**project**

**document**

Here is a proposal for refactoring blackjack4. The issues?

• 1800+ lines of code!

• convoluted table cell management processes

• bugs -- some trackable, others mysterious...

• bloated game functions

• confusing object structure

Rather than emulating a model/view/controller design, blackjack4 uses both "Sequencer" and "Game" objects to manage various states of the game (entering player data, dealing, betting, calculations, etc.) These two objects handle model-like, view-like and controler-like functions. Ouch!

Cell Management

An 18 by 12 table is fundamental to the design (for several reasons -- including the desire to try a non-standard approach). Manipulating cell regions throughout the game requires tracking row and column spans on cells to the left or above any targeted region. For example, any 18-cell table row that includes a cell that spans 3 columns is actually a 16-cell row. (18 - (3 - 1) = 16). The column span count is alway reduced by one since the 3-column cell must itself be counted.

Counting colspans in a target row is easily managed by iterating through cells of the row and testing for colspan values > 1, but rowspans create a greater challenge. A rowspaned region that extends into the target row will be missing cells completely. The rowspan itself may declared in some row above. One option would be to iterate through the whole table and count rowspans. Instead, a rowspan "tracker" is initialized at the start of the game and updated whenever a rowspan is assigned. The tracker will know of any rowspans affecting any row of the table.