GROUP MAJOR PROJECT

Date Given: May 31, 2020
Date Due: July 29, 2020
Weight: 40%
Group size: Five persons per group
Presentation Time Allotted: 10 minutes

This coursework will constitute 40% of the final module mark, and must be submitted by date mentioned on your course outline. Mark penalties will be applied for late submission (5% per working day), and extensions are *only* allowed in cases of *bone fide* and documented extenuating circumstances. You are required to do a 10-minute presentation of the program.

Each group is required to create a Github account to share codes among the team. https://github.com/

Introduction

Write a program to allow registered users to log in before they can do the Quiz. To log in, a user must correctly enter their UserID (string) and password (string). The user only get three (3) tries to enter their data correctly. After 3-failed attempts they are shut out of the system and your program will exit.

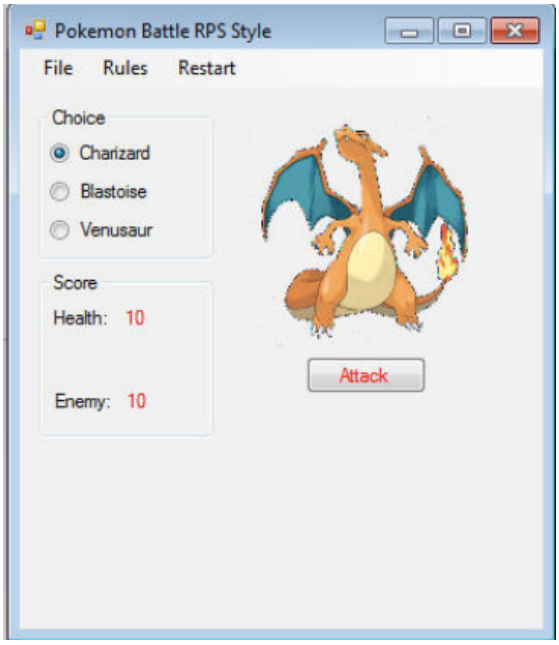
Project

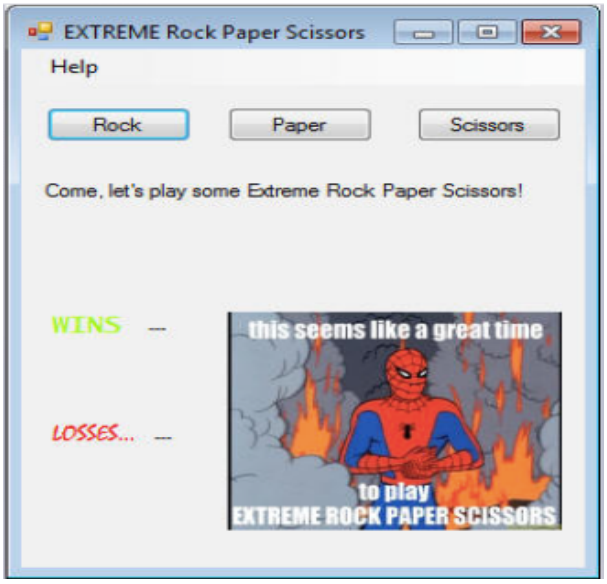
Create a rock-paper-scissors style application that has three (3) menu options File, Rules and Restart. Your program should have three pictures change depending on what object is selected. You do not have to use rock-paper-scissor pictures, but clearly state on the system which object beats which. Have a counter keep track of your total wins, losses, and ties.

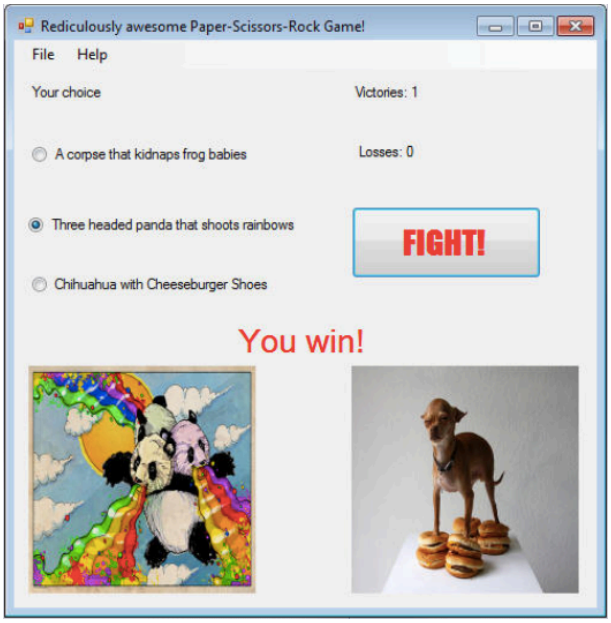
Use a radio button for the user selection. Use a standard button to submit the user choice. Display both user and the computer selection after the submit button has been clicked. Clearly identify who won the match. You must somewhere identify which object beats which (If different from the traditional Rock-Paper-Scissors).

