



UNIVERSITY *of* NICOSIA

Response to the External Evaluation Report

Programme of Study

Master of Science in Computer Science

(Concentrations: (a) Cyber Security (b) Mobile Systems (c)
Blockchain Technology) - Distance Learning

Nicosia

15 June 2017

I. Introduction

We refer to the Report of the external evaluation committee for the evaluation-accreditation of the program of study: ‘Master of Science in Computer Science- Distance Learning’, which was prepared following a site visit at the University of Nicosia by the members of the committee on June 8, 2017.

We would like to thank the External Evaluation Committee (EEC) for their professional and thorough work during the on-site evaluation of the MSc Computer Science (Distance Learning) programme. We would also like to express our appreciation for the collegial and constructive approach with which they conducted their evaluation. As noted by the committee, almost all full-time faculty teaching in the programme were present during the evaluation. Most of the part-time faculty were also present and the committee had separate meetings with them, as well as with two current students of the Computer Science Department.

We would like to note that the report of the committee is *extremely positive* with 87 out of the 114 (applicable) quality standards/indicators receiving a perfect score of 5, and a near perfect score of 4 out of 5 in the majority of the remaining indicators. The resulting average score is *4.66 out of 5* which is in line with the committee’s concluding remark that the *“Department of Computer Science of the University of Nicosia has developed a programme addressing important and emerging areas within the wider field of Computer Science”* and that the *“programme is comparable to national and international programmes of the same level”*.

We do appreciate the committee’s recommendations for improvement, which will enhance the quality of our program and we will be addressing those in the corresponding section of this response.

The response to the external evaluation report is structured as follows: In section II we highlight the positive comments made by the committee which reflect the quality of our programme. In section III, we address the recommendations made by the committee, in each of the categories. Finally, we conclude in section IV with a summary of our response. Appendices A to D include

the changes to some of the recommendations that were too long to be inserted in the corresponding subsection of section III.

II. Positive comments made by the committee

We have selected the highlights of the positive comments which are quoted below:

- p. 7: “The institution follows national standards that require capping student cohorts at 30 students per class. This is a very good practice that ensures appropriate level of support”.
- p.7: “The programme is supported by up to date resources and the institution ensures that the programme runs smoothly.”
- p. 7: “The programme team demonstrated their ability to enforce institutional policies in terms of communication with students, provision of feedback and progress monitoring.”
- p.8: “The programme team has in place a plan to ensure that different courses and course sections are delivered in a consistent way. Students receive opportunities for formative assessment in the form of quizzes and verbal exercises, with immediate feedback.”
- p. 8: “There is an appropriate balance between fulltime and part time staff delivering this programme. There is evidence that the vast majority of staff have relevant research output in the related areas of teaching. The entire team consists of individuals with appropriate backgrounds, level of experience and skills in the field.”
- p. 8: “The team seems to have sufficient contingency plans for unexpected personnel changes and the programme is coordinated by a capable and experienced member of staff. The programme has been presented by a well-prepared team who is led in an efficient manner with confidence and commitment to the high quality of the programme’s delivery.”
- p.8: “The programme is based on clear learning outcomes that are explicitly described at programme and course level. A detailed mapping of the learning outcomes against the programme’s courses is provided and the programme delivery is based on the identified learning outcomes. The team has shown evidence of carefully aligning these learning outcomes with the topics included in the curriculum.”

- p.9: “The number of courses and the depth of learning provided in each course are appropriate for the programme and the level of study. Sufficient provision is in place for students of different backgrounds and capabilities.”
- p. 9: “The programme is managed effectively as there are in place clear procedures and organization structures. There are specific policies that govern the design and delivery of curricula as well as the evaluation of the programme.”
- p.9: “The programme is designed in a way that equips its graduates with the necessary skills for the relevant sector. There is sufficient evidence for the necessity of such graduates in the national and international job market.”
- p.11: “The Department of Computer Science of the University of Nicosia has developed a distance learning (DL) Master programme addressing important and emerging areas within the wider field of Computer Science.”
- p.11: “This programme appears to be adequately supported by the full time and part time faculty members of the University of Nicosia and the overall infrastructure of the institution. The latter provides adequate support for offering distance learning programmes as the one that has been reviewed”.
- p. 11: “the reviewed Master programme is comparable to national and international programmes of the same level”

III. Recommendations by the committee

In this section we address the suggestions/recommendations of the committee on each criterion of each category.

1. Effectiveness of teaching work-available resources

1.1.1: “Admission regulations are not specific when it comes to non-Computer Science or Engineering graduates. The programme team explained that they are applying additional criteria in the case of such students. In particular, they indicated that they look for evidence of coverage of the following subjects: databases and data structures, programming, system analysis and design and networks. Such requirements should be clearly expressed in the admission criteria.”

Response/Action: The admission criteria have now been modified to clearly state the foundation courses that we are applying. Please see the modified “Student Admission Requirements” included in **Appendix A** of this document.

1.1.3.4: “Assessment settings is adequate. There is not however any process of mark moderation. Such a process should be introduced and be informed by a comparative analysis of general mark statistics at the module level (e.g., comparison of module average marks and their standard deviations) to detect significant deviations for particular student cohorts.”

Response/Action: Although this moderation is somehow already in practise by requiring the Head of the Department to review and sign all grades before submission, we agree with the committee that a more formal process will help us improve mark moderation even further. We will, therefore, introduce an Exam Board which will review the exams before they are delivered, and also provide various grade statistics after the grading, in order to detect possible abnormalities. The chair of the Examination Board will be the Programme Coordinator and members are the course leaders of each course taught in the program.

1.1.4.5: “Although there is support towards new staff members from more experienced colleagues and the Head of Department, there is no formal procedure followed for staff mentoring. There is an orientation programme that lasts for six hours at institutional level at the beginning of the year but this focuses more on rules and regulations.”

Response/Action: As the committee noted, we do follow an unofficial procedure by which new faculty members are being supported/advised/guided by the Head of the Department. We plan to establish an official mentoring procedure by assigning one mentor to each new faculty member. The mentor will be a senior faculty member, preferably a course leader in one of the courses to which the new faculty is assigned. The mentoring process will be under the supervision of the Head of Department.

1.3.1 “Although overall full-time and part time faculty numbers are adequate, there is not sufficient support for the specialization in Blockchain technologies. At least one full time faculty member should be appointed to provide adequate support for this area.”

Response/Action: We would like to clarify that there are now four faculty employed full-time by the institution, who support the Blockchain concentration: Dr. Dmitry Apraksin, Dr. Konstantinos Karasavvas, Prof. George Giaglis and Dr. Harald Gjermundrod.

1.3.7: “The ratio of full-time over part-time personnel needs to be increased in the case of blockchain technologies.”

Response/Action: All faculty supporting the blockchain technology concentration are now employed full-time (see response to point 1.3.1 above).

1.3.9: “The contractual teaching load for full time faculty members is 6 hours per week for senior staff and 9 hours per week for junior staff, as the institution confirmed to the External Evaluation Committee. This provides sufficient time for research. The institution should ensure that no extra teaching should be given to full time faculty members through additional overtime teaching contracts, as this would have a detrimental effect on the faculty’s research. This is particularly important given the fact that the programme of concern is delivered through distance learning and, therefore, creates additional demand upon faculty time (e.g., more hours of online student contact to address questions, content preparation and revision etc.).”

Response/Action: We plan to utilize additional faculty if needed so that we do not assign overtime workloads to existing faculty.

2. Program of study and higher education qualifications

2.1.1: “There are detailed learning outcomes for the entire programme, each concentration and every individual course. These are discussed in detail as part of the learning guides for each course. However, the team must consider the adoption of a more suitable use of words

when describing the work that is to be carried out, in order to reflect level 7 learning descriptors. There are appropriate guidelines the institution can follow, as it is aligned to the Bologna process and the three cycles. At this level students are expected to provide evidence of evaluation, synthesis, critical evaluation, construction of knowledge and critical thinking. It is important for the programme team to revisit the learning outcomes in order to ensure that all of them are phrased appropriately.”

Response/Action: The learning outcomes of the program and each concentration have been rephrased to reflect level 7 learning descriptors and are given in **Appendix B**.

2.2.3: “Whilst the programme includes interesting and coherently structured specializations, it does not require the students to select one of them and leaves too many credits accrued through electives (up to 54, i.e., more than half of the total ECTS for the programme). Some advice should be offered to students, who do not select one of the three concentrations, regarding meaningful combinations of electives. More, importantly however, we suggest that the final thesis module (i.e., COMP-599DL) should become compulsory for any student taking the programme. This would be necessary to ensure that students develop skills in combining knowledge from different modules (notably research methodology, literature study, programming, system evaluation) and apply it in solving some large scale computing problem independently.”

Response/Action: We would like to point out that the program follows the general University of Nicosia guidelines for Postgraduate programs that provides for the thesis module to be elective and to be substituted by two courses. Having said that and taking into consideration the committee’s suggestion, we will make the thesis a recommended course.

2.2.6: “The COMP-500DL module should cover basic statistical analysis and quantitative methods and user-centric system evaluation techniques.”

Response/Action: The COMP-500DL module has been modified to include basic statistical analysis as recommended. The amended course outline as well as the amended study guide for this course are both included in the **Appendix C** of this document.

2.2.7: “The syllabus of some practical development modules (COMP-528DL, COMP-523DL and COMP-521DL) is too ambitious and might not be possible to deliver and assess it effectively through distance learning (DL). Our suggestion would be to scope these modules down to ensure their adequacy for DL.”

Response/Action: We have followed the committee’s recommendation and scoped down the modules of the aforementioned courses (COMP-528DL, COMP-523DL and COMP-521DL). The amended course outlines as well as the amended study guides for each of these courses are included in the **Appendix C** of this document.

2.2.8: “Overall the content of individual modules is adequate, except that it does not cover privacy (this is important for the concentration on cyber security), and statistical analysis, quantitative methods and user-centric system evaluation techniques (see also comment 2.2.6 above). The required textbooks of some modules appear to be outdated (e.g., COMP-522DL, COMP-524DL, COMP-528DL, COMP-525DL).”

Response/Action: Regarding the first part of the comment, on Privacy, we would like to note that Privacy is covered in two courses and this is explicitly mentioned in the study guides. More specifically, an overview of privacy and related issues are covered in two courses as follows:

- COMP-514 Study Guide Section 12 (excerpt from the corresponding section):
Objectives: "The discussion proceeds with an overview of privacy, an important concept nowadays as the scale and interconnectedness of personal information collected and stored in information systems has increased dramatically. Legislations and directives are presented that aim to protect personal information and empower users to manage their personal data (e.g. General Data Protection Regulation)."
Expected Learning Outcome: "Define privacy and describe the US and EU legislations/directives that protect personal information."
- COMP527 Study Guide Section 1

Activities: Watch a debate on <https://www.youtube.com/watch?v=cCoQvL61FT0>. What do you think? Is it just hype spreading fear? Is it just another way for governments to violate privacy rights in the name of national security?

- *COMP527 Study Guide Section 5*

Summary: In this section we have studied CNE and CNA. There are controversial concepts, such as the surveillance of electronic communications, that one may claim that they breach privacy rights. Are there any tradeoffs between privacy and national security? Is it justifiable?

Regarding the second part of the comment, on required textbooks being “outdated”, we would like to note that these books, although they are dated in some cases, they are considered classic textbooks. However, we have now added some more recent textbooks in all modules that included dated books. The books to be added to each course are shown in **Appendix D**.

3. Research work and synergies with teaching

3.1.4: “Most of staff appear to have international peer reviewed publications. However, there are no annual or longer-term targets regarding this at either individual or group/subject area level.”

Response/Action: Perhaps it was not clearly communicated to the committee that short and long term plans regarding publications and research are specified in the self and peer evaluation reports that are submitted by all tenured faculty every two years, and by non-tenured faculty, every year.

3.1.9: “The training that students receive in research could be enhanced. As commented earlier students receive no training in statistical, quantitative and user-centric evaluation methods. Also, as they are not required to do a thesis, they may end up having limited exposure to research.”

Response/Action: There is significant exposure to current research through the seminar part of the course COMP500DL. Also in this course, students are required to read several papers and write a literature review. In addition, as stated in 2.2.6 above, COMP-500DL

has now been changed to include basic statistical analysis. Also the thesis module COMP-599DL will now be recommended.

4. Administration services, student welfare and support of teaching work

4.2.6: “The supportive material (books, presentations, webinars, exercises, teaching notes) are mainly provided to the students through the Moodle learning environment. Additional access to online books and journals is provided through digital library accounts given to the students upon registration. However, studying the course syllabus, a more updated list of suggested books and a more recent bibliography needs to be described in almost every course.”

Response/Action: See answer to 2.2.8 above including Appendix D with updated books.

5. Distance Learning programs

5.16: “The programme is supported by skilled staff. Most of them have a PhD.”

Response: We would like to note that **all** of the faculty of the program have a PhD.

IV. Conclusion

We thank the committee once again both for the positive evaluation, as well as for the constructive comments and suggestions and the fruitful discussion that we had with its members during the on-site visit. We also thank the committee for the time and thoroughness it dedicated to the evaluation of the program and for helping us improve the program through the suggestions made. We have already taken action and incorporated nearly all recommendations of the committee as indicated in section III of this response.

We would like to highlight again the high score evaluation and the very positive comments of the committee which we consider as recommendation for accreditation, having satisfied the committee's recommendations.

.....
Professor Athena Stassopoulou
Programme Co-ordinator

Appendix A

Admission Criteria for students:

General:

Applications for admission to the program will be considered only from candidates that fulfill the minimum entrance criteria as described below:

- A Bachelor Degree in Computer Science, Computer Engineering, or any other related field from a recognized university (i.e. American, European, or another recognized equivalent qualification) with a CPA of at least 2.0. Applicants with lower CPA will be considered on an individual basis. Applicants with a non-computer science degree (but related, e.g. MIS) may be admitted into the program but they may have to take up to 5 preparatory foundation courses, depending on the applicant's background. The courses are:
 - 4 courses before starting the programme: COMP-113 Programming Principles II, COMP-201 Systems Analysis and Design, COMP-302 Database Management Systems, COMP-358 Networks and Data Communications
 - 1 course during first semester: COMP-211 Data Structures
- Proficiency in the English Language: Students satisfy the English requirements if their first degree was taught in English. Otherwise, they would need to present at least a TOEFL score of 550 paper-based or 213 computer-based, or GCSE "O" Level with "C" or IELTS with a score of 6.0 or score placement at the ENGL-100 level of the University of Nicosia Placement Test.

