Visual Basic Game Program Specs

Design, pseudocode, and write a program that allows the user to play an entertaining, nontrivial game. You can write a "classic" game such as Hangman, Concentration, Blackjack or create one of your own design. The game must not present or suggest graphic violence (decapitation, blood, etc.) Do not include any inappropriate references or student names either. See Mr. Gage **before** you write the code if you have any questions. Your grade will be reduced if he deems anything inappropriate. Your game must follow all of the specifications cited below. You must turn in the code and a user manual (external documentation). You must also turn the source code with the complete project and associated files (graphics, text, sound). The game must execute properly from within the folder whether its executed from a floppy disk or your network folder.

You are also encouraged to see examples of VB games and programs that are found at sites like vbcode.com, vbgames.com, http://vbprogramming.8k.com/tutorials/main.htm and www.planet-source-code.com/vb. However, your final program must be your own work. Do not plagiarize code or algorithms from those sites. You must abide by the "Terms and Conditions" found in these code swap sites. If you do use any techniques or code segments from other sources, you must include citations and Web page addresses in your documentation.

You must first submit a detailed, typed **PROPOSAL** of your program in paragraph form. The proposal must explain how your program will work. Mr. Gage will keep this proposal on file and match it with your final program. If your final program does not match this proposal, you will lose points. However, if you feel that you need to make modifications at any point to your original design and proposal plans, you must immediately submit a revised proposal and obtain Mr. Gage's approval for the modifications in order to avoid losing points on your final grade.

Specifications - This assignment is worth a significant number of points. You will be graded in each of the following categories.

a/ splash screen

The splash screen (form) must include your first name (no last names) in a reasonable font size. It must also include the name of the game and a message indicating that how the user must continue (press a key or click the mouse.) You should make the splash screen interesting with graphics, color, and/or animation. Your project must be set to start up with Sub Main which must Show the splash form.

b/ user interaction

The game must involve user interaction. The game can be completely mouse-driven or it can require the user to type on the keyboard at various points. Perhaps, you can allow the user to enter his/her name and store that name in an external data file so that your program can "remember" the user and his/her previous score(s) the next time he/she plays.

c/ graphic image manipulation

Some graphics must be used in the program. This can be done with Image, Picture Box, Shape or other graphical controls. You may use clip art or graphics from the Web but abide by copyright laws. The graphics must be stored within your game's project folder.

d/ animation

You must include some animation in the program on the splash screen at least, if nowhere else. Your animation must be performed as efficiently as possible.

e/ user interface & accessibility

The user must have a pleasant experience running the program. Your program must include a standard Help menu with a submenu command Directions. Prompts and messages should be complete and understandable. The program should follow standard Windows conventions and provide access keys for command buttons and menu options. Be sure to set the TabIndex property values of text boxes and other controls properly. SetFocus must be used where appropriate and labels and prompt messages must guide novice computer users. The user must be able to exit the program at anytime, even if the game is not finished. You should also use ToolTipText properties where appropriate. Your program must also follow standard Windows conventions, especially those that we have studied in this course. For example, your menu commands and command buttons should have access keys that are consistent with those in commercial Windows programs. The order of your menu items should be consistent with commercial programs also. You should also use modal forms where appropriate. See Mr. Gage if you have questions regarding Windows conventions.

f/ menu

Your program must use a menu. As the first menu command,you must have a File menu with an Exit submenu command. You must have a Help menu at the far right of the menu. The Help menu must have Directions and About submenu options.

g/ 3 or more forms

Your project must include 3 or more forms, including the required splash screen. Besides the splash screen and your frmMain, you could also present the directions to your game on a form. You must use the form Hide, Show, Load, Unload events efficiently and correctly. You MUST unload all forms and close all sequential, random access, or binary access (audio, etc.) files when the project ends.

h/ standard code module

Your project must include a standard code module named stdCode (and saved as stdCode.bas). This code module must include Sub Main, which is the start up object for the project. The code module must also contain one or more functions and one or more general procedures.

i/ general function or general procedure

You must make use of one or more functions or one or more general procedures in your project. Your Main (i.e. Sub Main) procedure cannot count as this general procedure though.

