INF142/INF142A: Obligatorisk innlevering 1

vår 2013

This is the first of three mandatory assignments in INF142 during the spring semester of 2013. Your project will be marked, and the combined score of the three mandatory assignments will make up 30% of the final score that determine your mark in the course. Further, you need a passing score on each of the assignments in order to qualify for the final exam.

For INF142A, this is the only mandatory assignment during the spring semester of 2013. Your project will be marked, and the score will make up 20% of the final score that determines your mark in the course. Further, you need a passing score on this assignment in order to qualify for the final exam.

1 The Project

You are required to write Java code that implements a dice game involving two clients, Client A and Client B, and one Server. This involves three processes. You should base your implementation on either TCP or UDP . . . it's up to you.

The programs will implement a 'dice game' - Clients A and B play the game, and the server controls the steps of the game and shakes the 'dice'. In the next section a description of the game is given in words. Then, in the following section, a specification for the implementation is given. Finally, in the last section, an example run of the game is provided.

You may in the context of this assignment ignore complicating issues such as client or server authentication and other security issues (although such aspects may well be crucial in a real application.)

Hint: To avoid firewall problems, you may use the address "127.0.0.1" to run all three processes on the same computer.

Note: The game should work when all three processes are run on the same computer. This means that you cannot use the IP address to distinguish between Client A and Client B. The game should also work if the three processes are run on three different computers.

If you are not sure about coding and running in Java and, in particular, socket programming, then have a look at the Java tutorial pages located at `http://docs.oracle.com/javase/tutorial/`.

If you prefer, up to two students may cooperate; if you choose to do so, you still have to hand in your individual files, and you also have to provide an explicit reference to your co-worker.

Information about how to submit your project will be made available soon when I've worked out how best to do this . . . .

Deadline: Friday March 8, at 18.00.

2 A description of the game in words

There are two players in the game, player A and player B (these are Clients A and B in our setup). There are two unbiased dice, with each dice having numbers 1 to 6.

- Players A and B start the first game with running totals $T_A = T_B = 0$.
- **Step** $Z1$**:** Using a coin toss either player A or player B starts. Let's assume that player A wins the toss and starts.
- Players A and B start the game with $V_A = V_B = 0$.
- The player who won the coin toss starts, so player A selects three numbers between 2 and 12. Let's say that player A chooses $\{5, 5, 8\}$.
- Player B observes what player A has selected and then also selects three numbers between 2 and 12. Let's say that player B chooses $\{8, 9, 12\}$.
- Player B then chooses a bonus number between 0 and 10 that will be added to the score of the winner of the game. Let's say that player B chooses 6.
- Player A (the coin-toss winner) then shakes the two dice. Let's say that the resulting shake is a 6 and a 3. Then $6+3 = 9$, but 9 is not in player A's set, $\{5, 5, 8\}$, so nothing happens. But 9 is in player B's set, $\{8, 9, 12\}$, so 9 is removed from the set to leave $\{8, 12\}$, and 9 is added to B's running total to give $V_B = 9$.
- the last step is repeated again and again, until either one or both of their number sets is empty.
- If player A has an empty set but player B doesn't then add the bonus number to V_A. For our example the bonus number was 6 and $V_A = 5 + 5 + 8 = 18$, so $V_A + 6 = 24$. Let's say that player B had $\{12\}$ left in his set. Then $V_B = 8 + 9 = 17$.
  If, instead, player B has an empty set but player A doesn't then add the bonus number to V_B. For our example the bonus number was 6 and $V_B = 8 + 9 + 12 = 29$, so $V_B + 6 = 35$. Let's say that player A had $\{5, 8\}$ left in her set. Then $V_A = 5$.
- So the result of this round of the game is that player A has V_A points and player B has V_B points. V_A is added to the running total T_A and V_B is added to the running total T_B.

- If either or both of player A and player B want another round of the game then they return to step $Z1$.
- If $T_A > T_B$ then player A wins. If $T_B > T_A$ then player B wins.

In the Java implementation of this game, we let the Server do the dice shaking and control all aspects of the game. This is in contrast to the description of the game above.

On the next page a specification is given for the three process implementation.