Specific:

- A completed application form
- A Curriculum Vitae indicating the student's education, academic and professional experience, any publications, awards, etc.;
- Letters of Recommendation: Two recommendation letters from academic or professional advisors.

- Personal Statement: A letter highlighting the applicant's individual competences and strengths and providing his/her reflections regarding the expectations and value of the program as well as to his/her personal advancement and career development.

Appendix B

The updated program learning outcomes, as well as the learning outcomes of each concentration have been updated, according to the committee's recommendation to use Level 7 learning descriptors.

Program learning outcomes:

Upon successful completion of this program, the students should be able to:

1. Critically evaluate computer-based systems, processes and programs, and apply theory, practices and tools for their specification, design, implementation and maintenance;
2. Critically analyze and evaluate software solutions from the perspective of software architecture and design patterns;
3. demonstrate initiative and originality in the specification, design and implementation of computer-based systems, processes, and programs, as well as in the implementation of software project management to meet desired functional and quality requirements;
4. Critically analyze and evaluate the security risks that may be involved in the operation of computing equipment within a given context;
5. Deploy effectively the tools used for the construction and documentation of software, and in particular in the process involved in using computers to solve practical problems;
6. Communicate clearly and analytically using techniques derived from scientific practice in group meetings, presentations, lectures, written reports, and research papers;
7. Exhibit such skills that are required to participate in research and development work or to independently work in other qualified areas as well as be able to continue studies towards a doctoral degree

Specific Learning Outcomes per Specialization

Cyber Security

Upon successful completion of this concentration, the students should be able to:

1. Employ advanced skills to conduct attacks on networks and systems in a controlled setting to gain practical experience with attack methodologies;

2. Integrate complex key technologies to formulate a set of strategic approaches to defend and countermeasure attacks on networks and systems;
3. Critically analyze and evaluate the cyber warfare landscape, including targets, attack techniques and tools, defense tactics, legislations, and ethical dilemmas;
4. Innovatively practice security as an integral part in the system development lifecycle rather than an add-on feature.

Mobile Systems

Upon successful completion of this concentration, the students should be able to:

1. act with initiative in decision-making for the most appropriate methodology in analyzing and critically assessing existing Services and Protocols in Mobile environments;
2. Employ critical thinking for the key issues in designing applications in cross-platform operating systems and achieve a deep and systematic understanding of the technical aspects for platform dependent and platform independent applications;
3. synthesize novel implementations for the efficient utilization of components across platforms, which facilitates scalability and extensibility of mechanisms and applications on different platforms;
4. design and undertake substantial investigations to address significant areas of theory and/or practice, synthesize and construct new knowledge for various Mobile and Wireless networking issues and assess the key methodologies for monitoring Mobile Systems' performance;
5. build, test, and critically evaluate Mobile software solutions collaboratively (in a group) or autonomously, in order to develop a project in the area of Mobile Systems.

Blockchain Technologies

Upon successful completion of this concentration, the students should be able to:

1. systematically analyze decentralized digital currencies and the underlying blockchain technology;
2. analyze and critically assess the architecture of the bitcoin system, including the data structure used for the bitcoin blockchain;

3. use the Bitcoin Script language to develop different type of scripts using the provided API;
4. systematically analyze the horizontal and vertical application areas of blockchains, beyond digital currencies;
5. envision the disruptive potential of blockchains in combination with other technological developments (IoT, AI, etc.), in the field of money and commerce;
6. provide a deep understanding of smart contracts and their role in the emergence of decentralized applications;
7. critically assess and develop innovative DApps (Decentralized application development).

Appendix C

Course Outlines and Study guides of updated courses to include the committee's recommendation on criteria 2.2.6 and 2.2.7.



University of Nicosia

Course Code	Course Title	ECTS Credits
COMP-500DL	Research Seminars and Methodology	6
Department	Semester	Prerequisites
Computer Science	Fall/Spring	None
Type of Course	Field	Language of Instruction
Required	Computer Science	English
Level of Course	Year of Study	Lecturer(s)
2nd Cycle	1 st	Dr Athena Stassopoulou
Mode of Delivery	Work Placement	Co-requisites
Distance Learning	N/A	None

Objectives of the Course:

This course is designed to provide a deep and systematic understanding of the nature and conduct of CS research and to equip students with the ability to undertake independent research. The course will consist of three parts:

- a) Introduction to research: This will include lectures on research strategy and tactics, how to search and review literature, how to evaluate research, how to present research (both written and oral). The course will also include a section on introductory statistics for the purpose of testing hypotheses and evaluating fitted regression models.
- b) Research Seminars: The course will include CS seminars from faculty of the Department or/and other Universities as well as from research centers and the industry. Each seminar will focus on the area of specialization of the speaker and will clearly detail the approach used such as models, methodologies, algorithms, experimental set up and results etc.

- c) Literature Review: Students will undertake a literature review on a specific topic within one area of specialization (e.g. Distributed Systems, System Security, Networks, Mobile Computing, Artificial Intelligence, etc). The student will do a literature search and write the findings of this search in the form of a paper which will then be presented in class at a mini-conference. Students opting for the thesis option for their MSc, can use this part of the coursework to perform introductory-background work for their thesis.

Learning Outcomes:

After completion of the course students are expected to be able to:

- demonstrate an understanding of how established techniques of research and enquiry are used to extend, create and interpret knowledge in Computer Science.
- evaluate critically current research in Computer Science, and propose possible alternative directions for further work.
- test research hypotheses and evaluate statistical regression models.
- deal with complex issues at the forefront of the academic discipline of Computer Science in a systematic and creative manner and to communicate conclusions clearly to both specialists and non-specialists.
- demonstrate independence and originality in tackling and solving problems within the area of Computer Science, and to act autonomously in planning and implementing solutions in a professional manner.
- continue to advance their knowledge and understanding, and to develop new skills to a high level, and become independent life-long learners across the discipline of Computer Science.
- define and plan a piece of independent research.

Course Contents:

1. Introduction and overview
2. The nature of Computer Science research: what is research
3. Literature searches, information gathering
4. Reading and understanding research papers
5. Technical writing, referencing, bibliographies
6. Presentation skills, written and oral
7. Statistical Analysis
8. A series of research seminars from academia and industry

9. Review of legal, ethical, social and professional issues including data protection and standards

Learning Activities and Teaching Methods:

Lectures, Seminars, Assignments.

Assessment Methods:

Homework assignments	20%
Seminar Reports	30%
Research Report (individual)	30%
Presentation	20%

Recommended Textbooks/Reading:

Authors	Title	Publisher	Year	ISBN
P. D. Leedy and J. E. Ormrod	Practical Research: Planning and Design (9th edition)	Prentice Hall	2009	0136100872
W. Strunk Jr. and E. B. White	The Elements of Style (4th edition)	Longman	1999	9780205309023
Hossein Pishro-Nik	Introduction to Probability, Statistics, and Random Processes	Kappa Research, LLC	2014	0990637204
Raj Jain	The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling	Wiley	2007	ebook

Weekly Schedule:

See course/module Study Guide



UNIVERSITY *of* NICOSIA

SCHOOL OF SCIENCES AND ENGINEERING

MSc Computer Science

Study Guide

COMP-500DL: Research Seminars and Methodology

Prof. Athena Stassopoulou

Nicosia

2017

Contents

Contents.....	22
Introductory note.....	23
Section 1 – Introduction and Overview. The nature of Computer Science research: What is research?.....	24
Section 2 – Literature searches and information gathering.....	26
Section 3 – Reading and evaluating research papers.....	28
Section 4 – Technical writing, referencing, bibliographies.....	30
Section 5 – How to give a research talk.....	34
Section 6 – Creating an MSc Thesis Project.....	36
Section 7- Statistical Analysis.....	38
Sections 8-11 – Research Seminars.....	40
Section 12 – Review of legal, ethical, social and professional issues including data protection and standards.....	42

Introductory note

This Study Guide is a basic supplement for the distance learning course “Research Seminars and Methodology”, which is offered by the distance learning MSc programme in Computer Science. The aim of this Guide is to direct the students and help them into making systematic use of the educational material on which the teaching of the course is based.

The Guide must be used in common with the Course Outline and with the educational material (course textbooks, extensive notes of the lecturer, powerpoint presentations, articles and book chapters indicated for each lecture, etc.) which has been uploaded to the interactive internet-based platform of the course (moodle). The student must start his/her studying by the extended notes of the lecturer, which correspond with the powerpoint presentation of each lecture, and then, taking advice from the present Guide, he/she must extend his/her knowledge making use of the rest of the educational material of each lecture.

The Guide includes 12 sections. Each of these sections is composed of the following parts: objectives, expected learning outcomes, key words, bibliography, additional bibliography, activities, and synopsis. Conducting the activities is essential, since you will be helped into absorbing the material of each lecture as well as into developing critical thought. The activities and the proposed bibliography will be enriched during the semester.

Dr. Athena Stassopoulou, Professor of Computer Science, School of Science and Engineering

Section 1 – Introduction and Overview. The nature of Computer Science research: What is research?

Objectives

The aim of this section is to introduce the students into the objectives, the purposes, the requirements and the contents of the course. In particular, this section will introduce students to: the nature and conduct of CS research, the scope of graduate research (PhD and MSc), the types and characteristics of CS research, the research process, the research proposal, the ways by which research is measured and publication ethics.

Expected learning outcomes

After the completion of this section, the students are expected to:

- demonstrate an understanding of what is CS research and what is expected at graduate level research.
- describe the characteristics of research and what constitutes research-worthy projects.
- describe the contents of a research proposal
- describe all the steps of the research process
- explain the process of scientific publication
- explain the ways by which research is measured
- discuss the types of ethics complains

Key words

Applied research, theoretical research, research process, scientific publications

Bibliography

- [1] P. D. Leedy and J. E. Ormrod, “Practical Research: Planning and Design” (10th edition), Pearson, 2014.

[2] M. Berndtsson, J. Hansson, B. Olsson and B. Lundell, “Thesis Projects A Guide for Students in Computer Science and Information Systems”, (2nd edition, Springer, 2008).

[3] Wendy Hurrp, “Publishing Ethics and Copyright”, Elsevier, 2012.

Additional bibliography (Examples of seminal papers in Computer Science)

[1] Sergey Brin and Lawrence Page, “The Anatomy of a Large-Scale Hypertextual Web Search Engine”, Proceedings of the 7th international conference on World Wide Web, 1998.

[2] Tim Berners-Lee, “Information Management: A Proposal”, CERN, 1989.

[3] Turing, A.M., “Computing machinery and intelligence”. *Mind*, 59, 433-460, 1950.

Activities

1. Identify and read one seminal paper in Computer Science.
2. Summarize the paper in your own words (about half to 1 page maximum)
3. Explain the impact of the paper.
4. Study chapter

Summary

Research is defined as “the activity of a diligent and systematic inquiry or investigation in an area, with the objective of discovering or revising facts, theories, applications etc. The goal is to discover and disseminate new knowledge”. This section covers the characteristics of high-quality research and of research-worthy projects. It describes the contents of a research proposal and goes through all the steps of the research process. The process of scientific publication is elaborated along with the ways by which research is measured. Various issues on publication ethics are also addressed.

Section 2 – Literature searches and information gathering

Objectives

An essential part of research is the review of the relevant literature. As stated by Webster and Watson (2002) , “An effective review creates a firm foundation for advancing knowledge. It facilitates theory development, closes areas where a plethora of research exists, and uncovers areas where research is needed.”

The aim of this section is to describe the process of literature search and information gathering. More specifically, it addressed key issues in identifying *quality* literature relevant to the research study and gathering articles for an effective literature review.

Expected learning outcomes

After the completion of this section, the students are expected to:

- Understand the importance of literature review in the research process and the need for literature search.
- describe what the literature includes (i.e. journal articles, conference papers, books, dissertations, technical reports etc) and be able to identify the merits of each type of publication.
- Perform searches by querying quality literature databases available to them through the University library services, such as IEEE, ACM, Elsevier, Springer, etc.
- Perform searches using websites and other internet resources.
- Identify quality literature.
- Use keyword search as well as backward and forward search.
- Use the refWorks tool for managing their citations.

Key words

Literature search, effective literature review, literature categorization, literature databases, keyword search, backward and forward search.

Bibliography

- [1] Levy, Y., and Ellis, T. J., “A Systems Approach to Conduct an Effective Literature Review in Support of Information Systems Research”, *Informing Science Journal* (9), 2006.
- [2] Webster, J., and Watson, R. T., “Analyzing the past to prepare for the future: Writing a literature review.”, *MIS Quarterly*, 26(2), 13-23, 2002.
- [3] University of Nicosia Library and Information Center, <http://www.library.unic.ac.cy/> [last access May 9, 2017].

Activities

1. Connect to the University of Nicosia Library and perform keyword searches on a specific topic within your area of concentration.
2. Identify at least 4 quality publications that are relevant to your topic.
3. Create a database of your references using refWorks.
4. Create an example document and insert these citations in the document, experimenting with the various referencing styles (IEEE, Harvard, Chicago etc).

Summary

The identification of the relevant literature is an essential part of the research process. Equally important is the ability of the researcher to identify quality literature, a particularly challenging task to new researchers.

Section 3 – Reading and evaluating research papers

Objectives

The aim of this section is to introduce students to the fundamental task of reading and understanding research papers, for the purpose of performing literature surveys as well as for evaluating papers submitted to journals or conferences.

Expected learning outcomes

After the completion of this section, the students are expected to:

- Apply the three-pass approach to efficiently read and understand papers.
- Use efficient techniques to read papers for literature surveys.
- Describe the peer review process of evaluating papers.
- Understand the requirements for a paper to be “publishable”.
- Explain the questions that must be answered when evaluating a research paper.

Key-words

Three-pass approach, peer-review process, paper refereeing.

Bibliography

- [1] Keshyav, S., “How to read a paper”, ACM SIGCOMM Computer Communication Review 37 (3), 2007.
- [2] Smith, A.J., “*The Task of the Referee*”, Computer 23(4):65-71, 1990.

Additional bibliography

- [1] Benevenuto, F., Magno, G., Rodrigues, T., and Almeida, V., “Detecting spammers on Twitter”, in: Proceedings of the Seventh Annual Collaboration, Electronic messaging, Anti-abuse and Spam Conference (CEAS), 2010.

