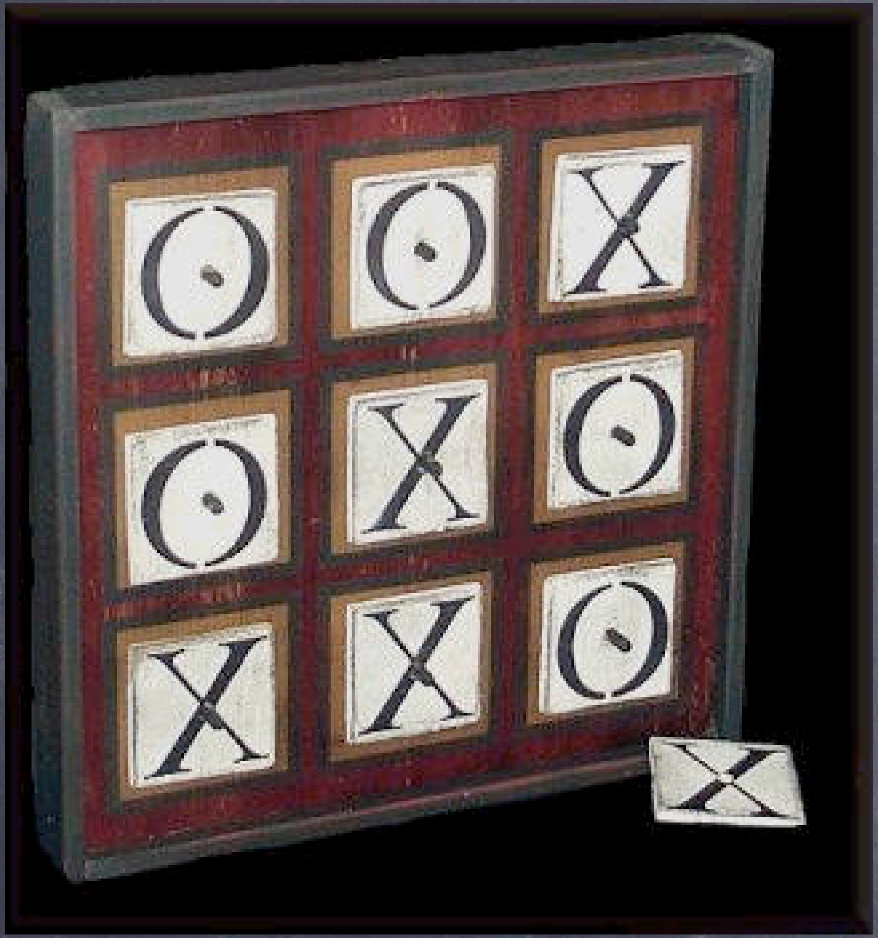
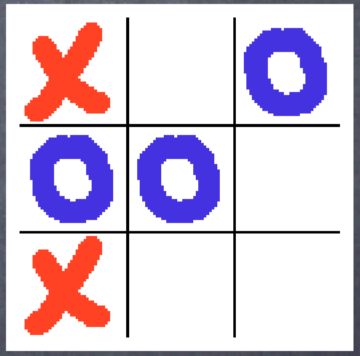


