Connor Overton

Jason Young

Demetrius Huff

Peter Copus

Alex Anstine

**Software Requirements Specifications**

**Purpose**

This project will entail creating a proof of concept (POC) “Competitive Coin Counter” game. The Money Museum at the FRBKC already has a game in which visitors use pictures of coins to compete against each other to add up to a designated amount. This is a simple but highly competitive game for visitors of all ages.

**Scope**

a) Explain what the software product(s) will, and, if necessary will not do;

The software product will take the form of an intuitive, educational, web-based game app.

b) Describe the application of the software being specified, including benefits, objectives, and goals;

The software will include 3 games to choose from a main menu, each of which will have 3 difficulty levels to cater to leaners of all ages. Each game will have a leaderboard system accessible to those who have logged in, although guest play will be available.

**Definitions, acronyms, and abbreviations**

This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS.

Coin Levels

There will be three levels of coin, consistent across all games, tied to difficulty level. Each level will be inclusive of the previous levels.

‘Level 1’

The basic coins we see in day to day use. The penny, the nickel, the dime, the quarter.

‘Level 2’

Semi-common or easily understood rare coins. The half-dollar, the dollar, the liberty five dollar coin.

‘Level 3’

Rare coins the average person will not know about. The Palladium Eagle twenty-five dollar coin, the American Buffalo fifty dollar coin.

Working Game Titles

‘Coin Match’

* A ‘match’ will be defined as the corresponding pair of two cards, one of which is a number in currency format, the other being a picture of the randomly generated coins which add up to that number.

‘Coin Drag’

‘Coin Ninja’

**Specific requirements**

The web app will be fully operational on Firefox, Chrome, Edge, and Safari.

**Sign-In Options Page:** Upon opening, the player will be greeted by a sign-in screen, prompting users to log in, register an account, or play as guest. The application will make use of a MySQL database running on an XAMPP server.

**Register:** The user will be prompted to enter an email, a username, and a password, including confirmation for that password. If the username and email are unique, the account will be registered. Once the user registers an account the program will store their name, email, and hashed password in the database. The server will also assign them a unique ID that will also be stored in the database. All of these attributes will be in a table titled ‘Users’. There will also be three columns for the users high score for each game.  Now that user is registered, they will go back to the login page. The user will input their email and password, at which point the server will compare the hashed password.

**Game Select Screen:** After choosing a play mode, the user will be taken to a game select screen where they can choose between ‘coin match,’ ‘coin drag,’ and ‘coin ninja.’

**Difficulty Select Screen:** Upon selection of any of the three games, the user will be taken to a difficulty select screen with easy, medium, and hard. Upon choosing a difficulty level, the player will be shown a page or pop up displaying the coins they will be playing with and their value. The available coins are the same per difficulty level for each game\* – Coin level 1 for difficulty 1, Coin level 2 for difficulty 2, etc. When they hit ‘ok’ they will be taken to their game. Score will be determined by time taken.

\*Not all coins will be used at once in the ‘coin drag’ game, so only those randomly chosen will be displayed on this page.

**Coin Match:** This game is equivalent to the game currently used by the FRBKC. The user will be presented with several cards face down, determined by the difficulty level (Difficulty 1 will show 8 cards, Difficulty 2 will show 14 cards, and difficulty 3 will show 20 cards.) They will turn over two cards, and if they don’t ‘match,’ they will both be flipped back over. If they do match, both cards will disappear. There will be a timer running at the top of the page, and a counter for the number of flips required to match all cards.

**Coin Drag:** This game tests addition speed and familiarity with coins. The users will be presented with a value to match at the top of the screen, and a simulated bin in which to drop coins. These coins will be presented in a row at the bottom of the screen and refresh as dragged to the bin. A group of four coins will be randomly selected from those available in a given difficulty level.

**Coin Ninja:** This game shall consist of the images of coins rising from the bottom of the screen in an arc, only to fall again off the screen. The player will be presented with a value, shown at the top of the screen, and will tap the coins whose value they want to subtract from that total – these coins will then display some graphic, perhaps exploding or being pulled into the goal value at the top of the screen and then decreasing its amount by that coin’s value. The round will end when the value reaches 0.

**Game Completion:** Upon completing any of these three games, the user will be presented with a leaderboard screen, showing the top 10 scores, with the option to scroll through more, and then their score listed with their placement and the users above and below them. If the user is a guest, this will still have the same visuals, but the word \*GUEST\* will be displayed in place of a username, the score will not be saved, and there will be a small message saying “Your score was not saved because you were a guest. Please Sign In to save your future scores.” Below this will be options to play again, or to return to the game select screen. If the user is a guest, there will also be buttons which can take them back to the sign in or register pages.