Activities

1. Read the paper “Detecting Spammers on Twitter” in depth.
2. Write a report of maximum one page which will include the following:
 - (a) Journal/Conference full title (include year of publication)
 - (b) Research paper title
 - (c) Author(s)
 - (d) Paper summary
 - (e) Motivation
 - (f) Review of literature
 - (g) Problem
 - (h) Solution
 - (i) Results
 - (j) Implication

Summary

Researchers read papers in order to be up to date and relevant in their field, to perform literature surveys, and to review the work of other researchers. In the case of performing literature surveys, this requires the reading of tens of papers, sometimes in an unfamiliar field. Therefore, learning to efficiently read papers is critical.

Section 4 – Technical writing, referencing, bibliographies

Objectives

The aim of this section is to introduce student to the principles of technical writing, be it research papers, technical reports, literature review, graduate theses etc. Emphasis is given to research papers and literature reviews. (Graduate theses will be specifically addressed in section 5).

Expected learning outcomes

After the completion of this section, the students are expected to:

- Understand the importance of writing as part of the research process
- Know the structure of research papers and what should be included in each section of a research paper
- Know the basic principles of writing a literature review and the structure of the review.
- Understand the significance of a thematic organization of the literature (concept-centric) which will facilitate further analysis of the existing literature and lay the foundations for the advancement of knowledge.
- Use a concept matrix, possibly augmented with other units of analysis for the purpose of the literature review.
- Understand the importance of language and writing style and apply the basic principles to their literature review assignment

Key words

Abstract, introduction, Related Work, Methodology, Experimental Results, thematic organization, concept-centric approach.

Bibliography

- [1] Zobel, J., “Writing for Computer Science”, Springer, 2014.
- [2] Webster, J., and Watson, R. T., “Analyzing the past to prepare for the future: Writing a literature review.”, *MIS Quarterly*, 26(2), 13-23, 2002.
- [3] “*What is a Literature Review*”, M. Harvey, Carnegie Mellon University, <https://www.cs.cmu.edu/~missy/WritingaLiteratureReview.doc> [last access: May 12, 2017].
- [4] Strunk, W., and White, E. B., “*The elements of style*”, 4th edition, Longman, 1999.

Additional bibliography

- [1] Adomavicius, G., “*Toward the Next Generation of Recommender Systems: A survey of the state-of-the-art and possible extensions*”, *IEEE Transactions on Knowledge and Data Engineering*, 17 (6), 2005.
- [2] Orhpanou, K., Stassopoulou, A., and Keravnou, E., “Temporal abstraction and temporal Bayesian networks in clinical domains: A survey”, *Artificial Intelligence in Medicine* 60, 133–149, 2014.

Activities

1. Download the two example abstracts and identify for each one, the answers to the questions:
 - a. What is the problem they are solving
 - b. Why this is an interesting problem
 - c. What methodology you used to solve it
 - d. Main results and conclusions
2. Term project: Write a literature survey on a topic of your interest. See attached assignment.

Summary

Writing is an essential part of the research process. Ideas need to be communicated to have an impact and be worthwhile. Writing also forces us to pin the details, to be clear and focused and to crystallize what we don't understand which will further help our research. This section covers the basic principles of technical writing.

Literature Review Assignment

You are asked to write a Literature Review on a specific topic within one area of a Computer Science specialization that you are interested in (see example below on topic/CS area). Your review will be written in the form of a research paper according to the IEEE conference style format which can be found in:

http://www.ieee.org/conferences_events/conferences/publishing/templates.html for Word and Latex.

The paper should be a minimum of 5 pages in the above format. (Latex is strongly recommended).

Instructions and advice:

- 1) Identify an area of specialization that you are interested in. Scan through all available sources to identify interesting, current research on a specific topic within the area. Students opting for the thesis option for their MSc, can use this literature survey to perform introductory-background work for their thesis.
- 2) After you have done some initial investigation, consult with a faculty member, who specializes in the area you are interested in (check the online list of faculty along with their areas of specialization).
- 3) With the help of the faculty member, you will clearly identify the topic and title of your review, as well as 3-4 relevant, recent journal or conference papers from where start.

- 4) Once you finalize the topic and theme of your review, submit a review proposal for approval.
- 5) Your review must include at least 8-10 papers published in refereed journals and conferences.
- 6) Your review should be organized thematically (see material on Writing Research Papers, section on Literature Review). You do not report on each paper you read one by one. Remember that your review is not a summary of studies, but a synthesis of information which requires comparing themes, methods and conclusions among the different works. (See example surveys in the Additional Bibliography section).

Section 5 – How to give a research talk

Objectives

The purpose of this section is to convey the structure and requirements of a research talk. It gives emphasis to both the content of the talk as well as the guidelines to presenting the content, both by use of the slides as well as orally. As part of the requirements of this course, students will need to present their literature review term project in a mini-conference which is organized as part of this course.

Expected learning outcomes

After the completion of this section, the students are expected to:

- Understand the basic principles of giving a good research talk.
- Identify what to include in the presentation slides.
- Use basic presentation principles in order to successfully deliver the content of their slides.

Bibliography

- [1] Peyton Jones, S. L., Hughes, J., and Launchbury, J., “How to give a good research talk”, SIGPLAN Notices 28(11), 1993.
- [2] Winston, P. H., “How to speak”, <http://people.csail.mit.edu/phw/index.html> [last access: 12 May, 2017].

Activities

1. The students will be able to utilize what they learned in this section by giving their own oral presentation of the literature review assignment.

Summary

Writing good papers and giving good presentations is fundamental to research excellence. Giving talks helps researchers crystalize their ideas, communicate them to others, receive feedback, meet other researches and build relations, and promote their work and have an impact.

Section 6 – Creating an MSc Thesis Project

Objectives

The purpose of this section is to convey the structure and requirements of the MSc Thesis Project, including the assessment criteria, writing the project proposal, the role of the advisors. Particular emphasis is given to the structure and content of the MSc thesis. The MSc Thesis Project is an Elective course to the proposed MSc Computer Science and in this section the students will become familiar with what it entails in order to make an informed decision of whether to opt for the course.

Expected learning outcomes

After the completion of this section, the students are expected to:

- Understand the requirements and scope of the MSc Thesis project.
- Know the assessment criteria to the project requirements: project implementation, thesis writing and oral presentation/defense.
- Understand what each of the project requirements entails.
- Understand the role of the advisor and the responsibilities of both the student and the advisor.
- Know the structure and content of the MSc Thesis.

Key words

MSc Thesis Project, MSc Thesis defense.

Bibliography

- [1] Shoaff, W., D. “How to Write a Master's Thesis in Computer Science”, Florida Institute of Technology, <http://cs.fit.edu/~wds/guides/howto/howto.html> [last access: May 12, 2017].

- [2] Berndtsson, M., Hansson, J., Olsson, B., and Lundell, B., “Thesis Projects A Guide for Students in Computer Science and Information Systems”, 2nd edition, 2008.

Activities

1. Briefly describe the requirements of an MSc Thesis Project
2. Give the outline of an MSc Thesis.
3. Describe the contents of each section of an MSc Thesis.
4. What skills are required for a successful MSc Thesis Project?

Summary

The MSc Thesis Project is often the first introduction of a student (possibly researcher-to-be) to research. It enables the student to specialize and provide him/her with substantial expertise on a specific topic. The requirements of the MSc Thesis Project consists of three parts: writing a significant program (suitable for MSc level), writing a thesis that describes the work done, and finally defending the work in an oral examination.

Section 7- Statistical Analysis

Objectives

Objectives of this section is to introduce the students to basic statistical analysis and basic design of experiments for the purpose of testing research hypotheses, and evaluating experimental results.

Expected learning outcomes

After the completion of this section, the students are expected to:

- know the main types of sample statistics, i.e. measures of central tendency (mean, median, mode) and their suitability depending on the type of data, as well as the measures of variation (range and standard deviation)
- know the important probability distributions: binomial, poisson, uniform, normal, exponential.
- know how to compute confidence intervals for the population mean
- perform hypothesis tests on the population mean (null hypothesis, alternative hypothesis, types of errors, p-value).
- be able to interpret a simple linear regression model to given data and evaluate the goodness of the model (p-value).

Key words

Measures of central tendency, measures of variation, probability distributions, confidence intervals, hypothesis testing, linear regression.

Bibliography

- [1] Hossein Pishro-Nik, "Introduction to Probability, Statistics, and Random Processes", Kappa Research, LLC, 2014.

- [2] Jain, R., “The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling”, Wiley 2007, ebook.

Activities

1. Given data :
 - (a) Distinguish between discrete and continuous sample data
 - (b) Use histograms and other visual aids to get an indication of a useful probability distribution for the data.
 - (c) Construct a normal quantile-quantile plot for the data. Does the distribution appear to be normal?
 - (d) Compute the sample mean, median, variance and standard deviation.
 - (e) Compute a 95% confidence interval for the mean.
 - (f) Test a specific value for the population mean.

2. Given the data:
 - (a) Interpret a fitted simple linear regression model to the data given.
 - (b) Use the p-value and residual plots to evaluate the goodness of the model.

Sections 8-11 – Research Seminars

Objectives

These 4 sessions (Sections 8-11) will include CS seminars from faculty of the Department or/and other Universities as well as from research centers and the industry. Each seminar will focus on the area of specialization of the speaker and will clearly detail the approach used such as models, methodologies, algorithms, experimental set up and results etc. Each research talk will be based on a research paper which the student will have to read in depth and describe in their own words. The objective is to expose students to current, state-of-the-art research in Computer Science as well as the application of research areas in the industry.

Expected learning outcomes

After the completion of this section, the students are expected to:

- Demonstrate familiarity with key research areas of Computer Science.
- Understand what a research project entails through these seminars: the literature reading, the writing, the implementation of the work, the analysis of the results, the presentation etc.
- Distinguish good presentation elements (written and oral) that they can utilize in their own work.

Key words

Depending on seminar

Bibliography

Paper of the speaker, plus any relevant literature.

Activities

For each seminar there will be the following assignment:

1. Read the paper on which the talk is based and write a report (maximum 1 page) which will include the following:
 - (a) Paper summary
 - (b) Motivation
 - (c) Review of literature
 - (d) Problem
 - (e) Solution (here you should describe the system in your own words)
 - (f) Experimental Results and Analysis (here you should briefly describe the experiments performed)
 - (g) Implication

Summary

Being exposed to current, state-of-the-art research is essential. The speakers will base their presentation on novel research work or the application of research work to the industry. Students will see the real-world application of the research process through the work of the speaker.

Section 12 – Review of legal, ethical, social and professional issues including data protection and standards

Objectives

In this section the student will be introduced to the major concepts of data protection and privacy from the point of view of research ethics.

Expected outcomes

After the completion of this section, the students are expected to:

- Know the definitions of the major concepts that surround the discussion and application of data protection and privacy rules.
- Use the knowledge acquired in this section as a guide for the identification of the privacy and data protection aspects of their research.
- Identify major EU laws that relate to the practice of information security.
- Identify the ethical aspects of the privacy and data protection issues within a research project.
- Identify the technical aspects of the privacy and data protection issues within a research project.

Key words

Data privacy, data protection, ethics.

Bibliography

- “Data protection and privacy ethical guidelines”, European Commission, 2009, http://ec.europa.eu/research/participants/data/ref/fp7/89827/privacy_en.pdf [last access: 15 May, 2017]

- Whitman, M. E., and Mattord, H. J., “Principles of Information Security” , 4th edition, Cengage Learning, 2011, chapter 3.
- Solove, D., “The Digital Person: Technology and Privacy in the Information Age”, New York University Press, 2004.
- ACM Code of Ethics and Professional Conduct, <https://www.acm.org/about-acm/acm-code-of-ethics-and-professional-conduct> [last access: 15 May, 2017].

Activities

- Briefly describe the terms: data privacy, data protection, informed consent, personal data, privacy, private information.
- What are the technical questions that should be asked in order to detect data protection and privacy issues within a project?
- Study the following paper and provide a brief summary:
 - Kenneally E., Bailey M., Maughan D. “A Framework for Understanding and Applying Ethical Principles in Network and Security Research.” In: Sion R. et al. (eds) Financial Cryptography and Data Security. FC 2010. Lecture Notes in Computer Science, vol 6054. Springer, 2010.

**COMP-521DL**
Cloud Computing

Course Code	Course Title	ECTS
COMP-521DL	Cloud Computing	10
Department	Semester	Prerequisites
Computer Science	Fall	COMP-515
Type of Course	Field	Language of Instruction
Elective	Computer Science	English
Level of Course	Year of Study	Lecturer
2 nd cycle	1 st or 2 nd	Dr Nicholas Loulloudes
Telephone	E-mail	Student Consultation Hours
N/A	loulloudes.n@unic.ac.cy	By Appointment

Objectives of the Course:

The main objectives of the course are to:

- Introduce the background and concepts of cloud computing
- Compare and contrast cloud computing with other computing paradigms
- Cover in detail the different technologies used in cloud computing including: virtualization, scalability, elasticity, and load balancing
- Expose the students to cloud services and platforms

- Make students aware of problems and challenges when designing and developing applications using cloud computing technology
- Expose the students to development tools/environments/frameworks to develop applications using cloud computing infrastructure
- Cover in detail how to secure cloud infrastructures and applications.
- Learn how to design and deploy Cloud applications over the infrastructure of currently established providers

Learning Outcomes:

After completion of the course students are expected to be able to:

1. Describe the concept, benefits, principals, architecture, and implementation technology of cloud computing
2. Compare and contrast the cloud computing with other computing paradigms
3. Explain in details aspects of the cloud computing including: virtualization, scalability, elasticity, and load balancing
4. Explain fundamental architecture, models, services, and platforms that are used in the cloud computing domain
5. Be aware of problems and challenges as to avoid them when designing and developing cloud based applications
6. Design and develop cloud based applications to be hosted by various cloud computing infrastructures
7. Summarize what is needed in order to secure the cloud infrastructure and applications hosted on a cloud infrastructure
8. Demonstrate the ability to select an appropriate technology/platform/environment in order to provide a cloud computing based application that fulfills the design requirements.

9. Explain in detail the concept of Big Data analytics and how fundamental concepts of it are supported by Cloud technologies
10. Understand how the demand for cloud resources changes in the course of time and the need to elastically adapt resource demand and offer based on the utilization.

