Computer Labs: Project Report

2º MIEIC

Pedro F. Souto (pfs@fe.up.pt)

December 19, 2020

Documents to Submit

Final Report we'll talk about it today

- Same as the project's deadline

Self-Evaluation a Google form similar to the ones used for self-evaluation of the labs

- 24 hours after the project's deadline

Demo-Form Yet another form that you must fill **before your project's demo**

- Must include the URL of a demo video
- ~~**You must bring to your project demo a printed copy of the filled form**~~ (This does not apply this year.)
  - This form will help us to better assess the ceiling of your project
  - This is one of the criteria we'll use to grade your demo (in other words: if you do not bring a printed copy of this form filled, then you'll be penalized)

Doxygen Documentation this must be submitted via Git

Video demonstrating your project (if, possible, use audio as well)

Project Report: Goals

Show you've met the goals of the course (taken from the LCOM's description in SIGARRA)

- use the hardware interface of the most common PC peripherals;
- develop low level software and embedded software;
- program in the C language (in a structured way);
- use various SW development tools.

Help us grading fairly your project

- There is a lot "under the hood" that may go unnoticed **unless you bring it to our attention**
- It is very hard to grade the work of each group member

Your help can prevent us from being unfair in our evaluation

The project report is worth much more than the nominal 20%

We use it to estimate the complexity of your project and therefore the ceiling of your grade

Project Report: General Structure

1. User instructions
2. Project status
3. Code organization/structure
4. Implementation details
5. Conclusions

Appendix: Installation instructions (optional)

Setion 1: User's Instructions

How to use your project

- use it also to provide an overview of your project's functionality
- use (and abuse of) images ("1 picture is worth 1000 words")

1. Instruções de utilização do programa

1.1. Ecrã inicial

Fig. 1 - Ecrã principal.

Ao iniciar o programa, é apresentado um ecrã de introdução do programa com informação sobre o projeto, os autores e o seu contexto. O utilizador pode avançar carregando em qualquer tecla.

1.2. Menu

De seguida, o utilizador é redirecionado para o menu principal onde poderá usar o rato para selecionar uma das 4 opções:

Single: inicia a ferramenta de desenho em modo de desenho singular;

Multi: inicia a ferramenta de desenho em modo de desenho colaborativo

Gallery: inicia o modo Galeria onde se podem ver os desenhos já gravados;

Quit: fecha o programa.

Fig. 2 - Menu.

1.3. Modo de desenho

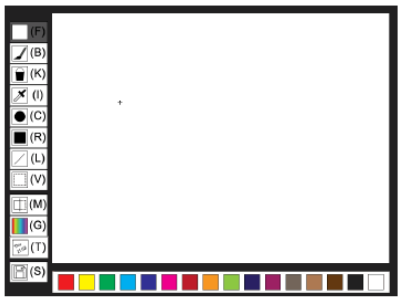


Fig. 3 - Modo de desenho.

Neste modo, o utilizador é convidado a desenhar livremente no ecrã, utilizando as ferramentas apresentadas. Junto ao ícone da ferramenta está, entre parêntesis, a tecla a utilizar para acionar a mesma ferramenta. Estão disponíveis as seguintes ferramentas:

1. "Blank" - Limpa o ecrã de desenho;
2. "Brush" - Permite pintar no ecrã uma linha utilizando o rato, com grossura alterável (pressionando as teclas '+' ou '-' no *numpad* ou junto à tecla *Enter*);
3. "Bucket" - Permite pintar o ecrã, de forma semelhante à ferramenta "Balde" do *paint* do *Windows*. O utilizador deverá selecionar no ecrã de desenho o local onde pintar;
4. Selecionador de cor - Ao clicar com o rato sobre uma cor (na tela de desenho), as ferramentas passam a trabalhar com essa cor. A cor de fundo do programa é alterada para a cor que está a ser usada;
5. Circulo - Desenha um círculo com dois cliques do rato, centro e extremidade;
6. Retângulo - Desenha um retângulo com dois cliques do rato, correspondentes a dois cantos opostos;

Section 2: Project Status (1/3)

What functionality did you really implement?

- May be easier to list which functionality mentioned in the previous section you did not implement;
  - If e.g. your GUI allows to choose some functionality that you did not implement
- **Must** include a table with the I/O devices you have used, what you have used them for, how did you use them (interrupt vs. polling). For example:

Device	What for	Int.
Timer	Controlling frame rate	Y
KBD	Menu navigation	Y
Mouse	Tool selection and drawing	Y
Video card	Application menus and screens display	N

Section 2: Project Status (2/3)

- Must include a subsection for each I/O device:
  1. describing the device's functionality actually used;
  2. **referring to the code** (function name) where you use it

Graphics card Should mention video mode (mention also resolution, color mode and number of colors) and whether you use:

- Double/triple buffering
- Moving objects (collision detection, animated sprites)
- Fonts
- VBE functions, e.g. to change the *palette* or for page flipping

Keyboard Whether it is used for:

- Game/application control
- Text input

Section 2: Project Status (3/3)

Mouse Whether your program uses:

- Position
- Buttons

and for each of them what do you use it for

RTC Whether it is used for:

- Reading date/time
- Generating an alarm
- Periodic interrupts

UART Should describe:

- Features used, e.g. interrupts or FIFOs
- Communication parameters used
- Data exchanged and exchange frequency

IMPORTANT **Unless you mention the features** you use in this section, you may not get credit for them

