18COA202 Embedded Systems Programming Coursework

Specification

In this coursework, you are given an Arduino Uno and LCD keypad shield. Your assignment is to develop a software that simulates a virtual pet.

- On starting, the software checks whether a saved virtual pet from a previous run is present in the EEPROM.
  - If a saved virtual pet is present, the software offers the choice between loading the saved virtual pet and starting with a new virtual pet.
  - If no saved virtual pet is present, a new virtual pet is started.
- As its default behaviour, the software displays all the statistics of the virtual pet (see next point) on the LCD. The exact way of displaying these is your choice.
- The virtual pet has the following statistics:
  - *Development stage*, which is 0 (egg), 1 (young), or 2 (adult).
  - *Happiness*, which is 0 (unhappy), 1 (content), or 2 (happy).
  - *Fullness*, which is 0 (ravenous), 1 (hungry), 2 (peckish), 3 (full), or 4 (overfull).
  - *Age* (the time since this virtual pet was started, in minutes and seconds).
- Starting values for the statistics are development stage 0, happiness 2, fullness 3, age 0 minutes and 0 seconds.
- When age reaches 5 seconds, the development stage is set to 1, and the following rules are applied:
  - Every 7 seconds, happiness is reduced by 1 (minimum 0).
  - Every 11 seconds, fullness is reduced by 1 (minimum 0). If fullness reaches 0, happiness is set to 0.
- If the development stage is at least 1, the user can perform the following actions with the buttons on the LCD keypad shield:
  - Feed the pet:
    - If fullness is less than 3, increase it by 1.
    - If fullness is 3, increase it by 1 and set happiness to 0.
  - Play with the pet: If happiness is less than 2 and fullness is at least 2, increase happiness by 1.
  - Grow the pet: The development stage is set to 2 if the age is at least 35 seconds, happiness is at least 1, and fullness is at least 3.
- Using buttons on the LCD keypad shield, the user can also enter a menu (displayed on the LCD) that offers the following functionality:
  - Save the statistics of the current virtual pet to the EEPROM and stop running. (This may overwrite another save, if it exists. You do not need to be able to save more than one pet.)
  - Delete the saved virtual pet.
  - Start a new virtual pet.
  - Leave the menu (return to default behaviour).
- Optional feature: While the user is in the menu, the simulation is paused (i.e., the countdowns for fullness and happiness are stopped, and the age does not increase).

- You can decide how you handle cases where age exceeds 9 minutes and 59 seconds. (For example, you can decide that the simulation ends, or that age does not increase after this point.) This allows you to exclude cases that make the design and implementation of the display and the save and load functionalities needlessly hard.

Assessment

Coursework type: individual coursework.

Submission date: Week 11, details will be announced later.

Demo: 5 minutes per student, in week 11; further details will be announced later.

Please remember that this coursework is an individual coursework assignment, not a group coursework assignment. While discussing this assignment with other students is permitted, you should not help each other more than an academic or teaching assistant would.

Be aware that we will use software to compare your submission to other submissions (from this year and previous years).

Deliverables

Your submission needs to include the following:

- Source code of your implementation.
- A written report.
- A video recording of your demo.

Source code: Submit this as actual source files, not just copy-pasted into the written report.

Written report: The report should contain the following:

- Simple instructions (a very short user manual).
- A list of features, and to which degree they were completed (including features that were omitted).
- A list of optional or additional features (if present), with a short description.
- At least one flowchart, finite state machine, or other diagram that describes the main behaviour of the system.

You can base your list of features on the list from the specification.

Video recording: Record your demo, in which you demonstrate all functions of your program.

Grading scheme

Most marks are expected to fall into a scale between 40 (“scraped pass”) and 80 (“impressive”). These are explained in the grading descriptors below.

To obtain higher marks, include impressive extra features or implement features in a way that is particularly impressive.

Design (40% of the module marks):

This partial mark judges whether the project includes all features that are required according to the specification.

- A (80) - All features from the specification have been realized at an excellent level. Additional features are also present.
- B (70) - All features from the specification have been realized at a level that is at least adequate. Some features are excellent or additional features are present.
- C (60) - All features from the specification have been realized at an adequate level.
- D (50) - Most features from the specification have been realized a mostly adequate level.
- E (40) - Sufficiently many features have been realized at a somewhat adequate level.

Usability (10% of the module marks):

This partial mark judges whether the software is intuitive to use.

- A (80) - The interface is highly intuitive.
- B (70) - The interface is easy to understand and requires almost no explanation.
- C (60) - The interface is easy to understand after some explanation.
- D (50) - The interface can be understood with some effort.
- E (40) - The interface is not intuitive and requires significant explanation.

Implementation (50% of the module marks):

This partial mark judges whether the implementation is of high quality and whether all features have been tested thoroughly. This also includes things like avoiding crashes or flickering on the display.

- A (80) - The project is the culmination of a full design process with the implementation carried out at a high-quality level.
- B (70) - The project implementation and testing have been well executed with some aspects at an excellent level
- C (60) - The project implementation and testing have been achieved at an adequate level.
- D (50) - Enough has been done, but nothing is very exciting and the overall achievement is mediocre.
- E (40) - The overall implementation and testing of the project is rudimentary.