Course Contents:

1. Introduction to Cloud Computing
2. Principles of Parallel and Distributed Computing
3. Cloud Concepts and Technologies
4. Virtualization in Depth
5. Cloud Architecture, Services and Platforms
6. MapReduce and Hadoop
7. Design of Cloud Applications
8. IaaS Providers (Amazon Web Services)
9. Cloud Application Development
10. Cloud Security
11. Big Data Analytics
12. Resource Elasticity

Learning Activities and Teaching Methods:

Distance Learning Lectures, Videos, Presentations, Tutorials, Theoretical Exercises and Assignments

Assessment Methods:

Assignments	20%
Course Project (programming, individual proj.)	20%

Final Examination	60%
-------------------	-----

Required Textbooks/Reading:

Authors	Title	Publisher	Year	ISBN
R. Buyya, C. Vecchiola, S. T. Selvi	Mastering Cloud Computing, Foundations and Applications Programming	Elsevier	2013	978-0-12-411454-8
A. Bahga, V. Madiseti	Cloud Computing: A Hands-On Approach	CreateSpace	2013	978-1494435141

Recommended Textbooks/Reading:

Authors	Title	Publisher	Year	ISBN
T. Erl, R. Puttini, Z. Mahmood	Cloud Computing: Concepts, Technology & Architecture	Prentice Hall	2013	978-0133387520
D. N. Chorafas	Cloud Computing Strategies	CRC Press	2010	978-1439834534
M. Yankelevich, M. Malamud, D. Mahaya	Pro Amazon EC2 and WS: Elastic Computing Cloud and Web Services Development with Java	APRESS	2011	978-1430224471

Course Requirements:

- All class materials such as slides, additional exercise sheets, announcements etc., will be available through the Moodle platform at: <http://moodle.lll.unic.ac.cy/>
- Published articles from Journals and Conference Proceedings are considered as part of the required reading material
- Students should strictly adhere to deadlines in the assigned work
- Homework assignments will be submitted **1 week** after their announcement, unless otherwise stated on their respective description sheet.
- The Course project will be submitted on the Final Examination date.
- The Final Examination will be comprehensive.
- **No make-up tests/exams will be given unless there is a serious reason. Documented evidence is required to support such cases.**
- Under NO circumstances will a make-up exam be given to anyone who has taken the regular test/exam.
- Anyone failing to take an exam will be assigned a grade zero for that particular exam.
- Plagiarism in all assessment work is a serious offense, which leads to severe punishment.

Grading Policy:

A	Excellent	93-100	4.0
A-		90-92	3.7
B+	Very Good	87-89	3.3
B		83-86	3.0
B-		80-82	2.7
C+	Good	77-79	2.3

<i>C</i>		73-76	2.0
<i>C-</i>		70-72	1.7
<i>D+</i>	Poor but Acceptable	67-69	1.3
<i>D</i>		63-66	1.0
<i>D-</i>		60-62	0.7
<i>F</i>	Failure	0-59	0.0

Schedule of Weekly Lectures and Assessment:

Please refer to course / module Study Guide



UNIVERSITY *of* NICOSIA

SCHOOL OF SCIENCES AND ENGINEERING

MSc IN COMPUTER SCIENCE

Study Guide

COMP-521DL: Cloud Computing

Dr. Nicholas Loulloudes

(loulloudes.n@unic.ac.cy)

Nicosia

2017

Table of Contents

Introductory Note	52
Section 1 - Introduction to Cloud Computing	53
Section 2 – Principles of Parallel and Distributed Computing	56
Section 3 – Cloud Concepts and Technologies	58
Section 4 – Virtualization in Depth	61
Section 5 – Cloud Architecture, Services and Platforms	63
Section 6 – MapReduce and Hadoop	65
Section 7 – Design of Cloud Applications	67
Section 8 – IaaS Providers (Amazon Web Services and Google Compute Engine)	70
Section 9 – Cloud Application Development	72
Section 10 – Cloud Security	74
Section 11 – Big Data Analytics	76
Section 12 – Resource Elasticity	78

Introductory Note

This Study Guide is a basic supplement for the distance learning course “Cloud Computing”, which is offered by the distance learning M.Sc. programme in Computer Science. The aim of this Guide is to direct the students and help them into making systematic use of the educational material on which the teaching of the course is based.

The Guide must be used in common with the Course Outline and with the educational material (course textbooks, PowerPoint presentations, bibliography and book chapters indicated for each lecture, instructions for the preparation of assignments and course project, etc.) which has been uploaded to the interactive internet-based platform of the course (Moodle). The student must start his/her studying from the PowerPoint presentation of each lecture, and then, taking advice from the present Guide, he/she must extend his/her knowledge making use of the rest of the educational material of each lecture.

This Guide includes 12 sections. Each of these sections is composed of the following parts: objectives, expected learning outcomes, key words, bibliography, additional bibliography, activities, and synopsis. Conducting the activities is essential, since you will be helped into absorbing the material of each lecture as well as into developing critical thought. The activities and the proposed bibliography will be enriched throughout the semester.

Section 1 - Introduction to Cloud Computing

Objectives

The aim of this first section, is to introduce students to the domain of Cloud Computing as well as the requirements and contents of this distance learning course. By utilizing day-to-day, exemplary applications across diverse domains (science, engineering, education, social networks, etc.) we shall briefly touch upon the fundamental concepts in the emergence of Cloud Computing and its prominent technologies. Students will receive principal exposure to the intrinsic properties of these environments, their impact on generating economies of scale and the driving factors that merit them as a study / research area of their own.

Expected Learning Outcomes

After the completion of this section, the students are expected to:

- Perceive the course content, timeline and structure of all the related sections.
- Understand the main aims of the course, so that these will constitute the central axes of their studying in regard to the following 11 sections.
- Acknowledge their requirements as per the course contract.
- Be aware of all resources (Required / Suggested textbooks, online libraries, Moodle, etc.) and contact endpoints (lecturer / teaching assistant emails) for the course.
- Acknowledge how to enrich their learning experience through the Moodle system functions (forum, wiki, Q&A section)
- Comprehend at a high-level the principles of Cloud Computing, architectures, system models and applications.
- Comprehend at a high-level the factors that motivate their in-depth study.

Key Words

Introduction, course contract, outline, requirements, Moodle, computing, Internetworking, WWW, resources, applications, Grid Computing, Online Social Networks, On-Demand Multimedia, Big Data, Artificial Intelligence, Machine Learning, NIST.

Bibliography

- A. Bahga, V. Madisetti, *Cloud Computing: A Hands-On Approach* (Independent Publ. Plat.), Chapter 1.
- R. Buyya, C. Vecchiola, S. T. Selvi, *Mastering Cloud Computing, Foundations and Applications Programming*, (Elsevier), Chapter 1.

Additional Readings

- P. Mell, T. Grance, “*The NIST definition of Cloud Computing*”, NIST SP 800-145

Activities

1. Read Chapter 1 from required textbook
2. During your activities try to pay attention to how many of your online interactions are potentially with Cloud-based system. In what perspective do these systems improve or make harder your daily life?
3. Start your semester diary (using the Moodle platform), where you will keep track of the Cloud-based systems that you are interacting with and note down when you encounter problems with these systems. For each problem you face, describe in your own words what went wrong, and what could possibly be the cause of it.
4. From additional readings, read the National Institute of Standards and Technology (NIST) definition of Cloud Computing.
5. Solve the review exercises that are assigned from Chapter 1 of the required textbooks

Section 2 – Principles of Parallel and Distributed Computing

Objectives

Regardless the fact that Cloud Computing is a relative modern technological trend (a buzzword), it has its roots in one the two dominant models of computing – parallel computing; more specifically, distributed and parallel computing which dates back in the early 50's. Through this chapter, students will be exposed to the fundamental principles of parallel and distributed computing and discuss those models and conceptual frameworks that serve as building blocks of modern cloud computing systems, services and applications. Among other, we will discuss the major elements of distributed and parallel computing, their differences and the technologies that facilitates those.

Expected Learning Outcomes

After the completion of this section, the students are expected to:

- Know the differences of parallel versus distributed computing
- Adequately explain what is parallel processing, the factors that influence its development and various processing approaches
- Enlist the established hardware architectures for parallel processing and their key characteristics
- Understand how different levels of parallelism can be achieved
- Explain the core elements of distributed computing, architectural styles and inter-process communication paradigms
- Comprehend the concepts of remote procedure call, distributed object frameworks and service-oriented computing.

Key Words

Sequential, Parallel, distributed, computing, processors, shared vs distributed memory, tasks, threads, data, CPU, workload, instruction, client-server, p2p, hardware, SISD, MISD, MIMD,

farmer-and-worker, algorithms, IPC, communication, platform, batch, pipe, flow, RPC, RMI, service, SOA, High Performance Computing, HPC.

Bibliography

- R. Buyya, C. Vecchiola, S. T. Selvi, *Mastering Cloud Computing, Foundations and Applications Programming*, (Elsevier), Chapter 2.

Additional Readings

- B. M. Leiner et.al, *Brief History of the Internet*, Oct. 2012.
- [Timeline of Computer History](#)

Activities

1. Read Chapter 2 from required textbook
2. Solve the review exercises that are assigned from Chapter 2 of the required textbooks
3. Read the article by B. M. Leiner. et.al, on the History of the Internet.
4. Browse the timeline of Computer History. Record in your semester diary, events that you think were highly notable (according to your own criteria) and expand on those records. Use the course forum on Moodle to publish these records to your colleagues under the topic “Notable Events in Computer History”.

Section 3 – Cloud Concepts and Technologies

Objectives

This section aims to provide an overview of the Cloud Computing founding concepts and its supporting technologies. At first, students will become familiar with the principle of virtualization, a core mechanism dating back in the era of mainframes, which facilitates abstraction of computing, memory and storage resources. Moreover, this section will introduce the techniques through which workloads are balanced across virtual resources, to enable service and/or application scalability and elasticity. Furthermore, it will expand upon these concepts and demonstrate how virtualization can be applied in network elements to establish service defined networks. Finally, we will overview how traditional identification, access, billing and auditing processes are adapted for Cloud computing.

Expected Learning Outcomes

After the completion of this section, the students are expected to:

- Have an adequate perception of the principal concepts that enable Cloud computing
- Gain a high-level understanding of the fundamentals of resource virtualization
- Be able to describe an application's deployment lifecycle
- Understand the concepts of load balancing, scalability and elasticity
- Comprehend how virtualization can be applied to networking elements to enable software defined networks.
- Understand the benefits of monitoring
- Understand how Identity and Access is enforced and managed in Cloud environments

Key Words

Virtualization, resources, scalability, elasticity, deployment, application lifecycle, refinement, replication, monitoring, software defined networks, SDN, network functions, VNF, NFV, access, identity, IAM, SLA, audit, billing, fixed, spot pricing.

Bibliography

- A. Bahga, V. Madiseti, *Cloud Computing: A Hands-On Approach* (Independent Publ. Plat.), Chapter 2.

Additional Bibliography

- T. Erl, R. Puttini, Z. Mahmood, *Cloud Computing: Concepts, Technology & Architecture* (Prentice Hall), Chapter 3.

Activities

1. Read Chapter 2 from the required textbook - *Cloud Computing: A Hands-On Approach*
2. Read Chapter 3 from the textbook in the additional bibliography - *Cloud Computing: Concepts, Technology & Architecture*
3. Solve the review exercises that are assigned from Chapter 2 of the required textbook

Section 4 – Virtualization in Depth

Objectives

This section extends upon the previous chapter and provides a more in-depth look to the core concept of virtualization. Virtualization is widely considered as a core element of Cloud computing, which in a nutshell enables the creation of many virtual environments sharing the same physical resources simultaneously. The aim of this section is to provide students with extended knowledge in characteristics of such virtualized environments, and a taxonomy of techniques currently available. Moreover, it defines the relationship of cloud computing and virtualization, as well as the pros and cons of sharing physical resources among virtual environments.

Expected Learning Outcomes

After the completion of this section, the students are expected to:

- Have a solid understanding of the relationship between virtualization and Cloud Computing
- Be familiar with the characteristics of virtual environments
- Enlist and discuss the pros and cons of virtualization
- Demonstrate the ability to taxonomize various virtualization techniques and adequately explain their individual execution processes
- Enlist reference implementations for each class of virtualization
- Be familiar with recent advances in virtualization / containerization technologies
- Demonstrate the ability to select an appropriate virtualization technology for given application domains

Key Words

Virtualization, security, managed execution, portability, taxonomy, machine reference model, virtual machine, instance, VMI, manager, VMM, hypervisor, para-virtualization, Xen, full-virtualization, VMware, containers, LXC/LXD, Docker

Bibliography

- R. Buyya, C. Vecchiola, S. T. Selvi, *Mastering Cloud Computing, Foundations and Applications Programming*, (Elsevier), Chapter 3.

Additional Bibliography

- T. Erl, R. Puttini, Z. Mahmood, *Cloud Computing: Concepts, Technology & Architecture* (Prentice Hall), Section 5.3.

Additional Readings

- Michael Pearce, Sherali Zeadally, and Ray Hunt. 2013. *Virtualization: Issues, security threats, and solutions*. ACM Comput. Surv. 45, 2, Article 17 (March 2013).
- Iliia Pietri and Rizos Sakellariou. 2016. *Mapping Virtual Machines onto Physical Machines in Cloud Computing: A Survey*. ACM Comput. Surv. 49, 3, Article 49 (October 2016).

Activities

1. Read Chapter 3 from the required textbook - *Mastering Cloud Computing, Foundations and Applications Programming*
2. Read Section 5.3 from the textbook - *Cloud Computing: Concepts, Technology & Architecture*
3. Solve the review exercises that are assigned from Chapter 3 of the required textbook
4. Read the articles by M. Pearce. et.al and I. Pietri. et. al., listed in the additional readings section
5. Given various applications from a range of domains, elaborate which is the most suitable virtualization / containerization technique. Post your discussion in the course forum on Moodle

Section 5 – Cloud Architecture, Services and Platforms

Objectives

This section presents Cloud Computing as an Internet-centric utility service that makes widely available - and on-demand - IT infrastructure, development platforms, applications and services to the masses. We will discuss in detail the architecture of the Cloud reference model through which Cloud offerings can be categorized in three main classes: Infrastructure-as-a -Service (IaaS), Platform-as-a-Service (PaaS), Software-as-a-Service (SaaS). Furthermore, we will examine the types of Cloud offerings (private, public, hybrid), as well as their intrinsic operational challenges.

