SCALE FOR PROJECT GUIMP (/PROJECTS/GUIMP)

You should evaluate 2 students in this team

★

Git repository

vogsphere@vogsphere.21

Introduction

In order to maintain high evaluation standards, you are expected to:

- Remain polite, courteous, respectful and constructive at every moment of the discussion. Trust between you and our community depends on your behaviour.

- Highlight the flaws and issues you uncover in the turned-in work to the evaluated student or team, and take the time to discuss every aspect extensively.

- Please take into account that discrepancies regarding the expected work or functionalities definitions might occur. Keep an open mind towards the opposite party (is he or she right or wrong?), and grade as honestly as possible. 42's pedagogy only makes sense if peer-evaluations are carried out seriously.

Guidelines

- You must grade only what exists in the GiT repository of the student or team.

- Remember to check the GiT repository's ownership: is it the student's or team's repository, and for the right project?

- Check thoroughly that no wicked aliases have been used to trick you into grading something other than the genuine repository.

- Any script supposed to ease the evaluation provided by one party must be thoroughly checked be the other party in order to avoid unpleasant situations.

- If the student in charge of the grading hasn't done the project
yet, it is mandatory that he or she reads it before starting the
evaluation.

- Use the available flags on this scale to tag an empty work, a non
functional work, a coding style ("norm") error if applicable,
cheating, and so on. If a flag is set, the grade is 0 (or -42 in
case of cheating). However, cheating case excluded, you are encouraged
to carry on discussing what went wrong, why, and how to address it,
even if the grading itself is over.

- The whole group has to be present.

Attachments

Subject (https://cdn.intra.42.fr/pdf/pdf/6093/GUImp.en.pdf)

Préliminaries

Preliminaries

Check the following points :
- There is a valid `author` file
- There is a `libui` folder with his own Makefile
- There is a `libui.a` created after compilation
- There is a Makefile at the root of the repository which compiles the 'guimp' executable

If the preliminaries are OK, we can continue the evaluation.

Remember that for the duration of the defence, no segfault,
no other unexpected, premature, uncontrolled or unexpected
termination of the program, else the final grade is 0.

Yes No

Libui part

This section of the scale will be used to evaluate the successful completion of the graphical library part of the project named `libui`, for that we will also use the 'guimp' part in order to prove the good implementation of the features. For this section please take the time to evaluate the quality of the implementation of all the features requested in the subject. Feel free to look at the code and ask the evaluateds to explain their implementation if it's necessary. If one of the feature is not used in the 'guimp' part the evaluateds are authorized to prove by other means, and this at the discretion of the corrector, that their library has this feature.

You said "Library" ?

Launch the Makefile of the `libui` part, and check that
the 'libui.a' is successfully created.

☑ Yes ☒ No

Windows

Check the following points (1 point per item):
- There are several windows of different sizes
- At least one window is resizable
- At least one window has an image or a background color
- Several windows have a given theme or style
- Different types of windows are present: generic, modal...

Rate it from 0 (failed) through 5 (excellent)

Simple elements

Check that the following elements are present (1 point per element):
- Buttons
- Menus
- Text
- Images
- Editable text fields

Rate it from 0 (failed) through 5 (excellent)

elements ++

Check that the following elements are present (1 point per element):
- Checkbox
- Slider
- Radio
- Drop-down list
- Progress bar

Rate it from 0 (failed) through 5 (excellent)

Style

Check that at least 3 of the above elements can take as parameter (1 point per element):
- One size
- A color
- A shape
- A background image
- A defined theme or style

Rate it from 0 (failed) through 5 (excellent)

The Niagara Falls

Check that several of the above elements can be put in "cascades".

examples :
- A button in a menu (verifiable by changing the position of the menu)
- A menu in a menu
- A menu in a menu, in a menu!
- An image and/or a text inside a button

Validate the point if at least 3 different cascades are present

☑ Yes ☒ No

Events

Judge the quality of implementation of the events (1 point per element):
- The libui must handle at least all standard mouse and keyboard events
(clicks, focus, one key, several simultaneous keys etc.)
- It must be possible to specify at least one type of event for an element
- Several types of event can be specified for a single element
- The action of the event can be standardized by the library (incremental, on/off etc.)
- The action of the event can be defined by the user (pointer to function e.g.)