Specification:

	Action/Message	Description	Comment
	Server action:	$T_A = 0$, $T_B = 0$	running totals
$Z1$	Server action:	Randomly assign $X = A, Y = B$ or $X = B, Y = A$.	
	Server → Client X	"Send me 3 numbers between 2 and 12"	
	Client X → Server	"5,5,8"	example
	Server action:	$C_X = \{5, 5, 8\}$	example
	Server → Client Y	"Client X chose $\{5, 5, 8\}$.	example
		Send me 3 numbers between 2 and 12"	
	Client Y → Server	"8,9,12"	example
	Server action:	$C_Y = \{8, 9, 12\}$	example
	Server → Client Y	"Send me a bonus number between 0 and 10"	
	Client Y → Server	"6"	example
	Server action:	$B = 6$	example
	Server action:	$V_X = 0$, $V_Y = 0$	
$Z2$	Server action:	Shake two dice and add the two numbers - call	
		the result R. (this means randomly select two	
		numbers between 1 and 6 and add them)	
		e.g. $R = 6 + 3 = 9$ or $R = 1 + 4 = 5$ . . . etc	example
	Server → Clients X, Y	"Dice sum is 6 + 3 = 9"	example
	Server action:	if R is in C_X then	
		remove R from C_X: $C_X \leftarrow C_X - R$	
		add R to V_X: $V_X \leftarrow V_X + R$	
		e.g. if $R = 5$, $C_X = \{5, 5, 8\}$, $V_X = 0$,	example
		then $C_X \rightarrow \{5, 8\}$, $V_X \rightarrow 5$	example
		if R is not in C_X then do nothing	
		e.g if $R = 9$, $C_X = \{5, 5, 8\}$ then do nothing	example
	Server action:	if R is in C_Y then	
		remove R from C_Y: $C_Y \leftarrow C_Y - R$	
		add R to V_Y: $V_Y \leftarrow V_Y + R$	
		e.g. if $R = 9$, $C_Y = \{8, 9, 12\}$, $V_Y = 0$,	example
		then $C_Y \rightarrow \{8, 12\}$, $V_Y \rightarrow 9$	example
		if R is not in C_Y then do nothing	
	Server action:	if neither C_X or C_Y is empty then JUMP TO $Z2$	
		if C_X is empty and C_Y is not empty then	
		$V_X \leftarrow V_X + B$	winner X gets bonus
		if C_Y is empty and C_X is not empty then	
		$V_Y \leftarrow V_Y + B$	winner Y gets bonus
		$T_X \leftarrow T_X + V_X$, $T_Y \leftarrow T_Y + V_Y$	update running total

Label	Action/Message	Description	Comment
	Server → Clients X, Y	"Client X got V_X, Client Y got V_Y Do you want a new game (Y/N)? "	

Then, either:

Label	Action/Message	Description	Comment
	Client X → Server Client Y → Server	"Y" "Y"	
	Server action:	JUMP TO $Z1$	

or:

Label	Action/Message	Description	Comment
	Client X → Server	"N"	
	Server → Clients X and Y	"Totals: Client A got T_A, Client B got T_B GOODBYE."	

or:

Label	Action/Message	Description	Comment
	Client Y → Server	"N"	
	Server → Clients X and Y	"Totals: Client A got T_A, Client B got T_B GOODBYE."	

3 An example run of the game

Activity	C_A	C_B	V_A	V_B	B	T_A	T_B
Initialize						0	0
Server chooses $X = A$							
Initialize			0	0			
Client A selects set	$\{5, 5, 8\}$						
Client B selects set		$\{8, 9, 12\}$					
Client B selects bonus					6		
Server shakes							
and gets $6 + 3 = 9$		$\{8, 12\} = \{8, 9, 12\} - 9$		9			
Server shakes							
and gets $1 + 4 = 5$	$\{5, 8\} = \{5, 5, 8\} - 5$		5				
Server shakes							
and gets $1 + 6 = 7$							
Server shakes							
and gets $4 + 4 = 8$	$\{5\} = \{5, 8\} - 8$	$\{12\} = \{8, 12\} - 8$	13	17			
Server shakes							
and gets $2 + 1 = 3$							
Server shakes							
and gets $3 + 2 = 5$	$\{-\} = \{5\} - 5$		18			24	17
Clients A and B both request a new game							
Server chooses $X = B$							
Initialize			0	0			
Client B selects set		$\{7, 7, 7\}$					
Client A selects set	$\{6, 7, 8\}$						
Client A selects bonus					10		
Server shakes							
and gets $2 + 4 = 6$	$\{7, 8\} = \{6, 7, 8\} - 6$		6				
Server shakes							
and gets $5 + 2 = 7$	$\{8\} = \{7, 8\} - 7$	$\{7, 7\} = \{7, 7, 7\} - 7$	13	7			
Server shakes							
and gets $3 + 1 = 4$							
Server shakes							
and gets $2 + 6 = 8$	$\{-\} = \{8\} - 8$		21			55	24
Client A requests new game							
Client B requests stop							

Game ended. Client A got 55 and client B got 24, so client A wins.