

# Language Study: ML

---

## Final Project: Develop a “perfect” Tic-Tac-Toe game

“Perfect” in this context means that the computer should never lose; it should always be able to tie the human player or win. Use the divide and conquer approach here, much like we do when applying recursion to solve a problem. Break the task into small parts and solve those individually. Here’s some advice on doing that:

### Part One

Represent the game with ML functions and constructs, including the game board.

- Use Integer values to represent X and O and blank.
- Be able to address the board in (row,column) tuples.
- Write functions to get values from and set values in the board.

You should be able to play a “symbolic” game by the end of this part. It may not look or feel like tic-tac-toe as you know it, but all of the necessary data points should be represented and should be able to be manipulated by the functions you wrote.

### Part Two

Develop a more natural interface to the game. Write a `ttt()` or `play()` (or whatever you want to name it) function that launches the game. The computer should

- display the board
- prompt the user for the X player’s move
- make the move
- display the board
- determine the O player’s move
- make the move
- display the board
- repeat

### Part Three

Bullet-proof the game. Rick Cook (NASA) wrote “Programming today is a race between software engineers striving to build bigger and better idiot-proof programs, and the universe trying to produce bigger and better idiots. So far, the universe is winning.” Defend your game against ID10T errors both intentional and malicious.

- Check all input for range validity.
- Check that the user is not moving into an already occupied space.

Raise and handle exceptions where possible.

### Part Four

- Check for a winner. You should probably do this after every move.
- Detect ties as soon as possible. You can certainly tell if it’s a tie after move 9 because the board will be full and there will be no winner. You can detect a tie after move 8 in most cases. There are some cases where you can detect a tie after move 7.

E-mail me your code **and** hand your printed documentation on or before class on December 13, 2004.