Expected Learning Outcomes

After the completion of this section, the students are expected to:

- Have an adequate perception of the Cloud Computing reference model
- Have a strong understanding of the three main classes of Cloud offerings, as well as other subsidiary classes that arise from their core elements
- Critically compare the different classes and types of Cloud offerings, and show competence in classifying well-known and established Internet applications / services under those
- Enlist service and platform offerings from key Cloud providers such as Amazon, Google and Microsoft and their features
- Critically assess the challenges in designing a Cloud service
- Understand the economic pillars of Cloud Computing (Clouconomics) as a utility service

Key Words

Utility, on-demand, architecture, reference, deliver, model, IaaS, PaaS, SaaS, RaaS, MaaS, public, private, hybrid, community, clouconomics, interoperability, standards, scalability, fault tolerance, security, privacy, trust.

Bibliography

- R. Buyya, C. Vecchiola, S. T. Selvi, *Mastering Cloud Computing, Foundations and Applications Programming*, (Elsevier), Chapter 4.

Additional Bibliography

- A. Bahga, V. Madisetti, *Cloud Computing: A Hands-On Approach* (Independent Publ. Plat.), Chapter 3.

Additional Readings

- F. Liu. et. al., “*NIST Cloud Computing Reference Architecture*”, NIST SP 500-292
- J. Weinman, “*Clouconomics: A Rigorous Approach to Cloud Benefit Quantification*”, The Journal of Software Technology, October 2011, Vol. 14, No. 4, pp. 10-18

Activities

1. Read Chapter 4 from the required textbook - *Mastering Cloud Computing, Foundations and Applications Programming*
2. Read Chapter 3 from the textbook - *Cloud Computing: A Hands-On Approach*
3. From additional readings, read the National Institute of Standards and Technology (NIST) Cloud Computing Reference Architecture
4. From additional readings, read the article by J. Weinman on Clouconomics

Section 6 – MapReduce and Hadoop

Objectives

This section aims to introduce students and provide a hands-on experience to the MapReduce programming model and its associated reference implementation – Hadoop - for processing and generating big data sets with a parallel, distributed algorithm on a cluster. Students will obtain a short introduction on the motivation behind the MapReduce model and consequently will become familiar with the MapReduce Job Execution (Nodes, Trackers and Workflows), as well as the various schedules available in Hadoop. Finally, students will learn how to setup and deploy a Hadoop cluster on their personal machines or a virtualized compute cluster.

Expected Learning Outcomes

After the completion of this section, the students are expected to:

- Have absorbed the relationship among the MapReduce programming model and Cloud Computing
- Comprehend in detail the MapReduce programming model and the motivation behind its existence.
- Have a complete understanding of the Hadoop MapReduce Job Execution Workflow
- Demonstrate the ability to execute and retrieve the results of simple MapReduce jobs.

Key Words

MapReduce, Apache, Hadoop, data analysis, split-and-combine, partition, steps, Execution Workflow, NameNode, DataNode, JobTracker, TaskTracker, Scheduler, FIFO, Fair, Capacity, Java, HDFS, I/O, Class

Bibliography

- A. Bahga, V. Madiseti, *Cloud Computing: A Hands-On Approach* (Independent Publ. Plat.), Chapter 4.

Additional Readings

- J. Dean and S. Ghemawat, “*MapReduce: Simplified Data Processing on Large Clusters*”, In *Communications of the ACM* (New York, NY 2008), Volume 51 Issue 1, pp. 107 – 113.

Activities

1. Read Chapter 4 from the required textbook - *Cloud Computing: A Hands-On Approach*
2. Read the article on MapReduce by J. Dean and S. Ghemawat
3. Solve the review exercises that are assigned from Chapter 4 of the required textbook
4. Experiment with the provided MapReduce programs, execute them and obtain the results.

Section 7 – Design of Cloud Applications

Objectives

Throughout this section, students will be presented to various methodologies for the efficient and effective design of Cloud applications. Specifically, students will be gain the necessary knowledge that will assist the understanding of key considerations when designing (and deploying) Cloud-centric applications such as the level of interaction among actors. Moreover, students will become familiar with reference architectures for cloud applications and their associated characteristics, pros and cons. Additionally, various well-established design methodologies will be studied to provide students with the necessary theoretical background and assessment skills for designing scalable and well-behaved Cloud applications.

Expected Learning Outcomes

After the completion of this section, the students are expected to:

- Critically asses the challenges when designing Internet-centric Cloud applications
- Have an adequate understanding of the various methodologies that facilitate efficient and effective designs under various domain workloads
- Compare and contrast the different reference architectures for cloud applications
- Present the available Data Storage approaches
- Demonstrate the ability to select the most appropriate reference architecture for a given workload

Key Words

Reference architecture, multi-tier, n-tier, 3-tier, microservices, SOA, MVC, RESTful WS, loose coupling, availability, scalability, elasticity, fault tolerance, resiliency, replication, redundancy, security, response, bursting, lifecycle, interaction.

Bibliography

- A. Bahga, V. Madiseti, *Cloud Computing: A Hands-On Approach* (Independent Publ. Plat.), Chapter 5

Additional Readings

- C. Fehling, F. Leymann, R. Retter, "Your Coffee Shop Uses Cloud Computing", *IEEE Internet Computing*, vol. 18, pp. 52-59, Sept.-Oct. 2014
- O. A. Ben-Yehuda, M. Ben-Yehuda, A. Schuster, and D. Tsafir. "The rise of RaaS: the resource-as-a-service cloud". *Communications. ACM* 57, 7 (July 2014), 76-84

Activities

1. Read Chapter 5 from the required textbook - *Cloud Computing: A Hands-On Approach*
2. Read the article by C. Fehling, F. Leymann and R. Retter.
3. Read the two articles by C. Fehling et. al, and O. A. Ben-Yehuda et. al. and then do the following: a) Use a few sentences to summarize each article; b) Give a short description of what each paper is trying to solve; c) Summarize the contribution of the paper;
4. Solve the review exercises that are assigned from Chapter 5 of the required textbook
5. In your semester diary, record what could be the reference architecture for popular web platforms like YouTube, Facebook, LinkedIn, Twitter, etc. In your view, what were the challenges that engineers were faced with when designing these platforms

Section 8 – IaaS Providers (Amazon Web Services)

Objectives

This section aims to provide students with a hands-on, technical experience with Amazon's Web Services (AWS). Students will learn how to create compute instances, with specific virtual resource requirements (CPU, Memory, storage, network), consequently access those instances, execute programs, retrieve results and terminate instances. Students will obtain free-usage credits by enrolling to accredited education programmes (i.e. GitHub student education pack), monitor their virtual instance credit usage and set quotas.

Expected Learning Outcomes

After the completion of this section, the students are expected to:

- Have a good overall understanding of AWS
- Adequately explain what is a machine image and how it differs from a virtual machine instance
- Explain what are instance types and how these change in terms of virtualized resources
- Demonstrate the ability to obtain access and request specific Elastic Compute (EC2) virtualized resources and start instances
- Monitor resource utilization from dashboards and terminate execution when required

Key Words

AWS, EC2, S3, SSH, SCP, Linux, AMI, instance, region, Compute Engine, credits, education, resource utilization, billing, pricing, security, firewall.

Bibliography

- R. Buyya, C. Vecchiola, S. T. Selvi, *Mastering Cloud Computing, Foundations and Applications Programming*, (Elsevier), Chapter 9

Additional Readings

- Amazon Web Services – Elastic Compute (EC2) - Online Documentation
- GitHub Student Education Pack
- C. Krintz, "*The AppScale Cloud Platform: Enabling Portable, Scalable Web Application Deployment*", IEEE Internet Computing, vol. 17, pp. 72-75, March-April 2013

Activities

1. Read Chapter 9 from the required textbook - *Mastering Cloud Computing, Foundations and Applications Programming*
2. Read online documentation on AWS EC2. Links provided on Moodle
3. View the How-To video tutorials on EC2 available on Moodle
4. Create a student education pack obtaining free-usage credits for AWS.
5. Follow-up the video tutorials for instantiating your own virtual machines (VM) to EC2
6. Follow the instructions on Moodle on transferring simple Java programs to your VM and executing them.

Section 9 – Cloud Application Development

Objectives

Following the design principles discussed on Section 7 and the ability gained in Section 8 to utilize IaaS providers such as AWS and Google, this section will provide the necessary material to students that will enable them to develop programmatically Cloud applications. It provides application development methodologies for the IaaS and PaaS service model and provides best practices using the Java and Python programming languages. Students will choose their development language of preference, select a problem domain, develop a solution and deploy the solution on one of the two infrastructures visited in the previous section. This section will also present the necessary tooling (IDE) and frameworks that provide seamless application lifecycle management.

Expected Learning Outcomes

After the completion of this section, the students are expected to:

- Have adequate understanding of application methodologies for IaaS and PaaS service models
- Demonstrate the ability to develop simple Cloud applications and deploy them on public infrastructures
- Demonstrate the ability to select the most suitable deployment platform, considering workload constraints as well as performance and pricing requirements.
- Critically assess the challenges of developing a scalable and resilient Cloud application, and expand that reasoning on the applications that were developed

Key Words

IDE, Development approaches, programming, Java, Python, REST, API, Eclipse, Cloud Management Framework, CAMF, portability, GitHub, DockerHub.

Bibliography

- A. Bahga, V. Madiseti, *Cloud Computing: A Hands-On Approach* (Independent Publ. Plat.), Chapter 8

Additional Readings

- K. Börner. 2011. "Plug-and-play macroscopes". *Communications. ACM* 54, 3 (March 2011), 60-69.
- N. Loulloudes, C. Sofokleous, D. Trihinas, M. D. Dikaiakos, G. Pallis, "Enabling Interoperable Cloud Application Management through an Open Source Ecosystem", *IEEE Internet Computing*, 19(3), May/June 2015
- S. Sofokleous, N. Loulloudes, D. Trihinas, G. Pallis, M. D. Dikaiakos. "c-Eclipse: An Open-Source Management Framework for Cloud Applications", In Proceedings of 20th International Conference on Parallel Processing (Euro-Par), 2014

Activities

1. Read Chapter 8 from the required textbook - *Cloud Computing: A Hands-On Approach*
2. Solve the exercises that are assigned from Chapters 8
3. Read the three articles by K. Börner, N. Loulloudes and S. Sofokleous and then do the following: a) Use a few sentences to summarize each article; b) Give a short description of what each paper is trying to solve; c) Summarize the contribution of the paper; d) Finally, give a small description of how the ideas of the paper is relevant to cloud application design & development
4. Develop and deploy your applications of choice
5. Use your semester diary to record any obstacles you come across, what solution you applied to overcome those and how you arrived at that solution. Also record any topic(s) that stimulated your interest

Section 10 – Cloud Security

Objectives

This section focuses on the practices employed in achieving a high-level of security in the Cloud, both for infrastructure (clusters) and deployed applications / services. It introduces the security challenges in the Cloud, as how these are considered in the Cloud Control Matrix (CCM) proposed by the Cloud Security Alliance (CSA). Students will also study the concepts of Identity and Access Management (IAM), Authentication (Single Sign-On) and Authorization, Key/Token management and Data protection through encryption.

Expected Learning Outcomes

After the completion of this section, the students are expected to:

- Have an adequate perception of the security challenges In Cloud computing
- Be familiar with the Cloud Control Matrix (CCM) and demonstrate the ability to apply it to an application of your choice
- Explain the concepts of encryption and the difference between symmetric and asymmetric encryption.
- Be aware of Public Key Infrastructures (PKI) and how these work
- Have a good understanding on the methodologies currently employed to guarantee Identity and Access management, as well as Authentication and Authorization.
- Demonstrate the ability to pinpoint and adequately describe how the above techniques function in the solutions offered by the two Cloud providers of Section

Key Words

Security, encryption, symmetric, asymmetric, privacy, CCM, CSA, IAM, identity, authentication, authorization, access, Single Sign-On, SSO, CAS, security key, PKI, token, auditing, expiration, data protection, vulnerability, attack, intrusion.

Bibliography

- A. Bahga, V. Madiseti, *Cloud Computing: A Hands-On Approach* (Independent Publ. Plat.), Chapter 12

Additional Bibliography

- T. Erl, R. Puttini, Z. Mahmood, *Cloud Computing: Concepts, Technology & Architecture* (Prentice Hall), Chapter 10

Activities

1. Read Chapter 12 from the required textbook - *Cloud Computing: A Hands-On Approach*
2. Read Chapter 10 from the textbook - *Cloud Computing: Concepts, Technology & Architecture*
3. Solve the exercises that are assigned from Chapters 12
4. Use your semester diary to record any security concerns you had while using a Cloud service, or when designing / developing a Cloud application. Post those concerns on the course forum on Moodle under the topic “Cloud Security Concerns”

Section 11 – Big Data Analytics

Objectives

Big Data are everywhere now. Wherever we look in our physical and digital worlds, there are endless possibilities of generating, transforming, analyzing and capitalizing upon hundreds or even thousands of data points. This section aims to make a short but meaningful introduction to the world of Big Data Analytics and present how this concept is widely considered as inseparable from Cloud computing. Students will have the possibility to learn various algorithms for clustering and classifying big data. Moreover, we will examine how the results of Big Data analysis can be useful to online recommendation systems even across heterogeneous application domains.

Expected Learning Outcomes

After the completion of this section, the students are expected to:

- Have an adequate understanding of what is coined as Big Data and be able to explain in their own words
- Be familiar with the relationship and co-existence of Big Data and Cloud technologies and how the latter support the former
- Demonstrate the ability to explain how the k-Means clustering algorithm works, and solve related problems
- Demonstrate the ability to explain how the Naïve-Bayes, Decision Trees and Random forests classification algorithms work, and solve related problems
- Understand the concept of recommendations systems and how the effectiveness of these is largely improved through Big Data analytics

Key Words

Big Data, volume, velocity, data intensive computing, analytics, data store, Clustering, k-Means, DBSCAN, MapReduce, Classification, Naïve-Bayes, Decision Trees, Random Forest, support vector machine, recommendation system

Bibliography

- A. Bahga, V. Madiseti, *Cloud Computing: A Hands-On Approach* (Independent Publ. Plat.), Chapter 9

Activities

- Read Chapter 9 from the required textbook - *Cloud Computing: A Hands-On Approach*
- Solve the exercises that are assigned from Chapters 9

Section 12 – Resource Elasticity

Objectives

This last section explores a defining characteristic of Cloud Computing which differentiates it significantly from previous proposed distributed computing paradigms. This characteristic is called resource elasticity and it is defined as the capacity of a Cloud system to workload changes by provisioning and de-provisioning resources in an intelligent and autonomic manner, such that at any time the available resources match the current demand as closely as possible. The driving factors of elasticity will be studied, as well as various models for predicting / monitoring demand and allocating adequate resources on the fly.

