FINAL PROJECT REPORT

FABIO MOREIRA, UP201806296

LUIS SILVA, UP201808912

FEUP – MIEIC

REPORT REGARDING THE PROJECT DONE

UNDER THE SUBJECT OF COMPUTER LABORATORY, IN 2019/20

INDEX

About:

This project was developed as part of the Computer Labs course of the Master's in Computing and Informatics Engineering, of the Faculty of Engineering of the University of Porto.

It consists in the development of a game (single player) like the known game *Hangman*. The user can play several times, where they try to guess a word, randomly generated, using the keyboard keys.

The goal is to correctly guess the greatest number of words in a row. However, the player will only have 5 failed guesses per word, as well as limited time.

1. USER INSTRUCTIONS

1.1. INITIAL MENU

When starting the program, the user will see the main menu. On the bottom part the current date will be displayed, as well as the current time. The player will also see the current high score (highest number of words guessed correctly in a row), as well as the instructions to play the game (press the left mouse button) and to exit (press the escape key).

Fig. 1 – Main Menu

1.2. GAME

When the game starts, the user will have 60 seconds to guess the word, by pressing the keyboard keys. If the letter exists in the word, it will appear in the corresponding position. Otherwise, the number of failed attempts will increment and the letter will appear in the screen, so that the player knows that they already tried that letter (pressing an "already guessed" key will do nothing, i.e. if in the situation below the letter A was pressed, nothing would happen).

The player will lose if they reach five failed attempts or if they run out of time, which is displayed on the screen on countdown.

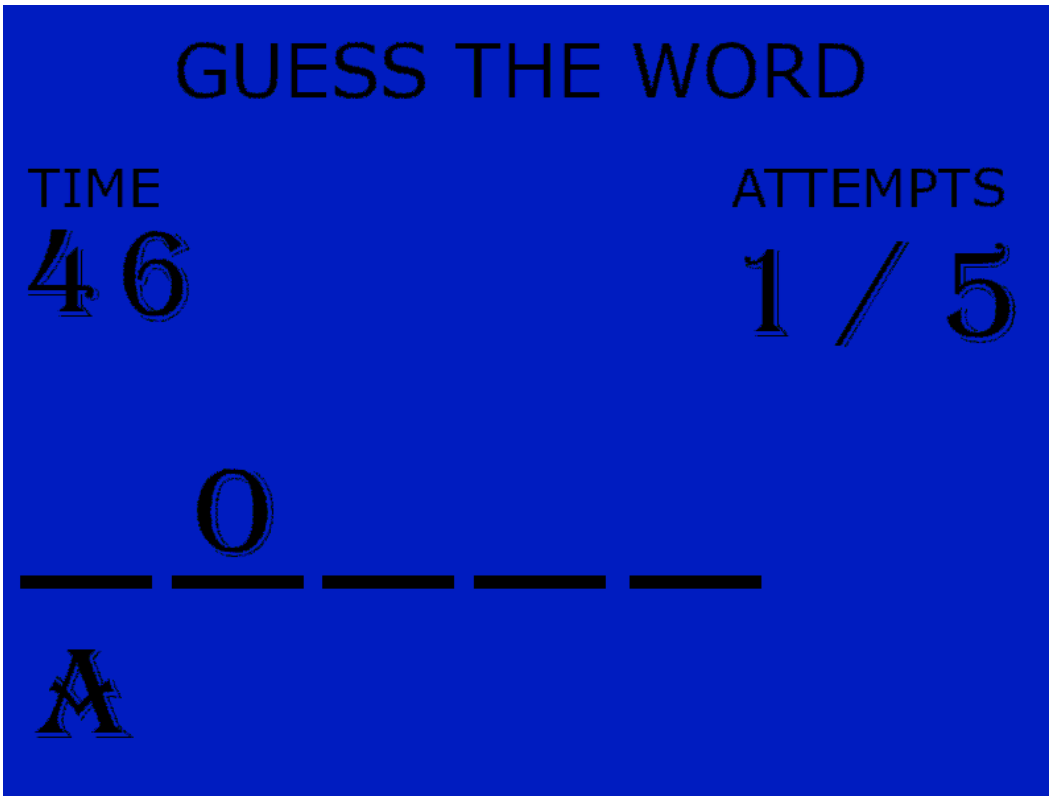


Fig.2 – Game screen

1.3. VICTORY/DEFEAT SCREEN

Fig.3 – Victory screen

Fig. 4 – Defeat screen

As you can see in the pictures above, when guessing the word correctly, the current score will be displayed on the screen. If it is higher than the high score, the message "You have a new high score" will appear, as seen in *Fig. 3.* The high score is saved in a text file "*highscores.*txt".

