DAT154 – Spring 2022 – Assignment 1

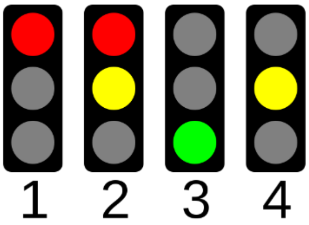
Purpose:

Graphics and timers in SDK.

Part 1: Make a simple traffic light

Make a static traffic light that is drawn on the screen. Use brush and circles. The traffic light is normal with red, yellow and green.

The logic for making the traffic light must be put into the WM_PAINT part of the program. The traffic light shall change state by pressing down the left mouse button (WM_LBUTTONDOWN). Study **InvalidateRect**, for example on Google.
Lights should change according to the normal pattern for traffic lights (Red -> Red+Yellow -> Green -> Yellow -> Red -> …)



[http://en.wikipedia.org/wiki/File:Traffic_lights_4_states.svg])

Part 2: Make two traffic lights.

Make two simple roads that cross each other and place two traffic light by the crossing. To simplify, traffic is only in one direction (top->bottom, left->right). The traffic lights should change when pressing down the left button (WM_LBUTTONDOWN).
Note that you don't need to draw the actual traffic yet, only lights and the road itself.
See the image on the next page for an example of how this may look when finished

Part 3: Timer

Study WM_TIMER and general about timers in the help system. Use timers to simulate the traffic lights so that the traffic lights change automatically. Select your own interval.

Part 4: Cars

You shall now introduce cars. Cars shall arrive from west by pressing left mouse button down and cars from north by pressing the right button. Cars may be drawn as a square or an X. Decide for yourself if you want to make the program object oriented or not.

The cars must follow the traffic rules, that is stop on red light at the crossing, and drive when the light is green, as well as not crashing into each other (While a stack of cars can be a legit data structure, it is not a desirable outcome in the real world). Use a timer for updating the position of the cars.

Part 5: Automation

Now change the program so cars from west arrive with a probability *pw* per second, and cars from north arrive with probability *pn*. *pw* and *pn* should be set in a dialog.
You now have a traffic simulator and can study how queues build up in the traffic crossing.

Part 6: WM_KEYDOWN

Read about WM_KEYDOWN. Make the program so that you change the intensities *pn* and *pw* with +-10% by pressing the following keys

- left/right arrow (west=*pw*)
- up/down (north = *pn*)

You can now change *pw* and *pn* by pressing these keys described above.

Just an example for how the final program can look. You don't have to try to make it look exactly like this, your program can be more or less fancy, bigger cars, larger roads, bitmaps, fewer lanes, whatever you want, this is just to give you a very rough idea.