

A348/A548 Midterm Exam

Today in class you will have to provide the design for the program described below. Your design should explicitly address the following issues:

- what variables will comprise your state?
- how does your program retrieve state?
- how does it determine that the state is empty (if that is the case)?
- what are the steps for initializing your state?
- what inputs does your program have?
- how do you update state starting from the current inputs?
- how does your program store state?
- how does your program report state?
- how does your program get ready for more input from the user?

Provide brief, explicit and articulate answers to these questions along with a flowchart or pseudocode for how they will be combined in your program. Then, in lab, you will have to implement it. The grade is 60% this written component (judged by the soundness of your design decisions) and 40% for the execution (implementation).

In specifying the design you don't have to commit yourself to an implementation, but feel free to do so if that seems to simplify your work. Your design document should be useful, so that if we gave it to another programmer (s)he would be able to follow your guidelines and implement it as described in the design, regardless of how (s)he chooses to store state (client or server side) and/or the language used (PHP or CGI/Perl).

As always, for the design, assume the user is program-friendly, so don't worry about accidental reloads etc. Just get the basic algorithm working, as described below.

You have to write a program that simulates a simplified casino slot machine. The first time you go to the program it displays a screen like this:

Your draw:

46	-27	-10
<input checked="" type="checkbox"/> keep	<input type="checkbox"/> keep	<input type="checkbox"/> keep

Welcome, the score is 0 - 0.

Press when you're ready to move on.

At this stage you can decide to keep one or more numbers, then push the Proceed button, to continue the game. In the example I keep 46 (by checking the box under it) and call the next move. The screen then changes to show my modified draw (the other two numbers

are new, random and new, but not 46 which is kept unchanged) along with a draw for the computer. My new numbers are -10 and -28, ridiculously similar to what I had before, but that's randomness, what can I say, and the computer draw is (10, 11, -43). At this time the two draws are compared, number by number, just like in tennis: $46 > 10$, $-10 < 11$, and $-28 > -43$, which means that two of my numbers are bigger, one smaller. I win, with 2 to 1, but only the win counts. So the score is now 1-0.

Your draw:

46	-10	-28
----	-----	-----

Computer draws (10, 11, -43). You win. Score now: 1 - 0.

Press when you're ready to move on.

At this stage I am expected to push Proceed. If I do that I start a new game, that is, I generate a new draw for me, again with the ability to keep some numbers (obviously the bigger the number the better) and call for the computer draw, again. Here's what happens

Your draw:

40	-12	39
<input checked="" type="checkbox"/> keep	<input type="checkbox"/> keep	<input checked="" type="checkbox"/> keep

The score currently is 1 - 0.

Press when you're ready to move on.

I like 40 and 39, but I want something else instead of -12. I push Proceed and win again.

Your draw:

40	35	39
----	----	----

Computer draws (4, -45, -16). You win. Score now: 2 - 0.

Press when you're ready to move on.

And so on, forever. That's the program you need to design and, later in lab, implement.

Good luck, don't rush and do well!