If the user runs out of attempts or time, the correct word will be displayed, and the score the player had up until that point (*Fig. 4*). The current score will be reset.

In either one of the situations, it is possible to exit the game (by pressing the *escape* key) or to (re)start the game (by pressing the left button of the mouse).

2. PROJECT STATE

Device	What for	Int.
Timer	Controlling round time	Y
KBD	Guess a letter or exit game	Y
Mouse	(Re)start the game	Y
Video Card	Screen display	N

Timer:

Functionality Used: Timer Interrupts

Function names where we use it: *mainMenu*, *gameFlow*

Keyboard:

Functionality Used: Keyboard Interrupts

Used for text input, to get what letter the user pressed, but also for game control, because pressing *ESC* in certain situations will terminate the game

Function names where we use it: *mainMenu*, *gameFlow*, *victoryScreen*, *gameOver*

Mouse:

Functionality Used: Keyboard Interrupts

Uses the mouse buttons, more specifically, the left one, to (re)start the game.

Function names where we use it: *mainMenu*, *victoryScreen*, *gameOver*.

Graphics Video Card:

The fonts are included in the necessary images.

Video Mode: 114h

Resolution: 800x600

Color Mode: RGB 5_6_5

Function names where we use it: All of them except *getRandomWord.*

RTC:

Used to display the date/time in the main menu.

Function names where we use it: *mainMenu*.

3. CODE ORGANIZATION AND STRUCTURE

3.1. FUNCTION DESCRIPTIONS

Code implemented to handle the game is present in the *proj.c* file.

Function: *mainMenu*
- Description: Function to display the main menu of the game. It will:
  Display current date and time
  Display the current high score
  Display the options and instructions (to play or exit)

- Developed by Fabio

Function: *getRandomWord*
- Description: Generates a random word from the available list of words

- Developed by Luis

Function: *gameFlow*
- Description: Main function to handle the game.
  It will control the remaining time as well as the attempts remaining.
  If the player loses (runs out of lives or time), it will call the function *gameOver*.
  If the player wins (gets all the letters right), it will call the function *victoryScreen*.
  As the player tries the different letters, it will draw them in the adequate position.

- Developed by Fabio

Function: *gameOver*
- Description: If the player ran out of times or lives, they will be presented with this screen.
  It will present the score they had before they lost (unless it was 0), and it will show what was the right word.
  It will also present the options to restart the game or quit

