGoalLibrary.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Deleted : /Game/server/immersiveAI/solutions

Modified : /Immersive AI.torsion.opt

Revision: 80

Author: gavin

Date: 12:04:33 PM, Tuesday, 29 August 2006

Message:

+ added more sophisticated goal/solution request processing, can now request a new goal, request a new goal avoiding the current goal, and request a new solution for the current goal.

Modified : /Engine/engine/immersiveAI/agent/iAIAgent.cc

Modified : /Engine/engine/immersiveAI/agent/iAIAgent.h

Modified : /Engine/engine/immersiveAI/iAIGoalLibrary.cc

Modified : /Engine/engine/immersiveAI/iAIGoalLibrary.h

Modified : /Engine/engine/immersiveAI/iAIGoalManager.cc

Modified : /Engine/engine/immersiveAI/iAIGoalManager.h

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Bandit.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Entertainer.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Soldier.cs

Modified : /Game/server/immersiveAI/goals/getFood.cs

Modified : /Game/server/immersiveAI/goals/getHealth.cs

Modified : /Game/server/immersiveAI/iAIManager.cs

Modified : /Immersive AI.torsion.opt

Revision: 79

Author: gavin

Date: 10:09:43 PM, Monday, 28 August 2006

Message:

more work on iAIGoalLibrary & iAIGoalManager

+ completed executing of goal determination & solution determination

+ fixed some fields on the iAIAgent

+ renamed states folder to solutions

Modified : /Engine/VS2005/Immersive AI.vcproj

Modified : /Engine/engine/immersiveAI/agent/iAIAgent.cc

Modified : /Engine/engine/immersiveAI/agent/iAIAgent.h

Modified : /Engine/engine/immersiveAI/iAIGoalLibrary.cc

Modified : /Engine/engine/immersiveAI/iAIGoalLibrary.h

Modified : /Engine/engine/immersiveAI/iAIGoalManager.cc

Modified : /Engine/engine/immersiveAI/iAIGoalManager.h

Modified : /Game/Immersive AI.exe

Modified : /Game/Immersive AI.torsion.exports

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Bandit.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Entertainer.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent_Soldier.cs

Modified : /Game/server/immersiveAI/goals/getFood.cs

Modified : /Game/server/immersiveAI/goals/getHealth.cs

Modified : /Game/server/immersiveAI/iAIGoalLibrary.cs

Modified : /Game/server/immersiveAI/iAIManager.cs

Deleted : /Game/server/immersiveAI/iAIStateManager.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Deleted : /Game/server/immersiveAI/seek

Added : /Game/server/immersiveAI/solutions(Copy from path: /Game/server/immersiveAI/states, Revision, 74

Modified : /Game/server/immersiveAI/solutions/buyFood.cs

Modified : /Game/server/immersiveAI/solutions/buyHealth.cs

Deleted : /Game/server/immersiveAI/states

Modified : /Immersive AI.torsion.opt

Revision: 78

Author: gavin

Date: 8:34:10 PM, Sunday, 27 August 2006

Message:

Add solution concept to GoalLibrary - to 'solve' a goal, eg. getFood, a solution is devised - either stealFood, seekFood, buyFood. The solution is then executed for its steps.

Modified : /Engine/engine/immersiveAI/iAIGoalLibrary.cc

Modified : /Engine/engine/immersiveAI/iAIGoalLibrary.h

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/iAIGoalLibrary.cs

Modified : /Immersive AI.torsion.opt

Revision: 77

Author: gavin

Date: 6:11:39 PM, Sunday, 27 August 2006

Message:

+ fixed agent variables being set as datablock statics, instead of intended members variables

+ split agent login into bandit, entertainer & soldier classes

---

Modified : /Engine/engine/immersiveAI/agent/iAIAgent.cc

Modified : /Engine/engine/immersiveAI/agent/iAIAgent.h

Modified : /Game/Immersive AI.exe

Modified : /Game/Immersive AI.torsion.exports

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Added : /Game/server/immersiveAI/agent/iAIAgent_Bandit.cs

Added : /Game/server/immersiveAI/agent/iAIAgent_Entertainer.cs

Added : /Game/server/immersiveAI/agent/iAIAgent_Soldier.cs

Modified : /Game/server/immersiveAI/iAIGoalLibrary.cs

Modified : /Game/server/immersiveAI/iAIManager.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 76

Author: gavin

Date: 10:51:02 AM, Saturday, 26 August 2006

Message:

Completed iAIGoalLibrary

- added more comments on usage of iAIGoal

- fixed some issues with ordering of goalname/agenttype information

---

Modified : /Engine/engine/immersiveAI/iAIGoalLibrary.cc

Modified : /Engine/engine/immersiveAI/iAIGoalLibrary.h

Modified : /Game/Immersive AI.exe

Modified : /Game/Immersive AI.torsion.exports

Modified : /Game/server/immersiveAI/iAIGoalLibrary.cs

Modified : /Immersive AI.torsion.opt

Revision: 75

Author: gavin

Date: 9:53:44 PM, Thursday, 24 August 2006

Message:

replaced to proper function

Modified : /Game/server/immersiveAI/iAIGoalLibrary.cs

Revision: 74

Author: gavin

Date: 9:33:27 PM, Thursday, 24 August 2006

Message:

+ Added a goal library, to register goals and function calls in a central location

+ removed old shield code

Modified : /Engine/VS2005/Immersive AI.vcproj

Added : /Engine/engine/immersiveAI/iAIGoalLibrary.cc

Added : /Engine/engine/immersiveAI/iAIGoalLibrary.h

Modified : /Engine/engine/immersiveAI/iAIGoalManager.cc

Modified : /Engine/engine/immersiveAI/iAIGoalManager.h

Modified : /Game/Immersive AI.exe

Modified : /Game/Immersive AI.torsion.exports

Modified : /Game/server/avatars/player.cs

Modified : /Game/server/core/game.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Added : /Game/server/immersiveAI/iAIGoalLibrary.cs

Modified : /Game/server/immersiveAI/iAIGoalManager.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 73

Author: gavin

Date: 10:54:46 AM, Wednesday, 23 August 2006

Message:

Started on iAIGoalManger

+ multithreaded working, with batch processing of requests

+ add stats & characteristics to iAIAgents

Added : /Documentation/Goals & States list.docx

Modified : /Engine/VS2005/Immersive AI.vcproj

Modified : /Engine/engine/immersiveAI/agent/iAIAgent.cc

Modified : /Engine/engine/immersiveAI/agent/iAIAgent.h

Added : /Engine/engine/immersiveAI/iAIGoalManager.cc

Added : /Engine/engine/immersiveAI/iAIGoalManager.h

Modified : /Game/Immersive AI.exe

Modified : /Game/Immersive AI.torsion.exports

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 72

Author: gavin

Date: 10:57:32 AM, Tuesday, 22 August 2006

Message:

Added : /Game/client/data/shapes/avatars/orc

Revision: 71

Author: gavin

Date: 10:57:06 AM, Tuesday, 22 August 2006

Message:

Added : /Game/client/data/shapes/avatars/orc(Copy from path: /Game/client/data/shapes/avatars/player, Revision, 70

Deleted : /Game/client/data/shapes/avatars/player

Revision: 70

Author: gavin

Date: 10:56:39 AM, Tuesday, 22 August 2006

Message:

Modified : /Game/client/data/shapes/avatars/player

Added : /Game/client/data/shapes/avatars/player/orc.cs

Revision: 69

Author: gavin

Date: 10:53:49 AM, Tuesday, 22 August 2006

Message:

f

Added : /Game/client/data/shapes/avatars/player

Revision: 68

Author: gavin

Date: 10:46:53 AM, Tuesday, 22 August 2006

Message:

clickity fixity

Modified : /Game/client/data/shapes/avatars/armygirl/armygirl.cs

Added : /Game/client/data/shapes/avatars/armygirl/player_dance.dsq

Added : /Game/client/data/shapes/avatars/armygirl/player_range.dsq

Modified : /Game/client/data/shapes/avatars/armyguy/armyguy.cs

Added : /Game/client/data/shapes/avatars/armyguy/player_dance.dsq

Added : /Game/client/data/shapes/avatars/armyguy/player_range.dsq

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/states/buyFood.cs

Modified : /Immersive AI.torsion.opt

Revision: 67

Author: tom

Date: 10:40:52 AM, Tuesday, 22 August 2006

Message:

Added : /Game/server/immersiveAI/iAIGoalManager.cs

Modified : /Game/server/immersiveAI/iAIStateManager.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 66

Author: tom

Date: 10:36:53 AM, Tuesday, 22 August 2006

Message:

updated buyFood state

Modified : /Game/server/immersiveAI/states/buyFood.cs

Revision: 65

Author: tom

Date: 10:36:26 AM, Tuesday, 22 August 2006

Message:

goals added

Added : /Game/server/immersiveAI/goals

Added : /Game/server/immersiveAI/goals/doAttack.cs

Added : /Game/server/immersiveAI/goals/doDefend.cs

Added : /Game/server/immersiveAI/goals/doExplore.cs

Added : /Game/server/immersiveAI/goals/doPatrol.cs

Added : /Game/server/immersiveAI/goals/doRest.cs

Added : /Game/server/immersiveAI/goals/getFood.cs

Added : /Game/server/immersiveAI/goals/getHealth.cs

Added : /Game/server/immersiveAI/goals/goHome.cs

Added : /Game/server/immersiveAI/goals/haveFun.cs

Added : /Game/server/immersiveAI/goals/haveTalk.cs

Revision: 64

Author: gavin

Date: 7:07:44 PM, Sunday, 20 August 2006

Message:

renamed to orc..

Deleted : /Game/client/data/shapes/avatars/player

Revision: 63

Author: gavin

Date: 7:06:50 PM, Sunday, 20 August 2006

Message:

renamed to orc

Added : /Game/client/data/shapes/avatars/player/orc.cs

Added : /Game/client/data/shapes/avatars/player/orc.dts

Deleted : /Game/client/data/shapes/avatars/player/player.cfg

Deleted : /Game/client/data/shapes/avatars/player/player.cs

Deleted : /Game/client/data/shapes/avatars/player/player.dts

Deleted : /Game/client/data/shapes/avatars/player/player.max

Revision: 62

Author: gavin

Date: 7:01:37 PM, Sunday, 20 August 2006

Message:

+ validated agent selected on iAIControlCentre

+ renamed avatar player to orc

+ added armygirl & armyguy avatars

+ added random selection of agents to gameworld

Added : /Game/client/data/shapes/avatars/armygirl

Modified : /Game/client/default.bind.cs

Modified : /Game/client/immersiveAI/iAIControlCentre.cs

Modified : /Game/server/avatars/player.cs

Modified : /Game/server/core/game.cs

Modified : /Game/server/core/weapon.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Immersive AI.torsion.opt

Revision: 61

Author: gavin

Date: 10:42:24 AM, Friday, 18 August 2006

Message:

Added melee

Modified : /Engine/engine/game/player.cc

Modified : /Engine/engine/game/player.h

Modified : /Engine/engine/game/shapeBase.cc

Modified : /Engine/engine/game/shapeBase.h

Modified : /Engine/engine/game/shapeImage.cc

Modified : /Game/Immersive AI.exe

Added : /Game/client/data/shapes/avatars/player/onehand_root.dsq

Added : /Game/client/data/shapes/avatars/player/onehand_root_alt.dsq

Added : /Game/client/data/shapes/avatars/player/onehand_slice.dsq

Added : /Game/client/data/shapes/avatars/player/onehand_swing.dsq

Added : /Game/client/data/shapes/avatars/player/onehand_thrust.dsq

Modified : /Game/client/data/shapes/avatars/player/player.cs

Added : /Game/client/data/shapes/avatars/player/player_h1jumpattack.dsq

Added : /Game/client/data/shapes/avatars/player/player_h1root.dsq

Added : /Game/client/data/shapes/avatars/player/player_h1slice.dsq

Added : /Game/client/data/shapes/avatars/player/player_h1stunde.dsq

Added : /Game/client/data/shapes/avatars/player/player_h1swing.dsq

Added : /Game/client/data/shapes/avatars/player/player_h1thrust.dsq

Added : /Game/client/data/shapes/items/sword

Added : /Game/client/data/shapes/items/sword/blade01.jpg

Added : /Game/client/data/shapes/items/sword/rune_blade01.dts

Modified : /Game/server/avatars/player.cs

Modified : /Game/server/core/game.cs

Modified : /Game/server/core/weapon.cs

Added : /Game/server/items/sword.cs

Modified : /Immersive AI.torsion.opt

Revision: 60

Author: gavin

Date: 6:36:51 PM, Wednesday, 16 August 2006

Message:

Completed initial Path options - changable spline rendering/displaying of paths for each agent

---

Modified : /Game/client/immersiveAI/iAIControlCentre.cs

Modified : /Game/client/immersiveAI/iAIControlCentre.gui

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 59

Author: gavin

Date: 1:36:38 PM, Sunday, 13 August 2006

Message:

Added more smooth turning for iAIAgent's. Fix for onStuck detection.

Modified : /Engine/engine/game/aiPlayer.cc

Modified : /Engine/engine/game/aiPlayer.h

Modified : /Engine/engine/immersiveAI/agent/iAIAgent.cc

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Immersive AI.torsion.opt

Revision: 58

Author: gavin

Date: 6:32:18 PM, Friday, 11 August 2006

Message:

Removed qSort from pathGrid, not needed anymore

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathGrid.cc

Modified : /Game/Immersive AI.exe

Modified : /Game/client/immersiveAI/iAIControlCentre.cs

Modified : /Game/client/immersiveAI/iAIControlCentre.gui

Modified : /Immersive AI.torsion.opt

Revision: 57

Author: gavin

Date: 10:44:09 AM, Friday, 11 August 2006

Message:

iAIControlCentre basic layout done

Modified : /Game/Immersive AI.exe

Modified : /Game/client/immersiveAI/iAIControlCentre.cs

Modified : /Game/client/immersiveAI/iAIControlCentre.gui

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/iAIManager.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 56

Author: gavin

Date: 6:40:07 PM, Thursday, 10 August 2006

Message:

Fix for CG water displaying underwater & correct fog issues

Modified : /Engine/engine/game/gameTSCtrl.cc

Modified : /Engine/engine/sceneGraph/sceneGraph.cc

Modified : /Engine/engine/sceneGraph/sceneGraph.h

Modified : /Engine/engine/terrain/fluidRender.cc

Modified : /Engine/engine/terrain/sky.cc

Modified : /Engine/engine/terrain/terrRender.cc

Modified : /Game/Immersive AI.exe

Modified : /Game/client/data/CG/water_fp20.cg

Modified : /Game/client/data/CG/water_vp11.cg

Revision: 55

Author: gavin

Date: 9:43:41 PM, Wednesday, 9 August 2006

Message:

Initial work on in game gui - bound to F6.

Modified : /Engine/VS2005/Immersive AI.vcproj

Added : /Engine/engine/gui/game/fxGuiSnooper.cc

Added : /Engine/engine/gui/game/guiObjectView.cc

Added : /Engine/engine/gui/game/guiObjectView.h

Modified : /Game/Immersive AI.exe

Modified : /Game/client/default.bind.cs

Added : /Game/client/immersiveAI

Added : /Game/client/immersiveAI/iAIControlCentre.cs

Added : /Game/client/immersiveAI/iAIControlCentre.gui

Modified : /Game/client/init.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Game/server/immersiveAI/states

Modified : /Immersive AI.torsion.opt

Revision: 54

Author: gavin

Date: 8:09:56 PM, Wednesday, 9 August 2006

Message:

Completed smoothing algorithm.

Modified : /Engine/engine/game/main.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathGlobal.h

Modified : /Game/Immersive AI.exe

Modified : /Game/server/avatars/player.cs

Modified : /Game/server/core/game.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/iAIManager.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 53

Author: gavin

Date: 4:51:21 PM, Wednesday, 9 August 2006

Message:

Added template class for Binary Heap - further optimisations within it also

---

Modified : /Engine/VS2005/Immersive AI.vcproj

Added : /Engine/engine/immersiveAI/core

Added : /Engine/engine/immersiveAI/core/tBinaryHeap.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathGlobal.h

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 52

Author: gavin

Date: 1:57:33 PM, Wednesday, 9 August 2006

Message:

Performance improvements on iAIPathFind - using BinaryHeap for openList, instead of a vector... much much much better performance :D

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/iAIManager.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 51

Author: gavin

Date: 12:06:05 PM, Wednesday, 9 August 2006

Message:

Completed getClosestNode... for now

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathGlobal.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathGrid.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathGrid.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.cc

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/iAIManager.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 50

Author: tom

Date: 1:10:31 AM, Tuesday, 8 August 2006

Message:

added 19 states and their script files as well as the StateManager

---

Added : /Game/server/immersiveAI/iAIStateManager.cs

Added : /Game/server/immersiveAI/states

Added : /Game/server/immersiveAI/states/buyFood.cs

Added : /Game/server/immersiveAI/states/buyHealth.cs

Added : /Game/server/immersiveAI/states/dance.cs

Added : /Game/server/immersiveAI/states/defendSelf.cs

Added : /Game/server/immersiveAI/states/exploreArea.cs

Added : /Game/server/immersiveAI/states/fleeArea.cs

Added : /Game/server/immersiveAI/states/killOpponent.cs

Added : /Game/server/immersiveAI/states/patrolArea.cs

Added : /Game/server/immersiveAI/states/relax.cs

Added : /Game/server/immersiveAI/states/seekFood.cs

Added : /Game/server/immersiveAI/states/seekHealth.cs

Added : /Game/server/immersiveAI/states/seekHome.cs

Added : /Game/server/immersiveAI/states/sing.cs

Added : /Game/server/immersiveAI/states/sleep.cs

Added : /Game/server/immersiveAI/states/stealFood.cs

Added : /Game/server/immersiveAI/states/stealHealth.cs

Added : /Game/server/immersiveAI/states/talkToAnother.cs

Added : /Game/server/immersiveAI/states/talkToSelf.cs

Added : /Game/server/immersiveAI/states/woundOpponent.cs

Revision: 49

Author: gavin

Date: 11:26:03 PM, Monday, 7 August 2006

Message:

Work on getClosestNode; dQsort on main grid seems to be sorting fine now, but binary search is having... issues...

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathGrid.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathGrid.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.h

Modified : /Game/Immersive AI.exe

Modified : /Game/Immersive AI.torsion.exports

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 48

Author: gavin

Date: 4:04:52 PM, Monday, 7 August 2006

Message:

Rewrite of iAIPath. Nodes now C++ objects, not simObjects - results in 400% (atleast) performance increase.

SmoothPath and getClosestNode requires some optimisations/fixes.

Modified : /Engine/VS2005/Immersive AI.vcproj

Modified : /Engine/engine/game/objectTypes.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.h

Added : /Engine/engine/immersiveAI/seek/path/iAIPathGlobal.h

Added : /Engine/engine/immersiveAI/seek/path/iAIPathGrid.cc

Added : /Engine/engine/immersiveAI/seek/path/iAIPathGrid.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.h

Modified : /Engine/engine/platform/profiler.h

Modified : /Game/Immersive AI.exe

Modified : /Game/Immersive AI.torsion.exports

Modified : /Game/client/data/missions/simplePathTest.mis

Modified : /Game/client/data/missions/strongholdNight.mis

Modified : /Game/creator/editor/EditorGui.cs

Modified : /Game/server/immersiveAI/agent/iAIAgent.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 47

Author: gavin

Date: 3:55:29 PM, Saturday, 5 August 2006

Message:

Reverted to just terrain mapping - performance issues!

Modified : /Engine/VS2005/Immersive AI.vcproj

Modified : /Engine/engine/game/objectTypes.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.h

Deleted : /Engine/engine/immersiveAI/seek/path/iAIPathGrid.cc

Deleted : /Engine/engine/immersiveAI/seek/path/iAIPathGrid.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.h

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 46

Author: gavin

Date: 5:57:14 PM, Wednesday, 2 August 2006

Message:

Change to density of nodes around interiors

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Game/Immersive AI.exe

Revision: 45

Author: gavin

Date: 12:31:16 PM, Wednesday, 2 August 2006

Message:

total rewrite of the Seek :: Path creation. Now works by creating "grids" around interiors, then creating a master grid for the terrain. The interiors grids are then joined to the master terrain grid. This allows for future creation on internal interior maps, as the iAIPathMap now simply stores a vector of grids.

Modified : /Engine/VS2005/Immersive AI.vcproj

Modified : /Engine/engine/game/objectTypes.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.h

Added : /Engine/engine/immersiveAI/seek/path/iAIPathGrid.cc

Added : /Engine/engine/immersiveAI/seek/path/iAIPathGrid.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.h

Modified : /Game/Immersive AI.exe

Modified : /Game/Immersive AI.torsion.exports

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 44

Author: gavin

Date: 1:11:21 PM, Monday, 31 July 2006

Message:

Cleanup.

Added C++ Optimisations text file on some tips to optimise algorithms

---

Added : /Documentation/C++ Optimisations.txt

Modified : /Game/Immersive AI.exe

Deleted : /Game/server/immersiveAI/agent/agent.cs

Added : /Game/server/immersiveAI/agent/iAIAgent.cs(Copy from path: /Game/server/immersiveAI/agent/agent.cs, Revision, 43

Modified : /Game/server/immersiveAI/immersiveAI.cs

Deleted : /Game/server/immersiveAI/seek/path

Modified : /Immersive AI.torsion.opt

Revision: 43

Author: gavin

Date: 1:45:47 PM, Sunday, 30 July 2006

Message:

updates to agents; following paths now

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.cc

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/agent/agent.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 42

Author: gavin

Date: 3:22:28 PM, Wednesday, 26 July 2006

Message:

Fixed some dOxygen comments

Modified : /Engine/engine/immersiveAI/agent/iAIAgent.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Revision: 41

Author: gavin

Date: 1:33:58 PM, Wednesday, 26 July 2006

Message:

Few optimisations/bugfixes to iAIPath.

Added spline path rendering to iAIPath - toggles between linear & spline rendering: iAIPath.renderSpline = true/false;

Added iAIAgent class. Added basic routines for following a path.

Modified : /Engine/VS2005/Immersive AI.vcproj

Added : /Engine/engine/immersiveAI/agent

Added : /Engine/engine/immersiveAI/agent/iAIAgent.cc

Added : /Engine/engine/immersiveAI/agent/iAIAgent.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.h

Modified : /Game/Immersive AI.exe

Modified : /Game/Immersive AI.torsion.exports

Modified : /Game/creator/editor/EditorGui.cs

Modified : /Game/server/core/game.cs

Added : /Game/server/immersiveAI/agent

Added : /Game/server/immersiveAI/agent/agent.cs

Added : /Game/server/immersiveAI/iAIManager.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 40

Author: gavin

Date: 4:45:12 PM, Tuesday, 25 July 2006

Message:

Changes static const members to capitals, better coding practice :)

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.cc

Modified : /Game/Immersive AI.exe

Revision: 39

Author: gavin

Date: 10:25:01 AM, Tuesday, 25 July 2006

Message:

Fixed bug with smooth path algorithm

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.cc

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 38

Author: gavin

Date: 4:15:25 PM, Monday, 24 July 2006

Message:

removed smoothing for now... something weird

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 37

Author: gavin

Date: 4:12:37 PM, Monday, 24 July 2006

Message:

bugfix for count()-2 (not -1!)

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.cc

Modified : /Game/Immersive AI.exe

Revision: 36

Author: gavin

Date: 4:07:53 PM, Monday, 24 July 2006

Message:

Added smoothPath algorithm

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.h

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 35

Author: gavin

Date: 2:33:04 PM, Monday, 24 July 2006

Message:

Corrected some dOxygen comments.

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.h

Revision: 34

Author: gavin

Date: 2:14:25 PM, Monday, 24 July 2006

Message:

Added stub for path smoothing. Added optional parameter to smooth the path (default to true).

Fixed bug when colour could not be set for a path.

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.h

Modified : /Game/Immersive AI.exe

Modified : /Game/Immersive AI.torsion.exports

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 33

Author: gavin

Date: 10:49:57 PM, Sunday, 23 July 2006

Message:

Completed dOxygen comments for path files.

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.h

Modified : /Game/Immersive AI.exe

Revision: 32

Author: gavin

Date: 7:43:10 PM, Sunday, 23 July 2006

Message:

Began implementing full dOxygen comments. Added persistant fields for some variables, allows proper exposure to TorqueScript.

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.h

Modified : /Game/Immersive AI.exe

Modified : /Game/Immersive AI.torsion.exports

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 31

Author: gavin

Date: 12:27:36 PM, Sunday, 23 July 2006

Message:

Fixed problem with LAN connections - multiplayer working now :)

Modified : /Game/server/core/clientConnection.cs

Modified : /Game/server/core/game.cs

Modified : /Immersive AI.torsion.opt

Revision: 30

Author: gavin

Date: 12:21:13 PM, Sunday, 23 July 2006

Message:

Fixed errors when no CG shading available - disables now instead of console spam

---

Modified : /Engine/engine/sceneGraph/sceneState.cc

Modified : /Engine/engine/terrain/fluidRender.cc

Modified : /Game/Immersive AI.exe

Revision: 29

Author: gavin

Date: 12:06:27 PM, Sunday, 23 July 2006

Message:

Some code cleanup (what else is new!).

Changed theme of main menu to brain wallpaper. Added GarageGames splash on startup.

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Game/Immersive AI.exe

Deleted : /Game/client/data/missions/paths

Modified : /Game/client/menu/StartupGui.gui

Modified : /Game/client/menu/gfx/background.png

Modified : /Game/client/menu/gfx/buttons/Thumbs.db

Added : /Game/client/menu/gfx/garagegames.splash.jpg

Modified : /Game/client/menu/mainMenuGui.gui

Modified : /Game/creator/ui

Modified : /Immersive AI.torsion.opt

Revision: 28

Author: gavin

Date: 11:40:30 PM, Saturday, 22 July 2006

Message:

Optimised FindClosestNodeIndex - will only iterate over all nodes now if can't find a node within the closest 3x3 square

---

Modified : /Engine/VS2005/Immersive AI.vcproj

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Game/Immersive AI.exe

Revision: 27

Author: gavin

Date: 7:18:53 PM, Saturday, 22 July 2006

Message:

More code cleanup. Added proper deconstructor to iAIPathMap, so now you can go into different maps without it crashing! Few optimisations and comment corrections.

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathFind.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.h

Modified : /Game/Immersive AI.exe

Modified : /Game/Immersive AI.torsion.exports

Revision: 26

Author: gavin

Date: 5:59:07 PM, Saturday, 22 July 2006

Message:

Code Cleanup.

Added TypeMasks: iAIPathNodeObjectType, iAIPathObjectType & placeholder for iAIAgentObjectType.

Added ToggleDisplayBoundingBoxes to world editor

Modified : /Engine/engine/game/objectTypes.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.cc

Modified : /Game/Immersive AI.exe

Modified : /Game/Immersive AI.torsion.exports

Modified : /Game/creator/editor/EditorGui.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 25

Author: gavin

Date: 4:52:17 PM, Saturday, 22 July 2006

Message:

Rollback to just terrain path nodes; ran into memory issues or something... need to clean up code for this

---

Modified : /Engine/VS2005/Immersive AI.vcproj

Deleted : /Engine/engine/immersiveAI/seek/path/iAIPathGrid.cc

Deleted : /Engine/engine/immersiveAI/seek/path/iAIPathGrid.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.h

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 24

Author: gavin

Date: 4:31:41 PM, Friday, 21 July 2006

Message:

Heaps of grid bugfixes & changes. Now compiling for all interiors. Needs code cleanup still

---

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathGrid.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathGrid.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.h

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 23

Author: gavin

Date: 11:55:33 AM, Wednesday, 19 July 2006

Message:

Added node object box validity checking

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathGrid.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.h

Revision: 22

Author: gavin

Date: 10:46:29 AM, Wednesday, 19 July 2006

Message:

PathFinding graph rework. Creating now from smaller 'grids' into a larger PathMap; allows for closer mapping around interiors. Start of PathGrid class checkin

---

Modified : /Engine/VS2005/Immersive AI.vcproj

Added : /Engine/engine/immersiveAI/seek/path/iAIPathGrid.cc

Added : /Engine/engine/immersiveAI/seek/path/iAIPathGrid.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.h

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 21

Author: gavin

Date: 2:05:19 PM, Sunday, 16 July 2006

Message:

Fixed path object box. Completed iAIPath

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.h

Modified : /Game/Immersive AI.exe

Modified : /Game/Immersive AI.torsion.exports

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 20

Author: gavin

Date: 10:58:02 PM, Saturday, 15 July 2006

Message:

Add Path to script now ;)

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPath.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Game/Immersive AI.exe

Modified : /Game/Immersive AI.torsion.exports

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 19

Author: gavin

Date: 10:13:59 PM, Saturday, 15 July 2006

Message:

phew big update! Added the A* algorithm. Added a funky path object, that even draws on the screen ;)

Modified : /Engine/VS2005/Immersive AI.vcproj

Added : /Engine/engine/immersiveAI/seek/path/iAIPath.cc

Added : /Engine/engine/immersiveAI/seek/path/iAIPath.h

Added : /Engine/engine/immersiveAI/seek/path/iAIPathFind.cc

Added : /Engine/engine/immersiveAI/seek/path/iAIPathFind.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.h

Modified : /Game/Immersive AI.exe

Modified : /Game/creator/editor/EditorGui.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Revision: 18

Author: gavin

Date: 4:41:37 PM, Saturday, 15 July 2006

Message:

Added GetClosestNode functions - parse a Point3F, retrieve the closest node to that point!

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.h

Modified : /Game/Immersive AI.exe

Modified : /Game/Immersive AI.torsion.exports

Modified : /Game/creator/editor/EditorGui.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 17

Author: gavin

Date: 1:51:20 PM, Saturday, 15 July 2006

Message:

Changes PathMap to a script object

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Game/Immersive AI.exe

Modified : /Game/Immersive AI.torsion.exports

Modified : /Game/server/core/clientConnection.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 16

Author: gavin

Date: 1:39:55 PM, Saturday, 15 July 2006

Message:

Changed neighbours of a node to pointer to another node instead of idx in the master array

---

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.h

Modified : /Game/Immersive AI.exe

Revision: 15

Author: gavin

Date: 10:55:35 AM, Saturday, 15 July 2006

Message:

Added CG DRL!

Modified : /Engine/engine/platform/GLExtFunc.h

Modified : /Engine/engine/platform/platformVideo.cc

Modified : /Engine/engine/platformMacCarb/macCarbGL.cc

Modified : /Engine/engine/platformMacCarb/platformGL.h

Modified : /Engine/engine/platformWin32/platformGL.h

Modified : /Engine/engine/platformWin32/winGL.cc

Modified : /Engine/engine/platformX86UNIX/platformGL.h

Modified : /Engine/engine/platformX86UNIX/x86UNIXGL.cc

Modified : /Engine/engine/sceneGraph/sceneGraph.cc

Modified : /Engine/engine/sceneGraph/sceneGraph.cc.bak

Modified : /Engine/engine/sceneGraph/sceneGraph.h

Modified : /Engine/engine/sceneGraph/sceneGraph.h.bak

Modified : /Engine/engine/sceneGraph/sceneState.cc

Modified : /Engine/engine/sceneGraph/sceneState.h

Modified : /Engine/engine/synapseGaming/contentPacks/lightingPack/sgInterior.cc

Modified : /Engine/engine/synapseGaming/contentPacks/lightingPack/sgLighting.h

Modified : /Engine/engine/synapseGaming/contentPacks/lightingPack/sgNewMethods.cc

Modified : /Engine/engine/terrain/blender.cc

Modified : /Engine/engine/terrain/blender_asm.asm

Modified : /Engine/engine/terrain/fluidRender.cc

Modified : /Engine/engine/terrain/sky.cc

Modified : /Game/Immersive AI.exe

Added : /Game/client/data/CG

Added : /Game/client/data/CG/gammaramp_fp.cg

Added : /Game/client/data/CG/glow_blur_fp.cg

Added : /Game/client/data/CG/glow_combine_fp.cg

Added : /Game/client/data/CG/glow_darken_fp.cg

Added : /Game/client/data/CG/water_fp20.cg

Added : /Game/client/data/CG/water_vp11.cg

Deleted : /Game/client/data/water/water_fp20.cg

Deleted : /Game/client/data/water/water_vp11.cg

Revision: 14

Author: gavin

Date: 9:31:33 PM, Wednesday, 28 June 2006

Message:

Some optimisations - need some work with unreachable nodes...

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.cc

Modified : /Game/Immersive AI.exe

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 13

Author: gavin

Date: 8:35:03 PM, Tuesday, 27 June 2006

Message:

fixed bug with trying to remove a non-existant node!

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Game/Immersive AI.exe

Revision: 12

Author: gavin

Date: 7:04:45 PM, Tuesday, 27 June 2006

Message:

Collision detection working when nodemap is made for the mission! set in iAIPathMap::mCollisionMask. Doesn't generate nodes for inside interiors though, just avoids them *shrug*

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.h

Modified : /Game/Immersive AI.exe

Modified : /Game/Immersive AI.torsion.exports

Modified : /Game/server/core/clientConnection.cs

Modified : /Game/server/core/game.cs

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 11

Author: gavin

Date: 4:03:05 PM, Tuesday, 27 June 2006

Message:

Memory optimizations by changing Vector of neighbours to a U32, instead of other nodes!

