UnityCAPIR – ReadMe

# Short introduction

UnityCAPIR is delivered as the code base for CAPIR-powered games designed using Unity. For the rest of this document, basic knowledge of how to create a game with Unity and use C# scripts to specify the AI of the game is assumed; we are using C# instead of UnityScript (JavaScript) in our project.

* To download Unity, go to <http://unity3d.com/unity/download/>. UnityCAPIR should be compatible with the latest version of Unity (3.5.\*).
* To learn how to create games with Unity, visit <http://unity3d.com/support/documentation/>
  + The Basics section is extremely useful and is highly recommended.
* To learn how to script in Unity, visit <http://unity3d.com/support/documentation/ScriptReference/index.html>

As mentioned in CAPIR\_userman.doc, the process of designing a game using CAPIR engine is as follows:

1. Specify the game dynamics in C++ code. The aspects of the game that need extending can be referred from CAPIRSolver\_techdoc.
2. Specify the game dynamics in Unity code.
   1. Each NPC must routinely update a universal game state stored in the main game script, which will be used by CapirBox to determine what is the best cooperative action to recommend the AI-controlled helper.
3. Attach the C# scripts to respective character Prefabs.
4. While in game, whenever the helper needs an advice on what to do, it queries the CapirBox for the best helper action.
   1. This means game designers can mix and match the AI techniques; there is no need to rely entirely on Capir engine.

# Routines to extend for a new game

The following scripts make up the CAPIR engine in Unity.

1. MainCSharp (MainCSharp.cs): The main game script, which should be placed in a GameObject on the stage. This script contains information about the map, the game state and is responsible to initialize other GameObjects such as the characters, NPCs and objects in the game.
2. AgentScript: The script for the human-controlled character, namely the ***Agent***.
3. HelperScript: The script for the AI-controlled character, namely the ***Helper***.
4. NPCBehavior: The base script for NPCs’ behavior.

Following is a list of routines that need adjustment when creating new games. The rest of the code can be reused as is from SheepSaviorUnity, up to a change of constants related to NPC type (Sheep, Ghost, Fiery).

1. CapirBox: The main AI code.
   1. void readTMXFile(string filename): Initializes level-specific data in MainCSharp. In SheepSaviorUnity, the routine sets up MainCSharp.levelData (of type MapData) which contains data on the map layout and initial game states.
   2. int realToVirtualFunction(State state, int npcIndex): Extracts the virtual state of the subgoal indexed by ***npcIndex*** and returns its integer representation.
      1. If the NPC has no other property than its position, e.g. Sheep, the formula is:   
         F = (PA\*NF + PH)\*NF + PNPC with
         1. PA being the position of the Agent, denoted by the free grid square index of the Agent’s current position.
         2. PH being the position of the Helper.
         3. PNPC being the position of the Helper.
         4. NF is the number of free grid squares.
      2. Otherwise, the formula is F \* NumProperties + Property …
2. NPCBehavior: In order to keep the NPC’s dynamics as close as possible to the C++ code, it is advisable (but not obligatory) to extend the following routines of NPCBehavior script (the script for each NPC can be rewritten entirely, as long as it matches with the behavior implemented in CAPIRSolver C++ code.)
   1. void UpdateSolved(): Specifies when the NPC is considered solved, i.e. there is no need to consider the NPC in selecting actions, e.g. ghost dead, sheep herded.
   2. void ActionSeeingNone(): Sets action for the NPC when seeing no player around.
   3. void ActionSeeingHelper(): Sets action for the NPC when seeing the AI-controlled player.
   4. void ActionSeeingAgent(): Sets action for the NPC when seeing the human-controlled player.
   5. void ActionSeeingBoth(): Sets action for the NPC when seeing both players.

For more information, please refer to the provided API of UnityCAPIR.