

```
void PG( 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; // this reads the word "in"
```

```
                p1 >> column;
```

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

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

```
            }
```

```
        else
```

```
        {
```

```
            in >> junk; // reads the word "board"
```

```
            theBoard.displayBoard( o );
```

```
        }
```

```
    }
```

```
    else
```

```
    {
```

```

    } // end of player
  } // end of while

```

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

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

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

```
{
```

```
  checkerCount = 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] != " ")
```

```
bool dr = false;
```

```
if (checkerCount == rows * cols)
```

```
  dr = !win();
```

```
return dr;
```

```

bool dc ( checker, column)
{
    bool dropped = false;
    if ( column >= 0 && column < cols)
    {
        for ( int i = rows - 1; i >= 0 && ! dropped; i -- )
        {
            if ( board[i][column] == " ")
            {
                board[i][column] = checker;
                checkerCount ++;
                dropped = true;
            }
        }
    }
    return dropped;
}

```