ENGF0002 SCENARIO 1

Engineering Scenarios

- Objectives:
  - Practical experience working on a somewhat "real-world" scenario.
  - Experience working in teams.
  - "Small project" level of involvement to talk about e.g. in job interviews.
  - Apply knowledge gained in first-year classes.
  - Open-ended and creative.
- Mark goes under the ENGF0002 module code (40% of total ENGF0002 mark).

Scenario task

- Create a logic or math exercise practice tool.
- Scenario 1: Describe your proposed tool and some details of implementation.
  - Decide on a plan in groups.
  - Practice written communication skills.
- Scenario 2: Implement (a prototype of) your tool and demo it.
  - Practice programming skills, verbal communication skills.
- Aim for low amount of work (2-5 hours per person per week).

Designing a logic/math exercise

- Draw upon concepts and skills that were/are required for your classes:
  - COMP0147 (Discrete math), COMP0003 (Logic and automata), COMP0005 (Algorithms), COMP0012 (Compilers), ENGF0001 (Challenges), ENGF0002 (T1 - DAPS)
- Example topics:
  - Logic gates, propositional logic, truth tables, inference, etc.
  - Sets, functions, relations,
  - Combinatorics,
  - Linear algebra,
  - Algorithms, complexity analysis, etc.

Example ideas

1. Truth table filler: Given variables and a proposition, user has to fill out a truth table. Software then checks their answers and highlights mistakes, if any.
2. Binary relation visualizer: Graphical tool that allows a user to specify 2 sets and a relation between them, and indicates if the specified relation has various properties (reflexivity, symmetry, transitivity, onto / one-to-one / bijection for functions).
3. Regular expression tool: Tool for evaluating whether a string is in a regular expression's language or not.

Scenario 1 marking

- Deliverable: 5-8 page proposal of your tool.
- Deadline: Friday 12 February, 4:00 PM.
- All members of group get the same mark.
- Worth half of scenarios mark / 20% of ENGF0002 mark.
- See coursework brief for more details.

Things to think about

- How to break implementation up into different component/modules/sub-tasks?
- What features are in scope and what features are beyond the scope of your project?
    - Be wary of the limited amount of time for this project.
- How do users design exercises for your tool? What format should new exercises be in for your tool to load it?
- What tools (programming language, libraries) will you use?
- Can visual aids help in explaining how your tool will look or how it will be implemented?

Ask PGTAs for feedback and suggestions

- Module staff:
  - Yuzuko Nakamura – y.nakamura@ucl.ac.uk
  - Azeem Khan – azeem.khan@ucl.ac.uk
  - Harriet Apel – harriet.apel.19@ucl.ac.uk
  - Sergi Bray – sergi.bray.18@ucl.ac.uk
- Scheduled help session: Mondays 9-10 AM.
- Additional help session time(s) TBA.