Each of these pages will include a mute button in the lower right hand corner.

**- 2.1 External Interface Requirement**

The application will make use of a MySQL database running on an XAMPP server.

**Low Fidelity Sketches/Wireframes of potential designs**

This file is included in the .zip upload as a .epgz file and will require pencil.evolus to open.

**Functional Requirement**

Functional requirements should define the fundamental actions that must take place in the software in accepting and processing the inputs and in processing and generating the outputs. These are generally listed as “shall” statements starting with “The system shall…”

It may be appropriate to partition the functional requirements into subfunctions or subprocesses. This does not imply that the software design will also be partitioned that way.

(ID, Name, Description, Dependency)

A1, Sign-in type select

Description: The System shall prompt the user to sign-in (C1,) Register (B1,) or play as guest (D1.)

B1. Register

Description: The user shall be prompted to enter an email, a username, and a password, including confirmation for that password.

B2. Check Registration.

Description: The System shall ensure the entered username and email are unique values not currently present in the table.

B3. Create Object.

Description: the program shall store their name, email, and hashed password in the database. The server shall also assign them a unique ID that shall also be stored in the database. All of these attributes shall be in a table titled ‘Users’. There shall also be three columns for the users high score for each game.

C1. Sign in

Description: The system shall accept as input an email and password, and the system shall then compare the hashed password.

D1. Go to Game Select.

Description: This function shall bring the user to the game select screen. The game select screen shall record the users selection and take the user to the difficulty select screen (E1.)

E1. Difficulty Select

Description: This function shall record the users selection and send the user to the game corresponding to their selection in D1, either Coin Drag (F1,) Coin Match (G1,) or Coin Ninja (H1.)

F1. Drag Game Function

Description: The system shall run (F2) and display the total of the values of the selected coins. It shall also run a timer and a check-value function (F3.) The system shall display one of each coin and a bin, and shall allow the user to drag coins into the bin. When the timer runs out or the game has been completed five times, the system shall run (K1.)

F2. Coin Drag Initialize

Description: The system shall randomly select four to seven coins, based on the selected difficulty level, and multiply them each by a random number 0-6.

F3. Check-Value

Description: The System shall watch the bin and react when the value of the coins inserted is equal to the value given by F2. If they match, the system shall record a score multiplier, empty the bin, and run (F1).

G1. Coin Match Initialize

Description: The System shall randomly select between two and four coins, based on the selected difficulty level, and multiply them each by a random number 0-5. The System will repeat this process 4, 7, or 10 times based on difficulty level. The system shall create 8, 14, or 20 paired cards depending on the selected difficulty level using the initialized values, where the matching card is the value of the given coins, and then randomize their location in an array. The system shall then run a game function (G2.)

G2. Match Game Function

Description: The system will take the array of coins and values and create the proper number of ‘face-down’ card images. The user shall be able to tap on a card to temporarily reveal its value. When the user has flipped over two cards, if they do not match, the cards shall both be displayed for one second and then flip back over. If they do match, they shall both be removed, a multiplier shall be recorded for scoring purposes, and a new set of cards shall be created. This will continue until the time has run out or the game has been completed five times, and then the user shall be taken to an end game screen (K1.)

H1. Coin Ninja Initialize

Description: The system shall randomly multiply all coins within the difficulty range by a number between 0-6, then repeat some number of times. An array of the values of all these coins shall be passed into a game function (H2.)

H2. Ninja Game Function

Description: The system shall display the total of the values of the selected coins and run a timer. The system shall show randomly selected coins traveling in an arc from the bottom of the screen to a random height less than the maximum height of the screen. If these coins are clicked on, their values will be subtracted from the remaining value, and if the resulting value would go below zero there will instead be a time penalty applied. In addition to these coins, there will be some sort of negative objects which will also remove remaining time. If the value reaches 0 and time remains, the next number from the array shall become the new value to be reached and a multiplier shall be recorded for the score. When the timer reaches zero or the game has been completed five imtes, the game shall end and call the game end function (K1.)

K1 End Game Function

Description: The System shall determine the users score based on the score multiplier and the time remaining, if any was. It shall display this score, along with the users place in relation to other users, and offer to take the user back to the game select screen (D1) or to allow them to play the same game again (F1, G1, or H1.)