Tic-Tac-Toe in SpecC

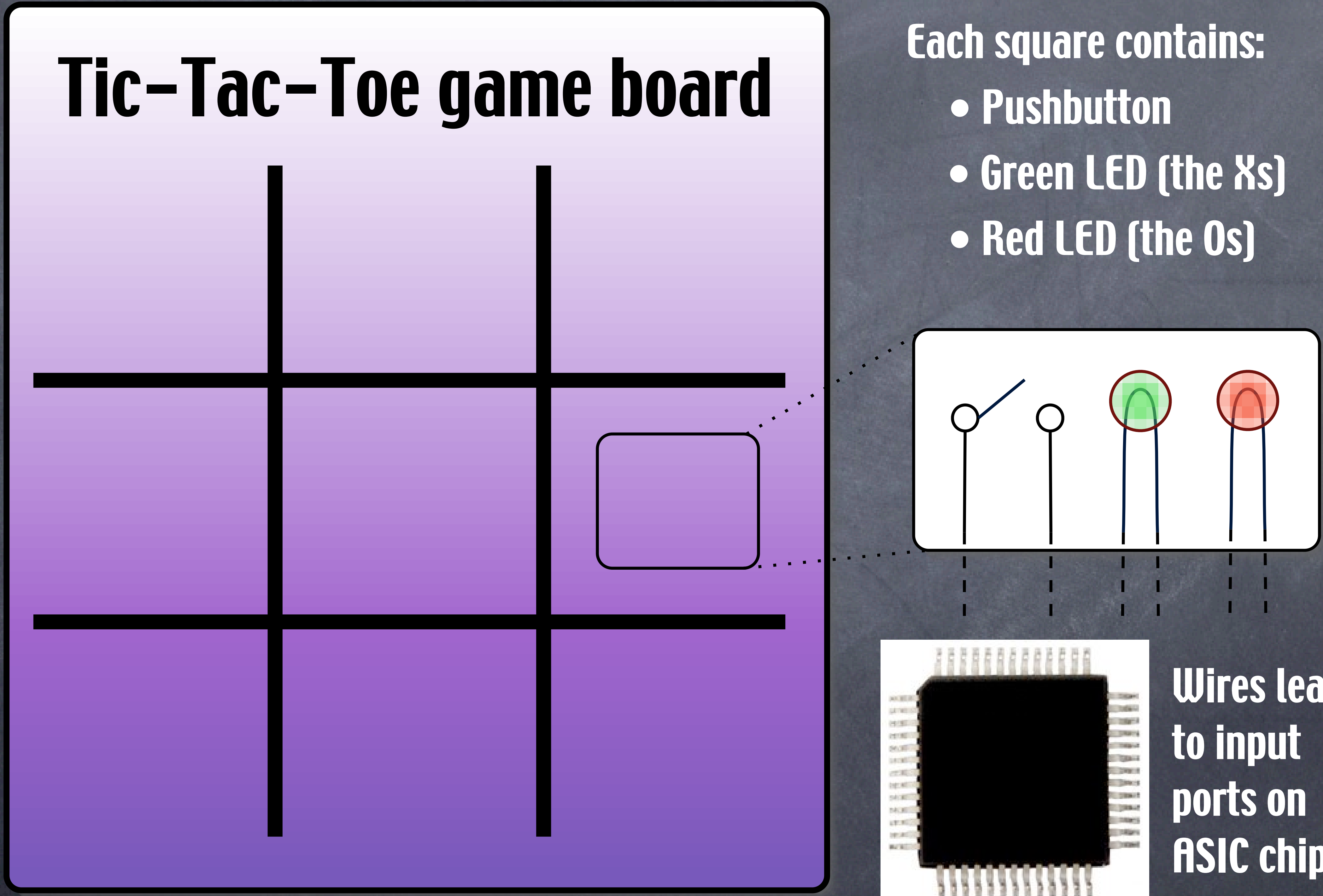
Trevor Harmon

Why Tic-Tac-Toe?

- Simplicity! Easy to understand and implement
- But not too trivial: Complexity can grow if desired
- Goal is to learn SpecC, not build fancy software
- You can play games while doing homework!

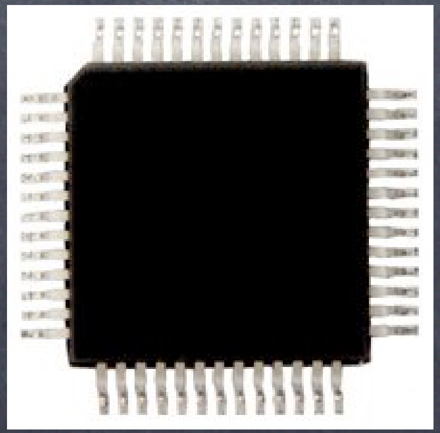


Tic-Tac-Toe in Hardware



Each square contains:

- Pushbutton
- Green LED (the Xs)
- Red LED (the Os)



