Assoc Prof Bernhard Schmidt

Object Oriented Programming I
Assignment 2, Instructions

Write a C++ program which allows the user to play a version of the game battleship against the computer. The game runs as follows.

- The game is played on an 8x8 grid. The rows of the grid are labeled A,B,...,H and the columns are labeled 1,2,...,8. A position in the grid, e.g., A5, is called a "square".
- A "ship" consists of vertically or horizontally adjacent squares. E.g., A2,A3,A4, is a vertical ship and C3,D3,E3 is a horizontal ship.
- The computer randomly sets up 5 ships: one ship of size 5, one of size 4, one of size 3, and two of size 2. Ships must not overlap. Example: The computer could choose A4,A5,A6,A7,A8 as ship 1, C4,D4,E4,F4 as ship 2, H1,H2,H3 as ship 3, F6,G6 as ship 4, and G8,H8 as ship 5.

  The positions of the ships are *not* revealed to the user.
- The user repeatedly throws "bombs" on a square to sink the ships. A bomb either hits a ship (indicated by "H") or hits water (indicated by "W").
- If the user input is in incorrect format or a square which has been hit already, the user should be informed and asked for correct input.
- If all squares of a ship are hit, the ship is sunk. To indicate this, the squares of the ship are labeled "S".
- The game ends when all ships are sunk. A message should be shown indicating how many bombs were needed. There is no winner or loser. The goal is to use as few bombs as possible.
- At each step, the current situation should be shown on the screen, as in the example on the next page.

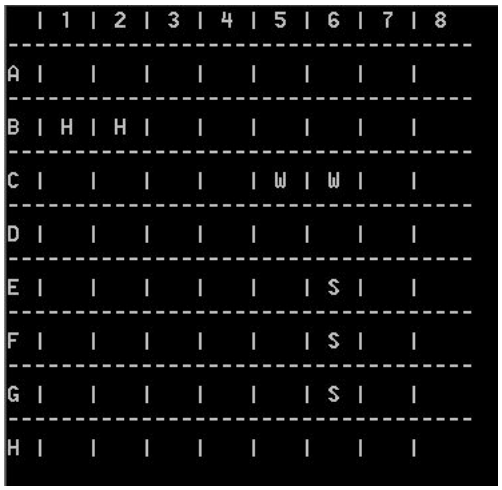

Figure 1: Output of the C++ program during a game

In the example in Figure 1, 7 bombs have been thrown. 5 of them hit ships (indicated by H or S), 2 of them hit water (indicated by W). The ship E6,F6,G6 was sunk already and thus the hits are indicated by S instead of H. The hits B1, B2 have not sunk a ship yet.