EECS22L DISCUSSION

Week7   Weiwei Chen

Let's meet our new team

- Please pick up the name tags and sit in teams
- Name tags
    - Get: at the beginning of the session
    - Return: at the end of the session
- How to work as a team
    - talk, brain storm, participate, and work to collaborate
    - Different jobs, i.e. leaders for the project maintenance, documentation, design, testing, …
    - Team account and online discussion forum
- Have a great team name!!

Project2: Optical Character Recognition (OCR)

- Some useful tools for this project
  - *cvs* is a great tool for collaboration
  - Linux tools to convert the image format, e.g. *jpegtopnm, pngtopnm, tifftopnm*, etc.
  - *gimp* or *eog* are tools for image display on Linux (X11 needed)
  - *gedit* is a conveniet text editor on Linux (X11 needed)
- Two deliverables due on next Monday
  - The User Specification
    - OCR_User_Spec.pdf
  - The Software Architecture Specification
    - OCR_SW_Spec.pdf

Project2: OCR User Spec

- Design the team application user specification
  - Goal: Specify the user experience!
    - What does the user want?
    - What does he/she need to provide? What does he/she get?
    - What does the software do? What features does it have?
  - Deliverable: Application User Specification
    - Input data including options and parameters
      - What? In which format? In which order? From which device?
    - Output
      - What? In which format? In which order? To which device?
    - Processing
      - What? (not how!) What happens? What is presented?
    - Which features do you want to have in your program?

Project2: OCR SW Spec

- □ Design the software architecture specification
  - ◻ Modules (in header files)
    - ◼ How many modules will you need?
      - ◼ Image processing
        - ◼ Rotate; resize; remove the stain; manage brightness, contrast, sharpness, saturation, …
        - ◼ Keep the user in the loop?
      - ◼ Character extraction, character feature detection (e.g. height, width)
      - ◼ Pattern matching
      - ◼ Dictionary / vocabulary construction
      - ◼ User interface (text-based or GUI)
      - ◼ File I/O
        - ◼ Load image files (single file or multiple files)
        - ◼ Dump C source code

Project2: OCR SW Spec (cont.)

- □ Design the software architecture specification
  - ◻ What are the APIs for each modules ?
    - ◼ Very important for the whole project
    - ◼ How to make coding and debugging efficient?
  - ◻ Data structures (in header files):
    - ◼ Very important for the whole project
    - ◼ How to represent the image?
    - ◼ How to represent character?
    - ◼ How to represent the dictionary / vocabulary?

Project2: OCR SW Spec (cont.)

- Design the software architecture specification
  - Algorithms (on paper first, maybe?)
    - How to preprocess the image
      - Rotate? Resize? Remove the stain? manage brightness, contrast, sharpness, saturation, …
      - How to use the user input?
    - How to extract the image for single characters (symbols)?
    - How to match the patterns?
    - How to build the dictionary?
  - Team member job assignment
  - Potential project schedule
  - Team communication policy, etc.

Project2 Prize

- Weiwei wrote a C program, prize.c, as the prize for project2
  - Unfortunately, the source code was ruined since Weiwei made a mistake by using the following command to compile the program
    - gcc prize.c –o prize.c
  - However, Weiwei is so old-school that she had her code printed on the paper (9 pages long).
  - Please use your OCR program to help Weiwei recover the program and win your prize.
- The first team who solves this puzzle will win the prize
- The prize will be revealed by running the program compiled from prize.c

Happy Coding!!!