DSED-01 UI Based Simple Game

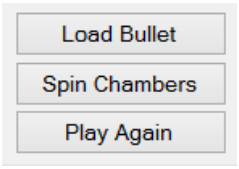
At the end of this project you will be able to

- Design and display UI interface
- Use Git Repository to store project
- Code using OOP

Instructions

This project involves playing Russian roulette with a gun, or any other method that meets the criteria specified below.

The program runs like this. (with a button for each step)



1 You **load** the bullet into the chamber of a revolver (not an automatic)

2 You **spin** the chamber to create a random place where the bullet is with a 1 in 6 chance.

3 You pull the **trigger** repeatedly until the gun fires through to the number where the bullet is stored.

4 The player has **2 chances** to shoot away during the game, this means that if the bullet is fired during that time they survive. If they shoot away twice and still the bullet has not been fired then they better make their will because the next shot they die.

What your project needs.

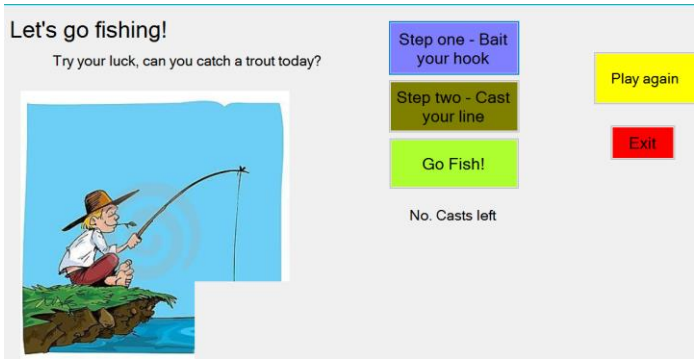
20%	**1 Form features**	
	1.1	At least one Image
	1.2	At least one sound
	1.3	Displays Win/Lose scores
30%	**2 Class Operations**	
	2.1	At least one class
	2.2	Includes Win/Lose class
	2.3	No variables in the code – all in the classes
20%	**3 Unit test**	
	3.1	At least two Unit Tests of major parts
30%	**4 Code Requirements**	
	4.1	All significant code to be commented
	4.2	Includes Win / Lose score
	4.3	Project to be hosted on Github (include your url)

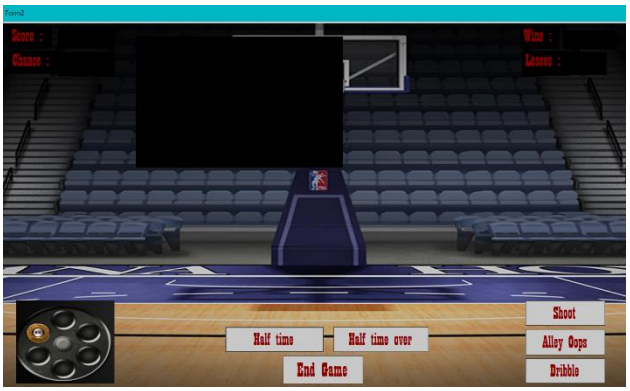
- You can and should use any examples we have made to date for help.
- You can use the net, (although that won't be much help).
- You can ask for help.
- But the SIGNIFIGENT MAJORITY of your program has to be your own work
- Look at the examples of past students work for ideas.

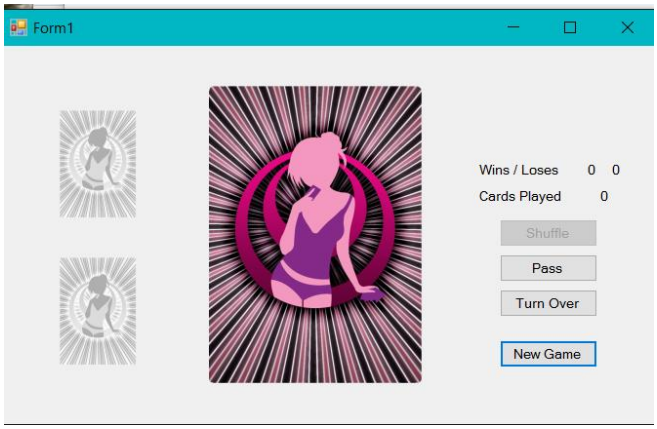
Be creative. **DO NOT make your program look like these images show.** Use your imagination and Coding Ninja skills to change it into something far more entertaining and surprise us.