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.h

Modified : /Game/Immersive AI.exe

Revision: 10

Author: gavin

Date: 3:41:50 PM, Tuesday, 27 June 2006

Message:

Added terrain creation of path finding nodes. Still needs to detect when obstructed by terrain height, other objects etc

---

Modified : /Engine/VS2005/Immersive AI.vcproj

Modified : /Engine/engine/editor/worldEditor.cc

Modified : /Engine/engine/editor/worldEditor.h

Deleted : /Engine/engine/immersiveAI/seek/path/iAIPathManager.cc

Deleted : /Engine/engine/immersiveAI/seek/path/iAIPathManager.h

Added : /Engine/engine/immersiveAI/seek/path/iAIPathMap.cc

Added : /Engine/engine/immersiveAI/seek/path/iAIPathMap.h

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.cc

Modified : /Engine/engine/immersiveAI/seek/path/iAIPathNode.h

Deleted : /Engine/engine/immersiveAI/seek/path/iAIPathNodeGraph.cc

Deleted : /Engine/engine/immersiveAI/seek/path/iAIPathNodeGraph.h

Modified : /Game/Immersive AI.exe

Modified : /Game/Immersive AI.torsion.exports

Modified : /Game/client/data/missions/stronghold.mis

Modified : /Game/glu2d3d.dll

Modified : /Game/opengl2d3d.dll

Modified : /Game/server/immersiveAI/immersiveAI.cs

Modified : /Immersive AI.torsion.opt

Revision: 9

Author: gavin

Date: 2:50:13 PM, Sunday, 11 June 2006

Message:

Added new Immersive AI icon :)

Added : /Documentation/GFX/Icon

Added : /Documentation/GFX/Icon/Immersive AI.ico

Added : /Documentation/GFX/Icon/iAI.Icon.128x128.png

Added : /Documentation/GFX/Icon/iAI.Icon.128x128.psd

Added : /Documentation/GFX/Icon/iAI.Icon.16x16.png

Added : /Documentation/GFX/Icon/iAI.Icon.16x16.psd

Added : /Documentation/GFX/Icon/iAI.Icon.32x32.png

Added : /Documentation/GFX/Icon/iAI.Icon.32x32.psd

Added : /Documentation/GFX/Icon/iAI.Icon.48x48.png

Added : /Documentation/GFX/Icon/iAI.Icon.48x48.psd

Added : /Documentation/GFX/Icon/iAI.Icon.96x96.png

Added : /Documentation/GFX/Icon/iAI.Icon.96x96.psd

Modified : /Engine/VS2005/Immersive AI.ico

Modified : /Game/Immersive AI.exe

Revision: 8

Author: gavin

Date: 9:20:11 PM, Tuesday, 6 June 2006

Message:

Changed nav nodes to non-collision objects. Fixed bug with toggling display of nav nodes.

Deleted : /Game/client/data/shapes/items/markers/navmarker.dts

Deleted : /Game/client/data/shapes/items/markers/navmarker.ms3d

Deleted : /Game/client/data/shapes/items/markers/navmarker.mtl

Deleted : /Game/client/data/shapes/items/markers/navmarker.obj

Deleted : /Game/client/data/shapes/items/markers/navmarker.qc

Deleted : /Game/client/data/shapes/items/markers/navmarker_map.bmp

Deleted : /Game/client/data/shapes/items/markers/navmarker_skin.jpg

Modified : /Game/server/immersiveAI/seek/path

Modified : /Game/server/immersiveAI/seek/path/path.cs

Revision: 7

Author: gavin

Date: 8:30:25 PM, Tuesday, 6 June 2006

Message:

Added brain logo & wallpaper

Added : /Documentation/GFX/Brain.10.psd

Added : /Documentation/GFX/iAI.Wallpaper.Brain.png

Revision: 6

Author: gavin

Date: 2:08:52 PM, Tuesday, 6 June 2006

Message:

Added creator functions for path finding: showing nav net / nav markers

---

Modified : /Engine/engine/editor/worldEditor.cc

Modified : /Engine/engine/editor/worldEditor.h

Modified : /Game/Immersive AI.exe

Modified : /Game/creator/editor/EditorGui.cs

Modified : /Game/server/immersiveAI

Modified : /Game/server/immersiveAI/seek/path/path.cs

Modified : /Immersive AI.torsion.opt

Revision: 5

Author: gavin

Date: 1:41:58 PM, Tuesday, 6 June 2006

Message:

Added A* Path Finding. Fixed warning with CG water.

Modified : /Engine/VS2005

Modified : /Engine/VS2005/Immersive AI.vcproj

Modified : /Engine/engine/editor/worldEditor.cc

Modified : /Engine/engine/game/gameTSCtrl.cc

Added : /Engine/engine/immersiveAI

Added : /Engine/engine/immersiveAI/seek

Added : /Engine/engine/immersiveAI/seek/path

Added : /Engine/engine/immersiveAI/seek/path/iAIPathManager.cc

Added : /Engine/engine/immersiveAI/seek/path/iAIPathManager.h

Added : /Engine/engine/immersiveAI/seek/path/iAIPathNode.cc

Added : /Engine/engine/immersiveAI/seek/path/iAIPathNode.h

Added : /Engine/engine/immersiveAI/seek/path/iAIPathNodeGraph.cc

Added : /Engine/engine/immersiveAI/seek/path/iAIPathNodeGraph.h

Modified : /Game

Modified : /Game/Immersive AI.exe

Added : /Game/client/data/missions/paths

Modified : /Game/client/data/missions/stronghold.mis

Modified : /Game/client/data/missions/strongholdNight.mis

Modified : /Game/creator/data

Modified : /Game/glu2d3d.dll

Modified : /Game/opengl2d3d.dll

Modified : /Game/server/core/game.cs

Added : /Game/server/immersiveAI

Added : /Game/server/immersiveAI/immersiveAI.cs

Added : /Game/server/immersiveAI/seek

Added : /Game/server/immersiveAI/seek/path

Added : /Game/server/immersiveAI/seek/path/path.cs

Added : /Game/server/immersiveAI/seek/path/path_test.cs

Modified : /Immersive AI.torsion.opt

Revision: 4

Author: gavin

Date: 8:35:06 PM, Monday, 5 June 2006

Message:

Added Thumbs.db to ignore over all folders

Modified : /

Revision: 3

Author: gavin

Date: 8:28:24 PM, Monday, 5 June 2006

Message:

Added ignore list for *.dso and prefs/config files.

Modified : /

Revision: 2

Author: gavin

Date: 7:27:46 PM, Monday, 5 June 2006

Message:

Added Engine/Game - code cleaned and sorted into logical groups. Various stock TGE bug fixes. Added CG water reflection.

Added : /Engine

Added : /Game

Added : /Immersive AI.torsion

Added : /Immersive AI.torsion.opt

Revision: 1

Author: gavin

Date: 3:40:39 PM, Monday, 5 June 2006

Message:

Added documentation - research and design documents

Added : /Documentation

Added : /Documentation/Design

Added : /Documentation/Design/Classes.iAICombat.vsd

Added : /Documentation/Design/Classes.iAIManager.GoalManager.vsd

Added : /Documentation/Design/Classes.iAIManager.StateMachine.vsd

Added : /Documentation/Design/Classes.iAIManager.StateManager.vsd

Added : /Documentation/Design/Classes.iAISeek.vsd

Added : /Documentation/Design/GameAI.Design.doc

Added : /Documentation/Design/GameAI.Design.pdf

Added : /Documentation/Design/System.Diagrams.vsd

Added : /Documentation/Design/System.SubSystems.vsd

Added : /Documentation/GFX

Added : /Documentation/GFX/Thumbs.db

Added : /Documentation/GFX/iAI.large.png

Added : /Documentation/GFX/iAI.logo.png

Added : /Documentation/GameAI.ProjectPlan.mpp

Added : /Documentation/Research

Added : /Documentation/Research/GameAI.Research.v1.0.doc

Added : /Documentation/Research/GameAI.Research.v1.0.pdf

9 APPENDIX B – IAIAGENT.H/.CC SOURCE CODE

9.1 iAIAgent.h

```
//-----------------------------------------------------------------------------
// Immersive AI :: Agent
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------------

//-----------------------------------------------------------------------------
/// @file iAIAgent.h
//-----------------------------------------------------------------------------
/// @class iAIAgent
/// @author Gavin Bunney
/// @version 1.0
/// @brief The AI Agent within the game world.
///
/// The iAIAgent class holds all information about an agent within
/// the game world. It maintains its current goal list, state and
/// vital statistics.<br><br>
/// It is an extension to the standard Torque AIPlayer class. The
/// AIPlayer class handles the movement within the game world and
/// the process of aiming towards an object. All other agent logic is
/// contained within this iAIAgent class.<br><br>
///
/// TypeMask |= iAIAgentObjectType
//-----------------------------------------------------------------------------

#ifndef _IAIAGENT_H_
#define _IAIAGENT_H_

#include "game/aiPlayer.h"
#include "immersiveAI/seek/path/iAIPath.h"

class iAIAgent : public AIPlayer
{
	typedef AIPlayer Parent;
	friend class iAIGoalManager;

public:

	//-----------------------------------------------------------------------------
	/// @var DECLARE_CONOBJECT(iAIAgent)
	/// @brief TorqueScript object.
	//-----------------------------------------------------------------------------
	DECLARE_CONOBJECT(iAIAgent);

	//-----------------------------------------------------------------------------
	/// @fn iAIAgent()
	/// @brief Default constructor.
	//-----------------------------------------------------------------------------
	iAIAgent();

	//-----------------------------------------------------------------------------
	/// @fn ~iAIAgent()
	/// @brief Destructor.
	//-----------------------------------------------------------------------------
	~iAIAgent();

	//-----------------------------------------------------------------------------
	/// @fn static void initPersistFields()
	/// @brief Exposes some variables to TorqueScript.
```

```
      //-----------------------------------------------------------------------
      static void initPersistFields();

      //-----------------------------------------------------------------------
      /// @fn void setCurrentPath(iAIPath* path)
      /// @brief Sets the Agent's path to the parsed path.
      ///
      /// @param path Pointer to the iAIPath for this agent.
      //-----------------------------------------------------------------------
      void setCurrentPath(iAIPath* path) { this->mCurrentPath = path; }

      //-----------------------------------------------------------------------
      /// @fn iAIPath* getCurrentPath()
      /// @brief Returns the Agent's current path.
      ///
      /// @return iAIPath Pointer to the Agent's current path.
      //-----------------------------------------------------------------------
      iAIPath* getCurrentPath() { return this->mCurrentPath; }

      //-----------------------------------------------------------------------
      /// @fn void setAgentType(const char* agentType)
      /// @brief Set the type of agent.
      ///
      /// @param agentType Type of the agent.
      //-----------------------------------------------------------------------
      void setAgentType(const char* agentType) { dSprintf(this->mAgentType,
sizeof(this->mAgentType), "%s", agentType); }

      //-----------------------------------------------------------------------
      /// @fn const char* getAgentType()
      /// @brief Retrieve the type of agent.
      ///
      /// @return char* Type of the agent.
      //-----------------------------------------------------------------------
      const char* getAgentType() const { return this->mAgentType; }

      //-----------------------------------------------------------------------
      /// @fn void setHealth(const S32 health)
      /// @brief Set the current health value.
      ///
      /// @param health Health value to set to.
      //-----------------------------------------------------------------------
      void setHealth(const S32 health);

      //-----------------------------------------------------------------------
      /// @fn const S32 getHealth()
      /// @brief Retrieve the current health value.
      ///
      /// @return S32 Health value.
      //-----------------------------------------------------------------------
      const S32 getHealth();

      //-----------------------------------------------------------------------
      /// @fn void setHappiness(const S32 happiness)
      /// @brief Set the current happiness value.
      ///
      /// @param happiness Happiness value to set to.
      //-----------------------------------------------------------------------
      void setHappiness(const S32 happiness);

      //-----------------------------------------------------------------------
      /// @fn const S32 getHappiness()
      /// @brief Retrieve the current happiness value.
```

```
        ///
        /// @return S8 Happiness value.
        //-------------------------------------------------------------------
        const S32 getHappiness();

        //-------------------------------------------------------------------
        /// @fn void setFatigue(const S32 fatigue)
        /// @brief Set the current fatigue value.
        ///
        /// @param fatigue Fatigue value to set to.
        //-------------------------------------------------------------------
        void setFatigue(const S32 fatigue);

        //-------------------------------------------------------------------
        /// @fn const S32 getFatigue()
        /// @brief Retrieve the current fatigue value.
        ///
        /// @return S8 Fatigue value.
        //-------------------------------------------------------------------
        const S32 getFatigue();

        //-------------------------------------------------------------------
        /// @fn void setMoney(const F32 money)
        /// @brief Set the current money value.
        ///
        /// @param money Money value to set to.
        //-------------------------------------------------------------------
        void setMoney(const F32 money);

        //-------------------------------------------------------------------
        /// @fn const F32 getMoney()
        /// @brief Retrieve the current money value.
        ///
        /// @return F32 Money value.
        //-------------------------------------------------------------------
        const F32 getMoney();

        //-------------------------------------------------------------------
        /// @fn void setBoredom(const S32 boredom)
        /// @brief Set the current boredom value.
        ///
        /// @param boredom Boredom value to set to.
        //-------------------------------------------------------------------
        void setBoredom(const S32 boredom);

        //-------------------------------------------------------------------
        /// @fn const S32 getBoredom()
        /// @brief Retrieve the current boredom value.
        ///
        /// @return S32 Boredom value.
        //-------------------------------------------------------------------
        const S32 getBoredom();

        //-------------------------------------------------------------------
        /// @fn void setLevel(const S32 level)
        /// @brief Set the current level value.
        ///
        /// @param level Level value to set to.
        //-------------------------------------------------------------------
        void setLevel(const S32 level);

        //-------------------------------------------------------------------
        /// @fn const S32 getLevel()
```

```
      /// @brief Retrieve the current level value.
      ///
      /// @return S32 Level value.
      //-----------------------------------------------------------------------
      const S32 getLevel();

protected:

      //-----------------------------------------------------------------------
      /// @var char mAgentType[255]
      /// @brief Name of the agent's type.
      //-----------------------------------------------------------------------
      char mAgentType[255];

      //-----------------------------------------------------------------------
      /// @var iAIPath* mCurrentPath
      /// @brief Pointer to the Agent's current path.
      //-----------------------------------------------------------------------
      iAIPath* mCurrentPath;

      //-----------------------------------------------------------------------
      /// @var S32 mHappiness
      /// @brief Happiness level of the agent. 0 is really angry, 100 is
      ///        really happy.
      //-----------------------------------------------------------------------
      S32 mHappiness;

      //-----------------------------------------------------------------------
      /// @var S32 mFatigue
      /// @brief Fatigue level of the agent. 0 is really awake, 100 is
      ///        really tired.
      //-----------------------------------------------------------------------
      S32 mFatigue;

      //-----------------------------------------------------------------------
      /// @var F32 mMoney
      /// @brief Amount of money the agent has. 0 is really poor, 10 000 is
      ///        really rich.
      //-----------------------------------------------------------------------
      F32 mMoney;

      //-----------------------------------------------------------------------
      /// @var S32 mBoredom
      /// @brief Boredom level of the agent. 0 is really excited, 100 is
      ///        really bored.
      //-----------------------------------------------------------------------
      S32 mBoredom;

      //-----------------------------------------------------------------------
      /// @var S32 mLevel
      /// @brief Level of the agent. 1 is really weak, 100 is really
      ///        strong.
      //-----------------------------------------------------------------------
      S32 mLevel;

private:

      //-----------------------------------------------------------------------
      /// @fn void executeFunction(const char *name)
      /// @brief Executes a TorqueScript function
      ///
      /// @param name The name of the function to execute.
      //-----------------------------------------------------------------------
```

```
      void executeFunction(const char *name);

      //-----------------------------------------------------------------
      /// @fn void executeDatablockFunction(const char *name)
      /// @brief Executes a TorqueScript function contained within the
      ///        agent's datablock.
      ///
      /// @param name The name of the function to execute.
      //-----------------------------------------------------------------
      void executeDatablockFunction(const char *name);
};

#endif
```

9.2 iAIAgent.cc

```
//-----------------------------------------------------------------------
// Immersive AI :: Agent
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------
#include "iAIAgent.h"
#include "console/consoleInternal.h"
#include "core/realComp.h"
#include "math/mMatrix.h"
#include "game/moveManager.h"
#include "game/gameConnection.h"

IMPLEMENT_CO_NETOBJECT_V1(iAIAgent);

iAIAgent::iAIAgent()
{
      mTypeMask |= iAIAgentObjectType;
      this->setMoveTolerance(1.00f);

      this->mDamage = 0.00f;
      this->mHappiness = 100;
      this->mFatigue = 0;
      this->mMoney = 100.0;
      this->mBoredom = 0;
      this->mCurrentPath = 0;
}

iAIAgent::~iAIAgent()
{
      // nothing to destruct.. yet
}

void iAIAgent::initPersistFields()
{
      Parent::initPersistFields();
}

void iAIAgent::executeFunction(const char *name)
{
      Con::executef(2, name);
}

void iAIAgent::executeDatablockFunction(const char *name)
{
      Con::executef(this->getDataBlock(), 2, name, this->scriptThis());
}
```

```
void iAIAgent::setHealth(const S32 health)
{
	if ((health >= 0) && (health <= 100))
	{
		this->setDamageLevel(100 - health);
	} else
	{
		if (health < 0)
			this->setDamageLevel(100);
		else
			this->setDamageLevel(0);
	}
}

const S32 iAIAgent::getHealth()
{
	return (100 - this->getDamageLevel());
}

void iAIAgent::setHappiness(const S32 happiness)
{
	if ((happiness >= 0) && (happiness <= 100))
	{
		this->mHappiness = happiness;
	} else
	{
		if (happiness < 0)
			this->mHappiness = 0;
		else
			this->mHappiness = 100;
	}
}

const S32 iAIAgent::getHappiness()
{
	return this->mHappiness;
}

void iAIAgent::setFatigue(const S32 fatigue)
{
	if ((fatigue >= 0) && (fatigue <= 100))
	{
		this->mFatigue = fatigue;
	} else
	{
		if (fatigue < 0)
			this->mFatigue = 0;
		else
			this->mFatigue = 100;
	}
}

const S32 iAIAgent::getFatigue()
{
	return this->mFatigue;
}

void iAIAgent::setMoney(const F32 money)
{
	if ((money >= 0) && (money <= 10000))
	{
		this->mMoney = money;
```

```
	} else
	{
		if (money < 0)
			this->mMoney = 0;
		else
			this->mMoney = 10000;
	}
}

const F32 iAIAgent::getMoney()
{
	return this->mMoney;
}

void iAIAgent::setBoredom(const S32 boredom)
{
	if ((boredom >= 0) && (boredom <= 100))
	{
		this->mBoredom = boredom;
	} else
	{
		if (boredom < 0)
			this->mBoredom = 0;
		else
			this->mBoredom = 100;
	}
}

const S32 iAIAgent::getBoredom()
{
	return this->mBoredom;
}

void iAIAgent::setLevel(const S32 level)
{
	if ((level >= 1) && (level <= 100))
	{
		this->mLevel = level;
	} else
	{
		if (level < 1)
			this->mLevel = 1;
		else
			this->mLevel = 100;
	}
}

const S32 iAIAgent::getLevel()
{
	return this->mLevel;
}

ConsoleMethodGroupBegin(iAIAgent, ScriptFunctions, "iAIAgent Script
Functions");

ConsoleMethod( iAIAgent, getCurrentPath, S32, 2, 2,
                    "S32 iAIAgent.getCurrentPath() - Retrieves the current
path for the agent.")
{
	iAIPath *currentPath = object->getCurrentPath();
	if (currentPath)
		return (currentPath->getId());
	else
```

```
            return 0;
}

ConsoleMethod( iAIAgent, setCurrentPath, bool, 3, 3,
                  "bool iAIAgent.setCurrentPath(iAIPath path) - Sets the
current path for the agent.")
{
      // ensure pos passed
      if (dStrlen(argv[2]) != 0)
      {
            // find the path in the Sim and set the agent path
            iAIPath *path;
            if (Sim::findObject(dAtoi(argv[2]), path))
            {
                  object->setCurrentPath(path);
                  return true;
            } else
            {
                  Con::errorf("Immersive AI :: Agent :: Path [%d] not
found.", dAtoi(argv[2]));
                  return false;
            }
      } else
      {
            Con::errorf("Immersive AI :: Agent :: path not passed!");
            return false;
      }
}

//-----------------------------------------------------------------------
// Variable accessors/mutators
//-----------------------------------------------------------------------

ConsoleMethod( iAIAgent, setAgentType, void, 3, 3,
                  "void iAIAgent.setAgentType(string agentType) - Set the
type of agent.")
{
      object->setAgentType(argv[2]);
}

ConsoleMethod( iAIAgent, getAgentType, const char*, 2, 2,
                  "string iAIAgent.getAgentType() - Get the type of
agent.")
{
      const char *agentType = object->getAgentType();
      char *returnBuffer = Con::getReturnBuffer(dStrlen(agentType) + 1);
      dSprintf(returnBuffer, sizeof(returnBuffer), "%s", agentType);

      return returnBuffer;
}

ConsoleMethod( iAIAgent, setHealth, void, 3, 3,
                  "void iAIAgent.setHealth(S8 health) - Set the health
level.")
{
      object->setHealth(dAtoi(argv[2]));
}

ConsoleMethod( iAIAgent, getHealth, S32, 2, 2,
                  "S8 iAIAgent.getHealth() - Get the health level.")
{
      return object->getHealth();
}
```

```
ConsoleMethod( iAIAgent, setHappiness, void, 3, 3,
                    "void iAIAgent.setHappiness(S32 happiness) - Set the
happiness level.")
{
      object->setHappiness(dAtoi(argv[2]));
}

ConsoleMethod( iAIAgent, getHappiness, S32, 2, 2,
                    "S8 iAIAgent.getHappiness() - Get the happiness
level.")
{
      return object->getHappiness();
}

ConsoleMethod( iAIAgent, setFatigue, void, 3, 3,
                    "void iAIAgent.setFatigue(S32 fatigue) - Set the
fatigue level.")
{
      object->setFatigue(dAtoi(argv[2]));
}

ConsoleMethod( iAIAgent, getFatigue, S32, 2, 2,
                    "S8 iAIAgent.getFatigue() - Get the fatigue level.")
{
      return object->getFatigue();
}

ConsoleMethod( iAIAgent, setMoney, void, 3, 3,
                    "void iAIAgent.setMoney(F32 money) - Set the money
level.")
{
      object->setMoney(dAtof(argv[2]));
}

ConsoleMethod( iAIAgent, getMoney, S32, 2, 2,
                    "F32 iAIAgent.getMoney() - Get the money level.")
{
      return object->getMoney();
}

ConsoleMethod( iAIAgent, setBoredom, void, 3, 3,
                    "void iAIAgent.setBoredom(S32 boredom) - Set the
boredom level.")
{
      object->setBoredom(dAtoi(argv[2]));
}

ConsoleMethod( iAIAgent, getBoredom, S32, 2, 2,
                    "S8 iAIAgent.getBoredom() - Get the boredom level.")
{
      return object->getBoredom();
}

ConsoleMethod( iAIAgent, setLevel, void, 3, 3,
                    "void iAIAgent.setLevel(S32 level) - Set the agent
level.")
{
      object->setLevel(dAtoi(argv[2]));
}

ConsoleMethod( iAIAgent, getLevel, S32, 2, 2,
                    "S8 iAIAgent.getLevel() - Get the agent level.")
```

```
{
	return object->getLevel();
}

	ConsoleMethodGroupEnd(iAIAgent, ScriptFunctions);
```

10 APPENDIX C – IAIPATHGLOBAL.H SOURCE CODE

10.1 iAIPathGlobal.h

```
//-----------------------------------------------------------------------
// Immersive AI :: Seek :: iAIPathGlobal
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
/// @file iAIPathGlobal.h
//-----------------------------------------------------------------------
/// @class iAIPathGlobal
/// @author Gavin Bunney
/// @version 1.0
/// @brief Global macros for the Seek :: Path functions.
///
/// Holds a collection of the macros used within the various path
/// finding classes.
//-----------------------------------------------------------------------
#ifndef _IAIPATHGLOBAL_H_
#define _IAIPATHGLOBAL_H_

//-----------------------------------------------------------------------
/// @def IAIPATHGLOBAL_COLLISION_MASK
/// @brief Collision mask to detect for node clearance.
//-----------------------------------------------------------------------
#define IAIPATHGLOBAL_COLLISION_MASK                  (InteriorObjectType |
StaticShapeObjectType | VehicleObjectType | PlayerObjectType |
StaticTSObjectType)

//-----------------------------------------------------------------------
/// @def IAIPATHGLOBAL_MAX_SLOPE
/// @brief Max slope between two nodes.
//-----------------------------------------------------------------------
#define IAIPATHGLOBAL_MAX_SLOPE                              100.0f

//-----------------------------------------------------------------------
/// @def IAIPATHGLOBAL_MAX_SMOOTHED_SLOPE
/// @brief Max slope between path nodes for smoothing.
//-----------------------------------------------------------------------
#define IAIPATHGLOBAL_MAX_SMOOTHED_SLOPE        50.0f

//-----------------------------------------------------------------------
/// @def IAIPATHGLOBAL_NODE_CLEARANCE
/// @brief Amount of clearance in X, Y & Z around a node.
//-----------------------------------------------------------------------
#define IAIPATHGLOBAL_NODE_CLEARANCE                  Point3F(1.0, 1.0,
2.3)

//-----------------------------------------------------------------------
/// @def IAIPATHGLOBAL_MOVE_MODIFIER_UNTRAVERSAL
/// @brief MoveModifier for a node to be considered untraversal.
//-----------------------------------------------------------------------
#define IAIPATHGLOBAL_MOVE_MODIFIER_UNTRAVERSAL 100.0f

//-----------------------------------------------------------------------
/// @def IAIPATHGLOBAL_MOVE_MODIFIER_WATER
/// @brief MoveModifier when a node is in water.
//-----------------------------------------------------------------------
#define IAIPATHGLOBAL_MOVE_MODIFIER_WATER       70.0f
```

```
//-----------------------------------------------------------------
/// @def IAIPATHGLOBAL_GRID_BUFFER_INTERIOR
/// @brief Amount of clearance around an interior for a grid.
//-----------------------------------------------------------------
#define IAIPATHGLOBAL_GRID_BUFFER_INTERIOR           Point3F(10.0, 10.0,
0)

//-----------------------------------------------------------------
/// @def IAIPATHGLOBAL_GRID_DENSITY_INTERIOR
/// @brief Density of nodes within an interior.
//-----------------------------------------------------------------
#define IAIPATHGLOBAL_GRID_DENSITY_INTERIOR          2.0f

//-----------------------------------------------------------------
/// @def IAIPATHGLOBAL_GRID_DENSITY_TERRAIN
/// @brief Density of nodes on normal terrain.
//-----------------------------------------------------------------
//#define IAIPATHGLOBAL_GRID_DENSITY_TERRAIN          0.1f
#define IAIPATHGLOBAL_GRID_DENSITY_TERRAIN           0.4f

//-----------------------------------------------------------------
/// @def IAIPATHGLOBAL_GRID_RENDER_CLEARANCE
/// @brief Clearance above node position to render the grid.
//-----------------------------------------------------------------
#define IAIPATHGLOBAL_GRID_RENDER_CLEARANCE          Point3F(0, 0, 0.3f)

//-----------------------------------------------------------------
/// @def IAIPATHGLOBAL_GRID_RENDER_NODE_HEIGHT
/// @brief Height of rendered nodes.
//-----------------------------------------------------------------
#define IAIPATHGLOBAL_GRID_RENDER_NODE_HEIGHT   Point3F(0, 0, 1.0f)

//-----------------------------------------------------------------
/// @def IAIPATHGLOBAL_GRID_RENDER_COLOUR
/// @brief Colour of the rendered grid: R, G, B, Alpha
//-----------------------------------------------------------------
#define IAIPATHGLOBAL_GRID_RENDER_COLOUR        31, 102, 155, 255

//-----------------------------------------------------------------
/// @def IAIPATHGLOBAL_GRID_RENDER_BOX_COLOUR
/// @brief Colour of the rendered grid bounding box: R, G, B, Alpha
//-----------------------------------------------------------------
#define IAIPATHGLOBAL_GRID_RENDER_BOX_COLOUR    255, 0, 0, 255

//-----------------------------------------------------------------
/// @def IAIPATHGLOBAL_GRID_RENDER_NODE_COLOUR
/// @brief Colour of the rendered nodes on a grid: R, G, B, Alpha
//-----------------------------------------------------------------
#define IAIPATHGLOBAL_GRID_RENDER_NODE_COLOUR   255, 255, 0, 255

//-----------------------------------------------------------------
/// @def IAIPATHGLOBAL_PATH_RENDER_CLEARANCE
/// @brief Clearance above node to render the path.
//-----------------------------------------------------------------
#define IAIPATHGLOBAL_PATH_RENDER_CLEARANCE          Point3F(0, 0, 1.0f)

//-----------------------------------------------------------------
/// @def IAIPATHGLOBAL_PATH_RENDER_NODE_HEIGHT
/// @brief Height of rendered path nodes.
//-----------------------------------------------------------------
#define IAIPATHGLOBAL_PATH_RENDER_NODE_HEIGHT   Point3F(0, 0, 1.0f)

//-----------------------------------------------------------------
```

```
/// @def IAIPATHGLOBAL_PATH_RETRY_COUNT
/// @brief Number of retries to perform if a path is not found.
//-----------------------------------------------------------------
#define IAIPATHGLOBAL_PATH_RETRY_COUNT               2

//-----------------------------------------------------------------
/// @def IAIPATHGLOBAL_PATH_SMOOTH_ANGLE_THRESHOLD
/// @brief Angle (degrees) between two points to detect if it can
///        be removed in the path smoothing process. The angle
///        between the two points must be +/- threadhold around 90.
//-----------------------------------------------------------------
#define IAIPATHGLOBAL_PATH_SMOOTH_ANGLE_THRESHOLD    15.0

//-----------------------------------------------------------------
/// @def IAIPATHGLOBAL_INVALID_POSITION
/// @brief Used to detect for invalid position nodes.
//-----------------------------------------------------------------
#define IAIPATHGLOBAL_INVALID_POSITION               Point3F(-1000.0f, -
1000.0f, -1000.0f)


#endif
```

11 APPENDIX D – IAIPATH.H/.CC SOURCE CODE

11.1 iAIPath.h

```
//-----------------------------------------------------------------------
// Immersive AI :: Seek :: iAIPath
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
/// @file iAIPath.h
//-----------------------------------------------------------------------
/// @class iAIPath
/// @author Gavin Bunney
/// @version 1.0
/// @brief Represents a path from one point to another.
///
/// The iAIPath class holds a list of all nodes (positions) from one
/// start position to another end position.
/// <br><br>
/// TypeMask |= iAIPathObjectType
//-----------------------------------------------------------------------
#ifndef _IAIPATH_H_
#define _IAIPATH_H_

#include "iAIPathNode.h"
#include "sceneGraph/sceneState.h"

class iAIPath : public SceneObject
{
	typedef SceneObject Parent;

	friend class iAIAgent;

public:

	//-----------------------------------------------------------------------
	/// @var DECLARE_CONOBJECT(iAIPath)
	/// @brief TorqueScript object.
	//-----------------------------------------------------------------------
	DECLARE_CONOBJECT(iAIPath);

	//-----------------------------------------------------------------------
	/// @fn iAIPath()
	/// @brief Default constructor.
	//-----------------------------------------------------------------------
	iAIPath();

	//-----------------------------------------------------------------------
	/// @fn bool createPath(iAIPathMap* pathMap,
	///                     const Point3F start, const Point3F end,
	///                     const bool smoothPath = true)
	/// @brief Creates a path from the start node to the end node.
	///
	/// @param pathMap Pointer to the pathmap to generate path within.
	/// @param start Point to start the path from.
	/// @param end Point to end the path at.
	/// @param smoothPath Flag to smooth the path. Default true.
	/// @return Path creation success.
	//-----------------------------------------------------------------------
	bool createPath(iAIPathMap* pathMap, Point3F start, Point3F end,
const bool smoothPath = true);
```

```
      //-----------------------------------------------------------------
      /// @fn Point3F getNextPosition()
      /// @brief Retrieves the next position to goto and removes the last
      ///        visited node.
      ///
      /// @return Point3F of next node on the path.
      //-----------------------------------------------------------------
      Point3F getNextPosition();

      //-----------------------------------------------------------------
      /// @fn bool hasNextNode()
      /// @brief Test to see if this path has another node or not.
      ///
      /// @return True if the path has another node.
      //-----------------------------------------------------------------
      bool hasNextNode();

      //-----------------------------------------------------------------
      /// @fn U32 nodeCount()
      /// @brief Retrieve the number of nodes left in the path.
      ///
      /// @return U32 Number of nodes left in the path.
      //-----------------------------------------------------------------
      U32 nodeCount();

      //-----------------------------------------------------------------
      /// @fn bool onAdd()
      /// @brief Called on adding to Sim.
      //-----------------------------------------------------------------
      bool onAdd();

      //-----------------------------------------------------------------
      /// @fn bool onRemove()
      /// @brief Called on removal from Sim.
      //-----------------------------------------------------------------
      void onRemove();

      //-----------------------------------------------------------------
      /// @fn bool prepRenderImage(SceneState *state, const U32 stateKey,
      ///     const U32 startZone, const bool modifyBaseZoneState = false)
      /// @brief Called on scene render. Detects if the path needs to be
      ///        rendered, calls renderObject if need to render the path.
      ///
      /// @param state the current SceneState.
      /// @param stateKey key for the state.
      /// @param startZone zone for scene start.
      /// @param modifyBaseZoneState flag to modify the base zone state.
      /// @return scene state change.
      //-----------------------------------------------------------------
      bool prepRenderImage(SceneState *state, const U32 stateKey, const U32
startZone, const bool modifyBaseZoneState = false);

      //-----------------------------------------------------------------
      /// @fn void renderObject(SceneState *state, SceneRenderImage *image)
      /// @brief Renders the path.
      ///
      /// @param state the current SceneState.
      /// @param image scene to render in.
      //-----------------------------------------------------------------
      void renderObject(SceneState *state, SceneRenderImage *image);

      //-----------------------------------------------------------------
```

```
      /// @fn static void initPersistFields()
      /// @brief Exposes some variables to TorqueScript.
      //-----------------------------------------------------------------------
      static void initPersistFields();

protected:

      //-----------------------------------------------------------------------
      /// @fn void updateWorldBox()
      /// @brief Repositions and resizes the worldbox. Required to ensure
      ///        that path is rendered in the scene.
      //-----------------------------------------------------------------------
      void updateWorldBox();

      //-----------------------------------------------------------------------
      /// @var bool mTraversing
      /// @brief Used to set a flag if the GetNextNode has been called
      ///        previously or not.
      //-----------------------------------------------------------------------
      bool mTraversing;

      //-----------------------------------------------------------------------
      /// @var bool mShow
      /// @brief Render flag to show the path on scene pass.
      //-----------------------------------------------------------------------
      bool mShow;

      //-----------------------------------------------------------------------
      /// @var bool mRenderSpline
      /// @brief Render flag to render as a spline path. If set to false,
      ///        path will render linear.
      //-----------------------------------------------------------------------
      bool mRenderSpline;

      //-----------------------------------------------------------------------
      /// @var ColorI mPathColour
      /// @brief Colour of the rendered path.
      //-----------------------------------------------------------------------
      ColorI mPathColour;

      //-----------------------------------------------------------------------
      /// @var ColorI mPathNodeColour
      /// @brief Colour of the nodes on the rendered path.
      //-----------------------------------------------------------------------
      ColorI mPathNodeColour;

      //-----------------------------------------------------------------------
      /// @var Vector<iAIPathNode*> mPathNodes
      /// @brief Vector of all nodes in the path.
      //-----------------------------------------------------------------------
      Vector<iAIPathNode*> mPathNodes;

      //-----------------------------------------------------------------------
      /// @var iAIPathNode* mLastNode
      /// @brief Pointer to the last node which was returned.
      //-----------------------------------------------------------------------
      iAIPathNode* mLastNode;
};

      #endif
```

