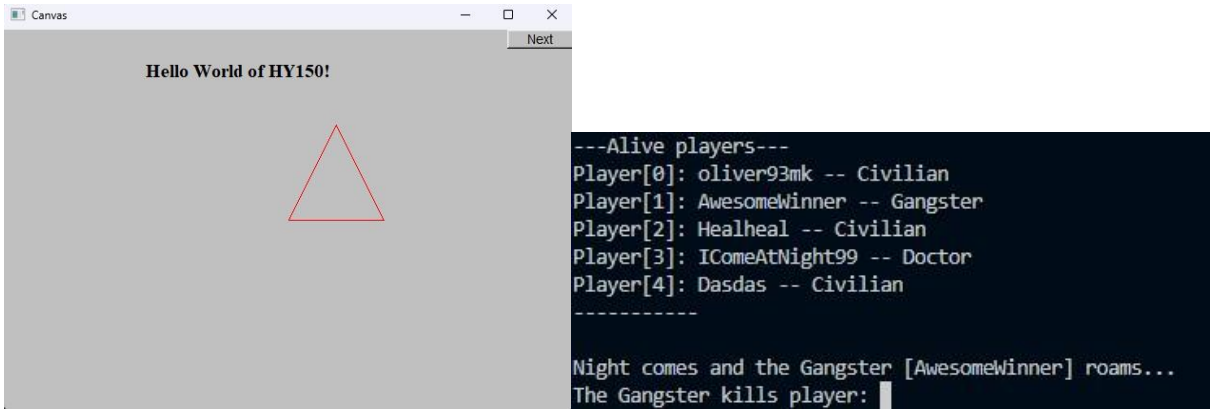
CS-150 2023

3rd Assignment
31/03/2023 - 28/04/2023

In this assignment you have to simulate the operation and display of **a round** of the Town of Salem game.

Part 1) *Build FLTK* (10%)

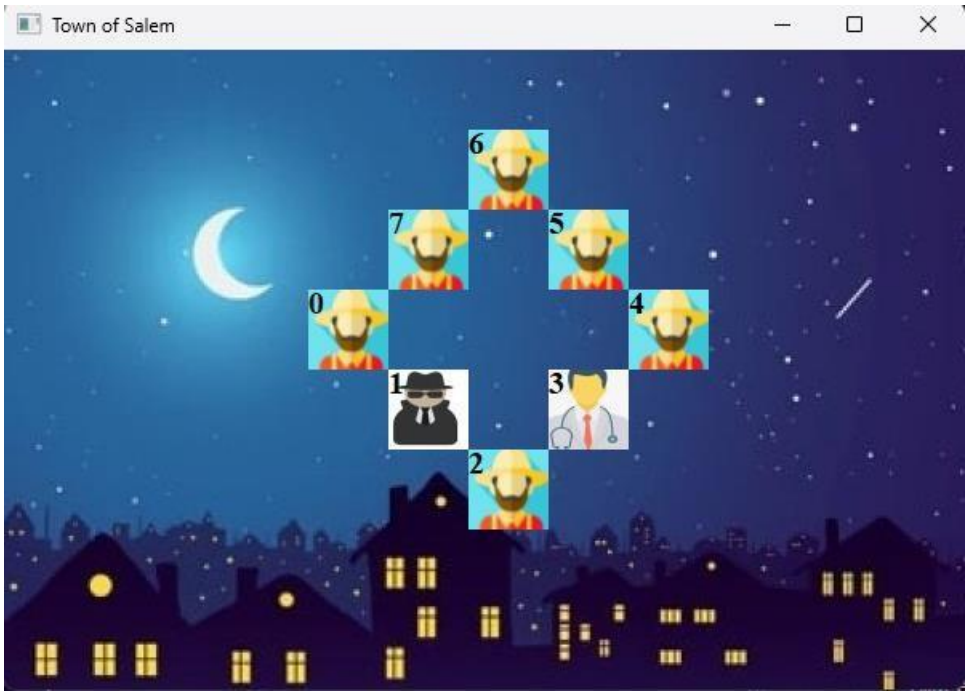
Build FLTK and GUI lib (the chapter12 example) together with your project. First, the example window should be visible and when you press the “Next” button the game you made in the 2nd exercise should continue running.