Expected Learning Outcomes

After the completion of this section, the students are expected to:

- Have adequate perception of the core aspects of resource elasticity and the motivating factors that drive its existence
- Understand scaling units for different resource types, and the scalability bounds on the amount of resources that can be allocated
- Be aware of the various resource elasticity patterns and dimensions (scale-out, scale-in, scale-up, etc.)
- Comprehend the challenges of rapidly estimating real-time changes in the workload and providing adequate resources
- Be aware of frameworks that enable qualitative and quantitative characterization of Cloud application's performance through modelling of its elastic properties.
- Demonstrate the ability to select the appropriate elasticity metrics (cloud and application level) to address the requirements of specific workload changes and resource provision calibration

Key Words

Resource, elasticity, provision, adaptive, automation, budget, scalability, metrics, cost, performance, time, requirements, infrastructure, monitoring, Nagios, benchmarking, SYBL, CELAR.

Bibliography

- N. Roman, S. Kounev, R. Reussner, “*Elasticity in Cloud Computing: What It Is, and What It Is Not*”, 10th International Conference on Autonomic Computing (ICAC), 2013.
- S. Dustdar, Y. Guo, B. Satzger, H.-L. Truong, “*Principles of Elastic Processes*”, IEEE Internet Computing, Volume: 15, Issue: 5, Sept.-Oct. 2011.
- G. Copil, D. Moldovan, H.-L. Truong, S. Dustdar, “*SYBL: an Extensible Language for Controlling Elasticity in Cloud Applications*” 2013, CCGrid 2013.
- I. Giannakopoulos, N. Papailiou, C. Mantas, I. Konstantinou, D. Tsoumakos and N. Koziris, “*CELAR: Automated Application Elasticity Platform*”, IEEE International Conference on Big Data, Washington DC, USA, 2014
- D. Trihinas, G. Pallis, M.D. Dikaiakos, “*Monitoring Elastically Adaptive Multi-Cloud Services*”, IEEE Transactions on Cloud Computing.

Activities

1. Read the first two articles by N. Roman, and S. Dustdar.
2. Select one of the remaining articles and then do the following: a) Use a few sentences to summarize each article; b) Give a short description of what each paper is trying to solve; c) Summarize the contribution of the paper; d) Finally, give a small description of how the ideas of the paper is relevant to resource elasticity
3. Use your semester diary to record your observations on public web services you use daily that might employ resource elasticity mechanisms to satisfy the constantly changing demand. Provide some examples, and describe Cloud and/or application metrics that could potentially be employed in each case.



Course Code	Course Title	ECTS Credits
COMP-523DL	Game Programming	10
Department	Semester	Prerequisites
Computer Science	Fall, Spring	None
Type of Course	Field	Language of Instruction
Elective	Computer Science	Greek/English
Level of Course	Year of Study	Lecturer(s)
2 st Cycle	2 st	Dr Andreas Savva
Mode of Delivery	Work Placement	Co-requisites
Distance Learning	N/A	None

Objectives of the Course:

The main objectives of the course are to:

- Introduce students the standards and innovative techniques for game programming.
- Provide the tools and inspiration that game developers need to excel.
- Feature cutting-edge, ready-to-use techniques contributed by industry veterans and experts.
- Introduce a plethora of time-saving, ready-to-use methods for the developer's tool box.
- Provide answers to the needs of passionate developers, eager newcomers, voracious production requirements, and to the demand for innovating and entertaining game-play.

- Provide students with practical ideas and techniques and get them ready to develop games that are more inventive, entertaining, and satisfying.
- Provide students the mathematical background of game development.
- Provide practical experience to computer game development.

Learning Outcomes:

Upon completion of the course students should be able to:

- Deal with the practicalities of writing a computer game.
- Explain the mathematics involved in computer games.
- Explain the AI algorithms and Physical Laws involved in generating computer games.
- Implement computer games for various platforms.
- Describe how Computer Graphics, AI, Physics and Networks are combined in developing computer games.

Course Contents:

1. Games: History and Society
 - The First Video Games
 - Games for the Masses, The Console Kings
 - Audience and Demographics, Societal Reaction to Games, Cultural Issues, Society within Games
2. Game Design
 - The Game Designer, A Model of Games
 - Game, Player and Experience
 - Play Mechanics, Interface, Game Systems
 - Design Work, Prototyping and Playtesting Cycles, Playtesting
3. Game Production and Project Management
 - Programming Teams
 - Methodologies, Common Practices and Quality
 - Leveraging Existing Code
 - Platforms
 - Concept Phase, Preproduction Phase, Production Phase, Postproduction
4. Programming Languages and Fundamentals
 - C++ and Game Development, Java, Scripting Languages
 - Data Structures, Object-Oriented Design in Games
 - Component Systems, Design Patterns
5. Game Architecture, Memory and Debugging
 - Bird's-Eye View of a Game, Initialization/Shutdown Steps, Main Game Loop, Game Entities
 - Memory Management, File I/O, Game Resources, Serialization
 - The Five-Step Debugging Process, Expert Debugging Tips, Tough Debugging Scenarios and Patterns
 - Understanding the Underlying System, Adding Infrastructure to Assist in Debugging, Prevention of Bugs
6. Mathematics and Physics in Games
 - Applied Trigonometry, Vectors and Matrices, Transformations, Geometry
 - Collision Detection, Overlap Testing, Intersection Testing
 - Simplified Geometry, Bounding Volumes
 - Terrain Collision Detection, Collision Resolution, Physics Simulations
 - Beyond Particles, Third-Party Physics Engines
7. Graphics and Animation
 - Introduction to 3D Modeling, Box Modeling with Polygons, NURBS, Subdivision Surfaces
 - 3D Sculpting, Reverse Engineering, BSP Modeling, Modeling Methodology
 - Texture Mapping, Mapping UV Coordinates
 - Animation, Motion Capture, Motion Extraction, Mesh Deformation, Inverse Kinematics, Collision Detection

- Real-Time Animation Playback, Character Animation, Facial Animation, Simulation Animation
8. Artificial Intelligence
- AI for Games, Game Agents
 - Finite-State Machines
 - Common AI Techniques, Search Space, Pathfinding
9. Audio and Network
- Programming Basic Audio
 - Programming Music Systems
 - Programming Advanced Audio
10. Game Industry
- Game Developers, Publishers, Platform Holders
 - Deal Dynamics, Payment Negotiation
 - Advertising, Media, Publicity Opportunities, Marketing
 - IP Protection, The IP Content of Video Games
 - Patents, Copyrights, Trademarks, Transfers of IP Rights
 - Video Game Content Regulation

Learning Activities and Teaching Methods:

Presentations, Textbook Exercises, Online Exercises

Assessment Methods:

Final Exam	60%
Project (Individual – Design & implementation of a game)	30%
Assignments (2)	10%

Required Textbooks/Reading:

Authors	Title	Publisher	Year	ISBN
Edited by Steve Rabin	Introduction to Game Development, 2 nd Ed.	Course Technology	2010	ISBN-13: 978-1-58450-679-9 ISBN-10: 1-58450-679-2 eISBN-10: 1-58450-705-5

Recommended Textbooks/Reading:

Authors	Title	Publisher	Year	ISBN
Edited by Adam Lake	Game Programming Gems 8	Course Technology, Cengage Learning	2011	ISBN-13:978-1-58450- 702-4 ISBN-10:1-58450-702- 0 eISBN-10:1-43545- 771-4
Eric Lengyel	Mathematics for 3D Game Programming and Computer Graphics, 3rd Edition	Course Technology, Cengage Learning	2012	ISBN-13: 978-1- 43545-886-4 ISBN-10: 1-43545- 886-9

Weekly Schedule

See course/module Study Guide



UNIVERSITY *of* NICOSIA

SCHOOL OF SCIENCES AND ENGINEERING

MSc IN COMPUTER SCIENCE

Study Guide

COMP-523DL: Game Programming

Dr Andreas Savva

Nicosia

2017

Contents

Contents	86
Introductory note	87
Section 1 – Games: History and Society	88
Section 2 – Game Design	90
Section 3 – Game Production and Project Management	92
Section 4 – Programming Languages and Fundamentals	94
Section 5 – Game Architecture, Memory and Debugging	96
Section 6 – Game Programming: Math and Collision Detection	98
Section 7 – Game Programming: Physics	100
Section 8 – Graphics, 3D Surface Modeling, and Textures	102
Section 9 – Animation	104
Section 10 – Artificial Intelligence (AI)	106
Section 11 – Audio and Networking	108
Section 12 – Game Industry and Legal Issues	110

Introductory note

This Study Guide is a basic supplement for the distance learning course “Game Programming”, which is offered by the distance learning MSc programme in Computer Science. The aim of this Guide is to direct the students and help them into making systematic use of the educational material on which the teaching of the course is based.

The Guide must be used in conjunction with the Course Outline and with the educational material (course textbooks, lecture notes, PowerPoint presentations, programming exercises etc.) which has been uploaded to the interactive internet-based platform of the course (Moodle). The student must start his/her studying by the lecture notes, which correspond with the PowerPoint presentation of each lecture, and then, taking advice from the present Guide, he/she must extend his/her knowledge making use of the rest of the educational material of each lecture.

The Guide includes 12 sections. Each of these sections is composed of the following parts: objectives, expected learning outcomes, keywords, bibliography, activities, and synopsis. Conducting the activities is essential, since the student will be helped into absorbing the material of each lecture as well as into developing critical thought. The activities and the proposed bibliography will be enriched during the semester.

Dr Andreas Savva, Associate Professor of Computer Science, School Sciences and Engineering.

Section 1 – Games: History and Society

Objectives

This section travels through time from the first recorded video game in 1958 all the way to the present. There are many ways to view and compare history, so it starts with a timeline approach, and then break out specific platforms, studios, people, and genres to effectively understand specific lines of innovation. It also examines the impact of computer games to societies and cultures.

Expected learning outcomes

After the completion of this section, the students are expected to:

- Describe the history of video games.
- Describe some of the most popular history games that reached above and beyond the status of mere success, such as “Space Invaders”, “Pac-Man”, and “Tetris”.
- Distribute video games in the appropriate genre.
- Discuss how video games are viewed in different societies.
- Describe how different societies react to the content of different video games.
- Discuss how games influence human culture.

Keywords

Video games, game genres, society, cultural issues, Entertainment Software Association, Youth Violence.

Bibliography

Edited by Steve Rabin, “Introduction to Game Development”, Second Edition, Course Technology, 2010 – Sections: 1.1, 1.2.

Activities

- Exercises at the end of each section in the textbook.
- Quiz.

Summary

True success and longevity in the video game world often rely on a combination of creativity, business acumen, and luck. Just as in any emerging media, there is an evolution that takes place, as genres are defined and capabilities are explored. The consoles and computers of the year 2000 enable ways of game playing that weren’t possible in the early 1980s, while some classic games still remain classic games despite featuring outdated technology. Ultimately, as advanced as video games have become and as much money as the industry generates, the medium must still be

considered in its infancy. This does not invalidate the lessons learned from the designers and companies that have made a success in it, but serves to inform the future.

The success of video games as a fiscal and cultural entity is not without controversy, though, as people struggle to understand this emerging media's effect on society. In some cases, a game may find a niche within a particular culture or society; in others, a game element may inadvertently cause offense. In particular, concerns over the effects of violent games on youth are prevalent, with few comprehensive studies done that can point to clear answers.

Section 2 – Game Design

Objectives

This section introduces a brief glimpse of topics and issues common to video game design. While there are many, equally valid, ways to view a game, this chapter will take a functional view, focused on shortening the conceptual distance between what the player experiences and what the game actually does.

Expected learning outcomes

After the completion of this section, the students are expected to:

- Describe the phases of game design.
- Describe the objectives and outcomes of a game.
- Define the term “uncertainty” in games and describe why it is necessary to the experience of playing games.
- Identify target groups to aim for before designing a game.
- Design the game according to target audience preferences.

Key words

Game Designer, Player, Play Mechanics, Actions, Interface, Game Systems, Content, Design Work, Prototyping, Playtesting.

Bibliography

Edited by Steve Rabin, “Introduction to Game Development”, Second Edition, Course Technology, 2010 – Sections: 2.1, 2.2.

Activities

- Exercises at the end of each section in the textbook.
- Quiz.
- Project – Start designing a computer game.

Summary

This section provides students a brief picture of concepts, vocabulary, tools, and other issues related to video game design. While traditional writing, such as that for novels or screenplays, shares much with game writing, game writing offers some unique challenges. It is a great deal more complicated than traditional storytelling, with writers having to deal with things like multiple storylines, player choice, and technical limitations. It is imperative that a writer

understands these challenges and knows techniques for dealing with them. As more games begin to incorporate stories and genres continue to merge, the challenges will increase. That is why it is of the utmost importance for game writers to be aware of the target audience for the game, as well as the funding and game engine limitations in order to keep their writing within the scope of the project.

Furthermore, there is a growing demand for the interactive story. While traditional storytelling techniques should be understood and incorporated into the game, it is not enough, and further methods for dealing with interactivity issues need to be incorporated into the writing. There are many methods for adding interactivity to the narrative game, and the writer needs to work with the developer to decide which methods will work best for the type of game being developed. Currently, hybrid approaches are the best way to grant the player the illusion of interactivity and player agency, while still keeping the game constrained enough to fit within the limitations of a project. The future, however, might allow for more experimental interactive storytelling techniques that combine artificial intelligence with narrative, creating entirely new gameplay experiences.

Section 3 – Game Production and Project Management

Objectives

There is more to programming a game than sitting in front of a computer equipped with an editor and a compiler and banging away at the keyboard all night long. Most commercial games are created by large teams of people, ranging from just a handful to hundreds of programmers, depending on the team size and scope of the program. This section explains how programming teams are organized and what techniques they commonly use to effectively coordinate the work of all the team members and create a great game. There are five phases in the process of making a game: concept, preproduction, production, postproduction, and aftermarket. Here, the first four phases are fully discussed.

