COMP-361 – Software Engineering Project
Course Outline

Instructors

Jörg Kienzle, McConnell 327
Phone: (514) 398-2049
Email: Joerg.Kienzle@mcgill.ca
Office hours: Wednesday: 13:00 - 14:20 or by appointment (send email)

Matthias Schöttle, McConnell 322
Phone: (514) 398-7071 ext. 00116
Email: mschoettle@cs.mcgill.ca
Office hours: Monday: 13:15 - 14:15 or by appointment (send email)

Teaching Assistants

Nirmal Kanagasabai, McConnell 322
Phone: (514) 398-7071 ext. 00116
Email: nirmal.kanagasabai@mail.mcgill.ca
Office hours: Thursdays 14:35 - 15:55 or by appointment (send email)

Course Contents Summary (from Minerva)

Software development process in practice: requirement elicitation and analysis, software design, implementation, integration, test planning, and maintenance. Application of the core concepts and techniques through the realization of a large software system.

Course Overview

This 1-year course gives students the opportunity to develop a considerable software system following a rigorous software development process. Over two semesters, students develop a strategic, turn-based, distributed, 2D game following a model-driven engineering approach in groups. The project takes the students through all the phases of a software's life-cycle, from requirements elicitation and specification, to architecture design and detailed design, to implementation, deployment and maintenance. Throughout the project, students follow a model-driven engineering approach, where models of the software that is being developed are built at different levels of abstraction using different modelling formalisms. The final implementation is done by the students using an object-oriented programming language of their choice.

During the first semester, lectures are held regularly to teach the foundations of software engineering and modelling. In the second semester, lectures are gradually replaced by group meetings, where the students discuss their problems, design decisions and implementation issues with the instructor.

Learning Outcomes

At the end of the project, the students of this class have gained experience in several areas related to software engineering:

- Development Skills
  - Understanding a (reasonably complicated) problem statement
  - Analyzing the needs that an application must fulfill
  - Specifying the requirements using a rigorous formalism
  - Deciding on a distributed software architecture (client/server or peer-to-peer)
  - Designing an object-oriented solution that fulfills the requirements
  - Implementing the solution with an object-oriented programming language
  - Code maintenance, i.e. adding additional functionality to existing code
  - Delivering the application in time
- Modelling Skills
  - Eliciting behavioural requirements with use case models
  - Elaborating domain models with class diagrams
  - Specifying system interfaces with object interaction diagrams
  - Specifying system protocols with use case maps
  - Specifying software design structure with class diagrams
  - Specifying software behaviour with sequence diagrams
- Team Skills
  - Contributing to the constitution and the growth of a small group of developers
  - Communicating efficiently within the group
  - Organizing work and programming within a group
- Tool Skills
  - Modelling tools
  - Compilers
  - Debuggers
  - Profilers
  - Version control software

Handouts and Optional Textbooks

All course slides and exercises will be available for download on the myCourses webpage. Apart from the slides, I recommend consulting the following textbooks when needed:

Software Engineering

- Hans Van Vliet: Software Engineering: Principles and Practice, 3rd edition. Wiley, 2008. ISBN: 0470031468

Model-Driven Engineering

- Craig Larman. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design. First Edition, Prentice Hall 1998. ISBN: 0137488807
- Craig Larman. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and the Unified Process. Second Edition, Prentice Hall 2002. ISBN: 0130925691
- Craig Larman. Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development. Third Edition, Prentice Hall 2005.

Software Development and Games

- David Brackeen, Bret Barker, Laurence Vanhelswue: Developing Games in Java. New Riders, 2003. ISBN: 1592730051
- Rudy Rucker: Software Engineering and Computer Games, Addison Wesley, 2003. ISBN: 0201767910

Design and UML

- E. Gamma, R. Helm, R. Johnson, and J. Vlissides: Design Patterns: Elements of Reusable Object-Oriented Software. Addison Wesley, 1994. ISBN: 0201633612
- James Rumbaugh, Ivar Jacobson and Grady Booch. The Unified Modeling Language Reference Manual. 2nd edition, Object Technology Series, Pearson, 2004. ISBN: 0-321-24562-8
- UML Specification for download at the OMG website (http://www.omg.org/spec/UML/)

Pre- and Co-requisites

- Pre-requisites: COMP-206 and COMP-250
- Co-requisites: COMP-303

Grading

The final grade for this course is based on 2 exams (written individually) and a project (done in groups of maximum 5 students). The grade weight and approximate time of each exam / submission is given below[1]:

- Exams 35% (individual grades):
  - 1st semester
    - 20% exam on requirements elicitation and modelling (December 2017)
  - 2nd semester
    - 15% exam on design modelling (February or early March 2018)
- Project part 65% (group grades):
  - 1st semester
    - 3% for the user interface sketch
    - 5% for the requirement elicitation documents (use case model)
    - 12% for the requirement specification documents (concept model, environment model, operation model, protocol model)
  - 2nd semester
    - 12% for the design document (design model, interaction model)
    - 10% for the demo
    - 23% for the acceptance test

A separate document describing the project and required submissions in detail is going to be distributed within the first week of classes.

[1] Note: In the event of extraordinary circumstances beyond the University's control, the content and/or evaluation scheme in this course is subject to change.

Language of Submission

In accord with McGill University's Charter of Students' Rights, students in this course have the right to submit in English or in French any written work that is to be graded.

Academic Integrity

McGill University values academic integrity. Therefore all students must understand the meaning and consequences of cheating, plagiarism and other academic offences under the Code of Student Conduct and Disciplinary Procedures (see www.mcgill.ca/integrity for more information).

L'université McGill attache une haute importance à l'honnêteté académique. Il incombe par conséquent à tous les étudiants de comprendre ce que l'on entend par tricherie, plagiat et autres infractions académiques, ainsi que les conséquences que peuvent avoir de telles actions, selon le Code de conduite de l'étudiant et des procédures disciplinaires (pour de plus amples reseignements, veuillez consulter le site www.mcgill.ca/integrity).

Final Exam Policy

Students should not make other commitments during the final exam period. Vacation plans do not constitute valid grounds for the deferral or the rescheduling of examinations. See the Centre Calendar for the regulations governing examinations: http://www.mcgill.ca/student-records/exams/regulations/

Students are required to present their I.D. Card (with photo) for entrance to their examination.

Conflicts

If you are unable to write your final examination due to scheduling conflicts, you must submit a Final Exam Conflict Form with supporting documentation at least one month before the start of the final examination period. Late submissions will not be accepted. For details, see http://www.mcgill.ca/student-records/exams/conflicts/

Students Rights and Responsibilities

Regulations and policies governing students at McGill University can be downloaded from the website:
http://www.mcgill.ca/deanofstudents/rights/

Email Policy

E-mail is one of the official means of communication between McGill University and its students. As with all official University communications, it is the student's responsibility to ensure that time-critical e-mail is accessed, read, and acted upon in a timely fashion. If a student chooses to forward University e-mail to another e-mail mailbox, it is that student's responsibility to ensure that the alternate account is viable. Please note that to protect the privacy of the students, the University will only reply to the students on their McGill e-mail account.