Technical Features of “Major Project”

# Product Documentation

## System Documentation

### Compatibility:

Devices: PC

Browsers: Chrome

Local files: html

Additional Hardware: Touchpad, Touchscreen, Laptop Keyboard (numbers key & enter key)

### Currently NOT Compatible/Untested On:

Devices: iPhone, Android, Windows, iPad

Browsers: Firefox, Opera, Safari, Bing

Additional Hardware: Keyboard, Stylus/Drawing Pad, Mouse

# Files

The files use the languages HTML, CSS and JavaScript. The central file (the main webpage) is “index.html” and the database file (ie. Results page) is “database.html”. The linked styling is “majorProject.css”, the project also uses JQuery with JavaScript, it is in a separate folder called “css”.

There are two JavaScript files for the Major Project that is currently used. “game.js” is for all the global and local functions and variables that also interact with the database, which is other file, “database.js”. These files are both located in the “js” folder.

There is also an “images” folder which contains all the images used in the game or once considered to be used.

# Functions

There are various “global” functions – generic functions that pass multiple parameters that are used many times. “startTimer” is the function that starts the countdown timer in the top right corner of the webpage. “checkSecond” is a variation of “startTimer” for the countdown timer to that the timer is correct and valid. Once the timer runs out, the webpage would redirect to the database.html page. “pickRandomObjects” is a function where three random index numbers (items) are generated from the array of items and is pushed to the array of random numbers for further use. “displayObject” is a function that displays the random generated items and their corresponding information on the webpage when it is opened by reading the array of random numbers. It is called multiple times within in the code to display the new updated information of items purchased or sold each time. “appear” is called when the user makes an error that is illogical. A message pops up on the screen to notify the user of their error and “disappear” is called to close the message off. Similarly, “appear2” is called when a required value is not entered by the user, a message pops up to remind the user to enter the value and “disappear2” follows so the user can close the message off. Whenever “appear” is called, “disappear” is also called, which is the same for “appear2” and “disappear2”.

The “purchase” and “sell” function stands for purchasing or selling a chosen item respectively, both with parameters item and amount. Item is meant for any one of the five items in arrItems and amount stands for the value the user entered for purchaseAmount and sellAmount. Each function contains logic to add or subtract the items’ price from the current currency and to change the amount of stock available or owned. The new values are then updated to the array of items under the corresponding chosen item. “purchase” is called whenever “btnPurchase” is clicked and “sell” is called whenever “btnSell” is clicked.

Functions “addResult” and “getResult” is used to store the result (ie. currency) and retrieve the stored values in the database in restdb.io and display them on the database.html page. “addResult” is called once “btnFinish” is clicked in the game.js page and “getResult” is called in the database.js page.

The function “submit” is to create a local array that stores the corresponding values that is going to be stored in the database. Which is called under a local function “$(‘#btnFinish’).click(function() {});”.

“local” functions – functions that are used generally once and are coded under other algorithms. The local functions under “$(‘#btnFinish’).click(function() {});” in order, is used to redirect the screen to database.html at a delayed time (“setTimeout”). The value stored in the database is then called through “getResult” function in the database.js page. The function “$(‘body’).keydown(function (event) {};)” matches the key number of enter key on the keyboard with the user input variables. Once the key is clicked, the input value is then stored to be called for later use. “$(“#myBtn”).click(function () {});” function is used to initialise the alert tab on the webpage and start the “startTimer” function after the user clicked “myBtn”. “$(“#btnNext”).click(function () {});” is to set how many rounds one game is going to have, and after that many rounds, the webpage automatically redirects to the database.html page.

“item.onclick = function (e) {};” with parameter “e” matches the element “id” of the array of items with the item the user chose. In this case, one of “btnUmbrella”, “btnRouge”, “btnJewellery”, “btnDessert” and “btnRice”.

# Variables

All variables and elements use camel case with the first term and capitalised letter for the second term if it exists without space between them (ie. itemClicked). All global variables are declared at the top of the file (or the top of the related function). All local variables are declared at the top in the algorithm.

## Arrays

There are multiple relevant arrays in the program. “arrItems” is an array that contains all the items that can be displayed in the “displayScreen” screen.

“pickedObjects” contains all the items that are to be displayed in the “displayScreen” screen. This array contains index numbers to make the HTML to display the correct items and ensure that there are no duplicates displayed.

“timeArray” is the countdown for the timer and “response” is the data stored in the database.

## Strings

“apikey” and “url” is used to connect to the database.

“str” is used to store the input of the user’s name.

In the “startTimer” function, “presentTime” returns the specified value of the ID attribute, “m” and “s” is the minutes and seconds in the countdown.

In the “pickRandomObjects” function, “numbersUsed” is a local string used to record and ensure that the used index numbers are not the same.

In database.js, “score” is used to display the relevant information of the scoreboard on the “displayScreen” screen.

# Databases

The game uses one database which is the majorproject (where the result will be called to display in descending order in the scoreboard). The majorproject database return the data as strings.

The Inventory database has a “UserName” (string that stores the user’s name and displays it in the form of User Name = $(“#myInput”).val()), “Currency” (currency (var) at the end of the game, the user sees it as Currency = currency).