11.2 iAIPath.cc

```
//-----------------------------------------------------------------------
// Immersive AI :: Seek :: iAIPath
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

#include "sceneGraph/sceneGraph.h"
#include "game/cameraSpline.h"

#include "iAIPath.h"
#include "iAIPathMap.h"
#include "iAIPathFind.h"
#include "iAIPathGlobal.h"

IMPLEMENT_CO_NETOBJECT_V1(iAIPath);

iAIPath::iAIPath()
{
	this->mTypeMask |= iAIPathObjectType;
	this->setPosition(Point3F(0,0,0));

	this->mTraversing = false;
	this->mShow = false;
	this->mRenderSpline = true;
	this->mLastNode = 0;

	// default path colour is orangey
	this->mPathColour = ColorI(157, 93, 31, 255);

	// default node colour is redish
	this->mPathNodeColour = ColorI(157, 31, 60, 255);
}

bool iAIPath::createPath(iAIPathMap* pathMap, Point3F start, Point3F end,
const bool smoothPath)
{
	iAIPathNode* startNode = pathMap->getClosestNode(start);
	iAIPathNode* endNode = pathMap->getClosestNode(end);

	// check if start and end nodes in the same position
	if (startNode->mPosition == endNode->mPosition)
	{
		// just push on the end node
		this->mPathNodes.push_back(endNode);
		return true;
	}

	// get an instance of the singleton pathfinder
	iAIPathFind* pathFinder = iAIPathFind::getInstance();
	this->mTraversing = false;

	// find the path; if unable to find a path, loop until
IAIPATHGLOBAL_PATH_RETRY_COUNT is reached
	U32 retryCount = 0;
	while ((!(pathFinder->generatePath(startNode, endNode, this-
>mPathNodes, smoothPath))) && (retryCount <=
IAIPATHGLOBAL_PATH_RETRY_COUNT))
		++retryCount;

	// check that a path was found
	if (this->mPathNodes.size() > 0)
	{
```

```
            // update the world box, so that path will render
            this->updateWorldBox();
            return true;
      } else
      {
            Con::errorf("Immersive AI :: Seek :: Unable to find a valid
path from %f, %f, %f to %f, %f, %f", start.x, start.y, start.z, end.x,
end.y, end.z);
            return false;
      }
}

Point3F iAIPath::getNextPosition()
{
      // only pop previous node if we are already traversing!
      if (this->mTraversing)
      {
            // check that this isnt the last node
            if (this->mPathNodes.size() > 1)
            {
                  // update the last node
                  this->mLastNode = this->mPathNodes.front();

                  // remove the previous node from the list
                  this->mPathNodes.pop_front();
            }
      } else
      {
            this->mTraversing = true;
      }

      // return the node, if any left
      if (this->mPathNodes.size() > 0)
      {
            iAIPathNode* returnNode = this->mPathNodes.front();

            // if we are going to return the last node, clear the list
            if (this->mPathNodes.size() == 1)
            {
                  this->mPathNodes.clear();
                  this->mLastNode = 0;
            }

            return returnNode->mPosition;
      } else
      {
            return IAIPATHGLOBAL_INVALID_POSITION;
      }
}

bool iAIPath::hasNextNode()
{
      return (this->mPathNodes.size() > 0);
}

U32 iAIPath::nodeCount()
{
      return this->mPathNodes.size();
}

void iAIPath::updateWorldBox()
{
      // only need a both if there is a path ;)
```

```
      if (this->mPathNodes.size() > 0)
      {
            Point3F min = Point3F(this->mPathNodes.front()->mPosition);
            Point3F max = Point3F(this->mPathNodes.front()->mPosition);

            // iterate over all nodes and find the min & max
            for (U32 i = 0; i < this->mPathNodes.size(); ++i)
            {
                  if (this->mPathNodes[i]->mPosition.x < min.x)
                        min.x = this->mPathNodes[i]->mPosition.x;
                  if (this->mPathNodes[i]->mPosition.y < min.y)
                        min.y = this->mPathNodes[i]->mPosition.y;
                  if (this->mPathNodes[i]->mPosition.z < min.z)
                        min.z = this->mPathNodes[i]->mPosition.z;


                  if (this->mPathNodes[i]->mPosition.x > max.x)
                        max.x = this->mPathNodes[i]->mPosition.x;
                  if (this->mPathNodes[i]->mPosition.y > max.y)
                        max.y = this->mPathNodes[i]->mPosition.y;
                  if (this->mPathNodes[i]->mPosition.z > max.z)
                        max.z = this->mPathNodes[i]->mPosition.z;
            }

            // set position as halfway point
            this->setPosition(min + ((max - min) / 2));

            // create a box to encompass the entire path
            this->mObjBox.min.set(-(this->getPosition() - min));
            this->mObjBox.max.set(max - this->getPosition());

            // must reset world box & transform when changing object box
            this->resetWorldBox();
            this->setRenderTransform(mObjToWorld);
      }
}

bool iAIPath::onAdd()
{
      // call Parent, ensure worked
      if(!Parent::onAdd())
         return false;

      // create object box
      this->updateWorldBox();

      // add to scene
    gClientContainer.addObject(this);
    gClientSceneGraph->addObjectToScene(this);

      return true;
}

void iAIPath::onRemove()
{
      // remove from scene
      removeFromScene();
      Parent::onRemove();
}

bool iAIPath::prepRenderImage(SceneState *state, const U32 stateKey, const
U32 startZone, const bool modifyBaseZoneState)
{
```

```
	// render if there is a path to render and want to show it
	if ((this->mShow) && (this->mPathNodes.size() > 0))
	{
		// return if last state
		if (this->isLastState(state, stateKey)) return false;

		// set last state
		this->setLastState(state, stateKey);

		// see if object rendered
		if (state->isObjectRendered(this))
		{
			// get a SceneRenderImage to show on
			SceneRenderImage* image = new SceneRenderImage;
			image->obj = this;

			// insert into scene image
			state->insertRenderImage(image);
		}
	}

	return false;
}

void iAIPath::renderObject(SceneState *state, SceneRenderImage *image)
{
	// save matrix to restore canonical state
	glPushMatrix();

	// enable blend
	glEnable(GL_BLEND);

	// always rendering lines
	glBegin(GL_LINES);

	// see if we want a linear or spline path
	if (this->mRenderSpline)
	{
		CameraSpline pathSpline;

		// add the lastNode to the spline
		if (this->mLastNode)
		{
			pathSpline.push_back(new CameraSpline::Knot(
					this->mLastNode->mPosition,
					QuatF(0, 0, 0, 0),
					1.0f,
					CameraSpline::Knot::NORMAL,
					CameraSpline::Knot::SPLINE));
		}

		// iterate over all the nodes: add to spline and draw the stick
		for (U32 j = 0; j < this->mPathNodes.size(); j++)
		{
			if (this->mPathNodes[j])
			{
				// add a new knot for each path node
				pathSpline.push_back(new CameraSpline::Knot(
						this->mPathNodes[j]->mPosition,
						QuatF(0, 0, 0, 0),
						1.0f,
						CameraSpline::Knot::NORMAL,
						CameraSpline::Knot::SPLINE));
```

```
                  }

                  // draw the path node
                  glColor4ub(this->mPathNodeColour.red, this-
>mPathNodeColour.green, this->mPathNodeColour.blue, this-
>mPathNodeColour.alpha);
                  glVertex3fv(this->mPathNodes[j]->mPosition +
IAIPATHGLOBAL_PATH_RENDER_CLEARANCE);
                  glVertex3fv(this->mPathNodes[j]->mPosition +
IAIPATHGLOBAL_PATH_RENDER_CLEARANCE +
IAIPATHGLOBAL_PATH_RENDER_NODE_HEIGHT);
            }

            F32 iter = 0.0f;
            Point3F lastPoint = Point3F(0,0,0);

            // draw the entire spline
            while (iter < (pathSpline.size()-1))
            {
                  // get set k to the current knot value
                  CameraSpline::Knot k;
                  pathSpline.value(iter, &k);

                  // advance the spline iter
                  iter = pathSpline.advanceDist(iter, 2.0f);

                  // get the knot point information
                  Point3F newPoint;
                  k.mRotation.mulP(Point3F(0,0,0), &newPoint);
                  newPoint += k.mPosition;

                  // check if there is a last point info stored
                  if (lastPoint == Point3F(0,0,0))
                  {
                        lastPoint = newPoint;
                  } else
                  {
                        glColor4ub(this->mPathColour.red, this-
>mPathColour.green, this->mPathColour.blue, this->mPathColour.alpha);

                        // draw a line between the new point and the last
point
                        glVertex3fv(newPoint +
IAIPATHGLOBAL_PATH_RENDER_CLEARANCE);
                        glVertex3fv(lastPoint +
IAIPATHGLOBAL_PATH_RENDER_CLEARANCE);

                        // set the new last point to the new point
                        lastPoint = newPoint;
                  }
            }

      } else
      {
            // draw a path between the last node and the current start node
            if (this->mLastNode)
            {
                  glColor4ub(this->mPathColour.red, this-
>mPathColour.green, this->mPathColour.blue, this->mPathColour.alpha);
                  glVertex3fv(this->mLastNode->mPosition +
IAIPATHGLOBAL_PATH_RENDER_CLEARANCE);
                  glVertex3fv(this->mPathNodes[0]->mPosition+
IAIPATHGLOBAL_PATH_RENDER_CLEARANCE);
```

```
            }

            for (int j = 1; j < this->mPathNodes.size(); j++)
            {
                  if (this->mPathNodes[j])
                  {
                        // draw the path line
                        glColor4ub(this->mPathColour.red, this-
>mPathColour.green, this->mPathColour.blue, this->mPathColour.alpha);
                        glVertex3fv(this->mPathNodes[j-1]->mPosition +
IAIPATHGLOBAL_PATH_RENDER_CLEARANCE);
                        glVertex3fv(this->mPathNodes[j]->mPosition +
IAIPATHGLOBAL_PATH_RENDER_CLEARANCE);

                        // draw the node stick
                        glColor4ub(this->mPathNodeColour.red, this-
>mPathNodeColour.green, this->mPathNodeColour.blue, this-
>mPathNodeColour.alpha);
                        glVertex3fv(this->mPathNodes[j]->mPosition +
IAIPATHGLOBAL_PATH_RENDER_CLEARANCE);
                        glVertex3fv(this->mPathNodes[j]->mPosition +
IAIPATHGLOBAL_PATH_RENDER_CLEARANCE +
IAIPATHGLOBAL_PATH_RENDER_NODE_HEIGHT);
                  }
            }
      }

      // end of line drawing
      glEnd();

      // disable the blend
      glDisable(GL_BLEND);

      // restore canonical maxtrix state
      glPopMatrix();

      // ensure canonical state is restored
      AssertFatal(dglIsInCanonicalState(), "Error, GL not in canonical
state on exit");
}


void iAIPath::initPersistFields()
{
      Parent::initPersistFields();

      addGroup("Misc");
            addField("showPath", TypeBool, Offset(mShow, iAIPath), "Display
the path on rendering.");
            addField("renderSpline", TypeBool, Offset(mRenderSpline,
iAIPath), "Render the path as a spline. If set to false, will render as
linear.");
            addField("pathColour", TypeColorI, Offset(mPathColour,
iAIPath), "The colour of the rendered path.");
            addField("pathNodeColour", TypeColorI, Offset(mPathNodeColour,
iAIPath), "The colour of the nodes on the rendered path.");
      endGroup("Misc");
}

ConsoleMethodGroupBegin(iAIPath, ScriptFunctions, "iAIPath Script
Functions");

ConsoleMethod( iAIPath, createPath, bool, 4, 5,
```

```
                    "bool iAIPath.createPath(Point3F start, Point3F goal,
bool smoothPath = true) - Create a path between the two points.")
{
      // ensure pos passed
      if ((dStrlen(argv[2]) != 0) && (dStrlen(argv[3]) != 0))
      {
            // parse arguments into points and create the path
            Point3F start;
            Point3F goal;
            dSscanf(argv[2], "%f %f %f", &start.x, &start.y, &start.z);
            dSscanf(argv[3], "%f %f %f", &goal.x, &goal.y, &goal.z);

            // get the path map variable
            iAIPathMap* pathMap = 0;
            if (Sim::findObject(dAtoi(Con::getVariable("$iAIPathMap")),
pathMap))
            {
                  // see of the smoothPath parameter is set
                  if (dStrlen(argv[4]) != 0)
                        return (object->createPath(pathMap, start, goal,
dAtob(argv[4])));
                  else
                        return (object->createPath(pathMap, start, goal));
            } else
            {
                  Con::errorf("Immersive AI :: Seek :: Path - unable to
find the iAIPathMap");
                  return false;
            }
      } else
      {
            Con::errorf("Immersive AI :: Seek :: Path- not enough nodes
passed to CreatePath!");
            return false;
      }
}

ConsoleMethod( iAIPath, nextPosition, const char*, 2, 2,
                    "Point3F iAIPath.nextPosition() - Get the next position
on the path.")
{
      char *returnBuffer = Con::getReturnBuffer(256);

      Point3F nextPosition = object->getNextPosition();
      if (nextPosition != IAIPATHGLOBAL_INVALID_POSITION)
      {
            dSprintf(returnBuffer, 256, "%f %f %f", nextPosition.x,
nextPosition.y, nextPosition.z);
      } else
      {
            dSprintf(returnBuffer, 256, "");
      }

      return returnBuffer;
}

ConsoleMethod( iAIPath, hasNextNode, bool, 2, 2,
                    "bool iAIPath.hasNextNode() - Returns if the path has
another node.")
{
      return (object->hasNextNode());
}
```

```
ConsoleMethod( iAIPath, nodeCount, S32, 2, 2,
                    "U32 iAIPath.nodeCount() - Returns number of nodes left
in the path.")
{
      return (object->nodeCount());
}


      ConsoleMethodGroupEnd(iAIPath, ScriptFunctions);
```

12 APPENDIX E – IAIPATHFIND.H/.CC SOURCE CODE

12.1 iAIPathFind.h

```
//---------------------------------------------------------------------
// Immersive AI :: Seek :: iAIPathFind
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//---------------------------------------------------------------------

//---------------------------------------------------------------------
/// @file iAIPathFind.h
//---------------------------------------------------------------------
/// @class iAIPathFind
/// @author Gavin Bunney
/// @version 1.0
/// @brief Finds a path from one node to another.
///
/// A Singleton class which implements an A* pathfinding algorithm
/// to find the easiest and shortest path from one node to another.
//---------------------------------------------------------------------
#ifndef _IAIPATHFIND_H_
#define _IAIPATHFIND_H_

#include "iAIPathNode.h"
#include "iAIPathGlobal.h"

class iAIPathFind {

public:

      //-----------------------------------------------------------------
      /// @fn ~iAIPathFind()
      /// @brief Default deconstructor.
      //-----------------------------------------------------------------
      ~iAIPathFind() { };

      //-----------------------------------------------------------------
      /// @fn static iAIPathFind* getInstance()
      /// @brief Retrieves the instance of the singleton. Creates a new
      ///        instance if one doesn't exist.
      ///
      /// @return Singleton instance pointer.
      //-----------------------------------------------------------------
      static iAIPathFind* getInstance();

      //-----------------------------------------------------------------
      /// @fn bool generatePath(iAIPathNode* startNode,
      ///                       iAIPathNode* goalNode,
      ///                       Vector<iAIPathNode*> &replyList,
      ///                       const bool smoothPath = true)
      /// @brief Performs an A* path finding algorithm to find a path from
      ///        the parsed startNode to the goalNode. Path is returned in
      ///        the replyList.
      ///
      /// @param startNode Pointer to the start node.
      /// @param goalNode Pointer to the goal node.
      /// @param replyList Vector to place the returned path in.
      /// @param smoothPath Flag to smooth the path. Default true.
      /// @return Path creation success.
      //-----------------------------------------------------------------
      bool generatePath(iAIPathNode* startNode, iAIPathNode* goalNode,
Vector<iAIPathNode*> &replyList, const bool smoothPath = true);
```

```
private:

      //-------------------------------------------------------------------
      /// @fn iAIPathFind::iAIPathFind()
      /// @brief Default constructor.
      //-------------------------------------------------------------------
      iAIPathFind() { };

      //-------------------------------------------------------------------
      /// @var static iAIPathFind* mInstance
      /// @brief Instance of the singleton.
      //-------------------------------------------------------------------
      static iAIPathFind* mInstance;

      //-------------------------------------------------------------------
      /// @fn inline F32 estimateCostToGoal(iAIPathNode* node,
      ///                                   iAIPathNode* goal)
      /// @brief Performs a vector length from node to goal, to determine
      ///        the cost from the node to the goal.
      ///
      /// @param node Pointer to the start node.
      /// @param goal Pointer to the goal node.
      /// @return Cost to goal.
      //-------------------------------------------------------------------
      inline F32 estimateCostToGoal(iAIPathNode* node, iAIPathNode* goal);

      //-------------------------------------------------------------------
      /// @fn void smoothPath(Vector<iAIPathNode*> &replyList)
      /// @brief Smooths the path to further optimise the A* algorithm and
      ///        make AI agents traverse a path more naturally.
      ///
      /// @param replyList Vector of iAIPathNode's to smooth.
      //-------------------------------------------------------------------
      void smoothPath(Vector<iAIPathNode*> &replyList);

      //-------------------------------------------------------------------
      /// @fn bool smoothPathConnectionValid(const Point3F from,
      ///          const Point3F to)
      /// @brief Checks that a connection from one point to another is
      ///        valid.
      ///
      /// @param from Point to start from.
      /// @param from Point to go to.
      /// @return True if connection is valid.
      //-------------------------------------------------------------------
      bool smoothPathConnectionValid(Point3F from, Point3F to);

      //-------------------------------------------------------------------
      /// @fn void resetNodeVariables(Vector<iAIPathNode*> &affectedList)
      /// @brief Resets the path finding variables of the nodes.
      ///
      /// @param affectedList Vector of nodes which were changed during
      ///        the pathfinding algorithm.
      //-------------------------------------------------------------------
      void resetNodeVariables(Vector<iAIPathNode*> &affectedList);
};

      #endif
```

12.2 iAIPathFind.cc

```
//-----------------------------------------------------------------------
// Immersive AI :: Seek :: iAIPathFind
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

#include "game/gameConnection.h"
#include "dgl/splineUtil.h"
#include "platform/profiler.h"

#include "immersiveAI/core/tBinaryHeap.h"
#include "iAIPathFind.h"
#include "iAIPathMap.h"

iAIPathFind* iAIPathFind::mInstance = 0;

iAIPathFind* iAIPathFind::getInstance()
{
	// if an instance doesn't exist yet, create one!
	if (!mInstance)
	{
		mInstance = new iAIPathFind();
	}
	return mInstance;
}

static S32 BINARYHEAP_COMPARE pathNodeFitnessCompare( const void* a, const
void* b )
{
	// compare the fitness of a & b
	F32 aCol = ((iAIPathNode *)(a))->mFitness;
	F32 bCol = ((iAIPathNode *)(b))->mFitness;
	F32 diff = aCol - bCol;
	S32 reply = diff < 0 ? -1 : (diff > 0 ? 1 : 0);
	return reply;
}

bool iAIPathFind::generatePath(iAIPathNode* startNode, iAIPathNode*
goalNode, Vector<iAIPathNode*> &replyList, const bool smoothPath)
{
	PROFILE_SCOPE(iAIPathFind_generatePath);

	// openList is a binary heap
	BinaryHeap<iAIPathNode*> openList(iAIPathMap::smNodeCount,
pathNodeFitnessCompare);

	// list of all nodes which were affected during this pathfinding
	Vector<iAIPathNode*> affectedList;

	// add start node to open list
	startNode->mHeuristicCostToGoal = estimateCostToGoal(startNode,
goalNode);
	startNode->mFitness = startNode->mLowestCostFromStart + startNode-
>mHeuristicCostToGoal;
	startNode->mOpen = true;
	openList.push(startNode);

	// keep searching while nodes in open list
	while (openList.size() > 0)
	{
		// first element is the lowest cost
		iAIPathNode* currentNode = openList.front();
```

```
            // remove the first element from the openList
            openList.pop();

            // add the current node to affected list
            affectedList.push_back(currentNode);

            // set it as closed
            currentNode->mOpen = false;
            currentNode->mClosed = true;

            // iterate over all its neighbours
            for (U32 i = 0; i < currentNode->mNeighbours.size(); ++i)
            {
                  iAIPathNode* currentNeighbour = currentNode-
>mNeighbours[i];

                  // add current neighbour to affected list
                  affectedList.push_back(currentNeighbour);

                  // its its closed, or not walkable, ignore the neighbour
                  if ((currentNeighbour->mClosed) || (currentNeighbour-
>mMoveModifier >= IAIPATHGLOBAL_MOVE_MODIFIER_UNTRAVERSAL))
                        continue;

                  // if its not open, add it
                  if (!currentNeighbour->mOpen)
                  {
                        // set this neighbours parent as the current node
                        currentNeighbour->mParent = currentNode;

                        // set the lowest cost, heuristic and fitness
                        currentNeighbour->mLowestCostFromStart =
currentNeighbour->mParent->mLowestCostFromStart +
estimateCostToGoal(currentNeighbour, currentNeighbour->mParent);
                        currentNeighbour->mHeuristicCostToGoal =
estimateCostToGoal(currentNeighbour, goalNode);
                        currentNeighbour->mFitness = currentNeighbour-
>mLowestCostFromStart + currentNeighbour->mHeuristicCostToGoal +
currentNeighbour->mMoveModifier;

                        // set as open node
                        currentNeighbour->mOpen = true;

                        // add to open list
                        openList.push(currentNeighbour);
                  } else
                  {
                        // see neighbour already has the current node as
its parent
                        if (currentNeighbour->mParent != currentNode)
                        {
                              // see if this neighbour is a quicker path
                              F32 costFromThisNode = currentNode-
>mLowestCostFromStart + estimateCostToGoal(currentNeighbour, currentNode) +
currentNeighbour->mMoveModifier;
                              if (currentNeighbour->mLowestCostFromStart >
costFromThisNode)
                              {
                                    // neighbour is better
                                    currentNeighbour->mLowestCostFromStart
= costFromThisNode;
```

```
                                    currentNeighbour->mParent =
currentNode;
                              }
                        }

                        // close the node
                        currentNeighbour->mClosed = false;
                  }

                  // see if we have reached the end yet
                  if (currentNeighbour == goalNode)
                  {
                        // go back over all the nodes parents and construct
the path
                        iAIPathNode* currentTraceNode = goalNode;

                        // keep going whilst able to find a parent
                        while (currentTraceNode) {

                              // add the node to the reply list
                              replyList.push_front(currentTraceNode);

                              // set parent as parents parent
                              currentTraceNode = currentTraceNode->mParent;
                        }

                        Con::iAIMessagef("Immersive AI :: Seek :: Path
found!");

                        // smooth the path
                        if (smoothPath)
                        {
                              Con::iAIMessagef("Immersive AI :: Seek ::
Smoothing path... %d nodes to start", replyList.size());
                              this->smoothPath(replyList);
                              Con::iAIMessagef("Immersive AI :: Seek ::
Path smoothed... %d nodes now", replyList.size());
                        }

                        // reset the affected node pathfinding variables
                        this->resetNodeVariables(affectedList);
                        affectedList.clear();

                        // found a path - return happy
                        return true;
                  }
            }
      }

      // reset the affected node pathfinding variables
      this->resetNodeVariables(affectedList);
      affectedList.clear();

      // couldn't find a path!
      return false;
}

inline F32 iAIPathFind::estimateCostToGoal(iAIPathNode* from, iAIPathNode*
goal)
{
      return (goal->mPosition - from->mPosition).len();
}
```

```
void iAIPathFind::smoothPath(Vector<iAIPathNode*> &replyList)
{
	PROFILE_SCOPE(iAIPathFind_smoothPath);
	U32 iter = 0;
	bool erasedNode = false;
	while (iter < (replyList.size()-2))
	{
		erasedNode = false;

		// check height difference from a [iter] to b [iter+1]
		Point3F vec = replyList[iter]->mPosition - replyList[iter+1]-
>mPosition;
		F32 zSq = vec.z * vec.z;

		// only see if we can remove b if less than the max slope
		// prevents things like making a path from one mountain top to
another
		if (zSq < IAIPATHGLOBAL_MAX_SMOOTHED_SLOPE)
		{
			// determine angle difference
			F32 angle = mRadToDeg(mDot(replyList[iter]->mPosition,
replyList[iter+2]->mPosition) / (replyList[iter]->mPosition *
replyList[iter+2]->mPosition).len());

			// check if angle is within acceptable range
			if (((90-IAIPATHGLOBAL_PATH_SMOOTH_ANGLE_THRESHOLD) <
angle) && (angle < (90+IAIPATHGLOBAL_PATH_SMOOTH_ANGLE_THRESHOLD)))
			{
				// check that it is a valid connection (avoiding
terrain aswell)
				if (smoothPathConnectionValid(replyList[iter]-
>mPosition, replyList[iter+2]->mPosition))
				{
					// remove b [iter+1]
					replyList.erase(iter+1);
					erasedNode = true;
				}
			}
		}

		// only increment the iter if no nodes were deleted from the
path
		if (!erasedNode)
			iter++;
	}
}

bool iAIPathFind::smoothPathConnectionValid(Point3F from, Point3F to)
{
	// adjust positions to check slightly above terrain
	from.z += IAIPATHGLOBAL_NODE_CLEARANCE.z;
	to.z += IAIPATHGLOBAL_NODE_CLEARANCE.z;

	RayInfo dummy;

	// if we can't get from node to neighbour without colliding, it is
untraversal
	if (gServerContainer.castRay(from, to, IAIPATHGLOBAL_COLLISION_MASK |
TerrainObjectType, &dummy))
		return false;

	return true;
}
```

```
void iAIPathFind::resetNodeVariables(Vector<iAIPathNode*> &affectedList)
{
	PROFILE_SCOPE(iAIPathFind_resetNodeVariables);

	// iterate over the affected list and reset the path finding
variables
	for (U32 i = 0; i < affectedList.size(); i++)
	{
		affectedList[i]->mFitness = 0.0f;
		affectedList[i]->mLowestCostFromStart = 0.0f;
		affectedList[i]->mHeuristicCostToGoal = 0.0f;
		affectedList[i]->mParent = 0;
		affectedList[i]->mOpen = false;
		affectedList[i]->mClosed = false;
	}
	}
```

13 APPENDIX F – TBINARYHEAP.H SOURCE CODE

13.1 tBinaryHeap.h

```
//------------------------------------------------------------------------
// Immersive AI :: Core :: tBinaryHeap
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//------------------------------------------------------------------------

//------------------------------------------------------------------------
/// @file tBinaryHeap.h
//------------------------------------------------------------------------
/// @class BinaryHeap
/// @author Gavin Bunney
/// @version 1.0
/// @brief Binary Heap template object.
///
/// Implementation of a binary heap (template) object.
//------------------------------------------------------------------------
#ifndef _TBINARYHEAP_H_
#define _TBINARYHEAP_H_

#ifndef _PLATFORM_H_
      #include "platform/platform.h"
#endif

//------------------------------------------------------------------------
/// @def BINARYHEAP_COMPARE
/// @brief Compare function callback type for comparing elements in
///        the binary heap.
//------------------------------------------------------------------------
#define BINARYHEAP_COMPARE FN_CDECL

template<class T> class BinaryHeap
{

public:

      //------------------------------------------------------------------
      /// @fn BinaryHeap(const U32 heapSize, S32 (BINARYHEAP_COMPARE *fcmp)
      ///                                    (const void *, const void *))
      /// @brief Creates a new BinaryHeap with the parsed size and using
      ///        the parsed compare function.
      ///
      /// @param heapSize Size of the binary heap.
      //------------------------------------------------------------------
      BinaryHeap(const U32 heapSize, S32 (BINARYHEAP_COMPARE *fcmp)(const
void *, const void *));

      //------------------------------------------------------------------
      /// @fn ~BinaryHeap()
      /// @brief Deconstructor which frees the binary heaps memory.
      //------------------------------------------------------------------
      ~BinaryHeap();

      //------------------------------------------------------------------
      /// @fn S32 size() const
      /// @brief Returns the number of elements contained within the heap.
      ///
      /// @return S32 Number of elements in the heap.
      //------------------------------------------------------------------
      S32 size() const;
```

```
//-----------------------------------------------------------------------
/// @fn bool empty() const
/// @brief True if the heap contains no elements; False otherwise.
///
/// @return Heap contains no elements.
//-----------------------------------------------------------------------
bool empty() const;

//-----------------------------------------------------------------------
/// @fn T& front()
/// @brief Retrieves the front element of the binary heap.
///
/// @return T& Front element.
//-----------------------------------------------------------------------
T& front();

//-----------------------------------------------------------------------
/// @fn const T& front() const
/// @brief Retrieves the front element of the binary heap, enforcing
///        const.
///
/// @return T& Front element.
//-----------------------------------------------------------------------
const T& front() const;

//-----------------------------------------------------------------------
/// @fn void push(const T& x)
/// @brief Adds an element to the binary heap.
///
/// @param x Element to add to the heap.
//-----------------------------------------------------------------------
void push(const T& x);

//-----------------------------------------------------------------------
/// @fn void pop()
/// @brief Removes the front element from the binary heap.
//-----------------------------------------------------------------------
void pop();

//-----------------------------------------------------------------------
/// @fn T& operator[](U32 index)
/// @brief Index operator to retrieve the element at the specified
///        U32 index (array indexed from 1).
///
/// @param index of the element (array indexed from 1).
/// @return T& Element at specifed index.
//-----------------------------------------------------------------------
T& operator[](U32 index);

//-----------------------------------------------------------------------
/// @fn const T& operator[](U32 index) const;
/// @brief Index operator to retrieve the element at the specified
///        U32 index (array indexed from 1), enforcing const.
///
/// @param index of the element (array indexed from 1).
/// @return T& Element at specifed index.
//-----------------------------------------------------------------------
const T& operator[](U32 index) const;

//-----------------------------------------------------------------------
/// @fn const T& operator[](S32 i) const;
/// @brief Index operator to retrieve the element at the specified
```

```
        ///        S32 index (array indexed from 1), enforcing const.
        ///
        /// @param i Index of the element (array indexed from 1).
        /// @return T& Element at specifed index.
        //-------------------------------------------------------------------
        T& operator[](S32 i) { return operator[](U32(i)); }

        //-------------------------------------------------------------------
        /// @fn const T& operator[](S32 i) const;
        /// @brief Index operator to retrieve the element at the specified
        ///        S32 index, (array indexed from 1) enforcing const.
        ///
        /// @param i Index of the element (array indexed from 1).
        /// @return T& Element at specifed index.
        //-------------------------------------------------------------------
        const T& operator[](S32 i) const { return operator[](U32(i)); }

        //-------------------------------------------------------------------
        /// @fn const T& operator[](S32 i) const;
        /// @brief Index operator to retrieve the element at the specified
        ///        S32 index, (array indexed from 1) enforcing const.
        ///
        /// @param i Index of the element (array indexed from 1).
        /// @return U32 Max number of elements placeable in this heap.
        //-------------------------------------------------------------------
        U32 capacity() const;

        //-------------------------------------------------------------------
        /// @fn T* address() const;
        /// @brief Address of the binary heap.
        ///
        /// @return T* Memory address of the binary heap array.
        //-------------------------------------------------------------------
        T* address() const;

protected:

        //-------------------------------------------------------------------
        /// @var U32 mElementCount
        /// @brief Count of elements contained in the binary heap.
        //-------------------------------------------------------------------
        U32 mElementCount;

        //-------------------------------------------------------------------
        /// @var U32 mArraySize
        /// @brief Size of the binary heap array.
        //-------------------------------------------------------------------
        U32 mArraySize;

        //-------------------------------------------------------------------
        /// @var T* mArray
        /// @brief Array of data contained in the binary heap.
        //-------------------------------------------------------------------
        T* mArray;

        //-------------------------------------------------------------------
        /// @var S32 (BINARYHEAP_COMPARE *mCompareFunction)(const void *,
        ///                                                 const void *)
        /// @brief Pointer to the compare function used when sorting the
        ///        binary heap.
        ///
        /// When the function is executed, a result of -1, 0 or 1 is
        /// expected.<br>
```

```
	/// Where:<br>
	/// -1: element a &lt; element b<br>
	/// 0: element a == element b<br>
	/// 1: element a &gt; element b<br>
	//-------------------------------------------------------------------
	S32 (BINARYHEAP_COMPARE *mCompareFunction)(const void *, const void
*);
};

template<class T> inline BinaryHeap<T>::~BinaryHeap()
{
	this->mCompareFunction = 0;
	delete[] this->mArray;
}

template<class T> inline BinaryHeap<T>::BinaryHeap(const U32 heapSize, S32
(BINARYHEAP_COMPARE *fcmp)(const void *, const void *))
{
	// array size is +1 as heap is indexed from 1
	this->mArraySize = heapSize + 1;
	this->mArray = new T[this->mArraySize];
	this->mElementCount = 0;
	this->mCompareFunction = fcmp;
}

template<class T> inline T* BinaryHeap<T>::address() const
{
	return this->mArray;
}

template<class T> inline T& BinaryHeap<T>::front()
{
	return this->mArray[1];
}

template<class T> inline const T& BinaryHeap<T>::front() const
{
	return this->mArray[1];
}

template<class T> inline S32 BinaryHeap<T>::size() const
{
	return (S32)this->mElementCount;
}

template<class T> inline bool BinaryHeap<T>::empty() const
{
	return (this->mElementCount == 0);
}

template<class T> inline void BinaryHeap<T>::push(const T& x)
{
	++this->mElementCount;
	this->mArray[this->mElementCount] = x;

	// reorder the openList
	U32 iter = this->mElementCount;
	while (iter != 1)
	{
		// check if child <= parent
		if (this->mCompareFunction(this->mArray[iter], this-
>mArray[iter/2]) < 1)
		{
```

```
                  T temp = this->mArray[iter/2];
                  this->mArray[iter/2] = this->mArray[iter];
                  this->mArray[iter] = temp;
                  iter = iter / 2;
            } else
            {
                  break;
            }
      }
}

template<class T> inline void BinaryHeap<T>::pop()
{
      // remove the first element
      this->mArray[1] = this->mArray[this->mElementCount];
      --this->mElementCount;

      U32 u = 0, v = 1;
      while (u != v)
      {
            u = v;

            // if both children exist
            if ((2*u + 1) <= this->mElementCount)
            {
                  // check if parent is greater than each child
                  if (this->mCompareFunction(this->mArray[v], this-
>mArray[2*u]) > -1)
                        v = 2*u; // child 1
                  if (this->mCompareFunction(this->mArray[v], this-
>mArray[2*u+1]) > -1)
                        v = 2*u+1; // child 2
            } else
            {
                  // check if only 1 child
                  if (2*u <= this->mElementCount)
                  {
                        // check if parent is greater than child 1
                        if (this->mCompareFunction(this->mArray[u], this-
>mArray[2*u]) > -1)
                              v = 2*u;
                  }
            }

            // if parent is > one of its children, swap them
            if (u != v)
            {
                  T temp = this->mArray[u];
                  this->mArray[u] = this->mArray[v];
                  this->mArray[v] = temp;
            }
      }
}

template<class T> inline T& BinaryHeap<T>::operator[](U32 index)
{
      return this->mArray[index+1];
}

template<class T> inline const T& BinaryHeap<T>::operator[](U32 index)
const
{
      return this->mArray[index+1];
```

```
}

template<class T> inline U32 BinaryHeap<T>::capacity() const
{
    return this->mArraySize;
}

      #endif
```