Rate it from 0 (failed) through 5 (excellent)

Events ++

Judge the quality of implementation of these elements (1 point per element):
- Good cohabitation between event management and the elements in cascade or not.
- It is possible to have an interaction on an element that leads to the appearance and/or modification of another element
- A `HotKeys` system specific to each type of elements is available by default.
- A window can be scrollable if necessary according to the selected parameters.
- An element can be scrollable if applicable and necessary, depending on the chosen parameters.

Rate it from 0 (failed) through 5 (excellent)

Prefabs

Judge the quality of implementation of these features (1 point per feature):
- There is a file selection prefab
- There is a prefab of font selection
- There are other types of preabs like a status bar with info and images, a default menu for open/save/quit etc...
- Some elements (2pt) and by default the window (1pt) are able to manage drag-n-drop

Rate it from 0 (failed) through 5 (excellent)

GUImp part

This part of the scale will focus on the program named `guimp`, a 2D drawing software inspired by the famous `The Guimp`. First, please check that in ALL the source code of the part `guimp` there is no any direct function call to the SDL, system graphics framework or any other graphic management system, if this is the case, the defense is finished and put the `Forbidden functions` flag.

Basical tools

Check the correct implementation of these tools (1 point per tool):
- The brush
- The eraser
- Drawing lines, rectangles, squares and circles (empty and full!)
- A magnifying glass that zooms AND unzooms!
- A moving hand

Attention if the tool requires it, it must have an associated menu for
manage its variables like line thickness, radius for a circle
or its opacity. If the tool is not variable do not count the point !
(A fixed size eraser is not very useful..)

Rate it from 0 (failed) through 5 (excellent)

Tools ++

Check the correct implementation of these tools (1 point per tool):
- The famous Painting pot tool (2pt)
- A brush menu that allows you to set "stickers"
- A functional pipette
- A button that erases the entire working area or the selected area

If the tool does not work smoothly, do not count the point.

Rate it from 0 (failed) through 5 (excellent)

Menus

Check the correct implementation of these menus (1 point per menu):
- A stylish menu for color selection : HSV circle, RGB sliders...
- A menu to import several images to the work area
- A line thickness management menu
- A menu to put text at a given location, with the choice of the
color, font and size (2pt)

Rate it from 0 (failed) through 5 (excellent)

Details

Check the correct implementation of these features (1 point per feature):
- The mouse cursor changes according to the selected tool
- It is possible to import several types of image format including at least PNG and JPEG
- Click on the red cross on the window border to close the corresponding window,
and exit the program properly if this is the rendering window.
- You can save the created image in at least two different image formats (2pt)

Rate it from 0 (failed) through 5 (excellent)

Bonus part

This part of the scale will focus on the BONUS part of the subject. Attention if the mandatory part is not 100% validated the evaluation ends here!

Simple bonuses

Are the following bonuses present and functional? (1 point per bonus) :
- Menus can have tabs
- An interface style configuration file like a CSS
- The interface is resizable
- Other stylish stuff we never even imagined.

Rate it from 0 (failed) through 5 (excellent)

Stylish bonuses

Are the following bonuses present and functional? (1 point per bonus) :
- the possibility of Undo/Redo aka ctrl+z/y
- You can copy and paste a given selection
- We can manage different layers and this in a coherent way
- Interace is responsive
- Some other REALLY stylish stuff we never even imagined!

Rate it from 0 (failed) through 5 (excellent)

Ratings

Don't forget to check the flag corresponding to the defense

✔ Ok ★ Outstanding project

Empty work Incomplete work No author file Invalid compilation Norme

Cheat Crash Incomplete group Leaks Forbidden function

Conclusion

Leave a comment on this evaluation

Finish evaluation