Final Project: Who Wants To Be A Millionaire? Game.

Algorithm.

```
Define CreateText function, parameters: window, text, coordinates, color, font, style and size.
    Create a text based on coordinates, color, font size, style and size given by the user.
    Return said text in order to be drawn later.
Define mainMenu function, parameters: window
    Draw an image background from “bg1.png” file.
    Draw the buttons for “Play”, “High Scores” and “Exit” following the sketch 1’s design.
    Fill color = Black
    Outline color = White.
    Create text for each button, invoking the CreateText function.
        “Play” for the first button.
        “High Scores” for the second button.
        “Exit” for the third and last button.
    Draw the WWTBAM logo from the “logo.png” file.
    Obtain the user’s click on one of the buttons, assign a different value for each option, return said option to be assigned to a variable in the main function.
Define Interface function: window
    Draw an image background from “bg1.png”
    Draw the Question Table as shown on the sketch 2, following the fill color and outline set in the mainMenu function.
    Draw the WWTBAM logo from the “logo.png” file.
Define QuestionFormat function, parameters: window, question and coordinates (x and y.)
    Declare variable question as a string.
    Invoke the CreateText function, and assign the return value obtained to the variable question.
    Return variable question
```

```
Define Answer function, parameters: window, answer and coordinates.
    Declare variable answer as a string.
    Invoke the CreateText function, assign the return value obtained to the variable answer.
    Return variable question.
Define QuestionPaste, parameters: window, question, option1, option2, option3, option4.
    Invoke the QuestionFormat function, and assign the return value obtained to question1 variable. Use the parameter question as the “question” parameter in QuestionFormat function.
    Draw question1
    Add a sleep timer in order to make each option appear 1s after the last one.
    Invoke the Answer function for each option, assigning the return value obtained to ans1, ans2, ans3 and ans4 variables respectively. Use the parameters option1, 2, 3 and 4 as the “answer” parameter in Answer function.
    Draw ans1, ans2, ans3 and ans4, with a sleep timer of 1s between each drawing.
Define Click function, parameters: window.
    Declare and initialize variable “a” as an integer.
    Obtain the Mouse Click for each of the answer options, assigning a value ranging from 1 to 4 to variable a.
    Return variable a.
Define HiScoresTable function parameters: window, score, hiscore, a.
    Declare variable “b” as an integer.
    Draw an image background from “bg1.png” file.
    Draw the WWTBAM logo from the “logo.png” file.
    Draw the scores table following the sketch 3’s design.
    Invoke the CreateText function to create the texts for current score and high score. Use parameter score and hiscore as a replacement for CreateText’s “text” parameter.
    Draw the “Play” and “Exit” buttons following the sketch 3’s design.
    Invoke the CreateText function to create text for “Play” and “Exit” buttons.
    Depending on the stage of the game, make the “Play” button have a different text. Final answer must be “Play Again”, for the rest “Continue Playing.”
    Obtain the click for either “Exit” or “Play” button, assign a different value for each, and store said value into variable b.
    Return variable b.
Define HighScores function, parameters: window, hiscore.
    Declare variable b as an integer.
    Draw an image background using “bg1.png” file.
    Draw the WWTBAM logo using “logo.png” file.
    Draw the High Scores table following the sketch’s 4 design.
    Invoke the CreateText function to create the high score text, us hiscore as a replacement for CreateText’s “text” parameter.
    Draw the “Play” and “Exit” buttons.
    Invoke the CreateText function to create text for each button, having the “Play” button display “Back to Main Menu” as text.
    Obtain the click for either of the buttons, assign a different value for each and store said value into variable b.
    Return variable b.
Define main function.
    Declare and initialize variables.
        Choice, answer, scorec, playon, playag, scores, play, hscoresc as integers.
        Question1 and hscore as strings.
    Draw a window of dimensions 700 x 550.
    Open the “hscores.txt” as an infile and read its content.
    Assign the value to variable hscore.
    Create a loop to keep the program running until told to stop.
        Play the main theme song for the main menu opening from the “theme.wav” file.
        Invoke mainMenu function and assign its return value to variable choice.
        If choice’s value is 1, proceed to play the game:
            Invoke Interface function.
```

```
Draw the buttons for each answer option as seen in sketch 2.

Play the question ambient sound from the “qsound2.wav” file.

Create a loop to keep the program running until a choice has been made from the answer options.

Invoke Click function and assign its return value to answer variable.

Depending on whether the answer is right or wrong:

    If the answer is right:

        Play the correct answer sound from “qwin.wav” file. If this is the final question play the final correct answer sound from “finalwin.wav” file instead.

        Add the answer’s point value following the given points table: 100, 200, 500, 1000, 2000. Assign this value to the scores variable.

        Invoke the HiScoresTable function and assign its return value to variable playag.

        If playag new value is 2, end the program.

        If playag value is anything else, continue on with the program and repeat all of the above steps for each of the 5 questions.

    If the answer is wrong:

        Play the wrong answer sound from “qlose.wav” file.

        Invoke the HiScoresTable function and assign its return value to variable playag.

        If playag new value is 2, end the program.

        If playag value is anything else, continue on with the pgroam and repeat all of the above steps for each of the 5 questions.

    If the scores value is higher than the hscores value registered from the infile, assign scores’ value to hscore variable and rewrite the “hscores.txt” file with said value.

If the choice is 2, invoke the HighScores function, and assign its return value to hscoresc variable.

    If hscoresc value is 1, return to main menu.

    If hscoresc value is 2, exit the program.
If the choice is 3:
    Break the loop and close the window.
```