

CSE 115/503

April 12-16, 2010

Announcements

- Lab 7 due this week; Start Lab 8 this week
- Exam 4's handed back on Friday – pick yours up from me.
- Exam 5 Review Wednesday 4/14
- Exam 5 Friday 4/16
- Lab 8 due Monday 4/26
- Final exam review session: Wednesday 4/28 afternoon
- Final exam Thursday 4/29

Lab 8 – my solution

- 35 files, broken down as follows
 - 3 interfaces
 - IGame, IGameConstants: unchanged from Lab 7 skeleton
 - IGameEngine – changed to add methods for game play

Lab 8 – my solution

- 4 abstract classes
 - APlayer
 - Represents players in the game
 - AShipPlacementState (not changed from Lab 7)
 - Tells us whether we should place ships on the screen or not
 - AShip
 - Ships for the game
 - ACell
 - The cells on the game board

Lab 8 – my solution

- 27 concrete classes
 - App, Game, GameEngine, Position, ShipHolder
 - 3 listeners (one for all buttons, player's board, opponent's board)
 - 11 subclasses of Aship
 - 2 subclasses of AShipPlacementState
 - 3 subclasses of ACell (Ship, Hit, Miss)

Lab 8 – my solution

- Concrete classes (continued)
 - Board class (wrapper for a drawing canvas)
 - 2 subclasses of Aplayer (Human and Computer)
- 1 enumeration
 - Direction (from a lecture example)

Lab 8 – my solution

- File stats (including white space and comments)
- Longest file is Game at 229 lines
- Second longest is GameEngine at 213 lines
- Shortest file is ACell at 20 lines
- Average lines per file: 53

Lab 8 – my solution

- Only 4 classes use an if, for loop, or for-each loop
 - AShip
 - Uses a for loop to create the positions the ship occupies
 - Uses a for-each loop with an if statement in its body to see if a ship will be placed out of bounds

Lab 8 – my solution

- PlaceShipsState (nested if/elses)
 - if (the ship coming out of the holder is null)
 - Tell user to click on buttons first and then board
 - else
 - if (the ship will be out of bounds)
 - Tell user the ship can not be placed there
 - else
 - Tell engine to place the ship

Lab 8 – my solution

- Game Engine (if/else with nested if)
 - if (ship does not conflict with another ship)
 - Really put ship on screen
 - if(all ships are placed)
 - Change states to done placing ships state
 - else
 - Tell user ship can not be placed there

Lab 8 - my solution

- APlayer
 - For-each loop to add the ship positions to the hash map
 - For-each loop with an if-statement that actually looks for the conflicting ships
 - If-statement to see if number of hits is zero to tell that game is over

Lab 8 – my solution

- APlayer (one more)
 - If/else to see if cell coming out of hash map is null, if it is null, then a miss has occurred, if it isn't null then we need to process what has occurred, either a hit on a ship or a repeat click on a hit or miss

Lab 8 – my solution

- For a grand total of:
 - 1 for-loop
 - 3 for-each loops
 - 4 if/else statements
 - 4 if-statements
- People have told me we need a lot of if's in this project...

Project for Today

- Write the words for a number.
- That is, when you type in 387, the computer will print out “three hundred eighty-seven”