Wires lead to input ports on ASIC chip

Finding a Game Engine

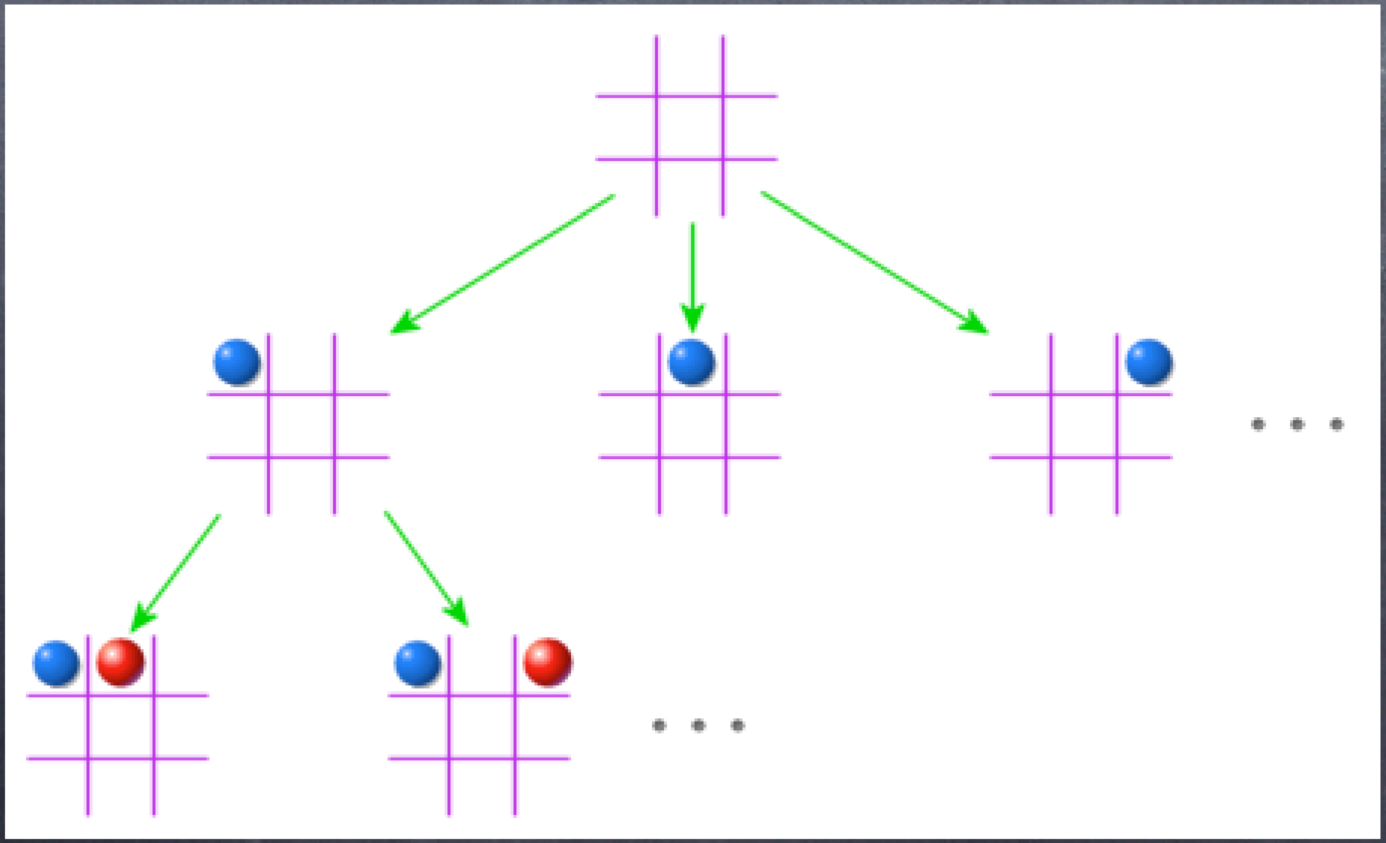
- Didn't want to reinvent the wheel
- Wanted to explore "pluggability" of SpecC
- Plenty of open-source tic-tac-toe algorithms available
- Found two good ones:
  a Java applet by Arthur van Hoff
  "Ultra Tic-Tac-Toe"

Arthur van Hoff's Engine

- Available in every Java SDK as a demo; license allows modification
- Very simple (brain-damaged?); relies on heuristics to choose next move
- Good for quick implementation and testing
- Required port from Java to SpecC (surprisingly easy)

“Ultra Tic-Tac-Toe”

- Open-source, highly configurable engine
- Claims to be one of the fastest recursive game-tree search algorithms available
- Relies on alpha-beta pruning