14 APPENDIX G – IAIPATHMAP.H/.CC SOURCE CODE

14.1 iAIPathMap.h

```
//-----------------------------------------------------------------------
// Immersive AI :: Seek :: iAIPathMap
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
/// @file iAIPathMap.h
//-----------------------------------------------------------------------
/// @class iAIPathMap
/// @author Gavin Bunney
/// @version 1.0
/// @brief Map of all nodes within the mission.
///
/// Holds a collection of all grids (collections of nodes) for the
/// current server map. The map creates grids and maps links between
/// them allowing for A* pathfinding.
//-----------------------------------------------------------------------
#ifndef _IAIPATHMAP_H_
#define _IAIPATHMAP_H_

#include "iAIPathMap.h"
#include "iAIPathGrid.h"
#include "iAIPathNode.h"

class iAIPathMap : public SimObject
{
      typedef SimObject Parent;
      friend class iAIPath;

public:

      //-----------------------------------------------------------------------
      /// @var DECLARE_CONOBJECT(iAIPathMap)
      /// @brief TorqueScript object.
      //-----------------------------------------------------------------------
      DECLARE_CONOBJECT(iAIPathMap);

      //-----------------------------------------------------------------------
      /// @fn iAIPathMap()
      /// @brief Default constructor.
      //-----------------------------------------------------------------------
      iAIPathMap();

      //-----------------------------------------------------------------------
      /// @fn ~iAIPathMap()
      /// @brief Deconstructor which clears all the grids.
      //-----------------------------------------------------------------------
      ~iAIPathMap();

      //-----------------------------------------------------------------------
      /// @fn bool initialize()
      /// @brief Initializes the pathmap for the current server map.
      ///
      /// @return initialization success.
      //-----------------------------------------------------------------------
      bool initialize();

      //-----------------------------------------------------------------------
```

```
      /// @fn bool createPathMap()
      /// @brief Creates the pathmap for the current server map.
      ///
      /// @return creation success.
      //-----------------------------------------------------------------------
      bool createPathMap();

      //-----------------------------------------------------------------------
      /// @fn void clearMap()
      /// @brief Clears the map.
      //-----------------------------------------------------------------------
      void clearMap();

      //-----------------------------------------------------------------------
      /// @fn void toggleDisplay()
      /// @brief Toggle displaying of the pathmap.
      //-----------------------------------------------------------------------
      void toggleDisplay();

      //-----------------------------------------------------------------------
      /// @fn iAIPathNode* getClosestNode(const Point3F position)
      /// @brief Retrieves the closest node to the parsed position
      ///
      /// @param position world point of node to find
      /// @return pointer to closest node
      //-----------------------------------------------------------------------
      iAIPathNode* getClosestNode(const Point3F position);

      //-----------------------------------------------------------------------
      /// @fn static U32 smNodeCount
      /// @brief Total count of nodes in the Path Map.
      //-----------------------------------------------------------------------
      static U32 smNodeCount;

protected:

      //-----------------------------------------------------------------------
      /// @var Vector<iAIPathGrid*> mGrids
      /// @brief Vector of all grids in the world.
      //-----------------------------------------------------------------------
      Vector<iAIPathGrid*> mGrids;

      //-----------------------------------------------------------------------
      /// @var bool mCompiled
      /// @brief Flag for when a pathmap has been compiled successfully.
      //-----------------------------------------------------------------------
      bool mCompiled;

      //-----------------------------------------------------------------------
      /// @var U32 mTerrainGridIndex
      /// @brief Holds the index of the terrain grid within mGrids.
      //-----------------------------------------------------------------------
      U32 mTerrainGridIndex;
};

      #endif
```

14.2 iAIPathMap.cc

```
//-----------------------------------------------------------------------
// Immersive AI :: Seek :: iAIPathMap
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

#include "game/missionArea.h"
#include "platform/profiler.h"
#include "terrain/terrData.h"
#include "game/gameConnection.h"
#include "interior/interiorInstance.h"

#include "iAIPathMap.h"
#include "iAIPathGlobal.h"
#include "iAIPathGrid.h"
#include "iAIPathNode.h"

IMPLEMENT_CONOBJECT(iAIPathMap);

U32 iAIPathMap::smNodeCount = 0;

iAIPathMap::iAIPathMap()
{
	this->mCompiled = false;
	this->mTerrainGridIndex = 0;
}

iAIPathMap::~iAIPathMap()
{
	this->clearMap();
}

bool iAIPathMap::initialize()
{
	// if not compiled, create the path map
	if (!this->mCompiled)
	{
		// create a new pathmap
		this->mCompiled = this->createPathMap();
	}
	return this->mCompiled;
}

bool iAIPathMap::createPathMap()
{
	Con::iAIMessagef("Immersive AI :: Seek :: Building PathMap...");

	// clear any current path map
	this->clearMap();

	// generate an avoid list of all the grids
	Vector<Box3F> avoidList;
	for (U32 i = 0; i < this->mGrids.size(); ++i)
		avoidList.push_back(this->mGrids[i]->getWorldBox());

	// calculate the entire mission area
	MissionArea *missionAreaPtr =
dynamic_cast<MissionArea*>(Sim::findObject("MissionArea"));
	if (!missionAreaPtr)
		return false;
```

```
	// set grid points are the initial mission area points in x&y to the
extent of the mission area
	Point3F gridStart = Point3F(missionAreaPtr->getArea().point.x,
missionAreaPtr->getArea().point.y, 100.0);
	Point3F gridEnd = Point3F(missionAreaPtr->getArea().point.x +
missionAreaPtr->getArea().extent.x, missionAreaPtr->getArea().point.y +
missionAreaPtr->getArea().extent.y, 100.0);

	// pointer for the terrain grid
	iAIPathGrid *terrainGrid = new iAIPathGrid();

	// generate a grid for the terrain, avoiding the avoid list
	if (terrainGrid->createTerrainGrid(gridStart, gridEnd, avoidList,
IAIPATHGLOBAL_GRID_DENSITY_TERRAIN))
	{
		// add to pathmap collection
		this->mGrids.push_back(terrainGrid);

		// set the type of the terrain grid to just a zone type;
		// needed to optimise the getClosestNode function
		// (so we don't always get the terrain grid as the collided
grid!)
		terrainGrid->mTypeMask |= PhysicalZoneObjectType;

		// add grid to the scene
		terrainGrid->registerObject();

		// set the terrain grid index variable
		this->mTerrainGridIndex = this->mGrids.size() - 1;
	}

	// iterate over all grids to calculate total node count
	for (U32 i = 0; i < this->mGrids.size(); ++i)
		iAIPathMap::smNodeCount += this->mGrids[i]->mNodes.size();

	Con::iAIMessagef("Immersive AI :: Seek :: PathMap Built!");
	return true;
}

void iAIPathMap::clearMap()
{
	// iterate over nodes and delete all
	for (U32 i = 0; i < this->mGrids.size(); ++i)
	{
		if ((this->mGrids[i]) && (!this->mGrids[i]->isDeleted()))
			this->mGrids[i]->deleteObject();
		this->mGrids[i] = 0;
	}

	// set as uncompiled
	this->mCompiled = false;
	this->mTerrainGridIndex = 0;
	iAIPathMap::smNodeCount = 0;
}

void iAIPathMap::toggleDisplay()
{
	// iterate over all grids
	for (U32 i = 0; i < this->mGrids.size(); ++i)
	{
		// toggle the display
		this->mGrids[i]->toggleDisplay();
	}
```

```
}

iAIPathNode* iAIPathMap::getClosestNode(const Point3F position)
{
      PROFILE_SCOPE(iAIPathMap_getClosestNode);
      iAIPathNode* closestNode = 0;

      RayInfo rInfo;

      // set the ray well above & below terrain
      Point3F start = Point3F(position.x, position.y, position.z +
1000.0f);
      Point3F end = Point3F(position.x, position.y, position.z - 1000.0f);

      // see what grid node is in
      if (gServerContainer.castRay(position, end, iAIPathGridObjectType,
&rInfo))
      {
            // cast the found grid and find its closest node
            if (iAIPathGrid* collidedGrid =
dynamic_cast<iAIPathGrid*>(rInfo.object))
                  closestNode = collidedGrid->getClosestNode(position);
      } else
      {
            // not in a grid, must be in terrain grid!
            closestNode = this->mGrids[this->mTerrainGridIndex]-
>getClosestNode(position);
      }

      return closestNode;
}

ConsoleMethodGroupBegin(iAIPathMap, ScriptFunctions, "iAIPathMap Script
Functions");

ConsoleMethod( iAIPathMap, initialize, bool, 2, 2,
                  "bool iAIPathMap.initialize() - Initializes the PathMap
for the current mission.")
{
      return object->initialize();
}

ConsoleMethod( iAIPathMap, toggleDisplay, void, 2, 2,
                  "void iAIPathMap.toggleDisplay() - Toggles displaying
of the pathmap.")
{
      object->toggleDisplay();
}

ConsoleMethod( iAIPathMap, closestNode, const char *, 3, 3,
                  "Point3F iAIPathMap.closestNode(Point3F pos) - Get the
closest node to the supplied position.")
{
      // ensure pos passed
      if (dStrlen(argv[2]) != 0) {

            // pass the args into a Point3F
            Point3F position;
            dSscanf(argv[2], "%f %f %f", &position.x, &position.y,
&position.z);

            // get the closest node
            iAIPathNode* closestNode = object->getClosestNode(position);
```

```
            // ensure found a closest index
            if (closestNode)
            {
                  char *returnBuffer = Con::getReturnBuffer(256);
                  dSprintf(returnBuffer, 256, "%f %f %f", closestNode-
>mPosition.x, closestNode->mPosition.y, closestNode->mPosition.z);

                  return returnBuffer;
            } else
            {
                  Con::errorf("Immersive AI :: Seek :: PathMap - no node
found near %f %f %f", &position.x, &position.y, &position.z);
                  return "";
            }
      } else
      {
            Con::errorf("Immersive AI :: Seek :: PathMap - no Point3F
parsed to ClosestNode!");
            return "";
      }
}

      ConsoleMethodGroupEnd(iAIPathMap, ScriptFunctions);
```

15 APPENDIX H – IAIPATHGRID.H/.CC SOURCE CODE

15.1 iAIPathGrid.h

```
//-----------------------------------------------------------------------
// Immersive AI :: Seek :: iAIPathGrid
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
/// @file iAIPathGrid.h
//-----------------------------------------------------------------------
/// @class iAIPathGrid
/// @author Gavin Bunney
/// @version 1.0
/// @brief Represents a small collection of nodes.
///
/// Holds a collection of nodes based around a set grid position.
/// Used in combination with other grids to form the iAIPathMap.
/// <br><br>
/// TypeMask |= iAIPathGridObjectType;
//-----------------------------------------------------------------------
#ifndef _IAIPATHGRID_H_
#define _IAIPATHGRID_H_

#include "iAIPathNode.h"
#include "iAIPathGrid.h"
#include "iAIPathMap.h"

class iAIPathGrid : public SceneObject
{
      friend class iAIPathMap;

      typedef SceneObject Parent;

public:

      //-----------------------------------------------------------------------
      /// @var DECLARE_CONOBJECT(iAIPathGrid)
      /// @brief TorqueScript object.
      //-----------------------------------------------------------------------
      DECLARE_CONOBJECT(iAIPathGrid);

      //-----------------------------------------------------------------------
      /// @fn iAIPathGrid()
      /// @brief Default constructor.
      //-----------------------------------------------------------------------
      iAIPathGrid();

      //-----------------------------------------------------------------------
      /// @fn ~iAIPathGrid()
      /// @brief Default deconstructor.
      //-----------------------------------------------------------------------
      ~iAIPathGrid();

      //-----------------------------------------------------------------------
      /// @fn bool onAdd()
      /// @brief Called on adding to Sim.
      //-----------------------------------------------------------------------
      bool onAdd();

      //-----------------------------------------------------------------------
```

```
      /// @fn bool onRemove()
      /// @brief Called on removal from Sim.
      //-----------------------------------------------------------------
      void onRemove();

      //-----------------------------------------------------------------
      /// @fn bool prepRenderImage(SceneState *state, const U32 stateKey,
      ///     const U32 startZone, const bool modifyBaseZoneState = false)
      /// @brief Called on scene render. Detects if the path needs to be
      ///        rendered, calls renderObject if need to render the path.
      ///
      /// @param state the current SceneState.
      /// @param stateKey key for the state.
      /// @param startZone zone for scene start.
      /// @param modifyBaseZoneState flag to modify the base zone state.
      /// @return scene state change.
      //-----------------------------------------------------------------
      bool prepRenderImage(SceneState *state, const U32 stateKey, const U32
startZone, const bool modifyBaseZoneState = false);

      //-----------------------------------------------------------------
      /// @fn void renderObject(SceneState *state, SceneRenderImage *image)
      /// @brief Renders the path.
      ///
      /// @param state the current SceneState.
      /// @param image scene to render in.
      //-----------------------------------------------------------------
      void renderObject(SceneState *state, SceneRenderImage *image);

      //-----------------------------------------------------------------
      /// @fn void toggleDisplay()
      /// @brief Toggles displaying of the grid.
      //-----------------------------------------------------------------
      void toggleDisplay();

      //-----------------------------------------------------------------
      /// @fn bool createTerrainGrid(const Point3F worldStart,
      ///                                       const Point3F worldEnd,
      ///                            Vector<Box3F> &avoidList,
      ///                            const F32 density = 1.0f)
      /// @brief Creates a grid for the parsed area and density, avoiding
      ///        the locations within the avoidList.
      ///
      /// @param worldStart The starting point in world coords for the grid
      /// @param worldEnd The ending point in world coords for the grid
      /// @param avoidList Vector of boxes (in world points) to avoid
      /// @param density Density of node coverage. Default 1.0f.
      /// @return creation success.
      //-----------------------------------------------------------------
      bool createTerrainGrid(const Point3F worldStart, const Point3F
worldEnd, Vector<Box3F> &avoidList, const F32 density = 1.0f);

      //-----------------------------------------------------------------
      /// @fn void clearGrid()
      /// @brief Clears the grid.
      //-----------------------------------------------------------------
      void clearGrid();

      //-----------------------------------------------------------------
      /// @fn iAIPathNode* getClosestNode(const Point3F position)
      /// @brief Retrieves the closest node to the parsed position
      ///
      /// @param position world point of node to find
```

```
      /// @return pointer to closest node
      //-------------------------------------------------------------------
      iAIPathNode* getClosestNode(const Point3F position);

      //-------------------------------------------------------------------
      /// @var Box3F mGridBox
      /// @brief Box encompassing the whole grid.
      //-------------------------------------------------------------------
      Box3F mGridBox;

private:

      //-------------------------------------------------------------------
      /// @fn bool isInAvoidList(const iAIPathNode *node,
      ///          const Vector<Box3F> &avoidList)
      /// @brief Checks if the given node is within the avoid list.
      ///
      /// @param node Node to check.
      /// @param avoidList Vector of Boxes to check within.
      /// @return True if node is in the avoid list space.
      //-------------------------------------------------------------------
      bool isInAvoidList(const iAIPathNode *node, const Vector<Box3F>
&avoidList);

      //-------------------------------------------------------------------
      /// @fn void updateWorldBox()
      /// @brief Updates the grids worldbox to encompass the entire grid.
      //-------------------------------------------------------------------
      void updateWorldBox();

protected:

      //-------------------------------------------------------------------
      /// @var Vector<iAIPathNode*> mNodes
      /// @brief Vector of all nodes in the grid.
      //-------------------------------------------------------------------
      Vector<iAIPathNode*> mNodes;

      //-------------------------------------------------------------------
      /// @var F32 mDensity
      /// @brief Density of nodes per unit of worldspace.
      //-------------------------------------------------------------------
      F32 mDensity;

      //-------------------------------------------------------------------
      /// @var U16 mNodesCountX
      /// @brief Count of nodes of pathmap grid, in X direction.
      //-------------------------------------------------------------------
      U16 mNodesCountX;

      //-------------------------------------------------------------------
      /// @var U16 mNodesCountY
      /// @brief Count of nodes of pathmap grid, in Y direction.
      //-------------------------------------------------------------------
      U16 mNodesCountY;

      //-------------------------------------------------------------------
      /// @var bool mCompiled
      /// @brief Flag for when a pathmap has been compiled successfully.
      //-------------------------------------------------------------------
      bool mCompiled;

      //-------------------------------------------------------------------
```

```
      /// @var bool mShow
      /// @brief Flag to render the grid.
      //-----------------------------------------------------------------
      bool mShow;
};

      #endif
```

15.2 iAIPathGrid.cc

```
//-----------------------------------------------------------------
// Immersive AI :: Seek :: iAIPathGrid
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------
#include "terrain/terrData.h"
#include "interior/interiorInstance.h"
#include "sceneGraph/sceneGraph.h"
#include "sceneGraph/sceneState.h"
#include "platform/profiler.h"

#include "iAIPathNode.h"
#include "iAIPathGrid.h"
#include "iAIPathMap.h"
#include "iAIPathGlobal.h"

IMPLEMENT_CONOBJECT(iAIPathGrid);

iAIPathGrid::iAIPathGrid()
{
      this->setPosition(Point3F(0,0,0));
      this->mTypeMask |= iAIPathGridObjectType;

      this->mCompiled = false;
      this->mDensity = 0.0f;
      this->mGridBox = Box3F(0,0,0, 0,0,0);
      this->mShow = false;
      this->mNodesCountX = 0;
      this->mNodesCountY = 0;
}

iAIPathGrid::~iAIPathGrid()
{
      while (!this->mNodes.empty())
      {
            this->mNodes.erase((U32)0);
      }
      this->mNodes.clear();
}

bool iAIPathGrid::onAdd()
{
      // call Parent, ensure worked
      if (!Parent::onAdd())
         return false;

      // create object box
      this->updateWorldBox();

      // add to scene
    gClientContainer.addObject(this);
    gClientSceneGraph->addObjectToScene(this);

      return true;
```

```
}

void iAIPathGrid::onRemove()
{
	// remove from scene
	removeFromScene();
	Parent::onRemove();
}

bool iAIPathGrid::prepRenderImage(SceneState *state, const U32 stateKey,
const U32 startZone, const bool modifyBaseZoneState)
{
	// render if showing the node
	if (this->mShow)
	{
		// return if last state
		if (this->isLastState(state, stateKey)) return false;

		// set last state
		this->setLastState(state, stateKey);

		// see if object rendered
		if (state->isObjectRendered(this))
		{
			// get a SceneRenderImage to show on
			SceneRenderImage* image = new SceneRenderImage;
			image->obj = this;

			// insert into scene image
			state->insertRenderImage(image);
		}
	}

	return false;
}

void iAIPathGrid::renderObject(SceneState *state, SceneRenderImage *image)
{
	// save matrix to restore canonical state
	glPushMatrix();

	// going to render some lines!
	glBegin(GL_LINES);

	// render the nodes
	for (U32 i = 0; i < this->mNodes.size(); ++i)
	{
		glColor4ub(IAIPATHGLOBAL_GRID_RENDER_NODE_COLOUR);
		glVertex3fv(this->mNodes[i]->mPosition +
IAIPATHGLOBAL_GRID_RENDER_CLEARANCE);
		glVertex3fv(this->mNodes[i]->mPosition +
IAIPATHGLOBAL_GRID_RENDER_CLEARANCE +
IAIPATHGLOBAL_GRID_RENDER_NODE_HEIGHT);

		// render neighbour links
		glColor4ub(IAIPATHGLOBAL_GRID_RENDER_COLOUR);
		for (U32 j = 0; j < this->mNodes[i]->mNeighbours.size(); ++j)
		{
			if (this->mNodes[i]->mNeighbours[j])
			{
				glVertex3fv(this->mNodes[i]->mPosition +
IAIPATHGLOBAL_GRID_RENDER_CLEARANCE);
```

```
                        glVertex3fv(this->mNodes[i]->mNeighbours[j]-
>mPosition + IAIPATHGLOBAL_GRID_RENDER_CLEARANCE);
                  }
            }
      }

      // render the gridbox
      glColor4ub(IAIPATHGLOBAL_GRID_RENDER_BOX_COLOUR);
      dglWireCube(Point3F(this->mObjBox.len_x()/2, this->mObjBox.len_y()/2,
this->mObjBox.len_z()/2), this->getBoxCenter());

      // end of line drawing
      glEnd();

      // restore canonical rendering state
      glDisable(GL_BLEND);
      glDisable(GL_TEXTURE_2D);

      // restore canonical maxtrix state
      glPopMatrix();

      // ensure canonical state is restored
      AssertFatal(dglIsInCanonicalState(), "Error, GL not in canonical
state on exit");
}

void iAIPathGrid::toggleDisplay()
{
      this->mShow = !this->mShow;
}

void iAIPathGrid::clearGrid()
{
      this->mNodes.clear();
      this->mCompiled = false;
      this->mDensity = 0.0f;
      this->mGridBox = Box3F(0,0,0, 0,0,0);
      this->mNodesCountX = 0;
      this->mNodesCountY = 0;
}

bool iAIPathGrid::createTerrainGrid(const Point3F worldStart, const Point3F
worldEnd, Vector<Box3F> &avoidList, const F32 density)
{
      // grab the terrain & ensure valid
      TerrainBlock *terrain =
dynamic_cast<TerrainBlock*>(Sim::findObject("Terrain"));
      if (!terrain)
            return false;

      // clear all the nodes
      this->clearGrid();

      // set the grid start & end bounds
      this->mGridBox = Box3F(worldStart, worldEnd);

      // check the gridbox is actually valid
      if (!this->mGridBox.isValidBox())
      {
            Con::errorf("Immersive AI :: Seek :: Grid build failed - grid
box not valid!");
            return false;
      }
```

```
	// the density is nodes per gridsize; default gridSize to 10.0f if
none found
	this->mDensity = density / Con::getFloatVariable("Server::gridSize",
10.0f);

	// density step needs to be the squareroot, as operates in both X & Y
	F32 densityStep = 1 / mSqrt(this->mDensity);

	// calculate the count of nodes in x & y
	this->mNodesCountX = this->mGridBox.len_x() * mSqrt(this->mDensity);
	this->mNodesCountY = this->mGridBox.len_y() * mSqrt(this->mDensity);

	// create all nodes
	for (U16 iterX = 0; iterX < this->mNodesCountX; ++iterX)
	{
		for (U16 iterY = 0; iterY < this->mNodesCountY; ++iterY)
		{
			Point3F nodePos = this->mGridBox.min;
			nodePos.x += densityStep * iterX;
			nodePos.y += densityStep * iterY;

			// transform point to terrain transform
			terrain->getWorldTransform().mulP(nodePos);
			nodePos.convolveInverse(terrain->getScale());
			F32 height;
			if (terrain->getHeight(Point2F(nodePos.x, nodePos.y),
&height))
			{
				nodePos.z = height;
				nodePos.convolve(terrain->getScale());
				terrain->getTransform().mulP(nodePos);
			}

			// create the node
			iAIPathNode *newNode = new iAIPathNode(nodePos, this,
iterX, iterY);
			this->mNodes.push_back(newNode);
		}
	}

	// join all the node neighbours
	for (U32 iter = 0; iter < this->mNodes.size(); ++iter)
	{
		U16 currentX = this->mNodes[iter]->mIdX;
		U16 currentY = this->mNodes[iter]->mIdY;

		// north
		if (currentY < (this->mNodesCountY - 1))
			this->mNodes[iter]->addNeighbour(this->mNodes[iter+1]);

		// south
		if (currentY > 0)
			this->mNodes[iter]->addNeighbour(this->mNodes[iter-1]);

		// east
		if (currentX < (this->mNodesCountX-1))
			this->mNodes[iter]->addNeighbour(this->mNodes[iter +
this->mNodesCountY]);

		// west
		if (currentX > 0)
```

```
                  this->mNodes[iter]->addNeighbour(this->mNodes[iter -
this->mNodesCountY]);

            // north-east
            if (currentX < (this->mNodesCountX - 1) && (currentY < (this-
>mNodesCountY - 1)))
                  this->mNodes[iter]->addNeighbour(this->mNodes[ iter +
this->mNodesCountY + 1]);

            // south-east
            if (currentX < (this->mNodesCountX - 1) && (currentY > 0))
                  this->mNodes[iter]->addNeighbour(this->mNodes[ iter+this-
>mNodesCountY-1]);

            // south-west
            if ((currentX > 0) && (currentY > 0))
                  this->mNodes[iter]->addNeighbour(this->mNodes[ iter -
this->mNodesCountY - 1 ]);

            // north-west
            if ((currentX > 0) && (currentY < (this->mNodesCountY - 1)))
                  this->mNodes[iter]->addNeighbour(this->mNodes[ iter -
this->mNodesCountY + 1 ]);
      }

      // cull invalid and alone nodes
      for (U32 gridIter = 0; gridIter < this->mNodes.size(); ++gridIter)
      {
            // remove the node if in an invalid position or has no
neighbours or in avoid list
            if ((!this->mNodes[gridIter]->isClear()) || (this-
>mNodes[gridIter]->mNeighbours.size() == 0) ||
                  this->isInAvoidList(this->mNodes[gridIter], avoidList))
            {
                  // remove it from its neighbours first
                  for (U32 j = 0; j < this->mNodes[gridIter]-
>mNeighbours.size(); ++j)
                  {
                        this->mNodes[gridIter]->mNeighbours[j]-
>removeNeighbour(this->mNodes[gridIter]);
                  }

                  // remove from grid list
                  this->mNodes.erase(gridIter);
            }
      }

      // set as compiled if any nodes in the grid
      this->mCompiled = (this->mNodes.size() > 0);

      // update the world box so renders properly
      this->updateWorldBox();

      return this->mCompiled;
}

bool iAIPathGrid::isInAvoidList(const iAIPathNode *node, const
Vector<Box3F> &avoidList)
{
      // iterate over all boxes in the avoid list
      for (U32 z = 0; z < avoidList.size(); ++z)
      {
```

```
            Box3F nodeBox = Box3F(node->mPosition -
(IAIPATHGLOBAL_NODE_CLEARANCE/2), node->mPosition +
(IAIPATHGLOBAL_NODE_CLEARANCE/2));

            // see if the avoid list box overlaps the nodes box
            if (avoidList[z].isOverlapped(nodeBox))
                  return true;
      }

      // didn't overlap on any boxes
      return false;
}

void iAIPathGrid::updateWorldBox()
{
      if (this->mCompiled)
      {
            Point3F min = Point3F(this->mNodes.front()->mPosition);
            Point3F max = Point3F(this->mNodes.front()->mPosition);

            // iterate over all nodes and find the min & max
            for (U32 i = 0; i < this->mNodes.size(); ++i)
            {
                  if (this->mNodes[i]->mPosition.x < min.x)
                        min.x = this->mNodes[i]->mPosition.x;
                  if (this->mNodes[i]->mPosition.y < min.y)
                        min.y = this->mNodes[i]->mPosition.y;
                  if (this->mNodes[i]->mPosition.z < min.z)
                        min.z = this->mNodes[i]->mPosition.z;


                  if (this->mNodes[i]->mPosition.x > max.x)
                        max.x = this->mNodes[i]->mPosition.x;
                  if (this->mNodes[i]->mPosition.y > max.y)
                        max.y = this->mNodes[i]->mPosition.y;
                  if (this->mNodes[i]->mPosition.z > max.z)
                        max.z = this->mNodes[i]->mPosition.z;
            }

            // set position as halfway point
            this->setPosition(min + ((max - min) / 2));

            // create a box to encompass the entire path
            this->mObjBox.min.set(-(this->getPosition() - min));
            this->mObjBox.max.set(max - this->getPosition());

            // must reset world box & transform when changing object box
            this->resetWorldBox();
            this->setRenderTransform(mObjToWorld);
      }
}

iAIPathNode* iAIPathGrid::getClosestNode(const Point3F position)
{
      PROFILE_SCOPE(iAIPathGrid_getClosestNode);

      F32 closestVec = 1000.0f;
      F32 currentVec = 1000.0f;
      iAIPathNode* closestNode = 0;

      // iterate over all nodes
      for (U32 i = 0; i < this->mNodes.size(); ++i)
      {
```

```
            // calculate the vector length
            currentVec = Point3F(position - this->mNodes[i]-
>mPosition).len();

            // check if its closest one found so far
            if (currentVec < closestVec)
            {
                  // save as closest so far
                  closestVec = currentVec;
                  closestNode = this->mNodes[i];
            }
      }

      return closestNode;
      }
```