j/ use of a data or sound file & program portability

You must use either a sequential access file, a random access file, and/or a binary access file in your project. You can write high scores, user names, or other data to a sequential access file. You can read data from a sequential or random access file. Or, you can play a binary access audio file. If your program reads from a file, you must make it very clear in the user manual and the games Help/Directions that the user must save certain files to the folder that the user is directed to create. If your program creates and writes to a file, make sure that it is stored in the folder that the user is directed to create when he first saves your program. You must use the App.Path property appropriately so that your project will execute when its folder is moved to any other location on the computer network or another computer. Both versions of your program, the VB project and the executable version, must be portable. See Mr. Gage if you have any questions about this requirement.

k/ required VB commands & structures

Your program must include at least one efficient, legitimate use of **at least 5** of the following in order to receive maximum credit. You must highlight statements and or evidence of the use of these 5 items in the code printout that you turn in.

Do loop, For loop, Select Case, Timer control, list (array), table (2-dimensional array), control array, horizontal/vertical scrollbar control, Time function, Timer function, and/or Rnd function.

l/ code and memory efficiency

Your program must use reasonably efficient logic and coding. For example, you must use loops, arrays, control arrays, Select Case statements, functions, general procedures, constants, etc. where appropriate. You must also use variable scope properly, using local and static variables where appropriate rather than module-level and global variables. Your program must also be as memory efficient as possible. Do not use unnecessary variables, arrays, and controls. You must also unload forms where necessary for the efficient operation of your program.

m/ proposal follow-through

Your final program must be reasonably close to your original proposal or the last approved modification that you submitted to Mr. Gage. Otherwise, you will lose points.

n/ originality

Your final program must be largely your own work. While you may model your game on a classic such as Hangman, you must add appropriate, interesting aspects to the game's logic and/or the interface of your project. If you learn techniques from other sources such as vbcode.com, you must implement them in your game in a fresh, original way. While you must follow the Terms and Conditions found at the site where you obtained snippets of code or ideas and cite their use and web site addresses in your documentation, you are still responsible for making the majority of your project you own handiwork and logic. See Mr. Gage if you have ANY questions in this regard.

o/ adequate internal documentation

The code for your program must be documented very well. Keep in mind that Mr. Gage may not know what you are attempting to do in certain sections of your code. The rule of thumb at this stage of your programming career is "the more internal documentation, the better."

p/ Coding Standards

Your program must follow our class coding standards. This means proper spacing and tabbing should be used throughout your program. Talk to Mr. Gage if you have any questions.

q/ external documentation (user manual)

A typed user manual must be submitted with the final program. The user manual must have a neat and attractive cover page with your name, the name of the program and graphical illustration. The user manual must be proofread for grammar and spelling mistakes. There is no minimum required length for the manual but it should include ALL of the instructions necessary to play the game. You could perhaps include strategy ideas for your game in the manual as well. If possible, you should include screen snapshots of key stages of your game. (Use the Ctrl/Print Screen key combination to copy the active window and paste it into MS Word or any graphics program.)

r/ beta-testing

You must have your program beta-tested by another student who has never taken Basic Programming at Mount Horeb. The beta-testers must fill out, sign, and date the official Beta-Testing Evaluation Form. You may have your program beta-tested before it is complete. But, enough of your game must work in order for the beta-tester to evaluate the basic parts. You will not be graded on how much the beta-tester liked your game but rather how honest and complete he or she is when they fill out the evaluation form.

s/ classroom productivity

You are expected to work on the game program assignment when class time is specifically devoted to the assignment. It is your responsibility to have finished planning (interface, properties, and pseudocode) your project outside of class (i.e. as homework). You should devote class time to testing and debugging. If you are unproductive during class periods devoted to the game program assignment, you will lose points.

Computer Program
Beta-Testing Evaluation Form

Name of Programmer: ____
Name of Program: ____
Description of Program:

Printed Name of Beta-Tester: ____
Signature of Beta-Tester: ____
Grade Level of Beta-Tester: ____
Date & Time of Beta-Testing: ____

Please answer the following questions. They answers should give the programmer constructive (helpful) criticism and advice so he/she can improve the program. Your comments will not affect his/her grade.

1. Did you have a pleasant experience running the program?

2. Did you find the instructions and messages given on the computer screen helpful? If not, explain.

3. Were you confused about any aspect of the program. If so, please explain.

4. Did the program "crash" when you tested it?

5. Would you like to see any improvements to the program? If so, describe them.

6. Make any other comments here: