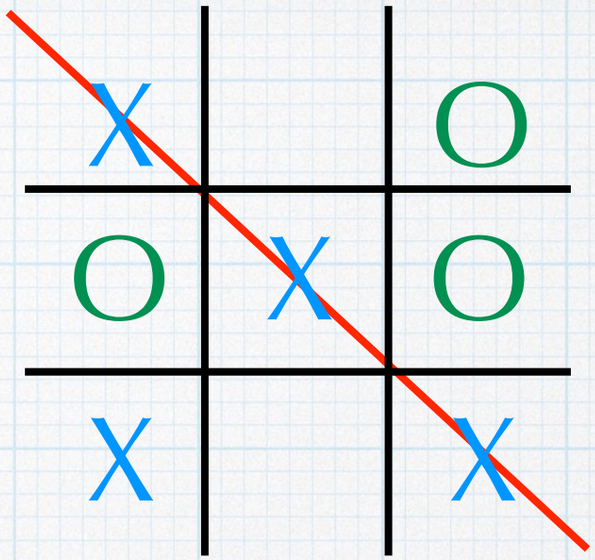
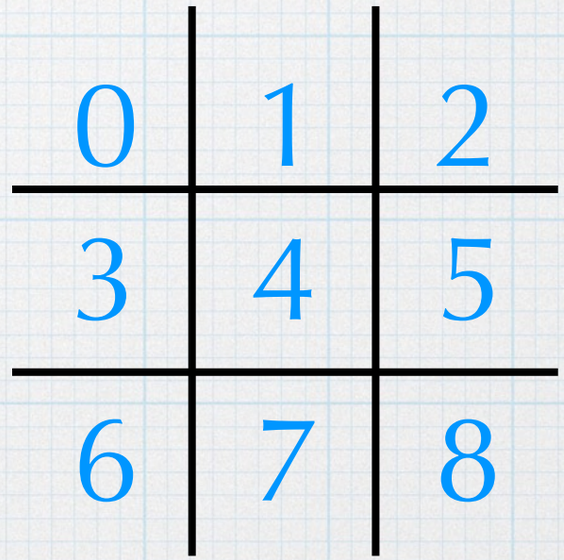
Tic-tac-toe



Programming project example: Let's build a simple tic-tac-toe game. You all know tic-tac-toe:

What is needed?

- variables for X & O positions on the board (an array)
- variable for the current player
- counter to keep track of the number of turns
- a loop to control the game play
- a function to draw the board with X's and O's
- a function for the users to enter choices
- a function to check for a win
- a function to swap players at each turn (maybe)



variables used in `main()`

```
char grid[9] = {'0', '1', '2', '3', '4', '5', '6', '7', '8' };
char currentPlayer = 'O';
int counter = 0, done = 0;
```

function: drawGrid

```c
#include <stdio.h>

void drawGrid( char*);

int main(){

   char grid[9] = {'X',' ', 'O', 'O', 'X', 'O', 'X', ' ', 'X' };

   drawGrid( grid );
}

/**  drawGrid
     Draws the current state of the tic-tac-toe grid    **/

void drawGrid( char a[] ){

   printf( "\n" );

   printf( " %c | %c | %c \n", a[0], a[1], a[2] );
   printf( "---|---|---\n");
   printf( " %c | %c | %c \n", a[3], a[4], a[5] );
   printf( "---|---|---\n");
   printf( " %c | %c | %c \n", a[6], a[7], a[8] );

   printf("\n");
}
```

```
 X |   | O
---|---|---
 O | X | O
---|---|---
 X |   | X
```

function: takeTurn

```
/**  takeTurn
 Tells the current player the available moves and enters the choice **/

void takeTurn( char a[], char *playerPtr  ){

   int available[8];
   int i, j = 0, choice;

   for(i = 0; i < 9; i++ )
      if( (a[i] != 'X') && (a[i] != 'O')  )
         available[ j++ ] = i;

   printf( "\n It is %c's turn.  Choose your move (", *playerPtr );

   for( i = 0; i < j; i++ )
      printf( " %d", available[i]);

   printf( " ): ");

   scanf( "%d", &choice);

   if( *playerPtr == 'X' )
      a[ choice ] = 'X';
   else
      a[ choice ] = 'O';
}
```

function: checkForWinner

```
int checkForWinner( char a[], int cnt, char* playerPtr  ){

   int i, gameOver = 0;

   for( i = 0; i < 3; i++ ){
      if( a[3*i] == a[3*i + 1] && a[3*i + 1] == a[3*i + 2]){
         printf( "We have a winner! It is %c.\n\n", *playerPtr );
         gameOver = 1;
      }

      if( a[i] == a[i + 3] && a[i + 3] == a[i + 6]){
         printf( "Game over, man! %c dominates.\n\n", *playerPtr );
         gameOver = 1;
      }
   }

   if( a[0] == a[4] && a[4] == a[8]){
      printf( "%c wins on a diagonal move!\n\n", *playerPtr );
      gameOver = 1;
   }

   if( a[2] == a[4] && a[4] == a[6]){
      printf( "That's it. %c has prevailed!\n\n", *playerPtr );
      gameOver = 1;
   }

   if( cnt == 8  && gameOver ==  0 ){
      printf( "Awww. It's a cat's game.\n\n" );
      gameOver = 1;
   }

   return gameOver;
}
```

function: swapPlayer

```
void swapPlayer( char *playerPtr ){

   if( *playerPtr == 'X' )
      *playerPtr = 'O';
   else
      *playerPtr = 'X';
}
```

function: main

```
#include <stdio.h>

int checkForWinner( char*, int, char* );
void drawGrid( char a[] );
void takeTurn( char*, char* );
void swapPlayer( char* );

int main(){

   char grid[9] = {'0', '1', '2', '3', '4', '5', '6', '7', '8' };
   char currentPlayer = 'O';
   int counter = 0, done = 0;

   drawGrid( grid );

   while( !done ){
      takeTurn( grid, &currentPlayer );
      drawGrid( grid );
      done = checkForWinner( grid, counter, &currentPlayer );
      swapPlayer( &currentPlayer );
      counter++;
   }
}
```

Other stuff

Potential changes/improvements

- Make it so that a user cannot over-write previous moves.
- Use a 2-D array to hold the Xs and Os.
- Have an option as whether X or O goes first, or choose randomly.
- Streamline some things. For example, it is not necessary to check for a winner until the fifth move.
- Jazz up the commentary.
- Improve the appearance of the board.
- Make a computer player option. (Requires some "artificial intelligence" in order for it to be "not lame".