Part 2) *Players* (30%)

Modify the window to initially have a background image and an image for each player that you will, again, read from the Players.txt file. Depending on the player's role, the corresponding image will appear. The roles are Doctor, Gangster, Civilian.

Images should always be placed in such a way as to form a circle. This circle must have a different form depending on the number of users (which is variable and determined by the file).



Part 3) *Input Box* (15%)

Create an input box and a submit button. The input box will always be given an integer that refers to one of the players shown on the screen and will be removed when you press the submit button. Depending on the state of the program the input will be used differently.

Part 4) *Game State* (15%)

Implement the logic of the game by dividing it into phases. The phases will be the same as those from the previous exercises, namely:

• The gangster chooses which player to leave

• The doctor chooses which player to visit

• The day started and player X left or the doctor saved player X

• Player X voted for Player Y

Add a label that will announce what phase the game is in or what the result of the previous phase was.

Finally, when a player leaves you should make sure to properly update the circle of players (removing his image and his index).

Tip1: When refreshing the state of a widget you should redraw the main window, using the redraw function.

Tip2: Corresponding to the attach function that adds a widget, there is the detach function that removes a widget.

Part 5) *Voting (20%)*

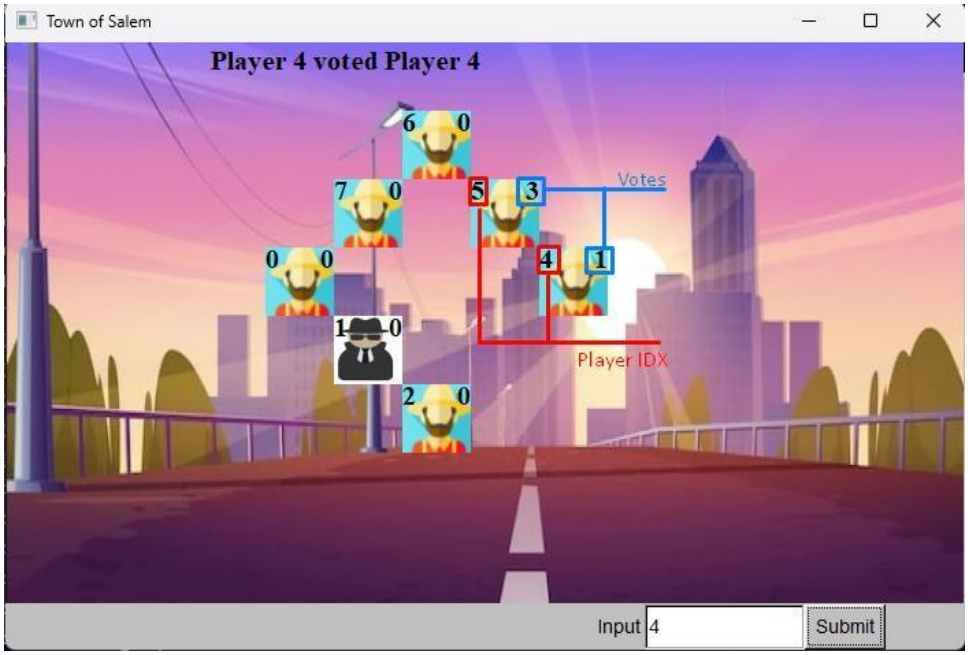
When the day starts, the voting process will start soon after. Add a label to each player that will show how many votes they have.

As the players vote, the number of votes should be updated on the corresponding label of the player who was voted. At the end of the process you should remove the player who had the most votes.

Finally, you should check if the player index the user gave you exists but you can assume it will always give you a number.

When the voting process is finished the user should be able to close the GUI window and continue the game from the command line (as implemented in the previous exercises). You can disregard this round and start over.

You don't need to take into account tie situations, just remove one of the players with the most votes.



Part 6) ***BackGround Change (10%)***

When the game phase changes from night to day, change the background to the corresponding image.

Tip: fltk paints the widgets in the order they were attached as the background will be done before other labels such as player images. That's why when you change the image it will be painted over the already existing widgets. To solve this you will need to refresh this order.

General Tip: Create widgets with dynamic memory allocation, that is, using new and pointers. It will make it easier for you to be able to use your widgets in different functions.
Note: For all the images you will need, use the ones provided by the course on elearn.