**Topic: Artificial Intelligence/ Machine Learning**

**Article: A DSL for Describing the Artificial Intelligence in Real-Time Video Games**

**Citation: T. Hastjarjanto, J. Jeuring and S. Leather, "A DSL for describing the artificial intelligence in real-time video games," 2013 3rd International Workshop on Games and Software Engineering: Engineering Computer Games to Enable Positive, Progressive Change (GAS), San Francisco, CA, 2013, pp. 8-14.**

**Summary of Article:**

Game AI is used to produce the illusion of intelligence in the behavior of computer-controlled agents. A domain-specific language (DSL) is a type of programming language (or specification language in software development and domain engineering) dedicated to a particular problem domain, a particular problem representation technique, and/or a particular solution technique. This paper introduces a novel language for describing the decision-making process of the AI in real-time video games. The authors develop a declarative, domain-specific language (DSL) embedded in the functional programming language Haskell for real-time video games, to describe the AI of a prototype real time video game. They want to develop an embedded DSL for game AI with which we can implement the behavior of computer-controlled agents in real-time video games such as Pacman, Space Invaders, etc. The DSL used in the paper offers a convenient and concise way to describe, include, transform, and reuse strategies in real-time video games, and does not require a graphical user interface to conveniently adjust, prototype, and maintain the game AI.

**Article Purpose:**

The purpose of this article is to introduce an approach that the authors think would be better than other languages for Game AI. It would be more generic and reusable not just within the same game but also for other video games. According to me, the authors want to show and highlight their study on this particular concept.

**Methodology**

The authors have implemented a small 2D real-time action game that contains agents that use GameAI to compute actions to complete the game. The game is between two teams where one team has to find a flag and reach to the destination by following an exit path and the other team has to prevent them from doing so. Based on the game AI description, the first step is to list all possible actions that can be performed. After choosing a particular action, an event is fired based on the action and state of the agent. Since there might be multiple AI agents playing simultaneously, there will be times when actions of one would affect the other. In such cases, a notification is sent to all agents which forces the agents to reconsider their moves and act accordingly.

The game has been implemented in Haskell using the OpenGL library to render the game on screen and to schedule the game loop. On top of OpenGL, GLUT is used which provides input handling. The authors also implemented the A\* path finding algorithm in the function shortestPath to generate a list of waypoints to navigate to a target destination.

**Conclusion**

This paper describes a domain-specific language for describing artificial intelligence in real-time video games. The authors have chosen to embed the DSL in Haskell, that would give implementation for free, and access to the full functionality of a programming language. The DSL proposed in the article can be an attractive alternative to existing methods for specifying game AI that mostly rely on using general purpose control flow elements or use tree- or graph-like data structures. The DSL used in the paper offers a convenient and concise way to describe, include, transform, and reuse strategies in real-time video games, and does not require a graphical user interface to conveniently adjust, prototype, and maintain the game AI.

**Article Strengths:**

The article shares in depth knowledge of the technology and the algorithm to be used for AI agents in the video game. It also has a lot of future recommendations which I think suggests the authors’ interest in the field and is also an encouragement for other researchers wanting to work on something similar. From a reader’s perspective, it was a good informative read that would encourage readers to dig in and research more on the concept.

**Article Weaknesses:**

Yes, the article creates a test game to implement and test their DSL but fails to provide results with proper data. They could have compared their approach with other approaches as well to conclude and convince why their DSL was better. It was all words and no actual data.

**Recommendation:**

The authors want to investigate if they can use the current DSL to describe other components of a game too, such as animations, the story line, or the behavior of groups of agents. Also, they want to further experiment with the current DSL by implementing other kinds of games. In the near future, they also hope to develop a serious game for practicing communication skills. The authors want to implement behavior trees, GOAP, and HTN, in the current DSL. There is also a scope to work on the relationship between actions and events, to determine if one can further integrate events and actions to remove the need for firing events and listing agents involved.

**Checklist:**

Number of Authors: 3

Number of Citations of Article: 2

Number of Citations to other articles: 25

Methodology Explained (Yes/No): Yes

Technology Explained (Yes/No): Yes

Experiments and Data Reviewed (Yes/No): Yes

Conclusion Exist (Yes/No): Yes

Recommendations Exist (Yes/No): Yes