David Landry

CS 290

Interesting Findings from Testing Sink the Fleet

1. Constructors all assume valid input. Very little input validation. None at all for CCell, CPlayer.
2. I had to adapt my testing a bit for CCell's inputCoordinates method. I had forgotten that that method is a user input method that takes in data in the grid format (e.g., A1).
3. The CShip class doesn't really do anything. It claims to be a wrapper class for a Ship enum, but the Ship cast method doesn't even do anything.
4. Different error handling for out-of-range row vs. out-of-range column. OOR row: reprompt row and column; OOR column: reprompt just column. Might consider total reprompt for invalid column.
5. CDirection: The direction type cast does not take any range into account.
6. CDirection: The input operator does not clear the buffer. Should it?
   Note: in stdio, the >> operator ignores leading whitespace and takes in all data before the next whitespace character, leaving it in the buffer. CDirection's >> operator seems to do the same thing.
7. CPlayer: There is an unhandled exception in the getCell method when accessing an undefined grid (e.g., if 0 is the ship grid and 1 is the hit/miss grid, program breaks if accessing the -1 grid)
8. CPlayer::setGridSize does not validate data.
9. Somewhere, there is validation for row data, but not column data. When accessing invalid cells in CPlayer, it accepts my invalid row data, but invalid column data crashes the program.
10. CPlayer::getGrid fails if the CPlayer m_gridSize property is lowercase.
11. CPlayer::saveGrid attempts to save even to an invalid file name or location.
12. CPlayer::setGrid is supposed to call clearGrid after setting ships, but it doesn't.
13. CPlayer as a whole, at the unit level, does not handle bad/invalid data well.
14. CShipInfo: Cell values will only properly instantiate when constructing a CCell object.
15. Control flow on CSinkTheFleet::play reveals workarounds for lack of validation. Validation is done primarily here.
16. Exploratory system testing reveals no game-breaking (or even game-annoying) bugs.