16 APPENDIX I – IAIPATHNODE.H/.CC SOURCE CODE

16.1 iAIPathNode.h

```
//---------------------------------------------------------------------
// Immersive AI :: Seek :: iAIPathMap
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//---------------------------------------------------------------------

//---------------------------------------------------------------------
/// @file iAIPathNode.h
//---------------------------------------------------------------------
/// @class iAIPathNode
/// @author Gavin Bunney
/// @version 1.0
/// @brief Represents a single path finding node.
///
/// A point within the game world, consisting of a location, id,
/// neighbours and various path finding variables.
//---------------------------------------------------------------------
#ifndef _IAIPATHNODE_H_
#define _IAIPATHNODE_H_

class iAIPathNode {

      friend class iAIPathMap;
      friend class iAIPathGrid;
      friend class iAIPath;
      friend class iAIPathFind;

public:

      //-----------------------------------------------------------------
      /// @fn iAIPathNode()
      /// @brief Default constructor.
      //-----------------------------------------------------------------
      iAIPathNode() { }

      //-----------------------------------------------------------------
      /// @fn ~iAIPathNode()
      /// @brief Default deconstructor.
      //-----------------------------------------------------------------
      ~iAIPathNode();

      //-----------------------------------------------------------------
      /// @fn iAIPathNode(const Point3F position, iAIPathGrid* pathGrid,
      ///                 const U16 idX, const U16 idY)
      /// @brief Constructor to create node at specified position.
      ///
      /// @param position Point in world coords to create the node.
      /// @param pathGrid Pointer to grid which the node is contained in.
      /// @param idX ID in X of the node within the grid.
      /// @param idY ID in Y of the node within the grid.
      //-----------------------------------------------------------------
      iAIPathNode(const Point3F position, iAIPathGrid* pathGrid, const U16
idX, const U16 idY);

      //-----------------------------------------------------------------
      /// @fn bool isClear()
      /// @brief Checks if the node is in a valid position, clear of
      ///          obstructions.
      ///
```

```
        /// @return Node clear of all obstructions.
        //-------------------------------------------------------------------
        bool isClear();

        //-------------------------------------------------------------------
        /// @fn bool addNeighbour(iAIPathNode* neighbour)
        /// @brief Adds a neighbour to this node's neighbour list.
        ///
        /// @return Neighbour added to node neighbour list.
        //-------------------------------------------------------------------
        bool addNeighbour(iAIPathNode* neighbour);

        //-------------------------------------------------------------------
        /// @fn bool hasNeighbour(iAIPathNode* neighbour)
        /// @brief Checks if the node has the parsed neighbour.
        ///
        /// @return Neighbour in this node's neighbour list.
        //-------------------------------------------------------------------
        bool hasNeighbour(iAIPathNode* neighbour);

        //-------------------------------------------------------------------
        /// @fn bool removeNeighbour(iAIPathNode* neighbour)
        /// @brief Removes the specified neighbour.
        ///
        /// @return Neighbour was removed successfully.
        //-------------------------------------------------------------------
        bool removeNeighbour(iAIPathNode* neighbour);

        //-------------------------------------------------------------------
        /// @fn void bool isNeighbourValid(const Point3F neighbourPosition)
        /// @brief Checks if a neighbour is accessible from this node.
        ///
        /// @param neighbourPosition The position of the neighbour to check.
        /// @return Neighbour in valid position
        //-------------------------------------------------------------------
        bool isNeighbourValid(const Point3F neighbourPosition);

        //-------------------------------------------------------------------
        /// @var Point3F mPosition
        /// @brief Position of the node in world coordinates.
        //-------------------------------------------------------------------
        Point3F mPosition;

        //-------------------------------------------------------------------
        /// @var U16 mIdX
        /// @brief ID in X within the node's grid.
        //-------------------------------------------------------------------
        U16 mIdX;

        //-------------------------------------------------------------------
        /// @var U16 mIdY
        /// @brief ID in Y within the node's grid.
        //-------------------------------------------------------------------
        U16 mIdY;

        //-------------------------------------------------------------------
        /// @var Vector<iAIPathNode*> mNeighbours
        /// @brief Vector of the nodes neighbours. Expect 8 neighbours, but
        ///        if edge of area or obstructed, will be less.
        //-------------------------------------------------------------------
        Vector<iAIPathNode*> mNeighbours;

        //-------------------------------------------------------------------
```

```
        /// @var iAIPathGrid* mParentGrid
        /// @brief Pointer to the grid which the node is contained within.
        //---------------------------------------------------------------------
        iAIPathGrid* mParentGrid;

        //---------------------------------------------------------------------
        /// @fn void updateMoveModifier()
        /// @brief Updates the node's move modifer, based on its position.
        //---------------------------------------------------------------------
        void updateMoveModifier();

        //---------------------------------------------------------------------
        /// @var F32 mMoveModifier
        /// @brief Utilised in A* algorithm; the level of difficulty at this
        //         node. 100.0f is untraversal, 0.0f is easiest.
        //---------------------------------------------------------------------
        F32 mMoveModifier;

        //---------------------------------------------------------------------
        /// @var F32 mFitness
        /// @brief Utilised in A* algorithm; fitness of this node from a
        ///        start node.
        //---------------------------------------------------------------------
        F32 mFitness;

        //---------------------------------------------------------------------
        /// @var F32 mLowestCostFromStart
        /// @brief Utilised in A* algorithm; lowest cost from the start node.
        //---------------------------------------------------------------------
        F32 mLowestCostFromStart;

        //---------------------------------------------------------------------
        /// @var F32 mHeuristicCostToGoal
        /// @brief Utilised in A* algorithm; best guess cost to goal from
this
        ///        node.
        //---------------------------------------------------------------------
        F32 mHeuristicCostToGoal;

        //---------------------------------------------------------------------
        /// @var iAIPathNode* mParent
        /// @brief Utilised in A* algorithm; the node traversed previously to
        ///        reach this node.
        //---------------------------------------------------------------------
        iAIPathNode* mParent;

        //---------------------------------------------------------------------
        /// @var bool mOpen
        /// @brief Utilised in A* algorithm; the node requires checking.
        //---------------------------------------------------------------------
        bool mOpen;

        //---------------------------------------------------------------------
        /// @var bool mClosed
        /// @brief Utilised in A* algorithm; if node has been checked, set as
        ///        closed.
        //---------------------------------------------------------------------
        bool mClosed;
};

        #endif
```

16.2 iAIPathNode.cc

```
//---------------------------------------------------------------------
// Immersive AI :: Seek :: iAIPathNode
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//---------------------------------------------------------------------

#include "sceneGraph/sceneGraph.h"
#include "sceneGraph/sceneState.h"

#include "iAIPathGlobal.h"
#include "iAIPathNode.h"
#include "iAIPathMap.h"

iAIPathNode::iAIPathNode(const Point3F position, iAIPathGrid* pathGrid,
const U16 idX, const U16 idY)
{
	this->mPosition = position;
	this->mIdX = idX;
	this->mIdY = idY;
	this->mParentGrid = pathGrid;

	this->mMoveModifier = 0.0f;
	this->mFitness = 0.0f;
	this->mLowestCostFromStart = 0.0f;
	this->mHeuristicCostToGoal = 0.0f;
	this->mParent = 0;
	this->mOpen = false;
	this->mClosed = false;
}

iAIPathNode::~iAIPathNode()
{
	while (!this->mNeighbours.empty())
	{
		this->mNeighbours.erase((U32)0);
	}
	this->mNeighbours.clear();
}

bool iAIPathNode::isClear()
{
	Point3F start = this->mPosition;
	Point3F end = this->mPosition + Point3F(0, 0,
IAIPATHGLOBAL_NODE_CLEARANCE.z);

	RayInfo dummy;

	// if collided with something, isn't clear
	if (gServerContainer.castRay(start, end,
IAIPATHGLOBAL_COLLISION_MASK, &dummy))
	{
		return false;
	}

	// must be valid
	return true;
}

bool iAIPathNode::isNeighbourValid(const Point3F neighbourPosition)
{
	// calculate vector in z
	Point3F vec = this->mPosition - neighbourPosition;
```

```
	F32 zSq = vec.z * vec.z;

	// ensure difference in node height is valid
	if (zSq > IAIPATHGLOBAL_MAX_SLOPE)
		return false;

	RayInfo dummy;

	// quick check from node to neighboour position
	if (gServerContainer.castRay(this->mPosition, neighbourPosition,
IAIPATHGLOBAL_COLLISION_MASK, &dummy))
		return false;

	// check 4 points around clearance box
	Point3F offset = -Point3F(IAIPATHGLOBAL_NODE_CLEARANCE.x/2, 0, 0);
	if (gServerContainer.castRay(this->mPosition + offset,
neighbourPosition + offset, IAIPATHGLOBAL_COLLISION_MASK, &dummy))
		return false;
	offset = Point3F(IAIPATHGLOBAL_NODE_CLEARANCE.x/2, 0, 0);
	if (gServerContainer.castRay(this->mPosition + offset,
neighbourPosition + offset, IAIPATHGLOBAL_COLLISION_MASK, &dummy))
		return false;
	offset = Point3F(0, 0, IAIPATHGLOBAL_NODE_CLEARANCE.z);
	if (gServerContainer.castRay(this->mPosition + offset,
neighbourPosition + offset, IAIPATHGLOBAL_COLLISION_MASK, &dummy))
		return false;
	offset = Point3F(IAIPATHGLOBAL_NODE_CLEARANCE.x/2, 0,
IAIPATHGLOBAL_NODE_CLEARANCE.z);
	if (gServerContainer.castRay(this->mPosition + offset,
neighbourPosition + offset, IAIPATHGLOBAL_COLLISION_MASK, &dummy))
		return false;

	// must be valid
	return true;
}

bool iAIPathNode::addNeighbour(iAIPathNode* neighbour)
{
	if (neighbour)
	{
		// check that neighbour is valid before adding
		if (isNeighbourValid(neighbour->mPosition))
		{
			this->mNeighbours.push_back(neighbour);
			return true;
		}
	}
	return false;
}

bool iAIPathNode::hasNeighbour(iAIPathNode* neighbour)
{
	if (neighbour)
	{
		// iterate over all neighbours
		for (U32 i = 0; i < this->mNeighbours.size(); i++)
		{
			// if position match, then it has that neighbour!
			if (this->mNeighbours[i]->mPosition == neighbour-
>mPosition)
				return true;
		}
	}
```

```
	return false;
}

bool iAIPathNode::removeNeighbour(iAIPathNode* neighbour)
{
	if (neighbour)
	{
		// iterate over all neighbours
		for (U32 i = 0; i < this->mNeighbours.size(); i++)
		{
			// if position match, then it has that neighbour!
			if (this->mNeighbours[i]->mPosition == neighbour-
>mPosition)
			{
				// remove the neighbour
				this->mNeighbours.erase(i);
				return true;
			}
		}
	}
	return false;
}

void iAIPathNode::updateMoveModifier()
{
	RayInfo dummy;
	// check if the node is in water
	if (gServerContainer.castRay(this->mPosition + Point3F(0,0,1000.0f),
this->mPosition - Point3F(0, 0, IAIPATHGLOBAL_NODE_CLEARANCE.z / 2),
WaterObjectType, &dummy))
		this->mMoveModifier = IAIPATHGLOBAL_MOVE_MODIFIER_WATER;
	else
		this->mMoveModifier = 0.0f;
	}
```

17 APPENDIX J – IAIGOALLIBRARY.H/.CC SOURCE CODE

17.1 iAIGoalLibrary.h

```
//-----------------------------------------------------------------------
// Immersive AI :: iAIGoalLibrary
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
/// @file iAIGoalLibrary.h
//-----------------------------------------------------------------------
/// @class iAIGoalLibrary
/// @author Gavin Bunney
/// @version 1.0
/// @brief Collection of goals.
///
/// Holds a library of all goals for the agents in the game world.
//-----------------------------------------------------------------------

#ifndef _IAIGOALLIBRARY_H_
#define _IAIGOALLIBRARY_H_

#include "immersiveAI/agent/iAIAgent.h"
#include "console/simBase.h"

//-----------------------------------------------------------------------
/// @struct iAIGoalSolution
/// @brief Represents a solution for a goal.
///
/// Holds the solution name and a condition function which is
/// executed, in script, to determine the weighting for the solution
/// at a particular point in time. The condition function must return
/// 0.00 <= return <= 1.00.
//-----------------------------------------------------------------------
struct iAIGoalSolution
{
      //-----------------------------------------------------------------
      /// @var char mName[50]
      /// @brief Name of the solution.
      //-----------------------------------------------------------------
      char mName[50];
};

//-----------------------------------------------------------------------
/// @struct iAIGoal
/// @brief Represents a goal for an agent type.
///
/// Holds the goal name, agent type the goal is for, and the name of
/// the condition function which is executed, in script, to determine
/// the weighting for the goal at a particular point in time. The
/// condition function must return 0.00 <= return <= 1.00.
//-----------------------------------------------------------------------
struct iAIGoal
{
      //-----------------------------------------------------------------
      /// @var char mName[50]
      /// @brief Name of the goal.
      //-----------------------------------------------------------------
      char mName[50];

      //-----------------------------------------------------------------
```

```
      /// @var char mAgentType[50]
      /// @brief Name of the agent type that goal is for.
      //-----------------------------------------------------------------------
      char mAgentType[50];

      //-----------------------------------------------------------------------
      /// @var Vector<iAIGoalSolution*> mSolutions
      /// @brief Collection of solution goals to solve this goal.
      //-----------------------------------------------------------------------
      Vector<iAIGoalSolution*> mSolutions;
};

class iAIGoalLibrary : public SimObject
{
      typedef SimObject Parent;
      friend class iAIGoalManager;

public:

      //-----------------------------------------------------------------------
      /// @var DECLARE_CONOBJECT(iAIGoalLibrary)
      /// @brief TorqueScript object.
      //-----------------------------------------------------------------------
      DECLARE_CONOBJECT(iAIGoalLibrary);

      //-----------------------------------------------------------------------
      /// @fn iAIGoalLibrary()
      /// @brief Default constructor.
      //-----------------------------------------------------------------------
      iAIGoalLibrary();

      //-----------------------------------------------------------------------
      /// @fn ~iAIGoalManager()
      /// @brief Deconstructor.
      //-----------------------------------------------------------------------
      ~iAIGoalLibrary();

      //-----------------------------------------------------------------------
      /// @fn void addGoal(const char* agentType, const char* goalName)
      /// @brief Add a goal to the library.
      ///
      /// @param agentType Type of agent the goal is for.
      /// @param goalName Name of the goal to add.
      /// @returns bool Add success.
      //-----------------------------------------------------------------------
      bool addGoal(const char* agentType, const char* goalName);

      //-----------------------------------------------------------------------
      /// @fn void addSolution(const char* agentType, const char* goalName,
      ///                      const char* solutionName)
      /// @brief Add a solution for a goal to the library.
      ///
      /// @param agentType Type of agent the goal is for.
      /// @param goalName Name of the goal to add the solution against.
      /// @param solutionName Name of the solution goal.
      /// @returns bool Add success.
      //-----------------------------------------------------------------------
      bool addSolution(const char* agentType, const char* goalName, const
char* solutionName);

      //-----------------------------------------------------------------------
      /// @fn bool removeGoal(const char* goalName, const char* agentType)
      /// @brief Remove a goal from the library.
```

```
///
/// @param goalName Name of the goal to add.
/// @param agentType Type of agent the goal is for.
/// @returns Remove success.
//-----------------------------------------------------------------------
bool removeGoal(const char* goalName, const char* agentType);

//-----------------------------------------------------------------------
/// @fn bool removeGoal(const char* goalName)
/// @brief Remove all goals that match the passed goal name.
///
/// @param goalName Name of all the goals to remove.
/// @returns Remove success.
//-----------------------------------------------------------------------
bool removeGoal(const char* goalName);

//-----------------------------------------------------------------------
/// @fn void clear()
/// @brief Empty the goal library.
//-----------------------------------------------------------------------
void clear();

//-----------------------------------------------------------------------
/// @fn void sort()
/// @brief Sorts the library.
//-----------------------------------------------------------------------
void sort();

//-----------------------------------------------------------------------
/// @fn void printGoals()
/// @brief Output all goals in the library to the console.
//-----------------------------------------------------------------------
void printGoals();

//-----------------------------------------------------------------------
/// @fn void printGoals(const char* agentType)
/// @brief Output all goals in the library for the specified agent
///        type to the console.
///
/// @param agentType Type of agent to output the goals for.
//-----------------------------------------------------------------------
void printGoals(const char* agentType);

//-----------------------------------------------------------------------
/// @fn bool getGoals(const char* agentType,
///                   Vector<iAIGoal*> &returnList)
/// @brief Retrieve all goals for the specified agent type.
///
/// @param agentType Type of agent to retrieve goals for.
/// @param returnList Vector where the retrieved goals are placed.
/// @returns bool returnList is not empty.
//-----------------------------------------------------------------------
bool getGoals(const char* agentType, Vector<iAIGoal*> &returnList);

//-----------------------------------------------------------------------
/// @fn iAIGoal* iAIGoalLibrary::getGoal(const char* agentType,
///                                      const char* goalName)
/// @brief Retrieve the specified goal.
///
/// @param agentType Type of agent to retrieve goal for.
/// @param goalName Name of the goal to retrieve.
/// @returns iAIGoal* Pointer to the found goal.
//-----------------------------------------------------------------------
```

```
      iAIGoal* getGoal(const char* agentType, const char* goalName);

protected:

      //-----------------------------------------------------------------------
      /// @var Vector<iAIGoal*> mGoalLibrary
      /// @brief Collection of goals in the library.
      //-----------------------------------------------------------------------
      Vector<iAIGoal*> mGoalLibrary;

      //-----------------------------------------------------------------------
      /// @var bool mSorted
      /// @brief Flag set when library is sorted.
      //-----------------------------------------------------------------------
      bool mSorted;
};

#endif
```

17.2 iAIGoalLibrary.cc

```
//-----------------------------------------------------------------------
// Immersive AI :: iAIGoalLibrary
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
/// @file iAIGoalLibrary.cc
//-----------------------------------------------------------------------
/// @class iAIGoalLibrary
/// @author Gavin Bunney
/// @version 1.0
/// @brief Collection of goals.
///
/// Holds a library of all goals for the agents in the game world.
//-----------------------------------------------------------------------

#include "iAIGoalLibrary.h"
#include "console/consoleTypes.h"
#include "platform/platform.h"

IMPLEMENT_CONOBJECT(iAIGoalLibrary);

iAIGoalLibrary::iAIGoalLibrary()
{
      this->mSorted = false;
}

iAIGoalLibrary::~iAIGoalLibrary()
{
      this->clear();
}

bool iAIGoalLibrary::addGoal(const char* agentType, const char* goalName)
{
      iAIGoal *newGoal = new iAIGoal();

      // check string sizes valid
      if ((dStrlen(agentType) <= sizeof(newGoal->mAgentType)) &&
            (dStrlen(goalName) <= sizeof(newGoal->mName)))
```

```
	{
		// create the new goal
		dSprintf(newGoal->mAgentType, sizeof(newGoal->mAgentType),
"%s", agentType);
		dSprintf(newGoal->mName, sizeof(newGoal->mName), "%s",
goalName);

		// add to library
		this->mGoalLibrary.push_back(newGoal);

		return true;
	} else
	{
		Con::errorf("Immersive AI :: Goal Library :: Add goal failed -
Invalid parameter lengths.");
		return false;
	}
}

bool iAIGoalLibrary::addSolution(const char* agentType, const char*
goalName, const char* solutionName)
{
	iAIGoalSolution *newSolution = new iAIGoalSolution();

	// check string sizes valid
	if (dStrlen(solutionName) <= sizeof(newSolution->mName))
	{
		// create the new solution
		dSprintf(newSolution->mName, sizeof(newSolution->mName), "%s",
solutionName);

		// find the goal in the library
		for (U32 i = 0; i < this->mGoalLibrary.size(); ++i)
		{
			// compare the name and agent type names
			if ((dStrcmp(this->mGoalLibrary[i]->mAgentType,
agentType) == 0) &&
				(dStrcmp(this->mGoalLibrary[i]->mName, goalName) ==
0))
			{
				// found, add to library
				this->mGoalLibrary[i]-
>mSolutions.push_back(newSolution);
				return true;
			}
		}

		// reached here, means goal/agent type didnt exist!
		Con::errorf("Immersive AI :: Goal Library :: Add solution
failed - goal/agent combination doesn't exist.");
		return false;
	} else
	{
		Con::errorf("Immersive AI :: Goal Library :: Add solution
failed - Invalid parameter lengths.");
		return false;
	}
}

bool iAIGoalLibrary::removeGoal(const char* goalName, const char*
agentType)
{
	bool removedGoal = false;
```

```
	// iterate over all goals
	for (U32 i = 0; i < this->mGoalLibrary.size(); ++i)
	{
		// compare the name and agent type names
		if ((dStrcmp(this->mGoalLibrary[i]->mName, goalName) == 0) &&
(dStrcmp(this->mGoalLibrary[i]->mAgentType, agentType) == 0))
		{
			// clear any solutions
			this->mGoalLibrary[i]->mSolutions.clear();

			// erase the actual goal
			this->mGoalLibrary.erase(i);
			--i;
			removedGoal = true;
		}
	}

	return removedGoal;
}

bool iAIGoalLibrary::removeGoal(const char* goalName)
{
	bool removedGoal = false;

	// iterate over all goals
	for (U32 i = 0; i < this->mGoalLibrary.size(); ++i)
	{
		// compare the name and agent type names
		if (dStrcmp(this->mGoalLibrary[i]->mName, goalName) == 0)
		{
			// clear any solutions
			this->mGoalLibrary[i]->mSolutions.clear();

			// erase the actual goal
			this->mGoalLibrary.erase(i);
			--i;
			removedGoal = true;
		}
	}

	return removedGoal;
}

void iAIGoalLibrary::clear()
{
	// keep looping until the goal library is empty
	while (!this->mGoalLibrary.empty())
	{
		// loop while the current solutiosn isn't empty
		while (!this->mGoalLibrary.front()->mSolutions.empty())
		{
			// delete the front solution
			this->mGoalLibrary.front()->mSolutions.erase((U32)0);
		}

		// delete the front goal
		this->mGoalLibrary.erase((U32)0);
	}
}

static S32 QSORT_CALLBACK goalCompare( const void* a, const void* b )
{
```

```
	const iAIGoal *goalA = *((iAIGoal **) a);
	const iAIGoal *goalB = *((iAIGoal **) b);

	// sort by agent type first
	S8 result = dStrcmp(goalA->mAgentType, goalB->mAgentType);
	if (result == 0)
	{
		// then by goal name
		return dStrcmp(goalA->mName, goalB->mName);
	} else
	{
		return result;
	}
	return 0;
}

static S32 QSORT_CALLBACK goalSolutionCompare( const void* a, const void* b
)
{
	// sort by solution name
	return dStrcmp((*((iAIGoalSolution **) a))->mName,
(*((iAIGoalSolution **) b))->mName);
}


void iAIGoalLibrary::sort()
{
	// sort the solutions within the goals first
	for (U32 i = 0; i < this->mGoalLibrary.size(); ++i)
	{
		dQsort((void *)&(this->mGoalLibrary[i]->mSolutions[0]), this-
>mGoalLibrary[i]->mSolutions.size(), sizeof(iAIGoalSolution*),
goalSolutionCompare);
	}

	// sort the goals
	dQsort((void *)&(this->mGoalLibrary.front()), this-
>mGoalLibrary.size(), sizeof(iAIGoal*), goalCompare);

	this->mSorted = true;
}

void iAIGoalLibrary::printGoals()
{
	Con::iAIMessagef("iAIGoalLibrary -- Goals list:");
	for (U32 i = 0; i < this->mGoalLibrary.size(); ++i)
	{
		Con::iAIMessagef("[%d - %s] %s", i, this->mGoalLibrary[i]-
>mAgentType, this->mGoalLibrary[i]->mName);
		for (U32 j = 0; j < this->mGoalLibrary[i]->mSolutions.size();
++j)
		{
			Con::iAIMessagef("--- [%d] %s", j, this->mGoalLibrary[i]-
>mSolutions[j]->mName);
		}
	}
}

void iAIGoalLibrary::printGoals(const char* agentType)
{
	Con::iAIMessagef("iAIGoalLibrary :: Goal List for '%s'", agentType);
	for (U32 i = 0; i < this->mGoalLibrary.size(); ++i)
	{
```

```
            // only output the desired agent type goals
            if (dStrcmp(this->mGoalLibrary[i]->mAgentType, agentType) == 0)
            {
                  Con::iAIMessagef("[%d] %s", i, this->mGoalLibrary[i]-
>mName);
                  for (U32 j = 0; j < this->mGoalLibrary[i]-
>mSolutions.size(); ++j)
                  {
                        Con::iAIMessagef("--- [%d] %s", j, this-
>mGoalLibrary[i]->mSolutions[j]->mName);
                  }
            }
      }
}

bool iAIGoalLibrary::getGoals(const char* agentType, Vector<iAIGoal*>
&returnList)
{
      // check if not sorted and sort if need be
      if (!this->mSorted)
            this->sort();

      returnList.clear();

      // iterate over all goals in the library
      for (U32 i = 0; i < this->mGoalLibrary.size(); ++i)
      {
            // check for agent type
            if (dStrcmp(this->mGoalLibrary[i]->mAgentType, agentType) == 0)
            {
                  // add to return list
                  returnList.push_back(this->mGoalLibrary[i]);
            } else
            {
                  // if the return list is not empty, then must have gone
beyond agent type
                  if (!returnList.empty())
                  {
                        // break out
                        break;
                  }
            }
      }

      return (!returnList.empty());
}

iAIGoal* iAIGoalLibrary::getGoal(const char* agentType, const char*
goalName)
{
      if ((dStrlen(agentType) == 0) || (dStrlen(goalName) == 0))
            return 0;

      // iterate over all goals in the library
      for (U32 i = 0; i < this->mGoalLibrary.size(); ++i)
      {
            // check for agent type and goalName
            if ((dStrcmp(this->mGoalLibrary[i]->mAgentType, agentType) ==
0) &&
                  (dStrcmp(this->mGoalLibrary[i]->mName, goalName) == 0))
            {
                  return this->mGoalLibrary[i];
            }
```

```
	}

	// didn't find the goal
	return 0;
}

ConsoleMethodGroupBegin(iAIGoalLibrary, ScriptFunctions, "iAIGoalLibrary
Script Functions");

ConsoleMethod( iAIGoalLibrary, addGoal, bool, 4, 4,
			"bool iAIGoalLibrary.addGoal(string agentType, string
goalName) - Add a goal to the library.")
{
	// ensure all params passed
	if ((dStrlen(argv[2]) != 0) && (dStrlen(argv[3]) != 0))
	{
		object->addGoal(argv[2], argv[3]);
		return true;
	} else
	{
		Con::errorf("Immersive AI :: Goal Library :: Not all goal
params passed to add to the library!");
		return false;
	}
}

ConsoleMethod( iAIGoalLibrary, addSolution, bool, 5, 5,
			"bool iAIGoalLibrary.addSolution(string agentType,
string goalName, string solutionName) - Add a solution for a goal to the
library.")
{
	// ensure all params passed
	if ((dStrlen(argv[2]) != 0) && (dStrlen(argv[3]) != 0) &&
(dStrlen(argv[4]) != 0))
	{
		object->addSolution(argv[2], argv[3], argv[4]);
		return true;
	} else
	{
		Con::errorf("Immersive AI :: Goal Library :: Not all solution
params passed to add to the library!");
		return false;
	}
}

ConsoleMethod( iAIGoalLibrary, printGoals, void, 2, 3,
			"void iAIGoalLibrary.printGoals([optional] string
agentType) - Output all the goals to the console.")
{
	// see if want just a single agent type
	if (dStrlen(argv[2]) != 0)
	{
		object->printGoals(argv[2]);
	} else
	{
		object->printGoals();
	}
}

ConsoleMethod( iAIGoalLibrary, clear, void, 2, 2,
			"void iAIGoalLibrary.clear() - Clear all goals in the
library.")
{
```

```
	object->clear();
}

ConsoleMethod( iAIGoalLibrary, removeGoal, bool, 3, 4,
                    "bool iAIGoalLibrary.removeGoal(string goalName,
[optional] string agentType) - Remove a goal from the library.")
{
	// ensure at least 1 param passed
	if (dStrlen(argv[2]) != 0)
	{
		if (dStrlen(argv[3]) != 0)
			return (object->removeGoal(argv[2], argv[3]));
		else
			return (object->removeGoal(argv[2]));
	} else
	{
		Con::errorf("Immersive AI :: Goal Library :: Remove goal not
given enough parameters.");
		return false;
	}
}

ConsoleMethod( iAIGoalLibrary, sort, void, 2, 2,
                    "void iAIGoalLibrary.sort() - Sort the goal library.")
{
	object->sort();
}

ConsoleMethod( iAIGoalLibrary, getGoalList, const char*, 3, 3,
                    "string iAIGoalLibrary.getGoalList(string agentType) -
Get a space deliminated string of the goals for an agent type.")
{
	char *returnBuffer = Con::getReturnBuffer(512);
	dSprintf(returnBuffer, sizeof(returnBuffer), "");

	// ensure agent type parsed
	if (dStrlen(argv[2]) != 0)
	{
		// get the goals for the parsed agent type
		Vector<iAIGoal*> goalList;
		if (object->getGoals(argv[2], goalList))
		{
			// iterate over all the goals and add to return string
			for (U32 i = 0; i < goalList.size(); ++i)
			{
				if (i != 0)
					dStrcat(returnBuffer, " ");
				dStrcat(returnBuffer, goalList[i]->mName);
			}
		}
	}

	return returnBuffer;
}

ConsoleMethod( iAIGoalLibrary, getSolutionList, const char*, 4, 4,
                    "string iAIGoalLibrary.getSolutionList(string
agentType, string goalName) - Get a space deliminated string of the
solutions.")
{
	char *returnBuffer = Con::getReturnBuffer(512);
	dSprintf(returnBuffer, sizeof(returnBuffer), "");
```

```
      // ensure goal & agent type parsed
      if ((dStrlen(argv[2]) != 0) && (dStrlen(argv[3]) != 0))
      {
            // get a pointer to the goal
            iAIGoal* goal = object->getGoal(argv[2], argv[3]);

            // check valid goal found
            if (goal)
            {
                  // iterate over all the goal solutions and add to return
string
                  for (U32 i = 0; i < goal->mSolutions.size(); ++i)
                  {
                        if (i != 0)
                              dStrcat(returnBuffer, " ");
                        dStrcat(returnBuffer, goal->mSolutions[i]->mName);
                  }
            }
      }

      return returnBuffer;
}

      ConsoleMethodGroupEnd(iAIGoalLibrary, ScriptFunctions);
```

18 APPENDIX K – TORQUE SCRIPT: IMMERSIVE AI SOURCE CODE

18.1 immersiveAI.cs

```
//-----------------------------------------------------------------------
// Immersive AI
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

// agent types
$iAIAgentType_Bandit = "bandit";
$iAIAgentType_Bandit_Code = 1;
$iAIAgentType_Entertainer = "entertainer";
$iAIAgentType_Entertainer_Code = 2;
$iAIAgentType_Soldier = "soldier";
$iAIAgentType_Soldier_Code = 3;

// execute the main managers/libraries
exec("./agent/iAIAgent.cs");
exec("./agent/iAIAgentManager.cs");
exec("./goals/iAIGoalLibrary.cs");
exec("./goals/iAIGoalManager.cs");

// number of agents to spawn
$IAIAGENT_COUNT = 2;

// radius to search
$IAISEEK_RADIUS = 150;
$IAISEEK_CLOSE_ENOUGH = 10;

// time between check vitals loops
$IAIAGENT_CHECK_VITALS = 10000;

// vitals rates of decay
$IAIAGENT_VITALS_HAPPINESS = -10;
$IAIAGENT_VITALS_FATIGUE = 10;
$IAIAGENT_VITALS_BOREDOM = 10;

// time between thinking loops
$IAIAGENT_THINK = 2000;

// number of think ticks before an agent leaves a goal (if in same goal
whole time)
$IAIAGENT_THINK_TICK_LIMIT = 60;

//-----------------------------------------------------------------------
/// @fn immersiveAI_Initialize()
/// @brief Initializes the immersive AI system. Called when a game
///        is created.
///
//-----------------------------------------------------------------------
function immersiveAI_Initialize()
{
   // create & init the goal library
   $iAIGoalLibrary = new iAIGoalLibrary();
   MissionCleanup.add($iAIGoalLibrary);
   $iAIGoalLibrary.initialize();

   // create & init the goal manager
   new ScriptObject(iAIGoalManager) {};
   MissionCleanup.add(iAIGoalManager);
   iAIGoalManager.initialize();
```

```
   // create the path map
   $iAIPathMap = new iAIPathMap();
   MissionCleanup.add($iAIPathMap);

   // init pathmap for the current mission
   if ($iAIPathMap.Initialize())
   {
      // start the iAIAgentManager
      new ScriptObject(iAIAgentManager) {};
      MissionCleanup.add(iAIAgentManager);
      iAIAgentManager.initialize();
   }
}

//-----------------------------------------------------------------------
/// @fn getRandomPoint()
/// @brief Helper function to get a random position in the world.
///
/// @returns Point3F Random position in the world.
//-----------------------------------------------------------------------
function getRandomPoint()
{
   %randX = getRandom(-150, 600);
   %randY = getRandom(-150, 600);
   %randZ = getTerrainHeight(%randX SPC %randY) + 50;
   return %randX SPC %randY SPC %randZ;
}
```

19 APPENDIX L – TORQUE SCRIPT: AGENT SOURCE CODE

19.1 iAIAgent.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: iAIAgent
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

// global datablock for all agents
datablock PlayerData(iAIAgentData : PlayerBody)
{
   category = "iAIAgent";
};

// load all modules scripts
exec("./iAIAgent.Seek.cs");
exec("./iAIAgent.Combat.cs");

// load all agent types
exec("./iAIAgent_Bandit.cs");
exec("./iAIAgent_Entertainer.cs");
exec("./iAIAgent_Soldier.cs");


//-----------------------------------------------------------------------
/// @fn iAIAgent::think(%this)
/// @brief Main looping logic for an agent.
///
/// @param %this Datablock reference.
//-----------------------------------------------------------------------
function iAIAgent::think(%this)
{
   // check valid agent
   if ((!isObject(%this)) || (%this.getState() $= "Dead"))
      return;

   // run each modules think methods
   %this.think_seek();
   %this.think_combat();

   // increment ticks in current goal
   if (%this.currentGoal $= %this.lastTick_Goal)
   {
      %this.lastTick_ticks++;
   } else
   {
      // goal not the same as last think tick; reset tick counter
      %this.lastTick_ticks = 0;
   }

   %this.lastTick_Goal = %this.currentGoal;

   // check if ticks over limit
   if (%this.lastTick_ticks > $IAIAGENT_THINK_TICK_LIMIT)
   {
      %this.lastTick_ticks = 0;
      eval(%this.currentSolution @ "_onExit(\"" @ %this @ "\");");
   }

   // check if agent doesn't have a goal/solution
   if ((%this.currentGoal $= "") || (%this.currentSolution $= ""))
```

```
   {
      // request new goal and solution
      iAIGoalManager.requestNewGoal(%this);
   }

   // reschedule the think
   %this.schedule($IAIAGENT_THINK, think);
}

//-----------------------------------------------------------------------
/// @fn iAIAgent::checkVitals(%this)
/// @brief Updates agent vitals and throws a callback to the current
///        solutions _onCheckVitals to update as necessary.
///
/// @param %this Datablock reference.
//-----------------------------------------------------------------------
function iAIAgent::checkVitals(%this)
{
   // update the agents vitals
   %this.setHappiness(%this.getHappiness() + $IAIAGENT_VITALS_HAPPINESS);
   %this.setFatigue(%this.getFatigue() + $IAIAGENT_VITALS_FATIGUE);
   %this.setBoredom(%this.getBoredom() + $IAIAGENT_VITALS_BOREDOM);

   // callback to current solution onCheckVitals
   if (strlen(%this.currentSolution) > 0)
      eval(%this.currentSolution @ "_onCheckVitals(\"" @ %this @ "\");");

   // reschedule check vitals
   %this.schedule($IAIAGENT_CHECK_VITALS, checkVitals);
}