Expected learning outcomes

After the completion of this section, the students are expected to:

- Describe the three distinct areas of programming involved when creating a game: Game Code, Game Engine, and Team Organization.
- Explain what the “code-and-fix environment” is.
- Give the definition of the term “death-spiral”.
- Describe the “waterfall approach”, the “iterative methodology” and the “agile methodology”.
- Describe the Coding Standards of developing a software program.
- Design the implementation of a game so that it runs on the right platform, and if possible on a variety of platforms.
- Describe the process of creating a game from the point of view of the producer (or the project manager)
- Describe the phases of game production.

Keywords

Programming Teams, Game Code, Game Engine, Team Organization, Code-and-Fix Environment, Waterfall approach, Iterative methodology, Agile methodology, Game consoles, Multiplatform development.

Bibliography

Edited by Steve Rabin, “Introduction to Game Development”, Second Edition, Course Technology, 2010 – Sections: 3.1, 7.1.

Activities

- Exercises at the end of each section in the textbook.
- Quiz.
- Project – Continue on designing the game.

Summary

When it comes to writing code some common practices such as coding standards or use of source control can greatly help a team to work together. In particular, techniques that help improve the quality of the code – such as code reviews, good use of asserts, or unit and acceptance tests – will have a direct impact on the game itself. It is very important to research what code can be reused instead of writing your own. The game middleware industry is growing and becoming more important every year. That, combined with our choice of platform, will determine how development should be done and the areas on which the team should concentrate.

This section also discussed the job of the producer, especially his or her role in creating a game. From concept to preproduction, through production to postproduction and even into after-market, the producer manages and pushes and cajoles. The producer is the man or woman in the middle, under pressure from all directions and communicating outward, upward, and downward. It's a tremendous amount of work, but the reward is the satisfaction of seeing a finished product on the store shelf.

Section 4 – Programming Languages and Fundamentals

Objectives

You should always choose the right tool for the job, and a programming language is just that, a tool. Apart from a few physical limitations, you can almost get the job done with any language you want. However, if you choose the most appropriate language, the development will go much smoother, and you will get done faster. This section outlines the major languages used in game development, explains their strengths, and helps the developer decide which one to choose in which situation.

A firm understanding of programming fundamentals is required to program video games. This includes rudimentary data structures, object-oriented techniques, and a healthy repertoire of design patterns. While these topics are covered in many introductory programming books, this chapter will directly address how they apply to games, providing many useful insights along the way.

Expected learning outcomes

After the completion of this section, the students are expected to:

- Describe the advantages and disadvantages of the different programming languages in respect to implementing a game.
- Choose the ideal programming language for a particular game.
- Choose the right Data Structure for a particular game.
- Explain why it is best to call readymade functions rather than implementing our own ones.

Keywords

Game development, C++, Java, OpenGL, DirectX, Low-level and High level languages, Scripting languages, Data Structures, Arrays, Linked Lists, Object Oriented Design in games, Inheritance.

Bibliography

Edited by Steve Rabin, “Introduction to Game Development”, Second Edition, Course Technology, 2010 – Sections: 3.2, 3.3.

Activities

- Exercises at the end of each section in the textbook.
- Project – Select a programming language and start implementation.

Summary

C++ is the current language of choice for most game development. It combines the low-level control and high performance of C with the object-oriented approach and higher productivity of high-level languages. Knowing the language well is essential to take full advantage of it and avoid common pitfalls that are all too frequent in C++.

Java and C# have been making some inroads in game development over the years. With some of the recent updates to the Java language and libraries, the performance of Java programs has become much better. Combined with the fact that Java is easier and much cleaner than C++, as well as slightly higher level (which implies more productivity), it is easy to see why it has become a more attractive alternative.

More games every day are using scripting languages to write much of the highlevel game code. Whether it's done by designers or programmers, it allows for very fast development and iteration of game features. When using a scripting language, we can use one of the popular off-the-shelf ones (Python or Lua), adopt one for an existing game, or write our own.

To become an effective programmer, you need to know more than the specifics of a computer language. Data structures are fundamental to any structured program, and every game programmer should be well acquainted with the basics: arrays, lists, dictionaries, stacks, queues, and so forth. Additionally, a firm grasp of the object-oriented fundamentals should be considered essential in today's game development, with object-oriented languages such as C++ and Java being the norm.

Being familiar with the basics of object-oriented programming is not enough. It is important to apply it correctly and know when to choose a class inheritance hierarchy and when to use composition. Correctly identifying and using design patterns will also lead to faster development and a clearer architecture.

Section 5 – Game Architecture, Memory and Debugging

Objectives

A firm understanding of programming fundamentals is required to program video games. This includes rudimentary data structures, object-oriented techniques, and a healthy repertoire of design patterns. While these topics are covered in many introductory programming books, this chapter will directly address how they apply to games, providing many useful insights along the way.

The code necessary to create modern games is anything but simple. The days when the source code for a full game was just a couple of files and we didn't have to worry about overall structure and architecture are gone. In today's games, with million lines of code, it is vitally important to have a well-defined architecture in order to understand the source code, add new features, and finish the game on time.

Also every game needs to deal with the low-level details of the platform on which it runs. It needs to decide how memory will be used, how to load data from some type of storage, how to deal with the game resources, and so forth. Usually, that type of functionality is buried deep in the low-level systems of the game engine itself. Knowing how those systems work and the tradeoffs involved will make you a much more effective game programmer, even if you never have to implement them yourself. In this chapter, we will look at memory management, handling file I/O, organizing game resources, and the process of saving and loading game states.

Debugging a game, or any other piece of software, can be an extremely difficult task. For the most part, an experienced programmer can quickly identify and correct even the most baffling bug, but for the novice, it can quickly become a frustrating experience. To make matters worse, when you start looking for the source of a bug, you never know how long it will take to find. The key is not to panic and instead be disciplined and remain focused on the bug-finding process. This section will arm the student with techniques and knowledge to methodically find and prevent even the toughest bugs, while paying special attention to debugging issues and methods that are unique to games.

Expected learning outcomes

After the completion of this section, the students are expected to:

- Explain what the “Birds-Eye view of a game” is.
- Describe the Ad-Hoc, Directed Acyclic Graph (DAG), and Layered architectures.
- Describe and use where appropriate the “Resource Acquisition Is Initialization” philosophy.
- Describe the different steps involved in a game loop and the order in which they are executed.
- Define and explain what the “Game entities” are.
- Describe what the term “Memory Management” in respect to programming means.

- Define static and dynamic memory allocation.
- Explain what we mean by the terms “Game Resources” and “Serialization”.

Keywords

Game-specific code, Game-engine code, Birds-Eye view of a game, Game entities, Memory management, Static and Dynamic memory allocation, File I/O, Game resources, Serialization.

Bibliography

Edited by Steve Rabin, “Introduction to Game Development”, Second Edition, Course Technology, 2010 – Sections: 3.4, 3.5, 3.6.

Activities

- Exercises at the end of each section in the textbook.
- Project – Continue on Game development.

Summary

As projects grow in size and complexity, carefully considering the architecture of a game code base is becoming increasingly important, especially if you plan to reuse some of the code in future projects. At the highest level, games are usually a set of initialization/shutdown steps and one or more game loops. Each initialization step takes care of setting up any resources or systems needed by the game, and the corresponding shutdown step cleans up anything done by the initialization step. The game loop is executed once per frame, and it does all the tasks that have to be done each frame to make the game respond to the player: input gathering, simulation, collision, rendering, and so forth. This becomes increasingly complex with multicore architectures, as work must be divided up to be run in parallel, paying close attention to data access and synchronization.

Every game has some form of game entity. These are self-contained units of gameplay logic. They can be enemy units, animated scenery, the player avatar, or even just a trigger. Creating, managing, and updating these entities efficiently is very important for the smooth functioning of the game.

This section also presents what to expect from low-level systems such as memory management and file I/O. Memory allocation is often the cause of many bugs and crashes in games. A dynamic memory allocation system is illustrated to provide the student knowledge, safety, and control over memory allocations. The section provides the tools a student needs to debug games effectively.

Section 6 – Game Programming: Math and Collision Detection

Objectives

Mathematics has become an essential component of modern game development. As both the main processors and graphics processors in our gaming hardware become more powerful, the complexity of the mathematics used to model realistic environments and physical simulations increases without bound.

Collision detection will determine if and when two objects collide. Since it is not enough to merely detect the collision, collision resolution will figure out where each object should be once a collision is detected. Effectively, collision detection and resolution together will make objects solid so that they will never pass through each other. Calculating how objects move after the collision is the job of the physics, which is described in the next section.

This section provides an introduction to several fields of mathematics that are vital to today's game engines.

Expected learning outcomes

After the completion of this section, the students are expected to:

- Have a good knowledge of the mathematics involved a virtual 2D and 3D environment.
- Implement mathematical methods in order to apply physical laws in the 2D and 3D environment.
- Use the knowledge in trigonometry, vectors, matrices, geometry, to provide solutions to various aspects involved in the virtual world of a video game.
- Implement routines to visualize objects in 2D and 3D, and apply transformations, such as rotations, translations, scaling, to objects and to the camera, so that they are visualized from a different angle.
- Implement code to visualize objects using parallel and perspective projections.
- Use matrices to switch from one coordinate system to another.
- Use third-party physics routines to avoid collision without having to build them themselves.

Keywords

Coordinate systems, Trigonometry, Vectors, Matrices, Transformations, Geometry, rotation, translation, scaling, projection, camera, dot-product, cross-product, intersection, overlap, collision.

Bibliography

Edited by Steve Rabin, “Introduction to Game Development”, Second Edition, Course Technology, 2010 – Sections: 4.1, 4.2.

Activities

- Exercises at the end of each section in the textbook.
- Quiz
- Project – Continue on Game development.

Summary

This section introduces several mathematical aspects of computer game development, including trigonometry, vector and matrix arithmetic, coordinate transformations, and basic three-dimensional geometry. These concepts represent the foundations of many more advanced mathematical applications in game programming. Because they are so heavily relied upon in modern game making, a familiarity with such concepts can benefit both programmers and all members of a game development team

Collision detection and collision resolution enable game objects to behave as solid masses. Through either overlap testing or intersection testing, collisions can be detected. Once a collision has been detected, collision resolution corrects each object’s position and imparts appropriate velocities based on physical simulation.

Section 7 – Game Programming: Physics

Objectives

This section contains an introduction to physics simulation, with a focus on techniques that can run in real time. Following a technical discussion of a few fundamental algorithms that you can implement to add physics-based effects to your game, such as physics for particle systems, the chapter concludes with an overview of third-party physics engines that enable you to put sophisticated physics into your game without having to build a full custom physics engine. The concepts presented herein are not comprehensive, but are intended to provide sufficient information to enable you to begin developing your own physics engine and to understand the wide variety of realistic behaviors that are possible using third-party engines. The entire section can be useful in generating physics simulation within a runtime game engine, as well as within digital content creation tools.

Expected learning outcomes

After the completion of this section, the students are expected to:

- Have a good knowledge of the physics laws and theory involved in a virtual 2D and 3D environment.
- Implement techniques for physics simulations to run in real time.
- Implement algorithms to add physics-based effects to the game, such as physics for particle systems.
- Use third-party physics engines to put sophisticated physics into the game without having to build a full custom physics engine.
- Implement a general-purpose physics engine.

Keywords

Kinematics, Newton's Law, Collision response, numerical integration, force, particle systems, game engine.

Bibliography

Edited by Steve Rabin, "Introduction to Game Development", Second Edition, Course Technology, 2010 – Section: 4.3.

Activities

- Exercises at the end of each section in the textbook.
- Project – Continue on Game development.

Summary

Closed-form particle physics is extremely practical for games that require only simple physics. One significant benefit of these equations, if they are suitable, is that they are perfectly stable and will never cause floating-point overflow. In practice, these equations are only useful for spherical particles experiencing occasional collisions and at most a constant acceleration, such as that due to gravity (or piecewise-constant acceleration with restarts).

Numerical integration techniques remove the restriction that an object experiences only a constant force, making these techniques quite useful for implementing a general-purpose physics engine. These techniques are subject to stability concerns that you must consider carefully. Regardless of the stability considerations, these techniques open up a world of opportunity for physics simulation.

Section 8 – Graphics, 3D Surface Modeling, and Textures

Objectives

This section focuses on the rendering of three-dimensional scenes onto a flat screen of pixels. There are many ways to do this, but for games, the most common is to use custom hardware to render scenes made out of triangle-based meshes. While there are large differences in the wide range of graphics cards in desktop computers and the more special-purpose hardware in various consoles, there are also plenty of shared generalizations that travel well between most of the common platforms. Regardless of the low-level interface or API used, these common features tend to remain similar. With care, a developer can use these common features to write a graphics engine that works well on a variety of platforms.

In this section, the student will also learn about the process of creating 3D models. Some methods discussed are standard day-to-day work for a modeler, and some are less used or they are special-purpose methods, but are still worth mentioning. As an example of creating 3D objects, the generation of a few typical gaming models, including a car, an environment, and a low-polygon character are presented.

The kinds of texture maps that are often used, and the technical process of how an artist tells the computer how to apply a texture to a particular polygon on a model and explicitly how to display it on-screen, are also discussed.

Expected learning outcomes

After the completion of this section, the students are expected to:

- Explain different methods for 3D modeling.
- Describe the format of a 3D model in respect to vertices and faces.
- Describe the differences between interpolation and approximation modeling methods.
- Implement the most well known modeling methods such as: Hermite Interpolation, Natural Cubic splines, Bezier splines, B-splines, NURBS.
- Define the “Convex Hull” property, and the “Local Control” property in respect to surface modeling.
- Describe and implement “Subdivision surface” methods, such as Catmull-Clark method which is the most popular one.
- Describe a program’s UV mapping tools, the UV editor, and the controls within the editor window.

Keywords

3D modeling, Control points, Box modeling, polygons, splines, Bezier, B-splines, NURBS, Subdivision surfaces, 3D Sculpting, Reverse Engineering, BSP modeling, Texture mapping, UV coordinates.

Bibliography

Edited by Steve Rabin, “Introduction to Game Development”, Second Edition, Course Technology, 2010 – Sections: 6.2, 6.4.

Activities

- Exercises at the end of each section in the textbook.
- Quiz
- Project – Continue on Game development: Add transformation procedures.

