2BA2:
Programming Techniques
Assignment I
Mastermind

Conall O'Brien

01734351

conall@conall.net

November 28, 2003

1 Basic Design

The logic behind the game is rather simple. After initialising the game and generating the code, a simple loop iterates ten times, each time asking the user for input. Whil in the loop, the input is then checked for validity first, then two functions are used to run the necessary checks for exactly correct digits (bulls) and correct digits in wrong positions (cows).

2 The Loop

Basic error checking is done within the loop in regards to the user input. It is compared to the upper and lower bounds (1111 and 6666 respectively), before being passed into the comparison functions.

3 The Functions

3.1 generate_code : INTEGER

This function simply makes an object of type *STD_RAND* which is used to generate a random number, using a provided seed which is generated by the *get_seconds_from_epoch*. In order to ensure the generated number contains only valid digits, SmartEiffel API for the *STD_RAND* class is used to generate each digit with an upper limit of 6. Then the generated digit is added to the previously generated number, after the previous digits are shifted one digit left by multiplying by 10. Finally, the valid four digit number is returned to the main program.

3.2 compare_guess_for_exact_matches(guess : INTEGER; code : INTEGER) : INTEGER

This function compares the user's input to the code as integers, looking for an exact match. If an exact match is not found, it converts both numbers into variables of type *STRING*. Then each character of the code is compared to each character of the guess from the users. If a match occurs, a counter is incremented. Upon completion, the counter of matches is returned, after a basic post condition that the number returned is between 0 and 4.

3.3 compare_guess_for_wrong_places(guess : INTEGER; code : INTEGER) : INTEGER

This function compares the the number of occurrences of each digit contained in the user's input to the generated code. The total number of occurrences is then error checked to enure it is between 0 and 4 as it is returned by the function.

3.4 is_valid_input : BOOLEAN

This function is used as a condition to ensure the user input is valid by only containing the digits 1, 2, 3, 4, 5 and 6 and that the input is between the lower and upper bounds (1111 and 6666 respectively). If an invalid digit is found, the function returns false, else the entered input is valid and may be then tested.

3.5 get_seconds_from_epoch : INTEGER

This function calls SmartEiffel specific API that accesses the system clock. Once syncronised with the system clock, it returns the years, months, days, hours, minutes and seconds since UNIX Epoch (00:00:00 1st January 1970). After converting it to seconds via arithmetic and checking that the number is greater than 0, it is returned.

4 Source Code

5 Sample Output