//-----------------------------------------------------------------------
/// @fn iAIAgent::onReachDestination(%this, %agent)
/// @brief Global function called when an agent reaches their
///        destination.
///
/// @param %this Datablock reference.
/// @param %agent Instance of the agent.
//-----------------------------------------------------------------------
function iAIAgent::onReachDestination(%this, %agent)
{
   if (isObject(%agent) && !(%agent.getState() $= "Dead"))
   {
      if (%agent.combat_IsSideStepping == true)
      {
         %agent.combat_IsSideStepping = false;
      }
      if ((%agent.getCurrentPath() > 0) &&
(isObject(%agent.getCurrentPath())))
      {
         // if there is still nodes on the path, get the next one
         if ((isObject(%agent.getCurrentPath())) &&
(%agent.getCurrentPath().hasNextNode()))
         {

%agent.setMoveDestination(%agent.getCurrentPath().nextPosition());
         }
         else
         {
            // no next node, delete the path
            %agent.getCurrentPath().delete();

            // call solution onReachDestination
```

```
            eval(%agent.currentSolution @ "_onReachDestination(\"" @ %agent
@ "\");");
         }
      } else
      {
         // no current path, throw call back to solution onReachDestination
         eval(%agent.currentSolution @ "_onReachDestination(\"" @ %agent @
"\");");
      }
   }
}

//-----------------------------------------------------------------------
/// @fn iAIAgent::generatePath(%this, %destination)
/// @brief Generates a path from two world points to another. Adds
///        the created path to the game world and assigns to the
///        parsed agent.
///
/// @param %this Agent to generate the path for.
/// @param %destination Point3F destination location.
//-----------------------------------------------------------------------
function iAIAgent::generatePath(%this, %destination)
{
   // check if have current path
   if (isObject(%this.getCurrentPath()))
      %this.getCurrentPath().delete();

   // create the new path
   %newPath = new iAIPath();

   // check path able to be created
   if (%newPath.createPath(%this.getPosition(), %destination) == true)
   {
      // set the show variables according to agents settings
      %newPath.showPath = %this.path_show;
      %newPath.renderSpline = %this.path_spline;

      // add path to the world
      MissionCleanup.add(%newPath);

      // set the agent's current path to the new one
      %this.setCurrentPath(%newPath);
   } else
   {
      // warp a little bit towards destination, hopefully make a path
      //%this.setTransform($iAIPathMap.closestNode(%this.getPosition()));
      //%this.generatePath(%destination);
   }
}

//-----------------------------------------------------------------------
/// @fn iAIAgent::onDeath(%this)
/// @brief Called when an agent dies.
///
/// @param %this Agent which died.
//-----------------------------------------------------------------------
function iAIAgent::onDeath(%this)
{
   // delete any current path of the agent's
   if (isObject(%this.getCurrentPath()))
      %this.getCurrentPath().delete();

   // remove from control centre list
```

```
   iAICC_AgentList.removeRowById(%this.getId());
   iAIAgentManager.agentCount--;
}

//-----------------------------------------------------------------------
/// @fn iAIAgent::onMoveStuck(%this, %agent)
/// @brief Global function to call when an agent is stuck. Attempts
///        to unstick the agent!
///
/// @param %this Datablock reference.
/// @param %agent Instance of the agent.
//-----------------------------------------------------------------------
function iAIAgent::onMoveStuck(%this, %agent)
{
   if (isObject(%agent) && !(%agent.getState() $= "Dead"))
   {
      if (isObject(%agent.getCurrentPath()))
      {
         iAIMessage(%agent.getId() @ " is stuck");
         // check if able to warp to next node
         if (%agent.getCurrentPath().hasNextNode())
         {
            iAIMessage(%agent.getId() @ " going to try warping");

            // warp to next node
            %agent.setTransform(%agent.getCurrentPath().nextPosition());

            // set move destination to next node in path

%agent.setMoveDestination(%agent.getCurrentPath().nextPosition());
         } else
         {
            // no next node, check distance to destination
            %distance = VectorLen(%agent.getMoveDestination() -
%agent.getTransform());
            if ((%distance < 10) && (%distance > -10))
            {
               iAIMessage(%agent.getId() @ " is close enough");
               // must be near enough to destination, finish moving
               %agent.setMoveDestination(%agent.getTransform());
            } else
            {
               // still too far away, generate a new path
               iAIMessage(%agent.getId() @ " is too far... new path");

               // delete current path
               %agent.getCurrentPath().delete();

               // generate a new path
               %agent.generatePath(%agent.getMoveDestination());
            }
         }
      } else
      {
         iAIMessage(%agent.getId() @ " is stuck with no where to go!");
      }
   }
      }
```

19.2 iAIAgent.Combat.cs

```
//-----------------------------------------------------------------------
/// @fn iAIAgent::think_combat(%this)
/// @brief Main looping logic for the COMBAT module.
///
/// @param %this Datablock reference.
//-----------------------------------------------------------------------
function iAIAgent::think_combat(%this)
{
   // clear last attacked
   %this.lastAttackedBy = 0;

   if (%this.combat_InCombat == true)
   {
      // agent is in combat; randomly sidestep
      if (getRandom(0,1) == 1)
         %this.sideStep();

      // check for LOS
      if (haveLOS(%this, %this.hunt_Object))
      {
         // if not sidestepping, stop moving as have LOS now
         if (%this.combat_IsSideStepping == false)
            %this.stop();

         // fire
         %this.setImageTrigger(0, true);

         // increase inventory (never ending ammo)
         %this.incInventory("CrossbowAmmo", 1);

         // schedule the trigger stop
         %this.schedule(1500, setImageTrigger, 0, false);
      } else
      {
         // don't have LOS start moving towards player
         %this.setMoveDestination(%this.hunt_Object.getPosition());
      }
   }
}

//-----------------------------------------------------------------------
/// @fn iAIAgent::sideStep(%this)
/// @brief Makes an agent side step in a random direction.
///
/// @param %this Datablock reference.
//-----------------------------------------------------------------------
function iAIAgent::sideStep(%this)
{
   // set that are side stepping
   %this.combat_IsSideStepping = true;

   %sideStepLimit = 50;

   // get a random X&Y
   %xrand = getRandom(1, %sideStepLimit / 2);
   %yrand = getRandom(1, %sideStepLimit / 2);

   // start destination at current position
   %destination = %this.getTransform();

   // Word(0) is the x value
```

```
   %destination = setWord(%destination, 0, (getWord(%destination, 0) +
(%xrand)));

   // Word(1) is the y value
   %destination = setWord(%destination, 1, (getWord(%destination, 1) +
(%yrand)));

   // start moving
   %this.setMoveDestination(%destination);
}

//-----------------------------------------------------------------------
/// @fn iAIAgent::onDamage(%this, %attacker)
/// @brief Called when damage is applied to the agent.
///
/// @param %this Datablock reference.
/// @param %attacker Person attacking this agent.
//-----------------------------------------------------------------------
function iAIAgent::onDamage(%this, %attacker)
{
   if ((%this.getState() $= "Dead") || (!isObject(%attacker)))
      return;

   // dont care about self
   if (%attacker.getId() == %this.getId())
      return;

   // only interested if damage was from another agent/player
   if (!(
      (%attacker.getType() & $TypeMasks::iAIAgentObjectType) ||
      (%attacker.getType() & $TypeMasks::PlayerObjectType)))
      return;

   // check if already done interrupt checking for this attacker
   if (%this.lastAttackedBy == %attacker)
      return;

   %this.lastAttackedBy = %attacker;

   // evaluate to flee area
   %fleeArea = fleeArea_evaluate(%this);
   if (%fleeArea > 0)
   {
      %this.goalInterrupt = true;
      iAIGoalManager.assignGoal(%this, "doDefend", "fleeArea");
      return;
   }

   // check if not in combat
   if (%this.combat_InCombat == false)
   {
      // not in combat, so go and attack this player!
      // go into seekAndDestroy mode...
      %this.seekAndDestroy_Object = %attacker;
      %this.goalInterrupt = true;
      iAIGoalManager.assignGoal(%this, "doHunt", "seekAndDestroy");
      return;
   } else
   {
      // in combat already; check if new attacker is closer
      %currentTargetDist = VectorDist(%this.getPosition(),
%this.hunt_Object.getPosition());
```

```
      %newTargetDist = VectorDist(%this.getPosition(),
%attacker.getPosition());

      if (%newTargetDist < %currentTargetDist)
      {
         // new target is closer, throw interrupt and hunt new attacker
         %this.seekAndDestroy_Object = %attacker;
         %this.goalInterrupt = true;
         iAIGoalManager.assignGoal(%this, "doHunt", "seekAndDestroy");
         return;
      } else
      {
         // new attacker is further away, just ignore it
      }
   }
      }
```

19.3 iAIAgent.Seek.cs

```
//-----------------------------------------------------------------------
/// @fn iAIAgent::seekObject(%this, %objectName, %objectMask)
/// @brief Global function to seek an object in the game world. As
///        specified by the item name (as defined in the datablock)
///        and type mask.
///
/// @param %this Instance of iAIAgent.
/// @param %objectName Name of the object to find. Optional.
/// @param %objectMask Object type mask to search for.
/// @param %distanceAway Distance away to be close enough. Optional.
//-----------------------------------------------------------------------
function iAIAgent::seekObject(%this, %objectName, %objectMask,
%distanceAway)
{
   // setup seeking variables - seek logic happens in main think() loop
   %this.seeking = true;
   %this.seek_ObjectName = %objectName;
   %this.seek_ObjectMask = %objectMask;

   // if distance away specified, use it
   if (%distanceAway > 0)
      %this.seek_DistanceAway = %distanceAway;
   else
      %this.seek_DistanceAway = $IAISEEK_CLOSE_ENOUGH;

   %this.seek_ObjectPosition = "";
   %this.seek_Object = 0;
   %this.seek_OnObject = false;
   %this.seek_Found = false;
   %this.detecting = false;
}

//-----------------------------------------------------------------------
/// @fn iAIAgent::detectObject(%this, %objectName, %objectMask)
/// @brief Global function to detect an object in the game world.
///
/// @param %this Instance of iAIAgent.
/// @param %objectName Name of the object to find. Optional.
/// @param %objectMask Object type mask to search for.
//-----------------------------------------------------------------------
function iAIAgent::detectObject(%this, %objectName, %objectMask)
{
   // setup seeking variables - seek logic happens in main think() loop
   %this.detecting = true;
```

```
   %this.seeking = false;
   %this.seek_ObjectName = %objectName;
   %this.seek_ObjectMask = %objectMask;
}

//-----------------------------------------------------------------------
/// @fn iAIAgent::think_seek(%this)
/// @brief Main looping logic for the SEEK module.
///
/// @param %this Datablock reference.
//-----------------------------------------------------------------------
function iAIAgent::think_seek(%this)
{
   // check if searching for something
   if (%this.seeking == true)
   {
      // check if still not found the object needed
      if (%this.seek_Found == false)
      {
         // find the closest object
         seek_findClosest(%this, %this.seek_ObjectMask,
%this.seek_ObjectName);
      } else
      {
         // check if object is dead
         if (isDead(%this.seek_Object))
         {
             seek_cancelSeek(%this);
             return;
         } else
         {
            // check if the object has moved
            seek_checkMoved(%this, %this.seek_Object);
         }
      }
   }

   if (%this.detecting == true)
   {
      %closestObject = seek_closestObject(%this, %this.seek_ObjectMask,
%this.seek_ObjectName);
      if ((%closestObject > 0) && (isObject(%closestObject)))
      {
         %this.detect_Found = true;
         %this.detect_Object = %closestObject;
         %this.detecting = false;
         eval(%this.currentSolution @ "_onReachDestination(\"" @ %this @
"\");");
      }
   }
}

function seek_generatePath(%agent, %object)
{
   %agent.seek_Found = true;
   %agent.seek_Object = %object;
   %agent.seek_ObjectPosition =
$iAIPathMap.closestNode(%object.getPosition());

   // generate path and move to object location
   if (isObject(%agent.getCurrentPath()))
      %agent.getCurrentPath().delete();
   %agent.generatePath(%agent.seek_ObjectPosition);
```

```
   %agent.setMoveDestination(%agent.getCurrentPath().nextPosition());
}

function seek_foundObject(%agent, %object)
{
   %agent.seeking = false;
   %agent.seek_Found = true;
   %agent.seek_ObjectPosition = %object.getPosition();
   %agent.seek_OnObject = true;
   %agent.seek_Object = %object;

   // close enough, if object, warp to it
   if (%object.getType() & $TypeMasks::ItemObjectType)
   {
      %agent.setTransform(%object.getPosition());
   }

   // call solution onReachDestination
   eval(%agent.currentSolution @ "_onReachDestination(\"" @ %agent @
"\");");
}

function seek_cancelSeek(%agent)
{
   %agent.seeking = false;
   %agent.seek_Found = false;
   %agent.seek_ObjectPosition = "";
   %agent.seek_OnObject = false;
   %agent.seek_Object = 0;
}

function seek_closestObject(%agent, %objectMask, %objectName)
{
   %closestObject = 0;
   %closestDist = 10000;

   // search for objects
   InitContainerRadiusSearch(%agent.getPosition(), $IAISEEK_RADIUS,
%objectMask);

   // loop through all results
   while ((%foundObject = containerSearchNext()) != 0)
   {
      // check if agent
      if (%foundObject == %agent)
         continue;

      // check if object dead
      if (isDead(%foundObject))
         continue;

      // check name if name is set..
      if (strlen(%objectName) > 0)
      {
         // check name is same
         if (strcmp(%foundObject.getDataBlock().getName(), %objectName) !=
0)
         {
            continue;
         }
      }

      // check if new object closest
```

```
      %cDistance = VectorDist(%agent.getPosition(),
%foundObject.getPosition());
      if (%cDistance < %closestDist)
      {
         %closestObject = %foundObject;
         %closestDist = %cDistance;
      }
   }

   return %closestObject;
}

function seek_findClosest(%agent, %objectMask, %objectName)
{
   %closestObject = 0;
   %closestDist = 10000;

   // search for objects
   InitContainerRadiusSearch(%agent.getPosition(), $IAISEEK_RADIUS,
%objectMask);

   // loop through all results
   while ((%foundObject = containerSearchNext()) != 0)
   {
      // check if agent
      if (%foundObject == %agent)
         continue;

      // check if object dead
      if (isDead(%foundObject))
         continue;

      // check name if name is set..
      if (strlen(%objectName) > 0)
      {
         // check name is same
         if (strcmp(%foundObject.getDataBlock().getName(), %objectName) !=
0)
         {
            continue;
         }
      }

      // check if new object closest
      %cDistance = VectorDist(%agent.getPosition(),
%foundObject.getPosition());
      if (%cDistance < %closestDist)
      {
         %closestObject = %foundObject;
         %closestDist = %cDistance;
      }
   }

   // check if found an object
   if (isObject(%closestObject))
   {
      // check if close enough
      if (%closestDist < %agent.seek_DistanceAway)
      {
         // if seeking an agent, check the LOS
         if (isAgent(%closestObject))
         {
            if (haveLOS(%agent, %closestObject))
```

```
            {
               seek_foundObject(%agent, %closestObject);
            } else
            {
               seek_generatePath(%agent, %closestObject);
            }
         } else
         {
            // not an agent, must of found the object
            seek_foundObject(%agent, %closestObject);
         }
      } else
      {
         // not close enough, generate a path to the object
         seek_generatePath(%agent, %closestObject);
      }
   } else
   {
      // still didn't find the object
      // goto some random point if agent not already moving somewhere
      if (!((isObject(%agent.getCurrentPath())) &&
(%agent.getCurrentPath().hasNextNode())))
      {
         // generate a path to a random position
         if ((%agent.getCurrentPath() > 0) &&
(isObject(%agent.getCurrentPath())))
            %agent.getCurrentPath().delete();
         %agent.generatePath(getRandomPoint());

         // start moving along the path
         %agent.setMoveDestination(%agent.getCurrentPath().nextPosition());
      }
   }

   return %closestObject;
}

function seek_CheckMoved(%agent, %object)
{
   // get all objects in radius
   InitContainerRadiusSearch(%agent.getPosition(), $IAISEEK_RADIUS,
%object.getType());

   // find the one we are after
   %foundItAgain = false;
   while ((%foundObject = containerSearchNext()) != 0)
   {
      // check if found the object
      if (%foundObject == %object)
      {
         %foundItAgain = true;
         break;
      }
   }

   // check if didn't find it
   if (%foundItAgain == false)
   {
      // set object as not found
      %this.seek_Found = false;
      return;
   }
```

```
   %distanceToObject = VectorDist(%agent.getPosition(),
%object.getPosition());
   %objectPosition = $iAIPathMap.closestNode(%object.getPosition());

   // check if close enough
   if (%distanceToObject < %agent.seek_DistanceAway)
   {
      // if seeking an agent, check the LOS
      if (isAgent(%object))
      {
         if (haveLOS(%agent, %object))
         {
            seek_foundObject(%agent, %object);
         } else
         {
            seek_generatePath(%agent, %object);
         }
      } else
      {
         // not an agent, must of found the object
         seek_foundObject(%agent, %object);
      }
   } else
   {
      // not close enough, check if object moved
      if (%objectPosition !$= %agent.seek_ObjectPosition)
      {
         // object moved; need to make a new path to its new position
         %agent.seek_ObjectPosition = %objectPosition;

         // delete current path
         if (isObject(%agent.getCurrentPath()))
            %agent.getCurrentPath().delete();

         // generate a path to new position
         %agent.generatePath(%agent.seek_ObjectPosition);

         // start moving along the path
         %agent.setMoveDestination(%agent.getCurrentPath().nextPosition());
      } else
      {
         // hasn't moved, don't need to do anything :)
      }
   }
}

function isDead(%agent)
{
   if ((!isObject(%agent)) || (%agent == 0))
   {
      return true;
   }

   // if corpse or debris, must be dead
   if ((%agent.getType() & $TypeMasks::CorpseObjectType) |
      (%agent.getType() & $TypeMasks::DebrisObjectType))
   {
      return true;
   }

   // check if player/agent and state is dead
   if (((%agent.getType() & $TypeMasks::iAIAgentObjectType) ||
        (%agent.getType() & $TypeMasks::PlayerObjectType))
```

```
        && (%agent.getState() $= "Dead"))
   {
       return true;
   }

   // must be alive (or something that cannot be dead)
   return false;
}

function isAgent(%agent)
{
   if (%agent == 0)
   {
      return false;
   }

   if ((isObject(%agent)) && (
                    (%agent.getType() & $TypeMasks::iAIAgentObjectType) ||
                    (%agent.getType() & $TypeMasks::PlayerObjectType)))
      return true;
   else
      return false;
}

function haveLOS(%agent, %object)
{
   // check valid objects
   if (!((isObject(%agent)) && (isObject(%object))))
      return;

   // start at the agents eye level
   %start = %agent.getEyeTransform();

   // end at objects object box center
   %end = %object.getWorldBoxCenter();

   // cast the ray
   %result = containerRayCast(%start, %end,  $TypeMasks::TerrainObjectType
|
                                             $TypeMasks::InteriorObjectType
|
                                             $TypeMasks::ItemObjectType |
                                             $TypeMasks::PlayerObjectType |
                                             $TypeMasks::iAIAgentObjectType
|
                                             $TypeMasks::StaticObjectType,
                                             %agent);

   // get the object collided with
      %collidedWith = getWord(%result, 0);

   // check if object looking for
      if (%collidedWith == %object)
         return true;
      else
            return false;
      }
```

19.4 iAIAgent_Bandit.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: iAIAgent_Bandit
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------


datablock PlayerData(iAIAgent_Bandit : iAIAgentData)
{
   shapeFile = "client/data/shapes/avatars/orc/orc.dts";
};

//-----------------------------------------------------------------------
/// @fn iAIAgent_Bandit::spawn(%name, %spawnPoint)
/// @brief Creates and spawns a new Bandit into the game world, with
///        the parsed name and at the spawn point.
///
/// @param %name Name of the agent.
/// @param %spawnPoint Position in the world to spawn.
//-----------------------------------------------------------------------
function iAIAgent_Bandit::spawn(%name, %spawnPoint)
{
   // create the new agent
   %newAgent = new iAIAgent() {
      dataBlock = iAIAgent_Bandit;
   };
   MissionCleanup.add(%newAgent);

   // set the agent type
   %newAgent.setAgentType($iAIAgentType_Bandit);

   // set the agents name and place at spawn point
   %newAgent.setShapeName(%name SPC "[" @ %newAgent.getAgentType() SPC
%newAgent.getId() @ "]");
   %newAgent.setTransform(%spawnPoint);

   // set the agents inventory and mount a weapon
   %newAgent.setInventory(CrossbowAmmo, 1000);
   %newAgent.mountImage(CrossbowImage, 0);

   // retrieve the goal list for the agent type
   %newAgent.goalList = $iAIGoalLibrary.getGoalList($iAIAgentType_Bandit);

   return %newAgent;
}

//-----------------------------------------------------------------------
/// @fn iAIAgent_Bandit::onMoveStuck(%this, %obj)
/// @brief Called on agent being stuck in the game world.
///
/// @param %this Datablock reference.
/// @param %obj Instance of the agent.
//-----------------------------------------------------------------------
function iAIAgent_Bandit::onMoveStuck(%this, %obj)
{
   // no special actions required, call the global onMoveStuck
   iAIAgent::onMoveStuck(%this, %obj);
}

//-----------------------------------------------------------------------
/// @fn iAIAgent_Bandit::onReachDestination(%this, %obj)
/// @brief Called when an agent reaches their destination.
```

```
///
/// @param %this Datablock reference.
/// @param %obj Instance of the agent.
//-----------------------------------------------------------------------
function iAIAgent_Bandit::onReachDestination(%this, %obj)
{
   // no special actions required, call the global onReachDestination
   iAIAgent::onReachDestination(%this, %obj);
      }
```

19.5 iAIAgent_Entertainer.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: iAIAgent_Entertainer
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------
exec("client/data/shapes/avatars/armygirl/armygirl.cs");

datablock PlayerData(iAIAgent_Entertainer : iAIAgentData)
{
   shapeFile = "client/data/shapes/avatars/armygirl/armygirl.dts";
};

//-----------------------------------------------------------------------
/// @fn iAIAgent_Entertainer::spawn(%name, %spawnPoint)
/// @brief Creates and spawns a new Entertainer into the game world,
///        with the parsed name and at the spawn point.
///
/// @param %name Name of the agent.
/// @param %spawnPoint Position in the world to spawn.
//-----------------------------------------------------------------------
function iAIAgent_Entertainer::spawn(%name, %spawnPoint)
{
   // create the new agent
   %newAgent = new iAIAgent() {
      dataBlock = iAIAgent_Entertainer;
   };
   MissionCleanup.add(%newAgent);

   // set the agent type
   %newAgent.setAgentType($iAIAgentType_Entertainer);

   // set the agents name and place at spawn point
   %newAgent.setShapeName(%name SPC "[" @ %newAgent.getAgentType() SPC
%newAgent.getId() @ "]");
   %newAgent.setTransform(%spawnPoint);

   // set the agents inventory and mount a weapon
   %newAgent.setInventory(Sword, 1);
   %newAgent.mountImage(SwordImage, 0);

   // retrieve the goal list for the agent type
   %newAgent.goalList =
$iAIGoalLibrary.getGoalList($iAIAgentType_Entertainer);

   return %newAgent;
}

//-----------------------------------------------------------------------
/// @fn iAIAgent_Entertainer::onMoveStuck(%this, %obj)
/// @brief Called on agent being stuck in the game world.
///
/// @param %this Datablock reference.
```

```
/// @param %obj Instance of the agent.
//-----------------------------------------------------------------------
function iAIAgent_Entertainer::onMoveStuck(%this, %obj)
{
   // no special actions required, call the global onMoveStuck function
   iAIAgent::onMoveStuck(%this, %obj);
}

//-----------------------------------------------------------------------
/// @fn iAIAgent_Entertainer::onReachDestination(%this, %obj)
/// @brief Called when an agent reaches their destination.
///
/// @param %this Datablock reference.
/// @param %obj Instance of the agent.
//-----------------------------------------------------------------------
function iAIAgent_Entertainer::onReachDestination(%this, %obj)
{
   // no special actions required, call the global onReachDestination
   iAIAgent::onReachDestination(%this, %obj);
      }
```

19.6 iAIAgent_Soldier.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: iAIAgent_Soldier
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------
exec("client/data/shapes/avatars/armyguy/armyguy.cs");

datablock PlayerData(iAIAgent_Soldier : iAIAgentData)
{
   shapeFile = "client/data/shapes/avatars/armyguy/armyguy.dts";
};

//-----------------------------------------------------------------------
/// @fn iAIAgent_Soldier::spawn(%name, %spawnPoint)
/// @brief Creates and spawns a new Solder into the game world, with
///        the parsed name and at the spawn point.
///
/// @param %name Name of the agent.
/// @param %spawnPoint Position in the world to spawn.
//-----------------------------------------------------------------------
function iAIAgent_Soldier::spawn(%name, %spawnPoint)
{
   // create the new agent
   %newAgent = new iAIAgent() {
      dataBlock = iAIAgent_Soldier;
   };
   MissionCleanup.add(%newAgent);

   // set the agent type
   %newAgent.setAgentType($iAIAgentType_Soldier);

   // set the agents name and place at spawn point
   %newAgent.setShapeName(%name SPC "[" @ %newAgent.getAgentType() SPC
%newAgent.getId() @ "]");
   %newAgent.setTransform(%spawnPoint);

   // set the agents inventory and mount a weapon
   %newAgent.setInventory(CrossbowAmmo, 1000);
   %newAgent.mountImage(CrossbowImage, 0);

   // retrieve the goal list for the agent type
```

```
   %newAgent.goalList = $iAIGoalLibrary.getGoalList($iAIAgentType_Soldier);

   return %newAgent;
}

//-----------------------------------------------------------------------
/// @fn iAIAgent_Soldier::onMoveStuck(%this, %obj)
/// @brief Called on agent being stuck in the game world.
///
/// @param %this Datablock reference.
/// @param %obj Instance of the agent.
//-----------------------------------------------------------------------
function iAIAgent_Soldier::onMoveStuck(%this, %obj)
{
   // no special actions required, call the global onMoveStuck function
   iAIAgent::onMoveStuck(%this, %obj);
}

//-----------------------------------------------------------------------
/// @fn iAIAgent_Soldier::onReachDestination(%this, %obj)
/// @brief Called when an agent reaches their destination.
///
/// @param %this Datablock reference.
/// @param %obj Instance of the agent.
//-----------------------------------------------------------------------
function iAIAgent_Soldier::onReachDestination(%this, %obj)
{
   // no special actions required, call the global onReachDestination
   iAIAgent::onReachDestination(%this, %obj);
      }
```

19.7 iAIAgentManager.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: iAIAgentManager
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
/// @fn initialize(%this)
/// @brief Initializes the iAIAgentManager. Called when a new game is
///        created.
///
/// @param %this Instance of iAIAgentManager.
//-----------------------------------------------------------------------
function iAIAgentManager::initialize(%this)
{
   // set initial agent count to 0
   %this.agentCount = 0;

   // waits 15 seconds after game creation to start the thinking loop
   %this.schedule(15000, think);
}

//-----------------------------------------------------------------------
/// @fn think(%this)
/// @brief Main thinking loop. Called every 2seconds.
///        Detects if new agents need to be spawned.
///
/// @param %this Instance of iAIManager.
//-----------------------------------------------------------------------
function iAIAgentManager::think(%this)
{
```

```
   // check if need to spawn more agents
   if (%this.agentCount < $IAIAGENT_COUNT)
   {
      // spawn an agent!
      %this.agentCount++;
      %spawnedAgent = %this.spawn();
      iAIMessage("Spawned agent " @ %spawnedAgent.getId() @ " #" @
%this.agentCount @ "/" @ $IAIAGENT_COUNT);
   }

   // reschedule the thinking loop
   %this.schedule(2000, think);
}

//-----------------------------------------------------------------------
/// @fn spawn(%this, %agentTypeCode = random)
/// @brief Spawns a new agent into the game world. If %agentTypeCode
///        is not specified, a random agent type is chosen.
///
/// @param %this Instance of iAIManager.
/// @param %agentType Agent type to spawn in the world.
/// @returns iAIAgent Agent spawned.
//-----------------------------------------------------------------------
function iAIAgentManager::spawn(%this, %agentTypeCode)
{
   // check if agent type code not passed
   if (!(%agentTypeCode > 0))
   {
      // get a random agent type
      %agentTypeCode = getRandom(1, 3);
   }

   // create a new agent, based on type code
   switch(%agentTypeCode)
   {
      case $iAIAgentType_Bandit_Code: %newAgent =
iAIAgent_Bandit::spawn(createRandomName("client/data/names.nam"),
pickSpawnPoint());
      case $iAIAgentType_Entertainer_Code: %newAgent =
iAIAgent_Entertainer::spawn(createRandomName("client/data/names.nam"),
pickSpawnPoint());
      case $iAIAgentType_Soldier_Code: %newAgent =
iAIAgent_Soldier::spawn(createRandomName("client/data/names.nam"),
pickSpawnPoint());
   }

   // set the agent vitals
   %newAgent.setHappiness(100);
   %newAgent.setMoney(200.0);
   %newAgent.setHealth(100);
   %newAgent.setFatigue(0);
   %newAgent.setBoredom(0);

   // set default path options
   %newAgent.path_spline = false;
   %newAgent.path_show = true;

   // set as not in combat
   %newAgent.combat_InCombat = false;

   // schedule a check vitals
   %newAgent.schedule($IAIAGENT_CHECK_VITALS, checkVitals);
```

```
   // add to gui agent list
   iAICC_AgentList.addRow(%newAgent.getId(), %newAgent.getShapeName());

   // request a goal
   %newAgent.goalInterrupt = false;
   %newAgent.resumePrevious = false;
   %newAgent.lastTick_ticks = 0;
   %newAgent.lastTick_Goal = "";
   iAIGoalManager.requestNewGoal(%newAgent);

   // start the thinking loop
   %newAgent.schedule($IAIAGENT_THINK, think);

   return %newAgent;
      }
```

20 APPENDIX M – TORQUE SCRIPT: GOALS SOURCE CODE

20.1 iAIGoalLibrary.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: iAIGoalLibrary
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

function iAIGoalLibrary::initialize(%this)
{
   // execute all the goals
   exec("./doDefend/doDefend.cs");
   exec("./doExplore/doExplore.cs");
   exec("./doHunt/doHunt.cs");
   exec("./doRest/doRest.cs");
   exec("./getFood/getFood.cs");
   exec("./getHealth/getHealth.cs");
   exec("./goHome/goHome.cs");
   exec("./haveFun/haveFun.cs");

   // sort the library
   %this.sort();
      }
```

20.2 iAIGoalManager.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: iAIGoalManager
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
/// @fn initialize(%this)
/// @brief Initializes the GoalManager. Called when a new game is
///        created.
///
/// @param %this Instance of iAIGoalManager.
//-----------------------------------------------------------------------
function iAIGoalManager::initialize(%this)
{
   //
}

//-----------------------------------------------------------------------
/// @fn assignGoal(%this, %agent, %goal, %solution)
/// @brief Assigns the parsed goal & solution to the specific agent.
///        Executes the 'onEnter' method of the selected solution.
///
/// @param %this Instance of iAIGoalManager.
/// @param %agent Agent to assign to.
/// @param %goal Goal to assign to the agent.
/// @param %solution Solution for the goal.
/// @returns bool Assignment success.
//-----------------------------------------------------------------------
function iAIGoalManager::assignGoal(%this, %agent, %goal, %solution)
{
   // check valid parameters
   if (((!isObject(%agent)) || (%agent.getState() $= "Dead")) ||
       ((strlen(%goal) == 0) || (strlen(%solution) == 0)))
      return false;
```

```
   //%agent.setActionThread("look");//should reset their animations

   // check for a goal interrupt
   if (%agent.goalInterrupt == true)
   {
      %agent.goalInterrupt = false;
      %agent.resumePrevious = true;
   } else
   {
      if (%agent.resumePrevious == true)
      {
         %agent.resumePrevious = false;

         // resume previous only if previous is a differnt goal/solution!
         if (!((%agent.previousGoal $= %agent.currentGoal) &&
            (%agent.previousSolution $= %agent.currentSolution)))
         {
            iAIGoalManager.resumePreviousSolution(%agent);
            return;
         }
      }
   }

   // check if agent has a current goal
   if (strlen(%agent.currentGoal) != 0)
   {
      // set the previous goal/solution to the current ones
      %agent.previousGoal = %agent.currentGoal;
      %agent.previousSolution = %agent.currentSolution;
   }

   // set the new goal & solution
   %agent.currentGoal = %goal;
   %agent.currentSolution = %solution;

   // execute the solution
   %current = eval(%solution @ "_onEnter(\"" @ %agent @ "\");");

   return true;
}

//-----------------------------------------------------------------------
/// @fn evaluateList(%this, %agent, %wordList)
/// @brief Evaluates the list of words in the parsed space
///        deliminated list. Executes each words '_evaluate' method
///        passing the %agent variable.
///
/// @param %this Instance of iAIGoalManager.
/// @param %agent Agent to pass into the evaluate method.
/// @param %wordList String of space deliminated words to execute.
/// @param %avoidWord Word to avoid in the list.
/// @returns string Highest evaluated word.
//-----------------------------------------------------------------------
function iAIGoalManager::evaluateList(%this, %agent, %wordList, %avoidWord)
{
   // get number of words in the wordList
   %wordCount = getWordCount(%wordList);

   // hold the max value & max string found
   %maxValue = -1;
   %maxName = "";

   // iterate over all goals
```

```
   for (%i = 0; %i < %wordCount; %i++)
   {
      // get the current goal name from the goalList
      %currentWord = getWord(%wordList, %i);

      // check if to avoid the current word
      if (%currentWord $= %avoidWord)
         continue;

      // execute the goal evaluate method
      %current = eval(%currentWord @ "_evaluate(\"" @ %agent @ "\");");

      // check for less than 0, as not allowed into this goal
      if (%current < 0)
         continue;

