School of Applied Sciences And Technology
Department of IST
Program: CNT
CMPE1666 –ICA05-Struct With GDI

In this ICA you will operate a List of user-defined `struct` to draw and manipulate some lines.

Create a standard Forms application supported by the `GDIDrawer`.

Part A

You will want the following behaviour:

When the user left-clicks in the drawer window, that point will be saved, and the next click will be used to complete a line. The line will be drawn 5 pixels thick, in red, from the starting point to the ending point. The cycle will repeat, using a completely separate set of clicks to define the starting and ending points respectively. The points will need to be saved for subsequent activity in Part B.

To facilitate the above, you will use the following elements in your application:

- Create a new `struct` called `SLine` that consists of members for start and end as `Point`, line color as `Color`, and thickness as `byte`. You may put this structure definition within the form class definition.
- At form class level add a `CDrawer` variable (800 x 800, continuous update off). Add a `Point` variable to capture the line starting position. Add a `List<SLine>`, initialized, to hold all lines.
- Add a `Render` method to the form that can accept a `SLine` instance and render it to the drawer.
- Add a `Render` method to the form that takes no arguments and can render your entire list to the drawer, clearing before, and rendering after. This overload will leverage the parameterized `Render`.
- Add a 50ms, enabled timer to your application to poll the drawer for mouse click events. In the timer event, use `GetLastMouseLeftClick` to poll for a left-click event. If this is the first of two clicks, simply save the point to the starting point field at class level. If this is the second of two clicks, create a new `SLine`, using the starting point, current point, and fixed values for color and thickness. To manage what is the first and second click, create an enumeration as shown, and an instance of it to keep track of what state the application is in:

```
private enum eState { State_Idle, State_Armed };
```

- If you are waiting for the first click, the application will be in the '`State_Idle`' state. Once you have the first click, move to the '`State_Armed`' state. Once you have the second click, go back to the '`State_Idle`' state. For each change of state, the resulting state must be displayed in the main form.
- Once you have the second click, you may build and populate an instance of `SLine`, render it, and add it to the list. There is no need to completely render the entire list, just the one line you are adding.

Part B

Add the code necessary to poll for a right-click in the drawer. If the user right-clicks in the drawer, you will change the color of every existing line to a new random color and set the thickness to a random value between 1 and 15 inclusive.

Because the collection is made up of value types (all structures are value types), what is returned from the collection through virtually every operation is a copy of the value. You won't be able to change any of the individual properties of the elements as a result, as you are changing a copy local to the expression.

To get the change you want, you will need to iterate over the collection, and build a new collection with modified values. Do so.

When you are done, you may assign the class-level collection the object reference of the new collection.

Use the no parameter `Render` method to render the results.

Rubric- Max Marks: 30

This application will require visual inspection of functionality and code.

Mark loss is at your instructor's discretion but will be applied consistently across all students.

Item	Marks	Penalties
UI Design (4) • State of application (idle, armed, displayed on main form – see video) • GDI Drawer Window created	2 2	
Code Design and Implementation (26) • Tick event correctly polls drawer for new click activity (Both left and right clicks). • Left-click changes state on GUI. 1st click store points second click draws line • Tick event correctly transitions state on left-click. • Right-Click correctly changes all lines and redraws them	8 8 2 8	Enumeration not created and used correctly: -3 Point management for start and end of line not correctly implemented: -4 Collection of SLine not created and managed correctly: -4 Parameterized and non-parameterized versions of Render not correctly used: -8.
Documentation: • Programmer Block • Well commented code • Appropriate Variable Names • Proper spacing between blocks of code		Penalties: -1 to -6 based on instructor's judgement