You should pay attention to ensure that your code is well structured; appropriately commented and meaningful variable names have been used.

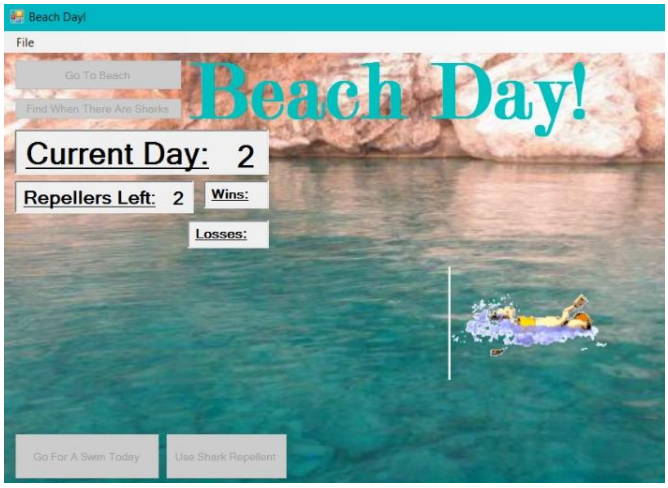
Examples of Students work

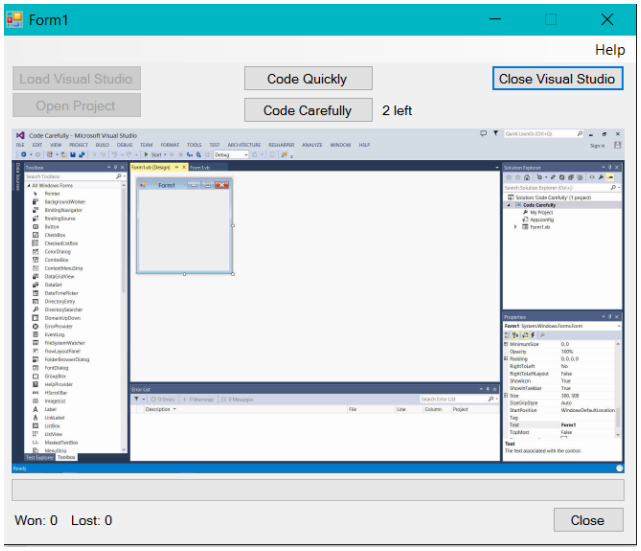






.





Marking Schedule

20%	**1 Form features**	
	1.1	At least one Image
	1.2	At least one sound
	1.3	Displays Win/Lose scores
30%	**2 Class Operations**	
	2.1	At least one class
	2.2	Includes Win/Lose class
	2.3	No variables in the code – all in the classes
20%	**3 Unit test**	
	3.1	At least two Unit Tests of major parts
30%	**4 Code Requirements**	
	4.1	All significant code to be commented
	4.2	Includes Win / Lose score
	4.3	Project to be hosted on Github (include your url)

Design Matrix

% of Grade	Excellent 100%	Adequate 80%	Poor 60%	Not Met 0%
Form Features				
20%	**No errors**, well thought out, and meets the specification.	**Minor details** of the specification are violated, program functions incorrectly for some inputs.	**Significant details** of the specification are violated, program often exhibits incorrect behaviour.	Program only functions correctly in very limited cases or not at all.
Mark	**20**	**16**	**13**	**0**
Class Operations				
30%	**No errors**, Classes are clean, understandable, and well-organized.	**Minor issues** with variable naming, or general organization.	**At least one major issue**, variable names, or organization.	**Major problems** with classes either totally absent or not used.
Mark	**30**	**24**	**19**	**0**
Unit Testing				
20%	**No Errors**. Both Unit tests work as advertised.	**Minor Issues**. Unit Tests work but don't cover all situations.	**One Unit test is missing**. The other works OK.	No Unit Tests.
Mark	**20**	**16**	**13**	**0**
Code Requirements				
30%	Code well commented, Hosted on Github. Good Win/Lose	Some Code comments, Hosted on Github. No Win/Lose	No code commenting Hosted on Github.No Win / Lose	No comment, no hosting..
Mark	**30**	**24**	**19**	**0**