TIC-TAC-LEGO: An Investigation into Coordinated Robotic Control

Ruben, Vuittonet, Jr.  
University of Alabama At Birmingham  
rvit@uab.edu

Jeff Gray  
University of Alabama At Birmingham  
gray@cis.uab.edu

**Embedded Systems**  
Special purpose computer systems encapsulated by the devices they control  

- Designed on one System  
- Deployed on Another Firmware

- Often deal with sensors & activators
- Often deal in real-time

- Information hiding and Safety Issues

- The goal?  
  To execute as quickly as possible in an asynchronous world using the smallest amount of code with the highest level of predictability

**The RCX® 2.0 Brick**  
Hilchi HR800 Processor. 16MHz, 32K RAM, 28K available for safe firmware use  

- Similar to the MIT Brick
- 3 Power Sources
- 1 Five-Character LCD Screen
- 3 Input Sensors
- 4 Interface Buttons
- On, Off, View, Run, Pgm
- 1 Infrared Port (Uploading firmware, software, and communication)
- Power Supply: 6 V 1.5 volt AA Batteries

The default firmware supports up to 5 programs in the RCX Brick

**CLASS DECOMPOSITION**

- Tic-Tac-Lego: a set of two embedded systems to coordinate the play of Tic-Tac-Toe
- Robotics and Reusability: Robotic provides ample opportunity for component abstraction. Robotics provides ample opportunity for component reuse.

**The Classes**  

- Arduino
- Decoder
- Locations
- Rotation
- Scanner
- Sensors
- TicTacRxc
- TicTacRxcMove
- TicTacRxcScanner
- TicTacRxcScanComm

**IeJOS** is an interpreter  
IeJOS is a Tiny JavaTM runtime. Replaces the Lego® Mindstorms firmware

- Memory footprint 17K
- 32K – 17K = 15K (important layer)

- IeJOS implements ODL Language
- Preemptive Threads
- Multi-dimensional arrays
- Recursion
- Synchronization
- Exceptions
- Well-documented API

The IeJOS firmware is installed through USB Tower to the RCX® via infrared

**Reading the Game Board**

- ScannerRobot Software Components
- TicTacRxcScan
- Display
- Location
- Rotation
- Scanner
- TicTacRxcMove
- TicTacRxcScanComm

The Scanning Process  
On startup, ScannerBot enters a loop to scan the board. When pressed, ScannerBot scans the board. After the scan, ScannerBot computes Next Move. Next move is transmitted to the MarkerBot

**Computation of the Next Move**

- Negamax utilized to compute next move
  1. Equivalent to Min-Max algorithm
  2. Tic-Tac-Toe is monotonic
  3. Recursive depth is < 9
  4. Not using u قوله allows exploration of parallelism

**Parallelism**

- Occurs when the number of remaining moves is 7
- ScannerBot plays X or O player
- The board is transmitted to the MarkerBot
- The MarkerBot and ScannerBot compute Next Move in parallel
- MarkerBot transmits its result to ScannerBot
- ScannerBot elects to use the appropriate result
- ScannerBot transmits the selected Next Move to MarkerBot

**Navigation Components & Behavior**

- Navigation Components Heavily Reused
- ScannerBot and MarkerBot
- Coordinate Component: Roll, roll, rotate, and place the ScannerBot
- Sensors: Motors, Encapsulation enables more precise navigation

**The Nature of Embedded Systems**

- Embedded Systems behavior for Tic-Tac-Toe
- Limited Interface to Embedded System
- Specific Classes Drive System Behavior
- TicTacRxc
- TicTacRxcScanner

**Constraints & Class Refactoring**

- The RCX 2.0s available memory is 32K
- The IeJOS Firmware is 17K
- The Code + the Firmware was > 32K

**Solution**

- Refactor
- Make data members public
- Remove accessor and mutator methods
- Code exposure is known aspect of embedded systems

- ScannerBot’s IeJOS byte code is 13.2K (17 * 1024 + 13572) = 30980 bytes = 32K

**SUMMARY & CONCLUSION**

- Embedded Systems provide effective solutions in many critical real-time processing situations
- In fact, the majority of processors are manufactured for embedded systems
- The Lego RCX 2.0 Mindstorms Robotic Invention Systems provides an out of the box opportunity to explore the capabilities and limitations of embedded systems programming
- The RCX provides the novice programmer entertaining opportunities to develop software engineering skills, including abstraction and encapsulation, while playing with a fun toy.