Graphical Interface for Constructing Parametrized Action Representations

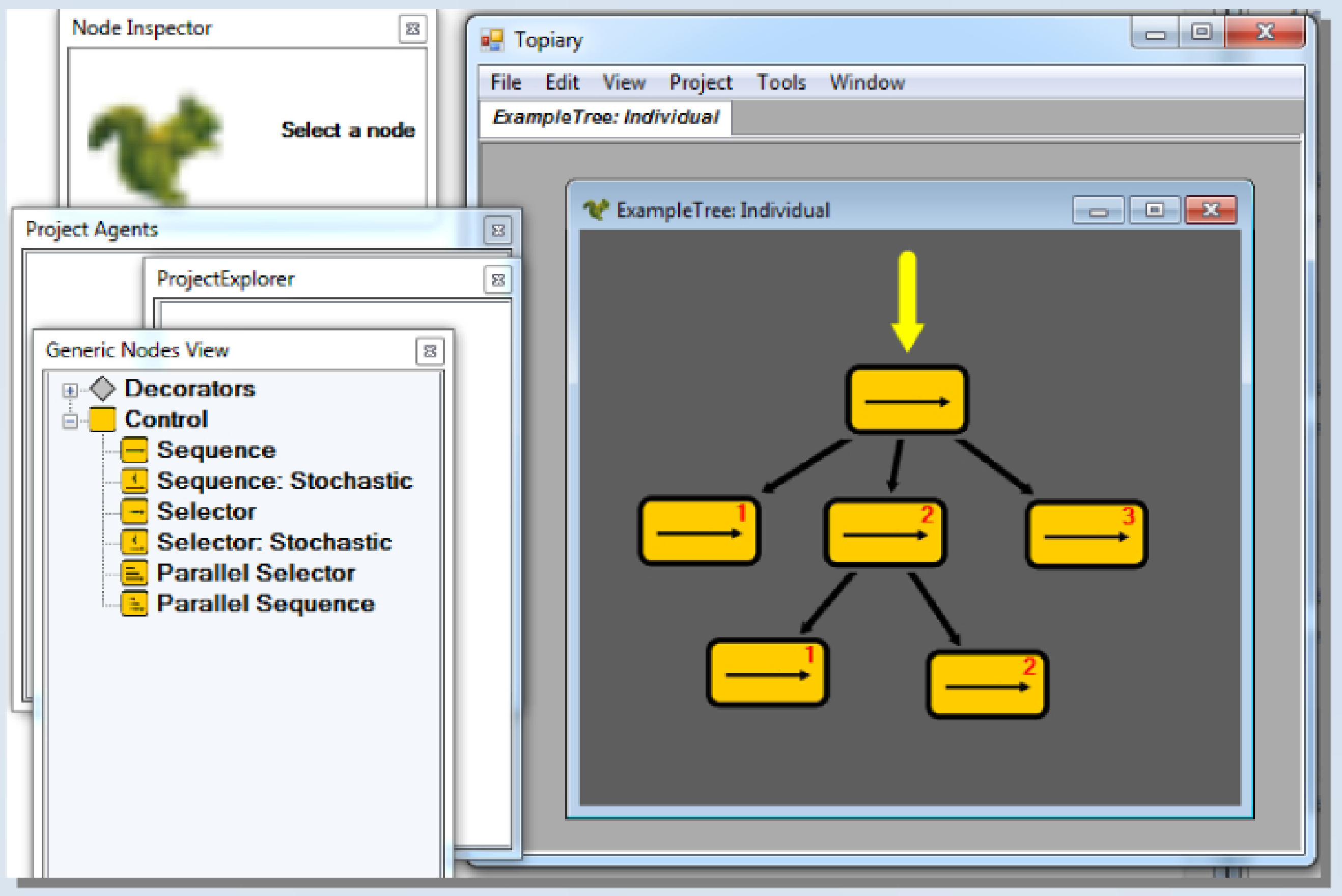
Robert Mead

Advisors: Alexander Shoulson, Norman Badler

Abstract

- Design an intuitive graphical interface to allow non-programmers to create and modify parametrized action representations

Screenshot:



Parametrized Action Representations (PARs)

- Parametrized Action Representations are used to describe actions in a generic fashion
- Each node executes, or fails.
- Generic nodes can control branching and sequence
- Agent nodes can execute an agent-specific programmed behavior, potentially with arguments
- An individual PAR can be used as a “Lookup Node” in another PAR, which will execute the entire subtree when called
- Variables propagate up – parameters of an agent node may become parameters of a lookup node containing its tree

Topiary

- Topiary is a Microsoft .NET Windows Forms application written in C#
- The application consists of a main window multiple-document interface containing a number of trees, each in a subwindow
- There are also several toolboxes with nodes, and a few inspectors for various properties
- The application adheres to a drag-drop paradigm. Nodes are placed by dragging them into a suitable canvas

Results and Future Work

At the end of the summer, we had a functional first version of the Topiary program that could be used to design complex trees in a reasonably straightforward way, and serialize them to XML. Utilizing and executing these trees within the Unity game engine is the next step of the process.

Further Reading

Shoulson, Alexander, Francisco Garcia, Matthew Jones, Robert Mead, and Norman I. Badler. "Parametrizing Behavior Trees." Motion in Games International Conference; Proceedings. 2011.