School of Electrical Engineering and Computer Science
The University of Newcastle
SENG2050/SENG6050 Introduction to Web Engineering
Assignment 2 (13%) – Semester 1, 2018

Deal or No Deal

1. Introduction

The main objective of this assignment is to implement an online version of the game – Deal or No Deal. The original rules for the game can be found on the web (for example, https://en.wikipedia.org/wiki/Deal_or_No_Deal (Australian_game_show)). In this implementation we will simplify the game to make it more suitable for online play. Following are the (modified) game rules.

There are 12 briefcases which are numbered 1through 12. Each briefcase contains a unique monetary amount from the following figures: $0.5, $1, $10, $100, $200, $500, $1000, $2000, $5000, $10000, $20000, $50000. The contestant is asked to nominate a number of cases (4 for the first round, 3 for the second round, 2 cases in the third round, 1 for round 4, and 1 for round 5) to be opened so that the monetary amount in those cases can be revealed.

The aim of the player is to hopefully not reveal the larger cash prizes. At the end of each round (for the first 4 rounds) - the "bank" will offer the contestant a monetary amount to try and convince them to make a "deal" and quit the game at that point. The amount offered is based on the following formula: the total amount of the money in the remaining cases divided by the number of cases remaining. If the contestant believes that they are in possession of big bucks or that the bank offer is not worth considering – they will choose "no deal" and the game will move on to another round. After the 5th round, there is a single briefcase remaining, the monetary amount contained in the briefcase is the prize the contestant gets.

The first page that is presented to the user will allow them to start a new game or load an existing game (see below). When a user starts a new game, the website will display 12 briefcases (randomly ordered) and allows the user to click on a briefcase to reveal the monetary amount in those cases. You will also display how many briefcases are yet to be opened in the current round. At the end of each round the user will be shown a page displaying the amount the bank is offering together with the largest amount unrevealed in the current stage. There should be an option of "deal" or "no deal" on the page. If the user chooses "deal", the game finishes. If the user chooses "no deal", they will move onto next round.

User will be given the option to save their game throughout the game. The game's state should be saved on the server (to prevent cheating) and referenced by a username provided by the user. Each user may have a single saved game. Lastly, saving the game should end the current game and loading it should clear the saved data. This is to prevent users from reloading their game if they are unhappy with their selection.

To make this game practical, you will need to make sure:

- It is multi-user safe

- Clicking the browser’s “refresh” button will not result in any inconsistencies such as a briefcase being opened twice, the user is asked to rethink the bank’s offer, etc. This is not to be done by disable the “refresh” button.
- Clicking the browser’s “back” button will not roll back the game, this is not to be done by disable the “back” button.
- If user hasn’t been active for 1 (one) minute, then the game is auto-saved.

In this assignment, you should use JSPs and Java Servlets together with JavaBeans. It is up to you how you implement session tracking.

2. Submission

You should create a zip file called cXXXXXXX_assignment2 (where XXXXXXX is your student number) with the following files and submit it through blackboard:

1. Assignment Cover Sheet.
2. A brief explanation of your program (readme.txt).
    - The application’s structure, i.e. relationships among objects etc.
    - What is the purpose of each of your objects?
    - How did you implement session tracking?
    - How did you implement game saving?
    - Please identify if you are a MIT student or an undergraduate student.
    - The URL the marker needs to visit to start the application.
3. Your entire application. This should include all the files that contribute to your web pages, including HTML, java source code, and images, etc.

Please use relative addresses (no host name) in your program rather than absolute addresses. Your application should work when deployed to Tomcat’s webapps directory with the name cXXXXXXX_assignment2 (where XXXXXXX is your student number) i.e. apache-tomcat/webapps/cXXXXXXX_assignment2/ and http://localhost:8080/cXXXXXXX_assignment2/

3. Marking Guidelines

The following is a general guide on how the marks are allocated to each part of the assignment and some common mistakes where you might lose marks. Total marks: 90

HTML and CSS: 10
- Is your HTML syntax valid (use a validator)?
- Do your tags make semantic sense?
- Are you using tags just for their visual effect?
- Are you using CSS to advise the browser about visual formatting?

JSP: 20
- Is your JSP syntax valid?
- Are you overloading your JSP with too much Java code?
- Is your JSP nicely formatted and commented?
- Can your JSP handle errors properly?

Java Beans: 20
- Does your "bean" conform to the Java Beans standard?
- Is your bean shared among the pages correctly?
- Does your bean contain too much HTML output?

- Is your Java nicely formatted and commented?

Requirements: 30

- Does your assignment meet the specifications above?

Session Tracking and Thread Safe: 10
Usability: 10