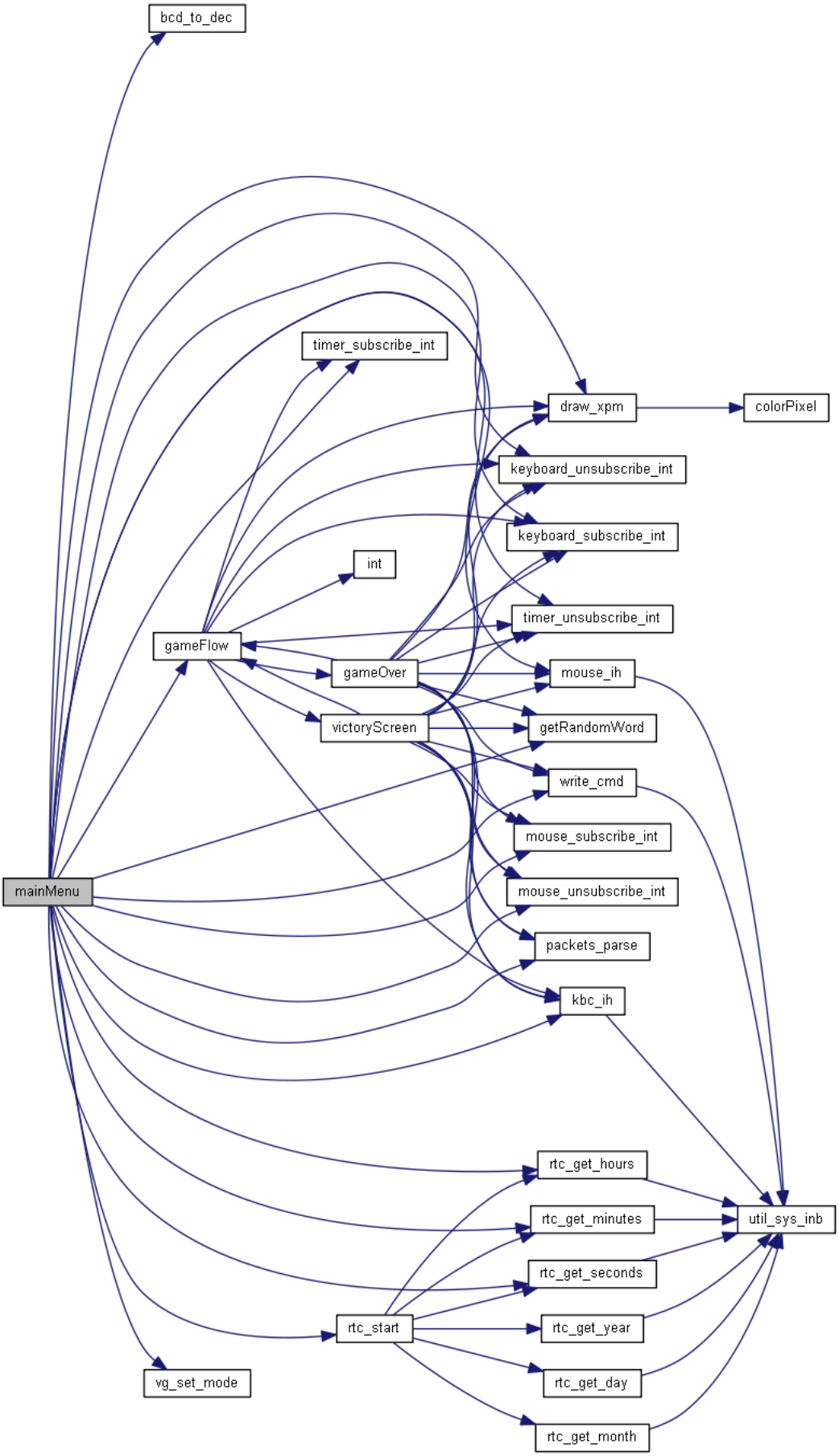
- Developed by Fabio

Function: *victoryScreen*
- Description: If the player got all the letters right, they will be presented with this screen.
  It will display the current score, and will let know the user if they beat the high score.
  It also presents the options to either keep playing or quit.

- Developed by Luis

Final contribution: Fabio: 65%, Luis 35%

3.2. FUNCTION CALL GRAPH:



4. IMPLEMENTATION DETAILS

RTC:

Even though the RTC is a simple peripheral, it has some singularities attached that need to be taken into account to avoid errors.

In this project, we got the date through polling, making sure that none of the other registers were changed while we accessed a certain one. Therefore, if the UIP flag was set in register A, it meant there was an update in progress and, therefore, we should not access the time/date registers.

5. CONCLUSIONS

POSITIVE:

It's a very interesting subject, where we have the opportunity to learn hands-on how to program and work with several peripherals, as well as increase our knowledge over many C programming topics.

If we struggle with one of the peripherals, we will only be graded on 3 of the LABs.

NEGATIVE:

We believe the 6 credits don't reflect the work we had throughout the semester, in comparison to other 6 credits subjects.

We don't know how we are doing throughout the semester since we didn't get the LABs grades while we were working on them. Even though we understand why they take long, it would give us a sense of security regarding our work.