      // check if best goal found
      if (%current > %maxValue)
      {
         // set best goal values
         %maxValue = %current;
         %maxName = %currentWord;
      }
   }

   return %maxName;
}

//-----------------------------------------------------------------------
/// @fn requestNewGoal(%this, %agent, %avoidCurrentGoal,
///                    %avoidCurrentSolution)
/// @brief Requests a new goal for the parsed agent.
///
/// @param %this Instance of iAIGoalManager.
/// @param %agent Agent to find a new goal for.
/// @param %avoidCurrentGoal Flag to avoid the current goal.
/// @param %avoidCurrentSolution Flag to avoid the current solution.
/// @returns bool Request success.
//-----------------------------------------------------------------------
function iAIGoalManager::requestNewGoal(%this, %agent, %avoidCurrentGoal,
%avoidCurrentSolution)
{
   // check valid parameters
   if ((!isObject(%agent)) || (%agent.getState() $= "Dead"))
      return false;

   // check if wanting to avoid the current goal
   %avoidWord = "";
   if (%avoidCurrentGoal == true)
      %avoidWord = %agent.currentGoal;

   // evaluate the agents goals
   %goal = %this.evaluateList(%agent, %agent.goalList, %avoidWord);

   // check goal found
   if (strlen(%goal) == 0)
      return false;

   // get a list of solutions for the goal
   %solutionList = $iAIGoalLibrary.getSolutionList(%agent.getAgentType(),
%goal);

   // check solution list found
```

```
   if (strlen(%solutionList) == 0)
      return false;

   // check if wanting to avoid the current solution
   %avoidWord = "";
   if (%avoidCurrentSolution == true)
      %avoidWord = %agent.currentSolution;

   // evaluate the solutions
   %solution = %this.evaluateList(%agent, %solutionList, %avoidWord);

   // check solution found
   if (strlen(%solution) == 0)
      return false;

   // assign and execute the goal and solution to the agent
   return (%this.assignGoal(%agent, %goal, %solution));
}

//-----------------------------------------------------------------------
/// @fn requestNewSolution(%this, %agent, %avoidCurrentSolution)
/// @brief Requests a new solution for the parsed agent; retaining
///        the current goal.
///
/// @param %this Instance of iAIGoalManager.
/// @param %agent Agent to find a new solution for.
/// @param %avoidCurrentSolution Flag to avoid the current solution.
/// @returns bool Request success.
//-----------------------------------------------------------------------
function iAIGoalManager::requestNewSolution(%this, %agent,
%avoidCurrentSolution)
{
   // check valid parameters
   if ((!isObject(%agent)) || (%agent.getState() $= "Dead") ||
       (strlen(%agent.currentGoal) == 0))
      return false;

   // get a list of solutions for the current goal
   %solutionList = $iAIGoalLibrary.getSolutionList(%agent.getAgentType(),
%agent.currentGoal);

   // check solution list found
   if (strlen(%solutionList) == 0)
      return false;

   // check if wanting to avoid the current solution
   %avoidWord = "";
   if (%avoidCurrentSolution == true)
      %avoidWord = %agent.currentSolution;

   // evaluate the solutions
   %solution = %this.evaluateList(%agent, %solutionList, %avoidWord);

   // check solution found
   if (strlen(%solution) == 0)
      return false;

   // assign and execute the solution to the agent
   return (%this.assignGoal(%agent, %agent.currentGoal, %solution));
}

//-----------------------------------------------------------------------
/// @fn resumePreviousSolution(%this, %agent)
```

```
/// @brief Resumes a previous solution for the given agent.
///
/// @param %this Instance of iAIGoalManager.
/// @param %agent Agent to resume for.
/// @returns bool Assignment success.
//-----------------------------------------------------------------------
function iAIGoalManager::resumePreviousSolution(%this, %agent)
{
   // check valid parameters
   if ((!isObject(%agent)) || (%agent.getState() $= "Dead"))
      return false;

   // check if agent has previous
   if ((strlen(%agent.previousGoal) != 0) &&
(strlen(%agent.previousSolution) != 0))
   {
      // assign the previous goal/solution
      return (iAIGoalManager.assignGoal(%agent, %agent.previousGoal,
%agent.previousSolution));
   } else
   {
      // didn't have a previous goal/solution
      return false;
   }
}

//-----------------------------------------------------------------------
/// @fn completedSolution(%this, %agent)
/// @brief Completes and requests a new goal for the given agent.
///
/// @param %this Instance of iAIGoalManager.
/// @param %agent Agent to complete for.
//-----------------------------------------------------------------------
function iAIGoalManager::completedSolution(%this, %agent)
{
   // check valid parameters
   if ((!isObject(%agent)) || (%agent.getState() $= "Dead"))
      return false;

   // some cleanup..
   %agent.setActionThread("look");
   if (isObject(%agent.getCurrentPath()))
      %agent.getCurrentPath().delete();

   // check if over think tick limit
   if (%this.lastTick_ticks > $IAIAGENT_THINK_TICK_LIMIT)
   {
      // request a different goal and solution
      %this.requestNewGoal(%agent, true, true);
   } else
   {
      // request a new goal and solution
      %this.requestNewGoal(%agent);
   }
      }
```

20.3 doDefend.cs

```
//---------------------------------------------------------------------
// Immersive AI :: Goal :: doDefend
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//---------------------------------------------------------------------

//---------------------------------------------------------------------
// Load all the solution files
//---------------------------------------------------------------------
exec("./fleeArea.cs");

//---------------------------------------------------------------------
// Add goal and solutions to library for each agent
//---------------------------------------------------------------------
// the defend goals are started from when an agent is attacked
$iAIGoalLibrary.addGoal($iAIAgentType_Entertainer, "doDefend");
$iAIGoalLibrary.addSolution($iAIAgentType_Entertainer, "doDefend",
"fleeArea");

$iAIGoalLibrary.addGoal($iAIAgentType_Soldier, "doDefend");
$iAIGoalLibrary.addSolution($iAIAgentType_Soldier, "doDefend", "fleeArea");


//---------------------------------------------------------------------
/// @fn doDefend_evaluate(%agent)
/// @brief Called to determine whether to assign this goal.
///
/// @param %agent Agent to determine for.
/// @returns F32 between 0.00 and 1.00
//---------------------------------------------------------------------
function doDefend_evaluate(%agent)
{
   // always return -1 to ensure never use this goal
   return -1;
      }
```

20.3.1 fleeArea.cs

```
//---------------------------------------------------------------------
// Immersive AI :: Goal :: doDefend :: Solution :: fleeArea
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//---------------------------------------------------------------------

//---------------------------------------------------------------------
/// @fn fleeArea_evaluate(%agent)
/// @brief Called to determine whether to assign this solution.
///
/// @param %agent Agent to determine for.
/// @returns F32 between 0.00 and 1.00
//---------------------------------------------------------------------
function fleeArea_evaluate(%agent)
{
   if (%agent.getAgentType() $= $iAIAgentType_Entertainer)
      return 1.0;

   // check if outside ranges to flee area
   if ((%agent.getHealth() >= 50) && (%agent.getFatigue() <= 50))
   {
      return 0.0;
   }
```

```
   // calculate the weightings of each parameter
   %healthWeight = (50 - %agent.getHealth()) / 50;
   %fatigueWeight = (%agent.getFatigue() - 50) / 50;

   // average the weightings (div 10 to allow for futher weightings)
   %avgWeight = ((%healthWeight + %fatigueWeight) / 2) / 10;

   // multiply by agent weight for this goal
   switch$(%agent.getAgentType())
   {
      case $iAIAgentType_Soldier: return (%avgWeight * 10);
      default : return 0.0;
   }
}

//-----------------------------------------------------------------------
/// @fn fleeArea_onEnter(%agent)
/// @brief Called on entry to this solution state.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function fleeArea_onEnter(%agent)
{
   iAIMessage("STATE: fleeArea being executed by " @ %agent);

   fleeArea_execute(%agent);
}

//-----------------------------------------------------------------------
/// @fn fleeArea_execute(%agent)
/// @brief Main execution state for the agent, called from onEnter.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function fleeArea_execute(%agent)
{
   // delete any current path
   if (isObject(%agent.getCurrentPath()))
      %agent.getCurrentPath().delete();

   // generate a path to a random position
   %agent.generatePath(getRandomPoint());

   // move along the path
   %agent.setMoveDestination(%agent.getCurrentPath().nextPosition());
}

//-----------------------------------------------------------------------
/// @fn fleeArea_onReachDestination(%agent)
/// @brief Callback from onReachDestination. When agent reaches
///        end point on path, function is called.
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
function fleeArea_onReachDestination(%agent)
{
   // reached the end of the flee path
   fleeArea_onExit(%agent);
}

//-----------------------------------------------------------------------
/// @fn fleeArea_onCheckVitals(%agent)
/// @brief Callback from checkVitals.
```

```
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
function fleeArea_onCheckVitals(%agent)
{
   %agent.setFatigue(%agent.getFatigue() + ($IAIAGENT_VITALS_FATIGUE *
1.5));
}

//-----------------------------------------------------------------------
/// @fn fleeArea_onExit(%agent)
/// @brief Called on exiting from this solution, called from execute.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function fleeArea_onExit(%agent)
{
   // update vitals
   %agent.setHappiness(%agent.getHappiness() - 20);
   %agent.setFatigue(%agent.getFatigue() + 20);

   iAIGoalManager.completedSolution(%agent);
      }
```

20.4 doExplore.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: Goal :: doExplore
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
// Load all the solution files
//-----------------------------------------------------------------------
exec("./exploreArea.cs");

//-----------------------------------------------------------------------
// Add goal and solutions to library for each agent
//-----------------------------------------------------------------------
$iAIGoalLibrary.addGoal($iAIAgentType_Bandit, "doExplore");
$iAIGoalLibrary.addSolution($iAIAgentType_Bandit, "doExplore",
"exploreArea");

$iAIGoalLibrary.addGoal($iAIAgentType_Entertainer, "doExplore");
$iAIGoalLibrary.addSolution($iAIAgentType_Entertainer, "doExplore",
"exploreArea");

//-----------------------------------------------------------------------
/// @fn doExplore_evaluate(%agent)
/// @brief Called to determine whether to assign this goal.
///
/// @param %agent Agent to determine for.
/// @returns F32 between 0.00 and 1.00
//-----------------------------------------------------------------------
function doExplore_evaluate(%agent)
{
   // check if outside ranges to do explore
   if ((%agent.getFatigue() >= 33) || (%agent.getHappiness() <= 50) ||
      (%agent.getBoredom() <= 50))
   {
      return 0.0;
   }
```

```
   // calculate the weightings of each parameter
   %fatigueWeight = (33 - %agent.getFatigue()) / 33;
   %happinessWeight = (%agent.getHappiness() - 50) / 50;
   %boredomWeight = (%agent.getBoredom() - 50) / 50;

   // average the weightings (div 10 to allow for futher weightings)
   %avgWeight = ((%fatigueWeight + %happinessWeight + %boredomWeight) / 3)
/ 10;

   // multiply by agent weight for this goal
   switch$(%agent.getAgentType())
   {
      case $iAIAgentType_Entertainer: return (%avgWeight * 1);
      case $iAIAgentType_Bandit: return (%avgWeight * 2);
      default : return 0.0;
   }
      }
```

20.4.1 exploreArea.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: Goal :: doExplore :: Solution :: exploreArea
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
/// @fn exploreArea_evaluate(%agent)
/// @brief Called to determine whether to assign this solution.
///
/// @param %agent Agent to determine for.
/// @returns F32 between 0.00 and 1.00
//-----------------------------------------------------------------------
function exploreArea_evaluate(%agent)
{
   // check if outside ranges to do explore
   if (%agent.getBoredom() <= 66)
   {
      return 0.0;
   }

   // calculate the weightings of each parameter
   %boredomWeight = (%agent.getBoredom() - 66) / 66;

   // average the weightings (div 10 to allow for futher weightings)
   %avgWeight = ((%boredomWeight) / 1) / 10;

   // multiply by agent weight for this goal
   switch$(%agent.getAgentType())
   {
      case $iAIAgentType_Entertainer: return (%avgWeight * 10);
      case $iAIAgentType_Bandit: return (%avgWeight * 10);
      default : return 0.0;
   }
}

//-----------------------------------------------------------------------
/// @fn exploreArea_onEnter(%agent)
/// @brief Called on entry to this solution state.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function exploreArea_onEnter(%agent)
{
```

```
   iAIMessage("STATE: exploreArea being executed by " @ %agent);

   // generate a random number of paths to explore along
   %agent.exploreArea_totalPaths = getRandom(1, 20);
   %agent.exploreArea_paths = 0;

   // nothing to do when entering the state, just execute
   exploreArea_execute(%agent);
}

//-----------------------------------------------------------------------
/// @fn exploreArea_execute(%agent)
/// @brief Main execution state for the agent, called from onEnter.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function exploreArea_execute(%agent)
{
   // check if visited enough paths
   if (%agent.exploreArea_paths >= %agent.exploreArea_totalPaths)
   {
      // exit the state
      exploreArea_onExit(%agent);
      return;
   }

   // generate a path to a random position
   %agent.generatePath(getRandomPoint());

   // move along the path
   %agent.setMoveDestination(%agent.getCurrentPath().nextPosition());
   %agent.exploreArea_paths++;
}

//-----------------------------------------------------------------------
/// @fn exploreArea_onReachDestination(%agent)
/// @brief Callback from onReachDestination. When agent reaches
///        end point on path, function is called.
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
function exploreArea_onReachDestination(%agent)
{
   // agent has reached the end of the path, return to execute method
   exploreArea_execute(%agent);
}

//-----------------------------------------------------------------------
/// @fn exploreArea_onCheckVitals(%agent)
/// @brief Callback from checkVitals.
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
function exploreArea_onCheckVitals(%agent)
{
   %agent.setFatigue(%agent.getFatigue() + ($IAIAGENT_VITALS_FATIGUE *
1.5));
}

//-----------------------------------------------------------------------
/// @fn exploreArea_onExit(%agent)
/// @brief Called on exiting from this solution, called from execute.
///
```

```
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function exploreArea_onExit(%agent)
{
   %agent.setBoredom(%agent.getBoredom() - 20);
   iAIGoalManager.completedSolution(%agent);
      }
```

20.5 doHunt.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: Goal :: doHunt
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
// Load all the solution files
//-----------------------------------------------------------------------
exec("./seekAndDestroy.cs");
exec("./patrolArea.cs");

//-----------------------------------------------------------------------
// Add goal and solutions to library for each agent
//-----------------------------------------------------------------------
$iAIGoalLibrary.addGoal($iAIAgentType_Bandit, "doHunt");
$iAIGoalLibrary.addSolution($iAIAgentType_Bandit, "doHunt",
"seekAndDestroy");
$iAIGoalLibrary.addSolution($iAIAgentType_Bandit, "doHunt", "patrolArea");

$iAIGoalLibrary.addGoal($iAIAgentType_Soldier, "doHunt");
$iAIGoalLibrary.addSolution($iAIAgentType_Soldier, "doHunt",
"seekAndDestroy");
$iAIGoalLibrary.addSolution($iAIAgentType_Soldier, "doHunt", "patrolArea");

//-----------------------------------------------------------------------
/// @fn doHunt_evaluate(%agent)
/// @brief Called to determine whether to assign this goal.
///
/// @param %agent Agent to determine for.
/// @returns F32 between 0.00 and 1.00
//-----------------------------------------------------------------------
function doHunt_evaluate(%agent)
{
   // check if outside ranges to do hunt
   if ((%agent.getHealth() <= 50) || (%agent.getFatigue() >= 33) ||
(%agent.getHappiness() <= 33))
   {
      return 0.0;
   }

   // calculate the weightings of each parameter
   %fatigueWeight = (33 - %agent.getFatigue()) / 33;
   %happinessWeight = (%agent.getHappiness() - 33) / 33;
   %healthWeight = (%agent.getHealth() - 50) / 50;

   // average the weightings (div 10 to allow for futher weightings)
   %avgWeight = ((%fatigueWeight + %happinessWeight + %healthWeight) / 3) /
10;

   // multiply by agent weight for this goal
```

```
   switch$(%agent.getAgentType())
   {
      case $iAIAgentType_Soldier: return (%avgWeight * 4);
      case $iAIAgentType_Bandit: return (%avgWeight * 2);
      default : return 0.0;
   }
      }
```

20.5.1 patrolArea.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: Goal :: doHunt :: Solution :: patrolArea
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
/// @fn patrolArea_evaluate(%agent)
/// @brief Called to determine whether to assign this solution.
///
/// @param %agent Agent to determine for.
/// @returns F32 between 0.00 and 1.00
//-----------------------------------------------------------------------
function patrolArea_evaluate(%agent)
{
   // check if outside ranges to patrol area
   if (%agent.getBoredom() >= 66)
   {
      return 0.0;
   }

   // calculate the weightings of each parameter
   %boredomWeight = (66 - %agent.getBoredom()) / 66;

   // average the weightings (div 10 to allow for futher weightings)
   %avgWeight = ((%boredomWeight) / 1) / 10;

   // multiply by agent weight for this goal
   switch$(%agent.getAgentType())
   {
      case $iAIAgentType_Soldier: return (%avgWeight * 7);
      case $iAIAgentType_Bandit: return (%avgWeight * 3);
      default : return 0.0;
   }
}

//-----------------------------------------------------------------------
/// @fn patrolArea_onEnter(%agent)
/// @brief Called on entry to this solution state.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function patrolArea_onEnter(%agent)
{
   iAIMessage("STATE: patrolArea being executed by " @ %agent);

   %agent.detect_Found = false;

   // pick 4 way points to run between in the patrol loop
   if (%agent.patrolArea_WP1 $= "")
      %agent.patrolArea_WP1 = getRandomPoint();
   if (%agent.patrolArea_WP2 $= "")
      %agent.patrolArea_WP2 = getRandomPoint();
   if (%agent.patrolArea_WP3 $= "")
```

```
      %agent.patrolArea_WP3 = getRandomPoint();
   if (%agent.patrolArea_WP4 $= "")
      %agent.patrolArea_WP4 = getRandomPoint();

   %nextWP = "";
   switch (%agent.patrolArea_currentWP)
   {
      case 0: %nextWP = %agent.patrolArea_WP1;
      case 1: %nextWP = %agent.patrolArea_WP2;
      case 2: %nextWP = %agent.patrolArea_WP3;
      case 3: %nextWP = %agent.patrolArea_WP4;
      default: %nextWP = %agent.patrolArea_WP1;
   }

   // generate a path to the next way point
   %agent.generatePath(%nextWP);
   %agent.setMoveDestination(%agent.getCurrentPath().nextPosition());

   // setup detecting of specific agent types
   switch$(%agent.getAgentType())
   {
      case $iAIAgentType_Soldier: %agent.detectObject("iAIAgent_Bandit",
$TypeMasks::iAIAgentObjectType);
      case $iAIAgentType_Bandit: %agent.detectObject("iAIAgent_Soldier",
$TypeMasks::iAIAgentObjectType);
      default : patrolArea_onExit(%agent);
   }
}

//-----------------------------------------------------------------------
/// @fn patrolArea_execute(%agent)
/// @brief Main execution state for the agent, called from onEnter.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function patrolArea_execute(%agent)
{
   %nextWP = "";
   switch (%agent.patrolArea_currentWP)
   {
      case 1: %nextWP = %agent.patrolArea_WP2;
      case 2: %nextWP = %agent.patrolArea_WP3;
      case 3: %nextWP = %agent.patrolArea_WP4;
      case 4: %nextWP = %agent.patrolArea_WP1;
   }

   // generate a path to the next way point
   %agent.generatePath(%nextWP);

   // move to next way point
   %agent.setMoveDestination(%agent.getCurrentPath().nextPosition());
}

//-----------------------------------------------------------------------
/// @fn patrolArea_onReachDestination(%agent)
/// @brief Callback from onReachDestination. When agent reaches
///        end point on path, function is called.
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
function patrolArea_onReachDestination(%agent)
{
   // check if found an object
```

```
   if (%agent.detect_Found == true)
   {
      // go into seekAndDestroy mode...
      %agent.goalInterrupt = true;
      iAIGoalManager.assignGoal(%agent, "doHunt", "seekAndDestroy");
      return;
   }

   %agent.patrolArea_currentWP++;
   if (%agent.patrolArea_currentWP > 4)
      %agent.patrolArea_currentWP = 1;
   patrolArea_execute(%agent);
}

//-----------------------------------------------------------------------
/// @fn patrolArea_onCheckVitals(%agent)
/// @brief Callback from checkVitals.
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
function patrolArea_onCheckVitals(%agent)
{
   %agent.setFatigue(%agent.getFatigue() + ($IAIAGENT_VITALS_FATIGUE *
1.5));
   %agent.setBoredom(%agent.getBoredom() + ($IAIAGENT_VITALS_BOREDOM *
1.2));
}

//-----------------------------------------------------------------------
/// @fn patrolArea_onExit(%agent)
/// @brief Called on exiting from this solution, called from execute.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function patrolArea_onExit(%agent)
{
   // gets $50 for patrolling
   %agent.setMoney(%agent.getMoney() + 50);

   %agent.detect_Found = false;
   %agent.detecting = false;
   %agent.detect_Object = 0;
   iAIGoalManager.completedSolution(%agent);
      }
```

20.5.2 seekAndDestroy.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: Goal :: doHunt :: Solution :: seekAndDestroy
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
/// @fn seekAndDestroy_evaluate(%agent)
/// @brief Called to determine whether to assign this solution.
///
/// @param %agent Agent to determine for.
/// @returns F32 between 0.00 and 1.00
//-----------------------------------------------------------------------
function seekAndDestroy_evaluate(%agent)
{
   // check if outside ranges to patrol area
   if (%agent.getHappiness() >= 66)
```

```
   {
      return 0.0;
   }

   // calculate the weightings of each parameter
   %happinessWeight = (66 - %agent.getHappiness()) / 66;

   // average the weightings (div 10 to allow for futher weightings)
   %avgWeight = ((%happinessWeight) / 1) / 10;

   // multiply by agent weight for this goal
   switch$(%agent.getAgentType())
   {
      case $iAIAgentType_Soldier: return (%avgWeight * 3);
      case $iAIAgentType_Bandit: return (%avgWeight * 7);
      default : return 0.0;
   }
}

//-----------------------------------------------------------------------
/// @fn seekAndDestroy_onEnter(%agent)
/// @brief Called on entry to this solution state.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function seekAndDestroy_onEnter(%agent)
{
   iAIMessage("STATE: seekAndDestroy being executed by " @ %agent);

   // find an agent to kill
   %agent.hunt_CloseFollowing = false;

   // check if need to explicity attack someone
   if (isObject(%agent.seekAndDestroy_Object))
   {
      %agent.seeking = true;
      %agent.seek_Found = true;
      %agent.seek_ObjectMask = %agent.seekAndDestroy_Object.getType();
      %agent.seek_DistanceAway = 100;
      seek_generatePath(%agent, %agent.seekAndDestroy_Object);
   } else
   {
      // set the seek depending on the agent type
      switch$(%agent.getAgentType())
      {
         case $iAIAgentType_Soldier: %agent.seekObject("iAIAgent_Bandit",
$TypeMasks::iAIAgentObjectType, 100);
         case $iAIAgentType_Bandit: %agent.seekObject("iAIAgent_Soldier",
$TypeMasks::iAIAgentObjectType, 100);
         default : seekAndDestroy_onExit(%agent);
      }
   }
}

//-----------------------------------------------------------------------
/// @fn seekAndDestroy_execute(%agent)
/// @brief Main execution state for the agent, called from onEnter.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function seekAndDestroy_execute(%agent)
{
   if ((!isObject(%agent)) || (%agent.getState() $= "Dead"))
```

```
      return;

   // check if object is dead
   if (isDead(%agent.hunt_Object))
   {
      seekAndDestroy_onExit(%agent);
      return;
   }

   // schedule execute again after 2seconds
   schedule(2000, 0, seekAndDestroy_execute, %agent);
}

//-----------------------------------------------------------------------
/// @fn seekAndDestroy_onReachDestination(%agent)
/// @brief Callback from onReachDestination. When agent reaches
///        end point on path, function is called.
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
function seekAndDestroy_onReachDestination(%agent)
{
   // check if finished seeking
   if ((%agent.seeking == false) && (%agent.hunt_CloseFollowing == false)
&& (%agent.seek_Found == true))
   {
      // now stay with the player..
      %agent.hunt_Object = %agent.seek_Object;
      %agent.hunt_CloseFollowing = true;

      // set as agent now in combat
      %agent.combat_InCombat = true;

      // stay aiming at the hunting agent
      %agent.setAimObject(%agent.hunt_Object, "0 0 2");

      // delete any current path
      if (isObject(%agent.getCurrentPath()))
         %agent.getCurrentPath().delete();

      seekAndDestroy_execute(%agent);
   }
}

//-----------------------------------------------------------------------
/// @fn seekAndDestroy_onCheckVitals(%agent)
/// @brief Callback from checkVitals.
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
function seekAndDestroy_onCheckVitals(%agent)
{
   %agent.setFatigue(%agent.getFatigue() + ($IAIAGENT_VITALS_FATIGUE *
1.5));
   %agent.setBoredom(%agent.getBoredom() + ($IAIAGENT_VITALS_BOREDOM *
1.2));
}

//-----------------------------------------------------------------------
/// @fn seekAndDestroy_onExit(%agent)
/// @brief Called on exiting from this solution, called from execute.
///
/// @param %agent Agent state is assigned to.
```

```
//-----------------------------------------------------------------------
function seekAndDestroy_onExit(%agent)
{
   // stops fireing and clear the aim
   %agent.setImageTrigger(0, false);
   %agent.clearAim();

   %agent.hunt_CloseFollowing = false;
   %agent.hunt_Object = 0;
   %agent.combat_InCombat = false;
   %agent.seekAndDestroy_Object = 0;

   // gets $50 for destroying
   %agent.setMoney(%agent.getMoney() + 50);

   iAIGoalManager.completedSolution(%agent);
      }
```

20.6 doRest.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: Goal :: doRest
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
// Load all the solution files
//-----------------------------------------------------------------------
exec("./relax.cs");
exec("./sleep.cs");

//-----------------------------------------------------------------------
// Add goal and solutions to library for each agent
//-----------------------------------------------------------------------
$iAIGoalLibrary.addGoal($iAIAgentType_Bandit, "doRest");
$iAIGoalLibrary.addSolution($iAIAgentType_Bandit, "doRest", "relax");
$iAIGoalLibrary.addSolution($iAIAgentType_Bandit, "doRest", "sleep");

$iAIGoalLibrary.addGoal($iAIAgentType_Entertainer, "doRest");
$iAIGoalLibrary.addSolution($iAIAgentType_Entertainer, "doRest", "relax");
$iAIGoalLibrary.addSolution($iAIAgentType_Entertainer, "doRest", "sleep");

$iAIGoalLibrary.addGoal($iAIAgentType_Soldier, "doRest");
$iAIGoalLibrary.addSolution($iAIAgentType_Soldier, "doRest", "relax");
$iAIGoalLibrary.addSolution($iAIAgentType_Soldier, "doRest", "sleep");

//-----------------------------------------------------------------------
/// @fn doRest_evaluate(%agent)
/// @brief Called to determine whether to assign this goal.
///
/// @param %agent Agent to determine for.
/// @returns F32 between 0.00 and 1.00
//-----------------------------------------------------------------------
function doRest_evaluate(%agent)
{
   // check if outside ranges to do rest
   if ((%agent.getFatigue() <= 33) || (%agent.getHealth() >= 50))
   {
      return 0.0;
   }

   // calculate the weightings of each parameter
   %fatigueWeight = (%agent.getFatigue() - 33) / 33;
```

```
   %healthWeight = (50 - %agent.getHealth()) / 50;

   // average the weightings (div 10 to allow for futher weightings)
   %avgWeight = ((%fatigueWeight + %healthWeight) / 2) / 10;

   // multiply by agent weight for this goal
   switch$(%agent.getAgentType())
   {
      case $iAIAgentType_Entertainer: return (%avgWeight * 2);
      case $iAIAgentType_Bandit: return (%avgWeight * 1);
      case $iAIAgentType_Soldier: return (%avgWeight * 3);
      default : return 0.0;
   }
      }
```

20.6.1 relax.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: Goal :: doRest :: Solution :: relax
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
/// @fn relax_evaluate(%agent)
/// @brief Called to determine whether to assign this solution.
///
/// @param %agent Agent to determine for.
/// @returns F32 between 0.00 and 1.00
//-----------------------------------------------------------------------
function relax_evaluate(%agent)
{
   // check if outside ranges to relax
   if ((%agent.getFatigue() <= 33))
   {
      return 0.0;
   }

   // calculate the weightings of each parameter
   %fatigueWeight = (%agent.getFatigue() - 33) / 33;

   // average the weightings (div 10 to allow for futher weightings)
   %avgWeight = ((%fatigueWeight) / 1) / 10;

   // multiply by agent weight for this goal
   switch$(%agent.getAgentType())
   {
      case $iAIAgentType_Entertainer: return (%avgWeight * 5);
      case $iAIAgentType_Bandit: return (%avgWeight * 4);
      case $iAIAgentType_Soldier: return (%avgWeight * 6);
      default : return 0.0;
   }
}

//-----------------------------------------------------------------------
/// @fn relax_onEnter(%agent)
/// @brief Called on entry to this solution state.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function relax_onEnter(%agent)
{
   iAIMessage("STATE: relax being executed by " @ %agent);
   relax_execute(%agent);
```

```
}

//-----------------------------------------------------------------------
/// @fn relax_execute(%agent)
/// @brief Main execution state for the agent, called from onEnter.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function relax_execute(%agent)
{
   // delete any current path
   if (isObject(%agent.getCurrentPath()))
      %agent.getCurrentPath().delete();

   // stop moving
   %agent.stop();

   // a nice relax animation
   %agent.setActionThread("Death8", true);

   // relax for half a minute
   schedule(30000, 0, "relax_onExit", %agent);
}

//-----------------------------------------------------------------------
/// @fn relax_onReachDestination(%agent)
/// @brief Callback from onReachDestination. When agent reaches
///        end point on path, function is called.
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
function relax_onReachDestination(%agent)
{
   //
}

//-----------------------------------------------------------------------
/// @fn relax_onCheckVitals(%agent)
/// @brief Callback from checkVitals.
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
function relax_onCheckVitals(%agent)
{
   %agent.setFatigue(%agent.getFatigue() - ($IAIAGENT_VITALS_FATIGUE*2));
   %agent.setBoredom(%agent.getBoredom() - ($IAIAGENT_VITALS_BOREDOM*2));
}

//-----------------------------------------------------------------------
/// @fn relax_onExit(%agent)
/// @brief Called on exiting from this solution, called from execute.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function relax_onExit(%agent)
{
   // cancel the relax animation
   %agent.setActionThread("look");
   %agent.setFatigue(%agent.getFatigue() - 50);
   %agent.setBoredom(%agent.getBoredom() - 50);
   %agent.setHappiness(%agent.getHappiness() + 15);

   iAIGoalManager.completedSolution(%agent);
```

```
      }
```

20.6.2 sleep.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: Goal :: doRest :: Solution :: sleep
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
/// @fn sleep_evaluate(%agent)
/// @brief Called to determine whether to assign this solution.
///
/// @param %agent Agent to determine for.
/// @returns F32 between 0.00 and 1.00
//-----------------------------------------------------------------------
function sleep_evaluate(%agent)
{
   // check if outside ranges to do sleep
   if ((%agent.getFatigue() <= 66) || (%agent.getBoredom() <= 50))
   {
      return 0.0;
   }

   // calculate the weightings of each parameter
   %fatigueWeight = (%agent.getFatigue() - 66) / 66;
   %boredomWeight = (%agent.getBoredom() - 50) / 50;

   // average the weightings (div 10 to allow for futher weightings)
   %avgWeight = ((%fatigueWeight + %boredomWeight) / 2) / 10;

   // multiply by agent weight for this goal
   switch$(%agent.getAgentType())
   {
      case $iAIAgentType_Entertainer: return (%avgWeight * 5);
      case $iAIAgentType_Bandit: return (%avgWeight * 6);
      case $iAIAgentType_Soldier: return (%avgWeight * 4);
      default : return 0.0;
   }
}

//-----------------------------------------------------------------------
/// @fn sleep_onEnter(%agent)
/// @brief Called on entry to this solution state.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function sleep_onEnter(%agent)
{
   iAIMessage("STATE: sleep being executed by " @ %agent);

   // generate path to the home
   %agent.generatePath($HomeLocation);
   %agent.setMoveDestination(%agent.getCurrentPath().nextPosition());
   %agent.atHouse = false;
   sleep_execute(%agent);
}

//-----------------------------------------------------------------------
/// @fn sleep_execute(%agent)
/// @brief Main execution state for the agent, called from onEnter.
///
/// @param %agent Agent state is assigned to.
```

```
//-----------------------------------------------------------------------
function sleep_execute(%agent)
{
   // check if at the house yet
   if (%agent.atHouse == false)
      return;

   // agent is at the house, so sleep
   %agent.setActionThread("Death8", true);

   // sleep for half a minute
   schedule(30000, 0, "sleep_onExit", %agent);
}

//-----------------------------------------------------------------------
/// @fn sleep_onReachDestination(%agent)
/// @brief Callback from onReachDestination. When agent reaches
///        end point on path, function is called.
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
function sleep_onReachDestination(%agent)
{
   // check if still on a path
   // arrived at the house
   %agent.atHouse = true;
   sleep_execute(%agent);
}

//-----------------------------------------------------------------------
/// @fn sleep_onCheckVitals(%agent)
/// @brief Callback from checkVitals.
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
function sleep_onCheckVitals(%agent)
{
   // check if at the house
   if (%agent.atHouse == true)
   {
      %agent.setFatigue(%agent.getFatigue() -
($IAIAGENT_VITALS_FATIGUE*4));
      %agent.setBoredom(%agent.getBoredom() -
($IAIAGENT_VITALS_BOREDOM*4));
   }
}