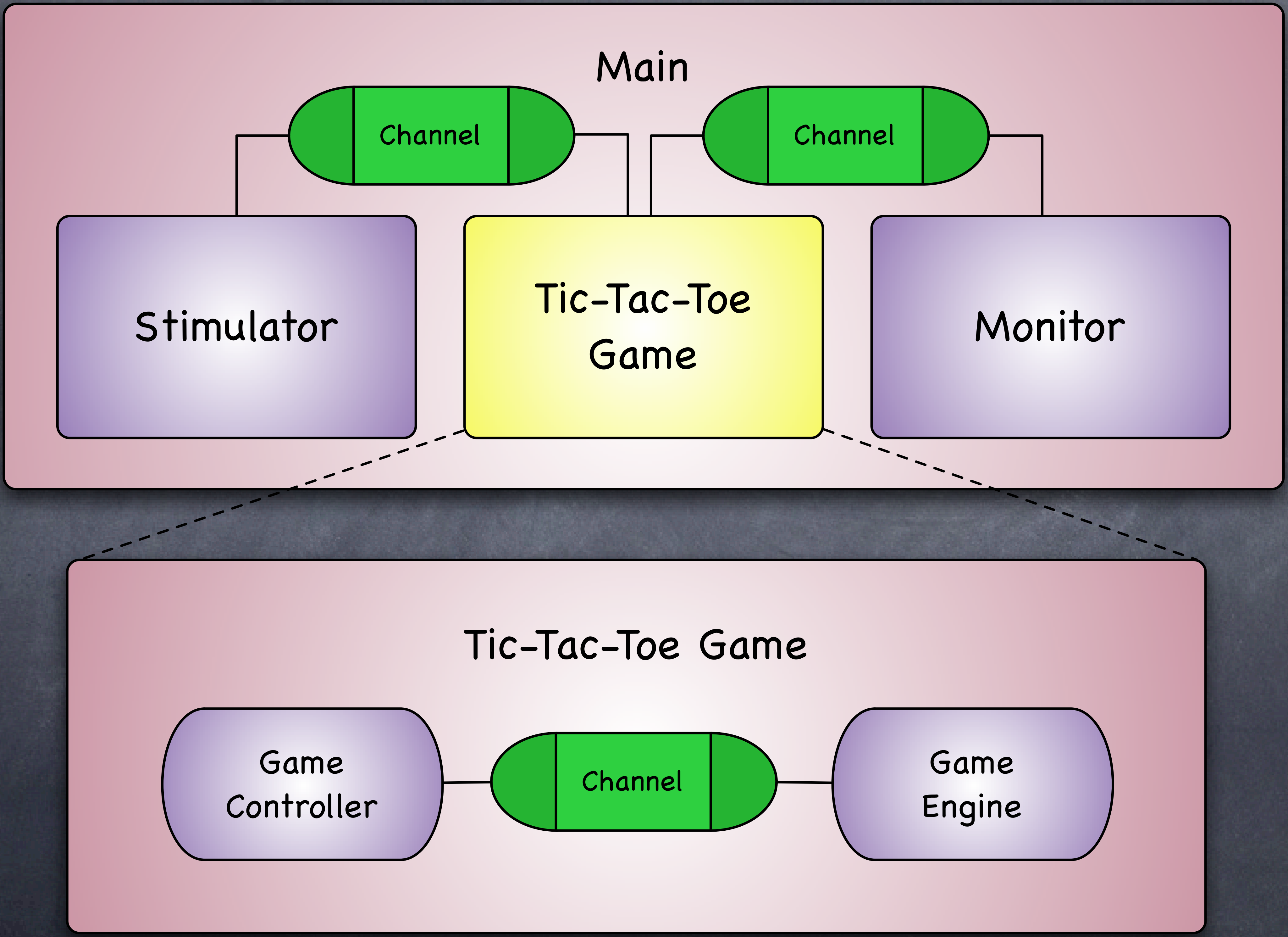
First Steps to SpecC

- Started with game engine API in ANSI C
- Interface:

```
void init();
void newGame();
void computerMove();
bool humanMove(int row, int column);
int  getStatus();
```

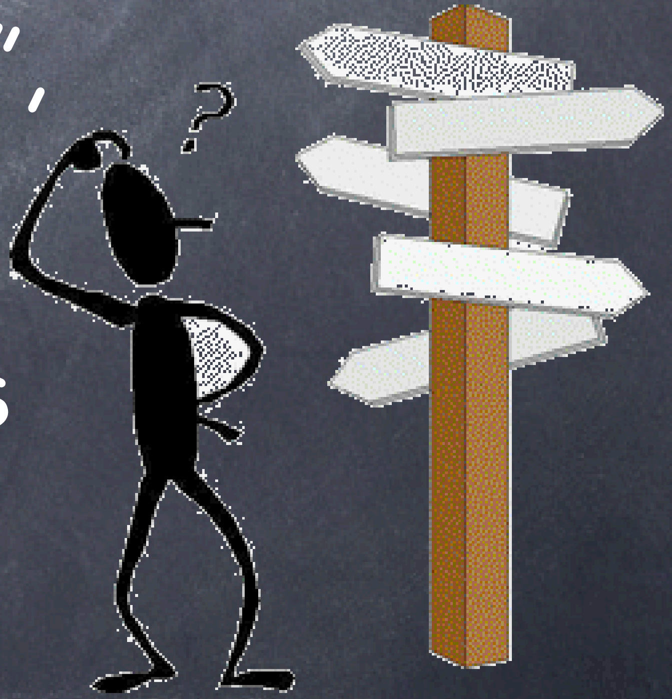
- Allows clean separation between game engine and user interface