Additions Program must include a counter to keep track of the total wins, losses, and ties. The program should prevent continuing play when the user has 10 losses or wins.

SAMPLE SCREEN







MARKING:

Marking scheme for the program:

Note that your overall mark will be affected by the following areas and should be borne in mind at all times:

It should be well designed.

This includes well-designed classes, cohesive functions, minimal use of global variables, effective use of parameters and so on.

Style and Readability
Programs are written not to just run, but also for others to read. Marks will be awarded for proper use of indentation, spacing, descriptive identifiers, and general neatness.

Documentation
There is both internal and external documentation. The internal documentation is contained within the program file(s) and includes a description at the top giving the inputs, overall processing, and outputs of the program. It also includes a comment section for each function, listing what is being passed into it via the parameters, what the function's main task is, and what values are being passed back out via the parameters or function name. Internal documentation also includes a description of each user-defined class. External documentation (readme file) is separate from the program and may consist of items like the specifications, record of testing and so on.

READ-ME FILE
Your read-me file should include the following:

1. The names of all the individuals who **worked** on the assignment.
2. Explain how your program works.
3. Include only source files.

Efficiency
Your program should not waste computer time or memory space. Make sure you don't hand in unnecessary codes.

Working Binary
The program must be able to compile and execute without errors.

SUBMISSION GUIDELINES

Your coursework must be uploaded by July 29, 2020. The following procedures must be conducted:

1. List all the members in your group.
2. DO NOT attach individual files and **exe files**. They MUST be zipped up as **PT_<your groupname>.zip.**

Submissions that are not collected due to your computer not working, Internet connection is down, viruses on computer, hard disk damaged, wedding, for example will incur 5% penalty.

To reiterate, all course works that are one working day late will be penalised 5%, two working days late 10% and so on (unless you are very ill or have valid case then a formal letter must be written supporting a request for extension).

PLAGIARISM

You are reminded of the policy regarding plagiarism **(also known as copying, cheating, or getting no marks)** *that:*

1. ALL codes will be checked thoroughly.

Please do not bother to plagiarize as it will just create problems for you, and more work for me, and I can **GUARANTEED** you, it **will** be discovered.

Programming Rubric – 100%

Criterion	Approx. % of Grade	Excellent (100%)	Adequate (80%)	Poor (60%)	Not Met (0%)
Program Specifications / Correctness	50%*	No errors, program always works correctly and meets the specification(s).	Minor details of the program specification are violated, program functions incorrectly for some inputs.	Significant details of the specification are violated, program often exhibits incorrect behavior.	Program only functions correctly in very limited cases or not at all.
Readability	20%	No errors, code is clean, understandable, and well-organized.	Minor issues with consistent indentation, use of whitespace, variable naming, or general organization.	At least one major issue with indentation, whitespace, variable names, or organization.	Major problems with at three or four of the readability subcategories.
Documentation	20%	No errors, code is well-commented.	One or two places that could benefit from comments are missing them **or** the code is *overly* commented.	File header missing, complicated lines or sections of code uncommented or lacking meaningful comments.	No file header or comments present.
Code Efficiency	5%	No errors, code uses the best approach in every case.	*N/A*	Code uses poorly-chosen approaches in at least one place.	Many things in the code could have been accomplished in an easier, faster, or otherwise better fashion.
Assignment Specifications	5%	No errors	*N/A*	Minor details of the assignment specification are violated, such as files named incorrectly or extra instructions slightly misunderstood.	Significant details of the specification are violated, such as extra instructions ignored or entirely misunderstood.