

```
void play ( p1 , p2, o )  
{
```

```
    string whoseTurn = "p1";
```

```
    p1 >> junk >> p1name;
```

```
    p1 >> junk >> p1a;
```

```
    p2 >> junk >> p2name;
```

```
    p2 >> junk >> p2a;
```

```
    while( ! theBoard.gameOver()  
           && ( ! p1.fail() || ! p2.fail() ) )  
    {
```

```
        if ( whoseTurn == "p1" )  
        {
```

```
            p1 >> command;
```

```
            if ( command == "drop" )
```

```
            {
```

```
                p1 >> junk; // read the word "in"
```

```
                p1 >> column;
```

```
                if ( theBoard.dropCheck( p1a, column-1 ) )
```

```
                    whoseTurn = "p2";
```

```
            }
```

```
        }  
        else
```

```
        {
```

```
            p1 >> junk;
```

```
            theBoard.display( out );
```

```
        }
```

```
    }
```

```
    else // p2 turn
```

```
    {  
    }  
} // game is over
```

---

```
string board[rows][cols];
```

	0	1	2	3	4	5	6
0	.	.	.	.			
1							
2							
3							
4							
5							

---

```
bool dc ( checker, col )  
{
```

```
    bool dropped = false;  
    if ( col >= 0 && col < cols )  
    {
```

```
        for ( int i = rows - 1; i >= 0 && !dropped; i-- )  
        {
```

```
            if ( board[i][col] == "  " )  
            {
```

```
                board[i][col] = checker;  
                dropped = true;
```

```

        checkerCount++;
    } // end of if
} // end of for
return dropped;
}

```

```
Board::Board()
```

```
{
```

```
    checkCount = 0;
```

```
    for (int i = 0; i < rows; i++)
```

```
        for (int j = 0; j < cols; j++)
```

```
            board[i][j] = " ";
```

```
}
```

---

```
if ( board[0][0] == board[0][1] &&
```

```
    board[0][1] == board[0][2] &&
```

```
    board[0][2] == board[0][3] &&
```

```
    board[0][0] != " ")
```