COMP397 – Web Game Programming

Assignment 1

CreateJS Slot Machine

Due class #6 (Saturday February 15, 2020) @ midnight.

Value 12%

Simple Slot machine **Maximum Mark: 57**

Overview: Use your accumulated knowledge of the programming language and tools described in class, the **slot machine code provided** and the graphics framework described in class to create a Simple Slot Machine game. You will need to translate the code from jQuery in order to make this work.

Assignment Profile:

- **Individual Assignment**: each student must complete this Assignment on their own to demonstrate knowledge of the programming language and graphics frameworks described in class, Game Development and the content thus far.
- **Code Provided**: very simple code for an old Web-based slot machine is provided. This code was created using JavaScript and the jQuery Library. Your Job is to remove jQuery, but use some of the JavaScript code to create a new Slot Machine Game.

Instructions :

(15 Marks: GUI, 19 Marks: Functionality, 5 Marks: Internal Documentation, 6 Marks: External Documentation, 4 Marks: Site Structure, 4 Marks: Version Control, 4 Marks: Cloud Deployment)

1. Your will use the code provided for some of the functionality for your Slot Machine Game. You will use the programming language in class and the graphics framework for the UI **(15 Marks: GUI, 19 Marks Functionality):**
    a. Your Game Should include a Play Scene where the Slot Machine Game can be played (2 Marks: Functionality).
    b. Add a **Slot Machine Graphic** object that gives the interface the look and feel of a Slot Machine (2 Marks: GUI).
    c. 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).

   d. 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).
   e. Add a **Button** control to your GUI that allows the user to reset the game (1 Mark: GUI, 1 Mark: Functionality).
   f. Add a **Button** control to your GUI that allows the user to quit the game (1 Mark: GUI, 1 Mark: Functionality).
   g. Add **Labels** (or other appropriate controls) that display the users **Money**, the current **Jackpot** and the current **Bet** (6 Marks: GUI, 3 Marks: Functionality).
   h. 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).
   i. 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).

2. Include **Internal Documentation** for your program and create an appropriate **site structure** for your project **(5 Marks: Internal Documentation, 4 Marks: Site Structure):**
   a. Ensure you include a program header that indicates: The Author's name, Student Number, Creation Date, Game App description and Revision History (3 Marks: Documentation).
   b. Ensure your program uses contextual variable names that help make the program human-readable (1 Marks: Documentation).
   c. Ensure you include **inline comments** that describe the conditional structure you will use for your decision tree matrix (1 Marks: Documentation)
   d. Ensure your project files are organized within an appropriate site Structure, separating the **View** (HTML Document and CSS Files) and your **Game Logic** (JavaScript and TypeScript files) (2 Marks: Site Structure).
   e. Use **TypeScript** to create a scalable, object-oriented application (2 Marks: Site Structure).
3. Include a simple **Game Design Document** (GDD) for your game that includes **(6 Marks: External Documentation)**:
   a. A company Logo (0.5 Marks: External Documentation).
   b. Table of Contents (0.5 Marks: External Documentation).
   c. Version History (0.5 Marks: External Documentation). Include a link to your GitHub repo here.
   d. Detailed Game Description – describing how your slot machine works (0.5 Marks: External Documentation).
   e. Game Play Mechanics (0.5 Marks: External Documentation).

f. Controls (0.5 Mark: External Documentation).
g. Interface Sketch (1 Mark: External Documentation). A wireframe of your Game Screen(s).
h. Screen Descriptions – Include screen shots for your game (1 Mark: External Documentation).
i. Scoring (0.5 Mark: External Documentation).
j. Art / Multimedia Index – Include examples of your image assets (0.5 Mark: External Documentation).

4. Share your files on **GitHub** and deploy to a **Cloud Service** (Microsoft Azure, Heroku, or GitHub pages) to demonstrate Version Control Best Practices **(4 Marks: Version Control, 4 Marks: Cloud Deployment).**

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).
c. Ensure your game is live and online. Deploy to a Cloud Service of your choice (4 Marks: Cloud Deployment).

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. (2 Bonus Marks).
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). (4 Bonus Marks).
D. Include additional "Bet Lines" (e.g. a primary and two secondary Bet Lines as well as Diagonal Bet Lines). (4 Bonus Marks).
E. Include a Start Scene and a Game Over Scene using the Finite State Machine Pattern (4 Bonus Marks).

Example Interfaces

super
SLOTS
000.094
03
0006
CREDITS
BET
WINNER PAID
25¢
RESET
PAY TABLE
BET ONE
BET MAX
SPIN

BAR
Jackpot!

DOUBLE DIAMOND
BAR
BAR
BAR
BAR
BAR
BAR
BAR
FREE
PAID
BET

EVALUATION CRITERIA

Feature	Description	Marks
GUI / Interface Design	Display elements meet requirements. Appropriate spacing, graphics, colour, and typography used.	15
Functionality	Site deliverables are me and site functions are met. No errors, including submission of user inputs.	19
Internal Documentation	Your site content is well organized and does not include any spelling errors. Grammar has also been checked so that the messages are clear and concise. Any images that are included are sharp and proportioned appropriately.	5
External Documentation (GDD)	Your External Game Design Document (GDD) is complete and includes a Logo, a Table of Contents, Revision History, Wireframes, Screen Shots, Scoring and Multi-media index.	6
Site Structure	Well organized site files. Separate HTML and CSS. Appropriate links to external documents and code. Code is error free.	4
Version Control	GitHub commit history demonstrating regular updates.	4
Cloud Deployment	Deploy site to Cloud Service or GitHub Pages	4
Total		**57**

SUBMITTING YOUR WORK

Your submission should include:

1. A Game Design Document (GDD) (MS Word or PDF).
2. A link to your project files on GitHub.
3. A link to your live site on a Cloud Service of your choice (GitHub pages recommended)
4. Your project files zipped and submitted to e-centennial

This assignment is weighted **12%** of your total mark for this course.

Late submissions:

- 20% deducted for each day late.

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.