Behavioral Layout



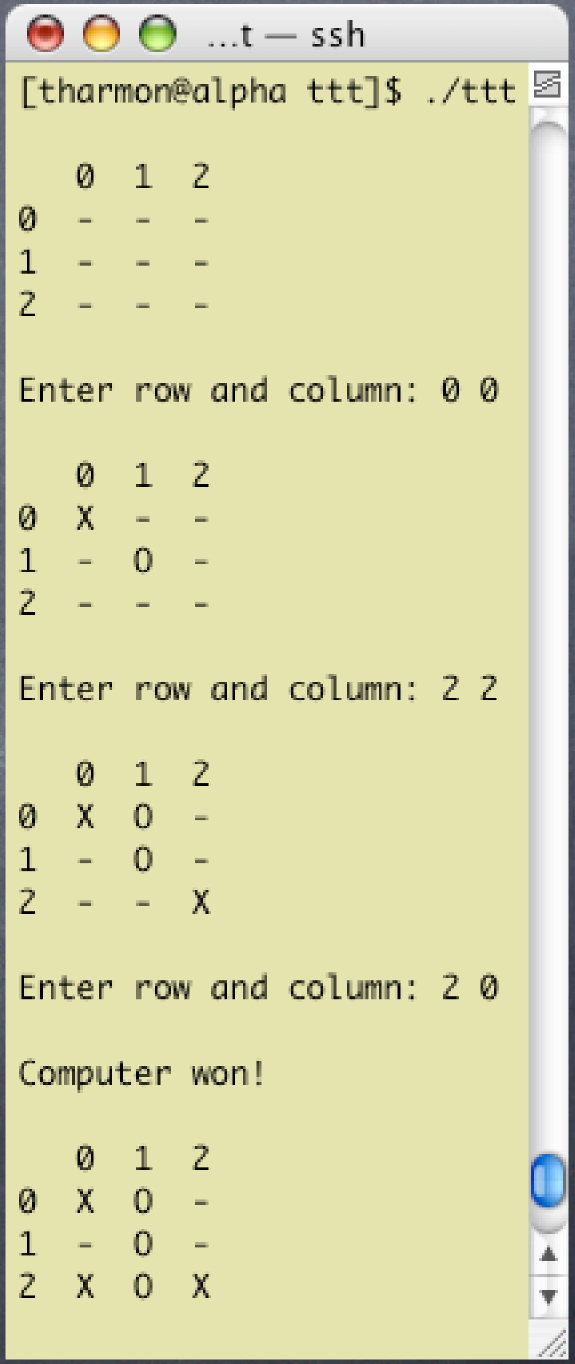
Problems Emerge

- Want to port clean, modular ANSI C game engine to SpecC
- Easier said than done: Communication between behaviors is through channels, not direct function calls
- SpecC provides concept of "interface", but this is for channels only
- For further discussion: Best practices for modular design in SpecC?



The Story So Far

- Both tic-tac-toe game engines have been ported to SpecC successfully
- But engine is not modularized as a behavior (hacked together inside the channel)
- Future work: cheesy lights and music
- Allow predictive processing in the background



```
[tharmon@alpha ttt]$ ./ttt

   0  1  2
0  -  -  -
1  -  -  -
2  -  -  -

Enter row and column: 0 0

   0  1  2
0  X  -  -
1  -  O  -
2  -  -  -

Enter row and column: 2 2

   0  1  2
0  X  O  -
1  -  O  -
2  -  -  X

Enter row and column: 2 0

Computer won!

   0  1  2
0  X  O  -
1  -  O  -
2  X  O  X
```

Lessons Learned

- Expertise with concurrent programming is essential
- Channels and interfaces are cool
- But SpecC needs more high-level encapsulation (e.g. explicit private methods in behaviors)
- Be careful when porting from C to SpecC: Your makefile might wipe you out!

Thank You

Image Sources

Tic-tac-toe game board:	primitivestenciling.com
Arthur van Hoff:	JavaWorld Magazine, 9/97
Alpha-beta:	Hamed Ahmadi Nejad
SourceForge logo:	sourceforge.net
Confused man:	seykota.com