Computing *Unassessed* Programming Tutorial 4

Susan Eisenbach

5 December 2007

Cows and Bulls, is played between two players, the *scorer* and the *guesser*. The scorer chooses a list of 4 numbers (repetitions are not allowed) from the numbers 1, 2, 3, 4, 5, 6, 7, 8 and 9.

For example a valid list is [1, 8, 2, 4].

The scorer keeps this list secret: it is called the **code**. The guesser now tries to **guess** the code. The scorer gives a **score** to each guess the guesser makes. This score is given in the form (no. of black pegs, no. of white pegs). One **black peg** is scored for each number that is correct and in the correct place. One **white peg** is scored for each colour that is correct but in the wrong place. For example if the scorer's secret code is

[1, 8, 2, 4] and the guesser guesses [1, 2, 3, 4] then the scorer should award a score of (2,1).

The two black pegs are for getting 1 and 4 in the right place, and the white peg is for 2 which is in position 3 in the guess and position 2 in the secret code. The 8 does not appear at all in the code and so earns nothing towards the score.

The guesser tries to guess the code in as few tries as possible by making use of the information provided by the scores of past guesses. A correct guess of course earns (4,0) and the game is then over.

Assume you have the following declarations:

```
int guess[ ] = new int[4] ;
int code[ ] new int[4] ; ;
```

1. Write a predicate `anotherGame` that asks the user whether another game is wanted and returns true if and only if the user replies with y or Y.

2. Write a method `readGuess`, which reads in 4 numbers for the guess from the keyboard and returns the guess. You can assume that only valid guesses are typed in.

3. Write a method `blackScore,` which takes as arguments a code and a guess and returns the number of positions that the code and guess are identical.

4. Write a method `whiteScore,` which takes as arguments a code and a guess and returns the number of matches of code to guess which are in different positions.

5. Write a method `printScore`, which takes as arguments a code and a guess and prints on the screen (b,w) where b is the black score and w is the white score.

1.

```
boolean anotherGame(){
 //post: returns true iff the user input 'y' or 'Y'
 char ans;
   print("Do you want to play another game (y/n) -->");
   ans = readChar();
   return ans == 'y' || ans == 'Y';
 }
```

2.

```
 int[ ] readGuess(){
 //pre: a legal guess is typed in
 int[] in = new int[4];
   for (int i=0; i < 4; i++){
         in[i] = readInt();
    }
    return in;
 }
```

3.

```
 int blackScore(int[ ] guess, int[ ] code){
 //pre: guess and score have been initialised
 //post: returns the number of places guess and code are the
 same
 int score = 0;
   for (int i=0; i < 4; i++){
     if (code[i] == guess[i]) {score = score+1 ;}
   }
   return score;
 }
```

4.

```
 int whiteScore(int[ ] guess, int[ ] code){
 int score = 0;
   for (int i=0; i < 4; i++){
         for (int j=0; j < 4; j++){
         if (i != j && code[i] == guess[j]) {score = score+1 ;}
         }
   }
   return score;
 }
```

5.

```
 void printScore(int[ ] g, int[ ] c){
    println("(" + blackScore(g,c) + ", " + whiteScore(g,c)+
 ")");
 }
```