//-----------------------------------------------------------------------
/// @fn sleep_oExit(%agent)
/// @brief Called on exiting from this solution, called from execute.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function sleep_onExit(%agent)
{
   %agent.atHouse = false;
   %agent.setActionThread("look");
   %agent.setHealth(%agent.getHealth() + 15);
   %agent.setFatigue(0);
   %agent.setBoredom(0);
   iAIGoalManager.completedSolution(%agent);
      }
```

20.7 getFood.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: Goal :: getFood
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
// Load all the solution files
//-----------------------------------------------------------------------
exec("./buyFood.cs");
exec("./seekFood.cs");
exec("./stealFood.cs");

//-----------------------------------------------------------------------
// Add goal and solutions to library for each agent
//-----------------------------------------------------------------------
$iAIGoalLibrary.addGoal($iAIAgentType_Bandit, "getFood");
$iAIGoalLibrary.addSolution($iAIAgentType_Bandit, "getFood", "seekFood");
$iAIGoalLibrary.addSolution($iAIAgentType_Bandit, "getFood", "stealFood");

$iAIGoalLibrary.addGoal($iAIAgentType_Entertainer, "getFood");
$iAIGoalLibrary.addSolution($iAIAgentType_Entertainer, "getFood",
"buyFood");
$iAIGoalLibrary.addSolution($iAIAgentType_Entertainer, "getFood",
"stealFood");

$iAIGoalLibrary.addGoal($iAIAgentType_Soldier, "getFood");
$iAIGoalLibrary.addSolution($iAIAgentType_Soldier, "getFood", "buyFood");
$iAIGoalLibrary.addSolution($iAIAgentType_Soldier, "getFood", "seekFood");

//-----------------------------------------------------------------------
/// @fn getFood_evaluate(%agent)
/// @brief Called to determine whether to assign this goal.
///
/// @param %agent Agent to determine for.
/// @returns F32 between 0.00 and 1.00
//-----------------------------------------------------------------------
function getFood_evaluate(%agent)
{
   // check if outside ranges to get food
   if ((%agent.getHealth() >= 80) || (%agent.getHappiness() >= 80))
   {
      return 0.0;
   }

   // calculate the weightings of each parameter
   %healthWeight = (80 - %agent.getHealth()) / 50;
   %happinessWeight = (80 - %agent.getHappiness()) / 80;

   // average the weightings (div 10 to allow for futher weightings)
   %avgWeight = ((%healthWeight + %happinessWeight) / 2) / 10;

   // multiply by agent weight for this goal
   switch$(%agent.getAgentType())
   {
      case $iAIAgentType_Bandit: return (%avgWeight * 2);
      case $iAIAgentType_Entertainer: return (%avgWeight * 1);
      case $iAIAgentType_Soldier: return (%avgWeight * 1);
      default : return 0.0;
   }
      }
```

20.7.1 buyFood.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: Goal :: getFood :: Solution :: buyFood
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

$FoodVendorLocation = "420.406 329.79 218.504";

//-----------------------------------------------------------------------
/// @fn buyFood_evaluate(%agent)
/// @brief Called to determine whether to assign this solution.
///
/// @param %agent Agent to determine for.
/// @returns F32 between 0.00 and 1.00
//-----------------------------------------------------------------------
function buyFood_evaluate(%agent)
{
   // check if outside ranges to buy food
   if (%agent.getMoney() <= 100)
   {
      return 0.0;
   }

   // calculate the weightings of each parameter
   %moneyWeight = (%agent.getMoney() - 100) / 100;

   // average the weightings (div 10 to allow for futher weightings)
   %avgWeight = ((%moneyWeight) / 1) / 10;

   // multiply by agent weight for this goal
   switch$(%agent.getAgentType())
   {
      case $iAIAgentType_Entertainer: return (%avgWeight * 7);
      case $iAIAgentType_Soldier: return (%avgWeight * 5);
      default : return 0.0;
   }
}

//-----------------------------------------------------------------------
/// @fn buyFood_onEnter(%agent)
/// @brief Called on entry to this solution state.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function buyFood_onEnter(%agent)
{
   iAIMessage("STATE: BuyFood being executed by " @ %agent);

   // generate a path to the vendor
   %agent.generatePath($FoodVendorLocation);
   %agent.setMoveDestination(%agent.getCurrentPath().nextPosition());
   %agent.atFoodVendor = false;

   buyFood_execute(%agent);
}

//-----------------------------------------------------------------------
/// @fn buyFood_execute(%agent)
/// @brief Main execution state for the agent, called from onEnter.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
```

```
function buyFood_execute(%agent)
{
   // check if at the health vendor yet
   if (%agent.atFoodVendor == false)
      return;

   // agent is at the vendor wait a bit
   // then buy it!
   schedule(5000, 0, "buyFood_onExit", %agent);
}

//-----------------------------------------------------------------------
/// @fn buyFood_onReachDestination(%agent)
/// @brief Callback from onReachDestination. When agent reaches
///        end point on path, function is called.
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
function buyFood_onReachDestination(%agent)
{
   // arrived at the food vendor
   %agent.atFoodVendor = true;

   // execute the buying of food
   buyFood_execute(%agent);
}

//-----------------------------------------------------------------------
/// @fn buyFood_onCheckVitals(%agent)
/// @brief Callback from checkVitals.
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
function buyFood_onCheckVitals(%agent)
{
    // check if at the food Vendor
   if (%agent.atFoodVendor == true)
   {
      //do nothing
   }
}

//-----------------------------------------------------------------------
/// @fn buyFood_onExit(%agent)
/// @brief Called on exiting from this solution, called from execute.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function buyFood_onExit(%agent)
{
   %agent.atFoodVendor = false;

   %agent.setMoney(%agent.getMoney()-50);
   // adding item...
   %agent.incInventory("FoodKit",1);
   // use item...
   %agent.use("FoodKit");
   // finished so request new goal
   iAIGoalManager.completedSolution(%agent);
      }
```

20.7.2 seekFood.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: Goal :: getFood :: Solution :: seekFood
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
/// @fn seekFood_evaluate(%agent)
/// @brief Called to determine whether to assign this solution.
///
/// @param %agent Agent to determine for.
/// @returns F32 between 0.00 and 1.00
//-----------------------------------------------------------------------
function seekFood_evaluate(%agent)
{
   // check if outside ranges to seek
   if (%agent.getFatigue() >= 50)
   {
      return 0.0;
   }

   // calcaulte weightings of each parameters
   %fatigueWeight = (50 - %agent.getFatigue()) / 50;

   // average the weightings (div 10 to allow for futher weightings)
   %avgWeight = ((%fatigueWeight) / 1) / 10;

   // multiply by agent weight for this solution
   switch$(%agent.getAgentType())
   {
      case $iAIAgentType_Bandit: return (%avgWeight * 5);
      case $iAIAgentType_Soldier: return (%avgWeight * 5);
      default : return 0.0;
   }
}

//-----------------------------------------------------------------------
/// @fn seekFood_onEnter(%agent)
/// @brief Called on entry to this solution state.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function seekFood_onEnter(%agent)
{
   iAIMessage("STATE: seekFood being executed by " @ %agent);

   seekFood_execute(%agent);
}

//-----------------------------------------------------------------------
/// @fn seekFood_execute(%agent)
/// @brief Main execution state for the agent, called from onEnter.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function seekFood_execute(%agent)
{
   // seek for the health patch
   %agent.seekObject("FoodItem", $TypeMasks::ItemObjectType);
   %agent.seek_OnObject = false;
}
```

```
//--------------------------------------------------------------------
/// @fn seekFood_onReachDestination(%agent)
/// @brief Callback from onReachDestination. When agent reaches
///        end point on path, function is called.
///
/// @param %agent Agent throwing the callback.
//--------------------------------------------------------------------
function seekFood_onReachDestination(%agent)
{
   // check if reached the food
   if (%agent.seek_OnObject == true)
   {
      seekFood_onExit(%agent);
   }
}

//--------------------------------------------------------------------
/// @fn seekFood_onCheckVitals(%agent)
/// @brief Callback from checkVitals.
///
/// @param %agent Agent throwing the callback.
//--------------------------------------------------------------------
function seekFood_onCheckVitals(%agent)
{
   %agent.setFatigue(%agent.getFatigue() + ($IAIAGENT_VITALS_FATIGUE*2));
}

//--------------------------------------------------------------------
/// @fn seekFood_onExit(%agent)
/// @brief Called on exiting from this solution, called from execute.
///
/// @param %agent Agent state is assigned to.
//--------------------------------------------------------------------
function seekFood_onExit(%agent)
{
   iAIGoalManager.completedSolution(%agent);
      }
```

20.7.3 stealFood.cs

```
//--------------------------------------------------------------------
// Immersive AI :: Goal :: getFood :: Solution :: stealFood
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//--------------------------------------------------------------------

$FoodVendorLocation = "420.406 329.79 218.504";

//--------------------------------------------------------------------
/// @fn stealFood_evaluate(%agent)
/// @brief Called to determine whether to assign this solution.
///
/// @param %agent Agent to determine for.
/// @returns F32 between 0.00 and 1.00
//--------------------------------------------------------------------
function stealFood_evaluate(%agent)
{
   // check if outside ranges to steal
   if (%agent.getHappiness() >= 50)
   {
      return 0.0;
   }

   // calcaulte weightings of each parameters
```

```
   %happinessWeight = (50 - %agent.getHappiness()) / 50;

   // average the weightings (div 10 to allow for futher weightings)
   %avgWeight = ((%happinessWeight) / 1) / 10;

   // multiply by agent weight for this solution
   switch$(%agent.getAgentType())
   {
      case $iAIAgentType_Bandit: return (%avgWeight * 5);
      case $iAIAgentType_Entertainer: return (%avgWeight * 3);
      default : return 0.0;
   }
}

//-----------------------------------------------------------------------
/// @fn stealFood_onEnter(%agent)
/// @brief Called on entry to this solution state.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function stealFood_onEnter(%agent)
{
   iAIMessage("STATE: stealFood being executed by " @ %agent);

   // generate a path to the club
   %agent.generatePath($FoodVendorLocation);
   %agent.setMoveDestination(%agent.getCurrentPath().nextPosition());
   %agent.atFoodVendor = false;

   stealFood_execute(%agent);
}

//-----------------------------------------------------------------------
/// @fn stealFood_execute(%agent)
/// @brief Main execution state for the agent, called from onEnter.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function stealFood_execute(%agent)
{
   // check if at the food vendor yet
   if (%agent.atFoodVendor == false)
      return;

   // agent is at the vendor shoot and steal  from it!

   //set the agents aim
   %agent.setAimObject(FoodVendor.getId());
   //set the agent to shoot the vendor
   %agent.setImageTrigger(0,true);
   //set the agent to stop shooting
   %agent.schedule(1500, setImageTrigger, 0, false);

   schedule(5000, 0, "stealFood_onExit", %agent);
}

//-----------------------------------------------------------------------
/// @fn stealFood_onReachDestination(%agent)
/// @brief Callback from onReachDestination. When agent reaches
///        end point on path, function is called.
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
```

```
function stealFood_onReachDestination(%agent)
{
   // arrived at the food vendor
   %agent.atFoodVendor = true;

   // execute the buying of food
   stealFood_execute(%agent);
}

//-----------------------------------------------------------------------
/// @fn stealFood_onCheckVitals(%agent)
/// @brief Callback from checkVitals.
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
function stealFood_onCheckVitals(%agent)
{
   // check if at the food Vendor
   if (%agent.atFoodVendor == true)
   {
      //do nothing
   }
}

//-----------------------------------------------------------------------
/// @fn stealFood_onExit(%agent)
/// @brief Called on exiting from this solution, called from execute.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function stealFood_onExit(%agent)
{
   %agent.atFoodVendor = false;

   //clear the aim
   %agent.clearAim();
   //add the taken item...
   %agent.incInventory("FoodKit",1);
   //use item...
   %agent.use("FoodKit");
   //finished so request new goal
   iAIGoalManager.completedSolution(%agent);
      }
```

20.8 getHealth.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: Goal :: getHealth
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
// Load all the solution files
//-----------------------------------------------------------------------
exec("./buyHealth.cs");
exec("./seekHealth.cs");
exec("./stealHealth.cs");

//-----------------------------------------------------------------------
// Add goal and solutions to library for each agent
//-----------------------------------------------------------------------
$iAIGoalLibrary.addGoal($iAIAgentType_Bandit, "getHealth");
```

```
$iAIGoalLibrary.addSolution($iAIAgentType_Bandit, "getHealth",
"seekHealth");
$iAIGoalLibrary.addSolution($iAIAgentType_Bandit, "getHealth",
"stealHealth");

$iAIGoalLibrary.addGoal($iAIAgentType_Entertainer, "getHealth");
$iAIGoalLibrary.addSolution($iAIAgentType_Entertainer, "getHealth",
"buyHealth");
$iAIGoalLibrary.addSolution($iAIAgentType_Entertainer, "getHealth",
"stealHealth");

$iAIGoalLibrary.addGoal($iAIAgentType_Soldier, "getHealth");
$iAIGoalLibrary.addSolution($iAIAgentType_Soldier, "getHealth",
"buyHealth");
$iAIGoalLibrary.addSolution($iAIAgentType_Soldier, "getHealth",
"seekHealth");

//-----------------------------------------------------------------------
/// @fn getHealth_evaluate(%agent)
/// @brief Called to determine whether to assign this goal.
///
/// @param %agent Agent to determine for.
/// @returns F32 between 0.00 and 1.00
//-----------------------------------------------------------------------
function getHealth_evaluate(%agent)
{
   // check if outside ranges to get health
   if (%agent.getHealth() >= 50)
   {
      return 0.0;
   }

   // calculate the weightings of each parameter
   %healthWeight = (50 - %agent.getHealth()) / 50;

   // average the weightings (div 10 to allow for futher weightings)
   %avgWeight = (%healthWeight / 1) / 10;

   // multiply by agent weight for this goal
   switch$(%agent.getAgentType())
   {
      case $iAIAgentType_Bandit: return (%avgWeight * 1);
      case $iAIAgentType_Entertainer: return (%avgWeight * 1);
      case $iAIAgentType_Soldier: return (%avgWeight * 1);
      default : return 0.0;
   }
      }
```

20.8.1 buyHealth.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: Goal :: getHealth :: Solution :: buyHealth
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

$HealthVendorLocation = "323.09 347.599 218.99";

//-----------------------------------------------------------------------
/// @fn buyHealth_evaluate(%agent)
/// @brief Called to determine whether to assign this solution.
///
/// @param %agent Agent to determine for.
/// @returns F32 between 0.00 and 1.00
```

```
//-----------------------------------------------------------------------
function buyHealth_evaluate(%agent)
{
   // check if outside ranges to buy health
   if (%agent.getMoney() < 100)
   {
      return 0.0;
   }

   // calculate the weightings of each parameter
   %moneyWeight = (%agent.getMoney() - 100) / 100;

   // average the weightings (div 10 to allow for futher weightings)
   %avgWeight = ((%moneyWeight) / 1) / 10;

   // multiply by agent weight for this goal
   switch$(%agent.getAgentType())
   {
      case $iAIAgentType_Entertainer: return (%avgWeight * 7);
      case $iAIAgentType_Soldier: return (%avgWeight * 5);
      default : return 0.0;
   }
}

//-----------------------------------------------------------------------
/// @fn buyHealth_onEnter(%agent)
/// @brief Called on entry to this solution state.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function buyHealth_onEnter(%agent)
{
   iAIMessage("STATE: buyHealth being executed by " @ %agent);

   // generate a path to the club
   %agent.generatePath($HealthVendorLocation);
   %agent.setMoveDestination(%agent.getCurrentPath().nextPosition());
   %agent.atVendor = false;

   buyHealth_execute(%agent);
}

//-----------------------------------------------------------------------
/// @fn buyHealth_execute(%agent)
/// @brief Main execution state for the agent, called from onEnter.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function buyHealth_execute(%agent)
{
   // check if at the health vendor yet
   if (%agent.atVendor == false)
      return;

   // agent is at the vendor wait a bit
   // then buy it!
   schedule(5000, 0, "buyHealth_onExit", %agent);
}

//-----------------------------------------------------------------------
/// @fn buyHealth_onReachDestination(%agent)
/// @brief Callback from onReachDestination. When agent reaches
///        end point on path, function is called.
```

```
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
function buyHealth_onReachDestination(%agent)
{
   // arrived at the health vendor
   %agent.atVendor = true;

   // execute the buying of health
   buyHealth_execute(%agent);
}

//-----------------------------------------------------------------------
/// @fn buyHealth_onCheckVitals(%agent)
/// @brief Callback from checkVitals.
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
function buyHealth_onCheckVitals(%agent)
{
   // check if at the health Vendor
   if (%agent.atVendor == true)
   {
      //do nothing for now
   }
}

//-----------------------------------------------------------------------
/// @fn buyHealth_onExit(%agent)
/// @brief Called on exiting from this solution, called from execute.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function buyHealth_onExit(%agent)
{
   %agent.atVendor = false;

   //removing monetary cost
   %agent.setMoney(%agent.getMoney() - 60);

   //adding item...
   %agent.incInventory("HealthKit",1);

   //use item...
   %agent.use("HealthKit");

   //finished so request new goal
   iAIGoalManager.completedSolution(%agent);
      }
```

20.8.2 seekHealth.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: Goal :: getHealth :: Solution :: seekHealth
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
/// @fn seekHealth_evaluate(%agent)
/// @brief Called to determine whether to assign this solution.
///
/// @param %agent Agent to determine for.
/// @returns F32 between 0.00 and 1.00
```

```
//-----------------------------------------------------------------
function seekHealth_evaluate(%agent)
{
   // check if outside ranges to seek
   if (%agent.getFatigue() >= 50)
   {
      return 0.0;
   }

   // calcaulte weightings of each parameters
   %fatigueWeight = (50 - %agent.getFatigue()) / 50;

   // average the weightings (div 10 to allow for futher weightings)
   %avgWeight = ((%fatigueWeight) / 1) / 10;

   // multiply by agent weight for this solution
   switch$(%agent.getAgentType())
   {
      case $iAIAgentType_Bandit: return (%avgWeight * 5);
      case $iAIAgentType_Soldier: return (%avgWeight * 5);
      default : return 0.0;
   }
}

//-----------------------------------------------------------------
/// @fn seekHealth_onEnter(%agent)
/// @brief Called on entry to this solution state.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------
function seekHealth_onEnter(%agent)
{
   iAIMessage("STATE: seekHealth being executed by " @ %agent);

   seekHealth_execute(%agent);
}

//-----------------------------------------------------------------
/// @fn seekHealth_execute(%agent)
/// @brief Main execution state for the agent, called from onEnter.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------
function seekHealth_execute(%agent)
{
   // seek for the health patch
   %agent.seekObject("HealthPatch", $TypeMasks::ItemObjectType);
   %agent.reachedHealthPatch = false;
}

//-----------------------------------------------------------------
/// @fn seekHealth_onReachDestination(%agent)
/// @brief Callback from onReachDestination. When agent reaches
///        end point on path, function is called.
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------
function seekHealth_onReachDestination(%agent)
{
   // check if reached health
   if (%agent.reachedHealthPatch == true)
   {
      seekHealth_onExit(%agent);
```

```
   }
}

//--------------------------------------------------------------------
/// @fn seekHealth_onCheckVitals(%agent)
/// @brief Callback from checkVitals.
///
/// @param %agent Agent throwing the callback.
//--------------------------------------------------------------------
function seekHealth_onCheckVitals(%agent)
{
   %agent.setFatigue(%agent.getFatigue() + ($IAIAGENT_VITALS_FATIGUE*2));
}

//--------------------------------------------------------------------
/// @fn seekHealth_onExit(%agent)
/// @brief Called on exiting from this solution, called from execute.
///
/// @param %agent Agent state is assigned to.
//--------------------------------------------------------------------
function seekHealth_onExit(%agent)
{
   iAIGoalManager.completedSolution(%agent);
      }
```

20.8.3 stealHealth.cs

```
//--------------------------------------------------------------------
// Immersive AI :: Goal :: getHealth :: Solution :: stealHealth
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//--------------------------------------------------------------------

$HealthVendorLocation = "323.09 347.599 218.99";

//--------------------------------------------------------------------
/// @fn stealHealth_evaluate(%agent)
/// @brief Called to determine whether to assign this solution.
///
/// @param %agent Agent to determine for.
/// @returns F32 between 0.00 and 1.00
//--------------------------------------------------------------------
function stealHealth_evaluate(%agent)
{
   // check if outside ranges to steal
   if (%agent.getHappiness() >= 50)
   {
      return 0.0;
   }

   // calcaulte weightings of each parameters
   %happinessWeight = (50 - %agent.getHappiness()) / 50;

   // average the weightings (div 10 to allow for futher weightings)
   %avgWeight = ((%happinessWeight) / 1) / 10;

   // multiply by agent weight for this solution
   switch$(%agent.getAgentType())
   {
      case $iAIAgentType_Bandit: return (%avgWeight * 5);
      case $iAIAgentType_Entertainer: return (%avgWeight * 3);
      default : return 0.0;
   }
}
```

```
//-----------------------------------------------------------------------
/// @fn stealHealth_onEnter(%agent)
/// @brief Called on entry to this solution state.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function stealHealth_onEnter(%agent)
{
   iAIMessage("STATE: stealHealth being executed by " @ %agent);

   // generate a path to the club
   %agent.generatePath($HealthVendorLocation);
   %agent.setMoveDestination(%agent.getCurrentPath().nextPosition());
   %agent.atHealthVendor = false;

   stealHealth_execute(%agent);
}

//-----------------------------------------------------------------------
/// @fn stealHealth_execute(%agent)
/// @brief Main execution state for the agent, called from onEnter.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function stealHealth_execute(%agent)
{
   // check if at the health vendor yet
   if (%agent.atHealthVendor == false)
      return;

   // agent is at the vendor shoot and steal  from it!

   //set the agents aim
   %agent.setAimObject(HealthVendor.getId());

   //set the agent to shoot the vendor
   %agent.setImageTrigger(0, true);

   //set the agent to stop shooting
   %agent.schedule(1500, setImageTrigger, 0, false);

   schedule(5000, 0, "stealHealth_onExit", %agent);
}

//-----------------------------------------------------------------------
/// @fn stealHealth_onReachDestination(%agent)
/// @brief Callback from onReachDestination. When agent reaches
///        end point on path, function is called.
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
function stealHealth_onReachDestination(%agent)
{
   // arrived at the health vendor
   %agent.atHealthVendor = true;

   // execute the buying of food
   stealHealth_execute(%agent);
}

//-----------------------------------------------------------------------
/// @fn stealHealth_onCheckVitals(%agent)
```

```
/// @brief Callback from checkVitals.
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
function stealHealth_onCheckVitals(%agent)
{
   // check if at the health Vendor
   if (%agent.atHealthVendor == true)
   {
      //do nothing
   }
}

//-----------------------------------------------------------------------
/// @fn stealHealth_onExit(%agent)
/// @brief Called on exiting from this solution, called from execute.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function stealHealth_onExit(%agent)
{
   %agent.atHealthVendor = false;

   //clear the aim
   %agent.clearAim();

   //add the taken item...
   %agent.incInventory("HealthKit",1);

   //use item...
   %agent.use("HealthKit");

   //finished so request new goal
   iAIGoalManager.completedSolution(%agent);
      }
```

20.9 goHome.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: Goal :: goHome
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
// Load all the solution files
//-----------------------------------------------------------------------
exec("./seekHome.cs");

//-----------------------------------------------------------------------
// Add goal and solutions to library for each agent
//-----------------------------------------------------------------------
$iAIGoalLibrary.addGoal($iAIAgentType_Bandit, "goHome");
$iAIGoalLibrary.addSolution($iAIAgentType_Bandit, "goHome", "seekHome");

$iAIGoalLibrary.addGoal($iAIAgentType_Entertainer, "goHome");
$iAIGoalLibrary.addSolution($iAIAgentType_Entertainer, "goHome",
"seekHome");

$iAIGoalLibrary.addGoal($iAIAgentType_Soldier, "goHome");
$iAIGoalLibrary.addSolution($iAIAgentType_Soldier, "goHome", "seekHome");

//-----------------------------------------------------------------------
/// @fn goHome_evaluate(%agent)
```

```
/// @brief Called to determine whether to assign this goal.
///
/// @param %agent Agent to determine for.
/// @returns F32 between 0.00 and 1.00
//-----------------------------------------------------------------------
function goHome_evaluate(%agent)
{
   // check if outside ranges to go home
   if ((%agent.getHealth() >= 66) || (%agent.getFatigue() <= 33) ||
      (%agent.getMoney() >= 200) || (%agent.getBoredom() <= 33))
   {
      return (getRandom(0, 2) / 100);
   }

   // calculate the weightings of each parameter
   %healthWeight = (66 - %agent.getHealth()) / 66;
   %fatigueWeight = (%agent.getFatigue() - 33) / 33;
   %moneyWeight = (%agent.getMoney() - 200) / 200;
   %boredomWeight = (%agent.getBoredom() - 33) / 33;

   // average the weightings (div 10 to allow for futher weightings)
   %avgWeight = ((%healthWeight + %fatigueWeight + %moneyWeight +
%boredomWeight) / 4) / 10;

   // multiply by agent weight for this goal
   switch$(%agent.getAgentType())
   {
      case $iAIAgentType_Entertainer: return (%avgWeight * 2);
      case $iAIAgentType_Bandit: return (%avgWeight * 1);
      case $iAIAgentType_Soldier: return (%avgWeight * 1);
      default : return 0.0;
   }
      }
```

20.9.1 seekHome.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: Goal :: goHome :: Solution :: seekHome
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

$HomeLocation = "120 280 250";

//-----------------------------------------------------------------------
/// @fn seekHome_evaluate(%agent)
/// @brief Called to determine whether to assign this solution.
///
/// @param %agent Agent to determine for.
/// @returns F32 between 0.00 and 1.00
//-----------------------------------------------------------------------
function seekHome_evaluate(%agent)
{
   // check if outside ranges to go home
   if (%agent.getHealth() >= 66)
   {
      return 0.0;
   }

   // calculate the weightings of each parameter
   %healthWeight = (66 - %agent.getHealth()) / 66;

   // average the weightings (div 10 to allow for futher weightings)
   %avgWeight = ((%healthWeight) / 1) / 10;
```

```
   // multiply by agent weight for this goal
   switch$(%agent.getAgentType())
   {
      case $iAIAgentType_Entertainer: return (%avgWeight * 10);
      case $iAIAgentType_Bandit: return (%avgWeight * 10);
      case $iAIAgentType_Soldier: return (%avgWeight * 10);
      default : return 0.0;
   }
}

//-----------------------------------------------------------------------
/// @fn seekHome_onEnter(%agent)
/// @brief Called on entry to this solution state.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function seekHome_onEnter(%agent)
{
   iAIMessage("STATE: seekHome being executed by " @ %agent);

   // generate a path to home
   %agent.generatePath($HomeLocation);
   %agent.setMoveDestination(%agent.getCurrentPath().nextPosition());
   %agent.atHome = false;

   seekHome_execute(%agent);
}

//-----------------------------------------------------------------------
/// @fn seekHome_execute(%agent)
/// @brief Main execution state for the agent, called from onEnter.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function seekHome_execute(%agent)
{
   // check if home yet
   if (%agent.atHome == false)
      return;

   // agent home... refresh stats
   %agent.setHealth(100);
   %agent.setMoney(%agent.getMoney() + 200);

   schedule(10000, 0, "seekHome_onExit", %agent);
}

//-----------------------------------------------------------------------
/// @fn seekHome_onReachDestination(%agent)
/// @brief Callback from onReachDestination. When agent reaches
///        end point on path, function is called.
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
function seekHome_onReachDestination(%agent)
{
   // arrived home
   %agent.atHome = true;

   // execute home functions
   seekHome_execute(%agent);
}
```

```
//-----------------------------------------------------------------------
/// @fn seekHome_onCheckVitals(%agent)
/// @brief Callback from checkVitals.
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
function seekHome_onCheckVitals(%agent)
{
   // check if home
   if (%agent.atHome == true)
   {
      %agent.setFatigue(%agent.getFatigue() -
($IAIAGENT_VITALS_FATIGUE*1.5));
      %agent.setBoredom(%agent.getBoredom() -
($IAIAGENT_VITALS_BOREDOM*1.5));
   }
}

//-----------------------------------------------------------------------
/// @fn seekHome_onExit(%agent)
/// @brief Called on exiting from this solution, called from execute.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function seekHome_onExit(%agent)
{
   %agent.atHome = false;
   iAIGoalManager.completedSolution(%agent);
      }
```

20.10 haveFun.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: Goal :: haveFun
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

//-----------------------------------------------------------------------
// Load all the solution files
//-----------------------------------------------------------------------
exec("./dance.cs");

//-----------------------------------------------------------------------
// Add goal and solutions to library for each agent
//-----------------------------------------------------------------------
$iAIGoalLibrary.addGoal($iAIAgentType_Bandit, "haveFun");
$iAIGoalLibrary.addSolution($iAIAgentType_Bandit, "haveFun", "dance");

$iAIGoalLibrary.addGoal($iAIAgentType_Entertainer, "haveFun");
$iAIGoalLibrary.addSolution($iAIAgentType_Entertainer, "haveFun", "dance");

//-----------------------------------------------------------------------
/// @fn haveFun_evaluate(%agent)
/// @brief Called to determine whether to assign this goal.
///
/// @param %agent Agent to determine for.
/// @returns F32 between 0.00 and 1.00
//-----------------------------------------------------------------------
function haveFun_evaluate(%agent)
{
   // check if outside ranges to have fun
   if ((%agent.getFatigue() <= 33) || (%agent.getHappiness() >= 33) ||
```

```
      (%agent.getBoredom() <= 33))
   {
      return (getRandom(0, 2) / 100);
   }

   // calculate the weightings of each parameter
   %fatigueWeight = (%agent.getFatigue() - 33) / 33;
   %happinessWeight = (33 - %agent.getHappiness()) / 33;
   %boredomWeight = (%agent.getBoredom() - 33) / 33;

   // average the weightings (div 10 to allow for futher weightings)
   %avgWeight = ((%fatigueWeight + %happinessWeight + %boredomWeight) / 3)
/ 10;

   // multiply by agent weight for this goal
   switch$(%agent.getAgentType())
   {
      case $iAIAgentType_Entertainer: return (%avgWeight * 3);
      case $iAIAgentType_Bandit: return (%avgWeight * 1);
      default : return 0.0;
   }
}
```

20.10.1 dance.cs

```
//-----------------------------------------------------------------------
// Immersive AI :: Goal :: haveFun :: Solution :: dance
// Copyright (c) 2006 Gavin Bunney & Tom Romano
//-----------------------------------------------------------------------

$ClubLocation = "405 331 200";

//-----------------------------------------------------------------------
/// @fn dance_evaluate(%agent)
/// @brief Called to determine whether to assign this solution.
///
/// @param %agent Agent to determine for.
/// @returns F32 between 0.00 and 1.00
//-----------------------------------------------------------------------
function dance_evaluate(%agent)
{
   // check if outside ranges to dance
   if (%agent.getFatigue() <= 33)
   {
      return 0.0;
   }

   // calculate the weightings of each parameter
   %fatigueWeight = (%agent.getFatigue() - 33) / 33;

   // average the weightings (div 10 to allow for futher weightings)
   %avgWeight = ((%fatigueWeight) / 1) / 10;

   // multiply by agent weight for this goal
   switch$(%agent.getAgentType())
   {
      case $iAIAgentType_Entertainer: return (%avgWeight * 10);
      case $iAIAgentType_Bandit: return (%avgWeight * 10);
      default : return 0.0;
   }
}
```

```
//-----------------------------------------------------------------------
/// @fn dance_onEnter(%agent)
/// @brief Called on entry to this solution state.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function dance_onEnter(%agent)
{
   iAIMessage("STATE: dance being executed by " @ %agent);

   // generate a path to the club
   %agent.setActionThread("look");
   %agent.generatePath($ClubLocation);
   %agent.setMoveDestination(%agent.getCurrentPath().nextPosition());
   %agent.atClub = false;

   dance_execute(%agent);
}

//-----------------------------------------------------------------------
/// @fn dance_execute(%agent)
/// @brief Main execution state for the agent, called from onEnter.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function dance_execute(%agent)
{
   // check if at the club yet
   if (%agent.atClub == false)
      return;

   // agent is at the club
   // dance baby, dance!
   %agent.setActionThread("dance");

   schedule(10000, 0, "dance_onExit", %agent);
   %agent.schedule(9000, setActionThread, "look");
}

//-----------------------------------------------------------------------
/// @fn dance_onReachDestination(%agent)
/// @brief Callback from onReachDestination. When agent reaches
///        end point on path, function is called.
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
function dance_onReachDestination(%agent)
{
   // arrived at the club
   %agent.atClub = true;

   // execute the dancing
   dance_execute(%agent);
}

//-----------------------------------------------------------------------
/// @fn dance_onCheckVitals(%agent)
/// @brief Callback from checkVitals.
///
/// @param %agent Agent throwing the callback.
//-----------------------------------------------------------------------
function dance_onCheckVitals(%agent)
```

```
{
   // check if at the club
   if (%agent.atClub == true)
   {
      %agent.setFatigue(%agent.getFatigue() + ($IAIAGENT_VITALS_FATIGUE *
2));
      %agent.setHappiness(%agent.getHappiness() -
($IAIAGENT_VITALS_HAPPINESS * 2));
   }
}

//-----------------------------------------------------------------------
/// @fn dance_onExit(%agent)
/// @brief Called on exiting from this solution, called from execute.
///
/// @param %agent Agent state is assigned to.
//-----------------------------------------------------------------------
function dance_onExit(%agent)
{
   %agent.atClub = false;

   %agent.setHappiness(%agent.getHappiness() + 33);
   %agent.setFatigue(%agent.getFatigue() - 33);
   %agent.setBoredom(%agent.getBoredom() - 33);

   // stop dancing
   %agent.setActionThread("look");
   iAIGoalManager.completedSolution(%agent);
}
```