CS3 Capstone Projects

Stage 3: Prototype and Demonstration

1. Guidelines for 3rd hand-in [25 Marks]

Refer to the slides on Prototypes. You may choose a horizontal or vertical prototype, or a little bit of both. You can also choose to throw-away or evolutionary prototype. Adopt whichever is appropriate, *clearly state what that choice is* and then justify your choice. (A paper prototype will not suffice—you must do some programming). We will mark you according to that choice —see below. Notice that if you omit to say what sort of prototype you have made, you are unlikely to get marks much above zero.

These can only be very general guidelines as the actual prototype very much depends on the particular project you are working on. For example if the user interface for your project is minimal (say, just one button labelled "Go!"), then a purely horizontal, "user interface only" prototype would not do. If you're unsure about specifics, the best person to ask is the proposer of your project.

In general we are checking that the project has "the bones" to start into the implementation phase. A possible approach is to take some of those classes that you specified in your design and implement some of their functionality. We will look for presence of various minimal requirements as well as quality. What counts as quality differs according to whether it is an evolutionary or a throw-away prototype. Quality can range between 0–5, meaning *absent, poor, basic, substantial, high, very high.*

Unless otherwise instructed you submit the working code on Vula.

1.1 Marking Evolutionary Prototypes

Quality in the case of an evolutionary prototype refers to the normal standards of software development, for example, programming conventions followed, adequate testing, use of inheritance, etc.

1. Statement and justification of choice of prototype (horizontal/vertical/combination) and throw-away or evolutionary. [5]
2. Class definitions (at least 3), [5]
3. Class member functions [5]
4. Class inheritance [5]
5. Scope (differs for vertical or horizontal, and if a combination then use combination) [5]
    - If vertical: achieved drill down to depth in one or two areas
    - If horizontal: achieved required breadth (e.g., tested basics of user interface).

1.2 Marking Throw-Away Prototypes

In the case of throw-away prototypes quality depends on the extent to which a critical aspect of the system was successfully tested. In this case we will look if your answer to Question 2 was really justified by the prototype.

1. Statement and justification of choice of prototype (horizontal/vertical/combination) and throw-away or evolutionary. [5]
2. Write up of implications from prototype testing for requirements (i.e., what were the questions and what were the implications of the prototype evaluations for the requirements?). [5]
3. Class definitions (at least 3), [5]
4. Class member functions [5]
5. Scope (differs for vertical or horizontal, and if a combination then use combination) [5]
    - If vertical: achieved drill down to depth in one or two areas
    - If horizontal: achieved required breadth (e.g., tested basics of user interface).

2. Demonstration to Client and Tutor

Please contact your client about arrangements for the demo, and ensure it is done by 21 August. The prototype demonstration is to obtain feedback from the client on how well your team is progressing and meeting requirements. It also helps the tutor in marking according to the scheme given above.