- But remember, you **must** specify the names of the relevant functions

Section 3: Code Organization/Structure (1/2)

For code you have developed

Must include one subsection per module (C ~~or assembly source~~ file), with:

1. "A one paragraph description" of the code contained in the module
   - Include also the relative weight (in %) of module in project
2. Information on who (group member) did what (in the module)

May also include a description of the

- main data structures per module;

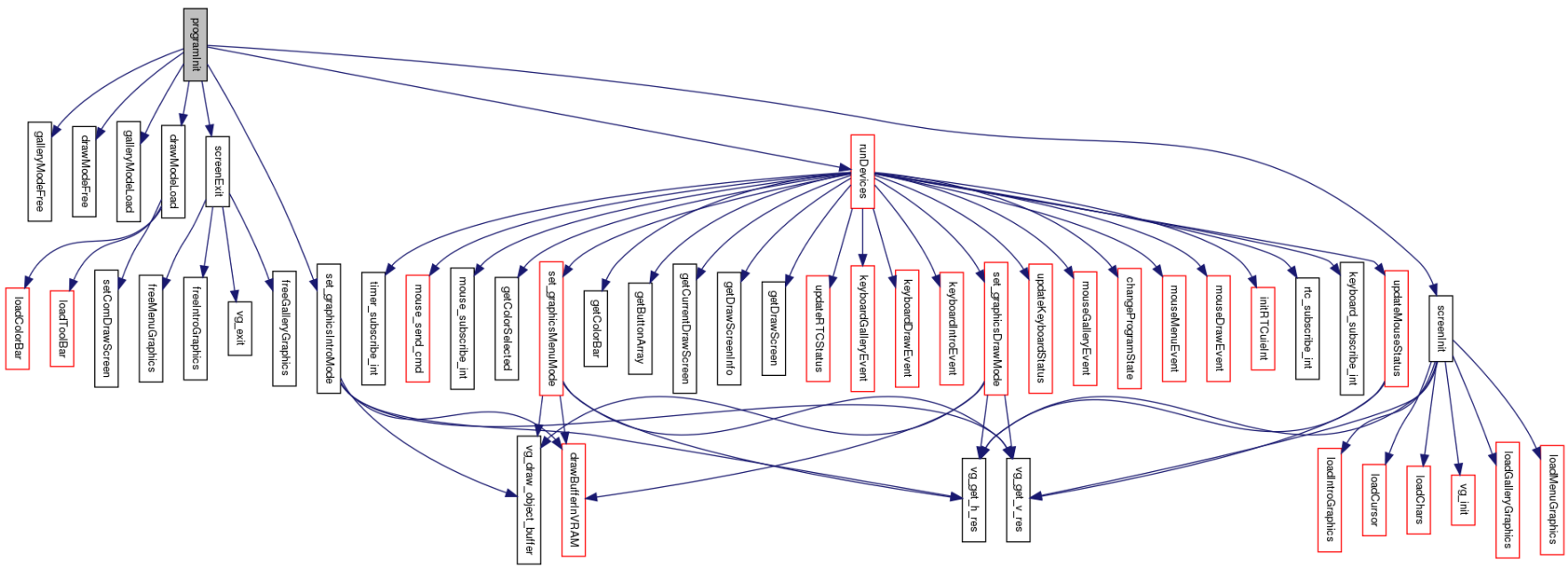
For code you have downloaded from the net

- Include one subsection per module/function, with:
  1. "A one paragraph description" of the code contained in the module
  2. A description of the changes you had to make, if any
  3. The URL you got the code from

Section 3: Code Organization/Structure (2/2)

Function call graph

- It can be generated automatically by Doxygen
  - May want to limit the depth so that the graph generated can be included in a figure in the report.
    - In this case, include the full Doxygen documentation in the Git repository
  - Include also a short description of the main functions
    - These must include functions that call `driver_receive()`



Section 4: Implementation Details

Implementation details this is where you can show your domain of the course's topics. In general, you should speak about topics:

- That were covered in the lectures, but that required some ingenuity in their application to your project (e.g. layering, event driven code, state machines, object orientation, frame generation, ~~assembly code,~~ ...)
  - Details regarding the use of the RTC and the UART are also important
- That were not covered in the lectures/labs and that you had to learn by yourself (and may be you wished we had talked about it) (e.g. collision detection, ~~call of assembly functions from C~~, ...)

Section 5: Conclusions

Course evaluation

- Mostly the bad (with suggestions for improvement)
- But also the good, if you feel like (so that we do not change things that you liked)

(The self-evaluation form includes also a field to provide this information, so you may not include this information in the report.)

Appendix (Optional)

Installation Instructions required only if we need to do something else other than invoke `make` in your project's top directory

- If you use files, avoid "hard-code" them: specify the directory with those files via the command line arguments.
  - This is a major source of "head-aches" and delays in the final demos.

Project Report: Final Recommendations

- Remember, the project report is worth 20% of your project's grade
  - And it also affects other aspects of your project's grade
  - In the self-evaluation form, there is a field to specify your contribution to the project report.
- Do not leave it for the last minute:
  - Start writing the report **now**
  - Write it incrementally
    - Most of the information we ask for is available rather early in the project
  - You can always review it, if later you change something already mentioned in the report
    - I.e. you can use an iterative approach
  - You can leave "refinements" to closer to the deadline
    - For example, do not worry much with the writing style in early versions
- If you want to provide us further information that does not fit in any of the sections enumerated, add new sections
  - But include all the sections mentioned (with the requested info)