

[0407] Top-Down Design Example (Tic-Tac-Toe)

Let's put Top-Down Design to the test. Let's use it to solve one of the greatest programming problems in the 21st Century...

TIC...
TAC...
TOE!!!!

(OK, granted, this is not a super-hard program - but pay attention anyway. Your homework assignment will be something similar...)

Design #1:

```
// Have two computer players play a random game of Tic-Tac-Toe
```

There. That wasn't too hard, was it? That is literally the top-level of our program. Granted it is written in what is called "pseudocode" which means it contained in a comment that describes things at the current "level" we are working at.

The point is we can take this "program" and show it to other people (i.e., our boss) and make sure they agree that this is what they really want. Remember, miscommunication is the source of some of the most expensive programming problems. OK. Onwards!

Design #2:

```
// Initialize the grid
// As long as we don't have a winner...
    // Show everyone the grid
    // Have the next player fill in an empty square (randomly).
    // Did they get three in a row?
    // Is it a Tie game?
    // No - change players and go back up to the top of the loop
// Yes! Exit the loop
// Tell everyone who won the game
```

OK, this time we got a little more detailed. But again, this is pseudocode. It is still very English-like and its main purpose is to ensure that every one agrees with the general outline of our program. Our boss can still look at this and tell us if they think it will work or not. Now let's add some real control statements and state variables:

Design #3:

```
// Initialize the grid
InitializeGrid()

gameOver = false
currentPlayer = FirstPlayer

// As long as we don't have a winner...
while (!gameOver)
{
    // Show everyone the grid
    DisplayGrid()

    // Have the current player fill in an empty square (randomly).
    squareSelected = SelectEmptySquare()
    grid[squareSelected] = currentPlayer

    // Did they get three in a row?
    if (FoundThreeInARow())
        gameOver = true;
    else if (TieGame())
        gameOver = true;
    else
        // Change players
        currentPlayer = OtherPlayer()
}

// Tell everyone who won the game
if (TieGame())
    WriteLine("Tie Game!")
else
    WriteLine("Player #{currentPlayer} won!")
```

Now we are much closer to creating real code. We could still give this to a non-programmer and probably get them to understand it if necessary. At this stage, we might also show this to other programming teams that might need to have their software interact with ours.

After those final reviews, we are ready to write actual code. Note that we should start with our approved pseudocode (as comments) and then go from there:

Try It Yourself!

Write the Tic-Tac-Toe program based on the pseudo-code above. After creating your own version, click here to see my solution: [\[407a\] Tic-Tac-Toe Solution](#)

<https://lwtech.instructure.com/courses/1841516/pages/407a-tic-tac-toe-solution>