

COMP2068 – Advanced Web Programming

Assignment 2

Web Slot Machine

Due class #7 (Friday October 17, 2014) @ midnight.

Value 15%

The slot machine game

Maximum Mark: 56

Overview: Use your accumulated knowledge of the JavaScript and Web technologies and the **slot machine code** (provided on GitHub as a template) and the Web API of your choice to create a HTML5 Slot Machine game.

Instructions :

(15 Marks: GUI, 17 Marks: Functionality, 8 Marks: Internal Documentation, 12 Marks: External Documentation, 4 Marks: Version Control)

1. Add three images to the GUI (one for each **Reel** of the slot machine) that have a default image (e.g. "SPIN") when the GUI initially loads. The images in the Reels will change every time the user presses the **Spin Button**. The images that appear in the image windows should be one of the symbols (e.g. fruits) that match the randomly generated symbols within your code. You will have to connect the functions contained in the **slot machine code** provided, or write your own code. (3 Marks: GUI, 6 Marks Functionality).
2. Add a **"Spin" Button** control to your GUI that allows the user to spin the reels of the slot machine each time the user clicks the button. This is where the action happens. The code to make your Reels Spin should be connected to this button (1 Mark: GUI, 3 Marks: Functionality).
3. Add a **Button** control to your GUI that allows the user to reset the game (1 Mark: GUI, 1 Mark: Functionality).
4. Add a **Button** control to your GUI that allows the user to quit the game (1 Mark: GUI, 1 Mark: Functionality).
5. Add **Labels** (or other appropriate controls) that display the users **Money**, the current **Jackpot** and the current **Bet** (6 Marks: GUI, 3 Marks: Functionality).
6. Ensure that when the user enters a Bet amount in the Bet Entry control that the program checks (validates) if he has enough money. If the user doesn't have enough money to make his bet, grey out (disable) the **Spin Button** control to prevent him from playing the game (2 Marks: Functionality).
7. Include conditions that allow the player to potentially win the Jackpot. You can use the example code provided for your odds. Reflect a Jackpot Win by displaying a special message (1 Mark: GUI, 1 Mark: Functionality).

8. Add a **Slot Machine Graphic** object that gives the interface the look and feel of a Slot Machine (2 Marks: GUI).
9. Include **Internal Documentation** for your program (**8 Marks: Internal Documentation**):
 - a. Ensure you include a program header that indicates: the Source file name, Author's name, Last Modified by, Date last Modified, Program description, Revision History (3 Marks: Internal Documentation).
 - b. Ensure you include a headers for all of your functions and classes (2 Marks: Internal Documentation)
 - c. Ensure your program uses contextual variable names that help make the program human-readable (1 Marks: Internal Documentation).
 - d. Ensure you include inline comments that describe your GUI Design and slot machine engine (2 Marks: Internal Documentation)
10. Include **External Documentation** for your program that includes (**12 Marks: External Documentation**):
 - a. A company Logo (1 Marks: External Documentation).
 - b. Table of Contents (1 Marks: External Documentation).
 - c. Version History (2 Marks: External Documentation).
 - d. Detailed Game Description – describing how your slot machine works (2 Marks: External Documentation).
 - e. Game Play Mechanics (1 Marks: External Documentation).
 - f. Controls (1 Mark: External Documentation).
 - g. Interface Sketch (1 Mark: External Documentation).
 - h. Screen Descriptions – Include screen shots for your game (1 Mark: External Documentation).
 - i. Scoring (1 Mark: External Documentation).
 - j. Art / Multimedia Index – Include examples of your image assets (1 Mark: External Documentation).
11. Share your files on **Github** to demonstrate Version Control Best Practices (**4 Marks: Version Control**).
 - a. Your repository must include **your code** and be well structured (2 Marks: Version Control).
 - b. Your repository must include **commits** that demonstrates the project being updated at different stages of development – each time a major change is implemented (2 Marks: Version Control).

Optional Game Features (i.e. Potential Bonus Marks).

- A. The Slot Machine activates **sound clips** each time the **Spin Button** is pressed, the player wins, etc.
- B. Create an **animation effect** for the Reels when the **Spin Button** is pressed.
- C. Include additional Reels (i.e. 5 Reels instead of just 3).
- D. Include additional “Bet Lines” (e.g. a primary and two secondary Bet Lines as well as Diagonal Bet Lines).

Example Interfaces



SUBMITTING YOUR WORK

Your submission should include:

1. An external document (MS Word or PDF).
2. A link to your GitHub Files. A link to your Live Game Site.

| Program Code & Functionality | | |
|--------------------------------------|---|----------|
| Technical Evaluation | | |
| Display / User Interface | The Display / User Interface elements meet the program requirements. All text is spelled correctly and appropriate space is allocated for user input. Graphics & Icons are appropriate and match the program's functions. | 15 |
| Functionality | The program's deliverables are all met and the program functions as it should. No errors appear as a result of execution. User Input does not crash the program. | 17 |
| Internal Documentation & Readability | A program header is present and includes the name of the program, the name of the student, a short revision history and a short description of the program. All procedures and classes include headers that describe their functionality and scope. Inline comments are used to indicate their function when code is new or unclear. Variable names are contextual wherever possible. | 8 |
| External Documentation | An external document (MS Word or PDF) has been created that includes a company logo, table of contents, version history, detailed program description, a sketch of the GUI and screenshot (if applicable), and other details outlined in the template provided. | 12 |
| Version Control | GitHub is used to track App development. A Commit history will demonstrate the App being updated at regular points in time that correspond with the milestones of the project. | 4 |
| Creative Evaluation | | Mark |
| Creativity | The program's GUI / UI is attractive. The programmer has added additional elements outside of the scope of the program that enhance functionality, usability and fun. | 0 |
| Total (/56) | | 56 |
| | | % 100.0% |

This assignment is weighted **15%** of your total mark for this course.

Late submissions:

- 10% deducted for each additional day.

External code (e.g. from the internet or other sources) can be used for student submissions within the following parameters:

1. The code source (i.e. where you got the code and who wrote it) must be cited in your internal documentation.
2. It encompasses a maximum of 10% of your code (any more will be considered cheating).
3. You must understand any code you use and include documentation (comments) around the code that explains its function.
4. You must get written approval from me via email.