Summary

Writing a full graphics engine for any cutting-edge hardware is extremely challenging. In most games, the rendering engine is a large module taking most of the memory and processing power of the system, and as such will have only a few specialized programmers who truly know how it works and what the performance characteristics are. This section provides the student a rough guide to the various capabilities and performance characteristics of most known engines. It also establishes enough of a framework to allow them to start using most of the common graphics APIs to create their own rendering engine.

Since a vast majority of game modeling is done with standard polygon editing, techniques specific to polygons is discussed. NURBS, subdivision surfaces, 3D sculpting, reverse engineering, and BSP modeling is also introduced. A step-by-step demonstration of how to model a character using polygons is presented. Three-dimensional modeling is both a technical and artistic endeavor. As you become more familiar with the tools, processes, and methods of creating models, the technical side will become second nature, and you can focus your attention on the much more important artistic side.

Texture maps are the color component of 3D models, and UV texture mapping is an intrinsic part that can make or break an object’s close-up appearance in a video game. The topics covered represent a large and valuable portion of any 3D artist’s employable skill set. A professional texture artist will learn all that he can about his program’s UV mapping tools, the UV editor, and the controls within the editor window.

Section 9 – Animation

Objectives

This section delves into the details of animating meshes for the purpose of character animation. This includes how to make them move and deform according to an animator’s instructions, how to store the animations and replay them efficiently, and how to blend multiple animations together. The primary focus will be on bone-based skeletal animation systems, since they are by far the most widely used systems today, used by most tools and runtime systems.

The process of creating animated 3D art for real-time games is also described. Animation for these games is the art of capturing a series of individual movements in digital form and then playing them back in real time. Just about everything in a game – from the user interface, to atmospheric effects, to characters and walking critters – will need to be animated. Even the camera may need to be animated through a 3D environment in a predictable and controlled manner. Animation establishes the character and personality of humanoid figures and both real and imagined creatures.

Expected learning outcomes

After the completion of this section, the students are expected to:

- Explain the Skeletal Hierarchy of an animated character.
- Define the term “Rig” in the organization of bones.
- Describe the differences between “Animation” and “Deformation”.
- Explain how “animation storage” is handled to save storage space.
- Explain Keyframes and Linear Interpolation animation.
- Explain the process of Motion Extraction and Mesh Deformation.
- Describe and compare Inverse Kinematics in respect to Forward kinematics.
- Describe the process of creating animated 3D art for real-time games, i.e. capturing a series of individual movements in digital form and then playing them back in real time.
- Describe important considerations for Facial Animation.

Keywords

Animation storage, Blending Animation, Motion Extraction, Mesh Deformation, Inverse Kinematics, Facial Animation, Motion Capture.

Bibliography

Edited by Steve Rabin, “Introduction to Game Development”, Second Edition, Course Technology, 2010 – Sections: 5.2, 6.7.

Activities

- Exercises at the end of each section in the textbook.
- Project – Continue on Game development: Add animation.

Summary

This chapter introduced the basic components of most animation systems. Animations are encoded using a variety of compression methods, primarily to reduce the memory footprint, but also to support playing and sampling them in real time, while blending them together in various ways. The overall motion of the animation is transferred from the animation system to the game's instance, so that the game logic can keep track of the animated figure. Inverse Kinematics (IK) corrects the animation to ensure that contact with other objects in the world is maintained even when blending and distorting the original animations. The pose is then transformed to world space, and finally the mesh is deformed using the animated bone positions and rendered on the screen.

These principles are shared by almost all animation systems. However, a real game's animation system will have many higher level systems layered upon these principles that are specific to the game, which is a juggling act between the needs of the animators and the needs of the game design. This demands flexibility and can result in many different methods for performing the same operation. As long as these are constructed using components of a single, shared, low-level animation system, the complexity of the code can be kept manageable, even when blending multiple techniques together.

The scope of work that an animator is expected to contribute on a project varies extensively. For character work and animation specifically, this section discussed the many facets of the production pipeline. It involves planning, modeling, texturing, rigging with vertex weighting, animating both body and face with keyframes, motion capture, or physics simulation. As the scope of design and the complexity of environments continue to increase for interactive games, the volume and variety of virtual inhabitants will also continue to increase. Learning the recommended workflow steps and approaches to setting up and animating this growing cast of digital actors will help secure more ongoing gratifying work for the art and animation staff.

Section 10 – Artificial Intelligence (AI)

Objectives

In many video games, the quality of the experience depends on whether the game presents a good challenge to the player. One way to present a good challenge is to offer computer opponents, or sometimes even allies, that are capable of playing the game intelligently. In most cases, this is not a trivial problem to solve, but fortunately, there is an entire field of study that can help us out – artificial intelligence (or AI for short). AI describes the intelligence embodied in any manufactured device. If we design a character or opponent in a video game that acts on its own, it is generally accredited with possessing AI.

Human-level AI is the stuff of dreams and science fiction. How do you take the accumulated common sense and expertise of a human and distill it into a computer? Unfortunately, this problem is currently unsolved, and it will likely be decades before we get close to understanding what it truly entails. Since general human-level intelligence is currently impossible to re-create, researchers chip away from dozens of different angles by solving much simpler problems. By sufficiently narrowing down the domain of an AI problem, it becomes possible to create behavior that is reasonable and believable, especially in the realm of video games.

This section first discusses the unique properties of game AI and how it differs from other AI fields. With believable characters being the centerpiece of most game AI, the concept of a game agent is introduced. Game agents perceive the world, react in intelligent ways, and potentially adapt to the player. As the most widely used architecture for game AI, various flavors of finite-state machines are then examined and compared. Intelligent movement for game agents is covered in-depth. The section also presents a survey of the most common and promising techniques in game AI today.

Expected learning outcomes

After the completion of this section, the students are expected to:

- Describe to what extent AI is possible in computer games today and what does the future foresees?
- Describe the role of the Game Agents, and the key steps through which they continually loop, which is known as “sense-think-act cycle”.
- Define and explain the term Finite-state-machine (FSM).
- Describe and implement current AI techniques, and discuss on future promising AI techniques.
- Explain the “Pathfinding” problem.
- Describe the terms “Breath-First”, and “Best-First”.

Keywords

Game Agents, Sense-think-act cycle, Finite-state-machines, pathfinding, Breath-First, Best-First.

Bibliography

Edited by Steve Rabin, “Introduction to Game Development”, Second Edition, Course Technology, 2010 – Sections: 5.3, 5.4.

Activities

- Exercises at the end of each section in the textbook.
- Project – Continue on Game development: Consider inclusion of AI.

Summary

Game AI is distinctively different from many other related AI fields. The goal is to create intelligent opponents, allies, and neutral characters that result in an engaging and enjoyable experience for the player. One of the most enduring techniques for endowing intelligence on agents is the ubiquitous finite-state machine. This simple computational model allows complex expertise to be expressed in a simple, easy-to-understand manner that is also convenient to debug. The actions and mindsets of an agent eloquently map to the states of an FSM, further allowing for simple, yet effective modeling of behavior. With the many enhancements developed for FSMs, it is easy to understand why they have become so universal within AI game development.

Finally, there are dozens of common and promising techniques for adding intelligence to games. Each game is unique and might require mixing and matching several different techniques. There is no single solution, and the resulting design is highly dependent on the exact requirements of the game. Therefore, it is critical that a developer becomes familiar with a broad range of techniques in order to experiment and make intelligent implementation decisions.

Section 11 – Audio and Networking

Objectives

The role of the audio programmer has become increasingly important as games have evolved to feature more complex sound and musical components. Rather than just supporting real-time playback of audio content, the audio programmer must also support the creation and integration of sound within the game engine. In some sense, programming success can be measured by the degree to which an audio designer can integrate audio without the direct need for programmer support. In most situations, audio events are triggered by in-game events: a character plays a specific animation, a weapon fires, an explosion occurs, and so forth. Others are triggered by locality, such as ambient effects. Still others may be launched via scripts, such as dialogue in a cut-scene.

As with visual rendering technology, audio programming has moved past the simple basics of vanilla sound mixing and playback. However, every audio system should be built on top of a fundamental understanding of these basic principles and systems. On most modern gaming platforms, the capabilities of mixing and rendering audio data on hardware is a given. Moreover, these platforms typically have a reasonably robust API to program these capabilities. It is rare for an audio programmer to have to write a low-level mixer or filter in software. Instead, this section will focus on midlevel programming; that is, how one should make use of existing APIs and hardware to create an audio engine.

It also introduces the concepts and terminology involved with network and multiplayer programming. It begins with an assessment of multiplayer game modes, followed by an exploration of network programming fundamentals, including network protocols, real-time data transfer, asynchronous environment guidelines, and game security.

Expected learning outcomes

After the completion of this section, the students are expected to:

- Discuss on the role of the audio programmer.
- Describe the basic physics of audio.
- Implement software programs with audio.
- Describe the audio pipeline and mixing features.
- Add playback and manipulation into programs.
- Describe the storage of audio – compress storage format.
- Explain the term “3D Audio”
- Integrate advance 3D Audio Environmental Effects.
- Describe the different terms for Multiplayer Modes.
- Describe the Protocols for transferring data between devises.
- Describe the different Network Layers.

Keywords

Audio programmer, Basic Audio, Music Systems, Advance Audio, Audio Pipeline, Mixing Features, 3D Audio, Audio Scripting, Multiplayer, Protocols, Network, Real-Time communication, Security.

Bibliography

Edited by Steve Rabin, “Introduction to Game Development”, Second Edition, Course Technology, 2010 – Sections: 5.5, 5.6.

Activities

- Exercises at the end of each section in the textbook.
- Quiz.
- Project – Continue on Game development: Include sound.

Summary

Audio programming from the perspective of a game developer is certainly no longer about fine-tuning some low-level audio mixing routines. Modern hardware and rendering libraries now take care of the basics. Instead, the lowest-level job will typically be integration of a complete audio scripting system into a game engine. The job of an audio programmer is to give as much power and control as possible to the audio content creators, while minimizing the time any programmer must think about low-level programming tasks. That leaves a world of much more advanced and exciting audio-related research topics to investigate and implement in tomorrow’s games.

Additionally, this section presents a survey of multiplayer development by looking at different multiplayer categories, from split-screen to real-time network connections. The core material focused on dissecting the OSI layers in an average network game. The OSI layers contained mediums, IP, TCP and UDP, sockets, packet presentation, and game logic related to controlling latency. Next, the real-time communication models of broadcasting, peer-to-peer, and client/server were analyzed for strengths and weaknesses, followed by tips for working in multithreaded environments. The chapter concluded with a glimpse into game security.

Section 12 – Game Industry and Legal Issues

Objectives

Delivering a big-budget game into a consumer’s hands is an increasingly complex, lengthy and costly process. Game industry veterans constantly debate the relative importance of developers as creative auteurs versus publishers as soulless businesspeople, or retailers as channel arbiters versus media as opinion-mongers. However, each “driver” entity on the highway to the consumer (developer, publisher, platform owner, retailer), is essential to the transaction, as the industry’s economic structure demonstrates.

“Adjunct” entities that feed into the channel also offer a plethora of service alternatives that reduce cost, save time, or improve quality. In this section, we examine the economics and roles of 11 entities that collaborate to bring a game to retail stores.

Expected learning outcomes

After the completion of this section, the students are expected to:

- Describe the role of Game Developers and Game Publishers.
- Describe the role of the publishers who are legally liable for the game’s quality to both consumers and the platform holder.
- Describe what the “Platform Holders” are.
- Describe Quality Assurance (QA) procedures.
- Describe Game Promotion procedures. What is the role of Public-Relations Firms, Advertising Agencies, and Merchandising Teams?
- Explain the role of Marketing in promoting and selling a game.

Keywords

Game Developers, Game Publishers, Platform Holders, Debugging, Testing, Quality Assurance, Marketing, Payment Negotiations, Research.

Bibliography

Edited by Steve Rabin, “Introduction to Game Development”, Second Edition, Course Technology, 2010 – Sections: 7.2 - 7.6.

Activities

- Exercises at the end of each section in the textbook.
- Quiz.
- Project – Final Result.

Summary

The video games industry is now in its fourth decade of providing interactive entertainment to the consumer market. Through the years, although industry entities have largely retained their roles in the channel, the balance of power (and flow of money) among them has fluctuated widely. Given the amount of money in play, a major stumble by any part of the value chain – or a paradigm shift in technology or business model – could turn the entire industry’s balance of power on its head.



Course Code	Course Title	ECTS Credits
COMP-528DL	Mobile Platforms and Software Development	10
Department	Semester	Prerequisites
Computer Science	Fall, Spring	COMP-525DL Mobile Computing
Type of Course	Field	Language of Instruction
Required	Computer Science	English
Level of Course	Year of Study	Lecturer(s)
2nd Cycle	2 nd	Dr. Constandinos Mavromoustakis
Mode of Delivery	Work Placement	Co-requisites
Distance Learning	N/A	None

Objectives of the Course:

The main objectives of the course are to:

- explore the basic Mobile Platforms and related Technologies (Wearable and Sensing Computing etc.) and explore the basic concepts of the development process for mobile applications hosted on different mobile devices
- thoroughly discuss and explore the basic concepts and develop skills in mobile application development, programming, graphic interface design and interaction design

- explore the basic concepts of mobile software development through fundamental programming principles with a focus on the mobile environment and the supported platforms
- design and implement applications on a mobile device using the model-view-controller design pattern and make students aware of the mobile software development using native and non-native Object Oriented libraries of the supported platforms
- design and develop applications using specifications by utilizing local and global abstraction features of the mobile platform (e.g. mobile devices or sensors) and quantify and design the effects of different software design decisions on mobile device restrictions and the impact of Cloud-oriented process offloading
- design and implement reusable user interface elements, that meet or exceed the design criteria of existing industry standard user interface libraries
- critically assess the limitations throughout developing mobile applications and applying software optimization methodologies for a certain case-application
- build, test, and deploy mobile solutions using appropriate technologies and collaborate in a group-based mobile development project

Learning Outcomes:

After completion of the course students are expected to be able to:

1. provide students with deep knowledge of the different types of application models/architectures used to develop mobile software applications
2. acquire deep knowledge of the components and structure of a mobile development frameworks and design and develop reusable user interface elements on a mobile device using the model-view-controller design pattern
3. design and develop applications with special features (local and global dependencies i.e. location-aware diversities, Cloud-oriented platform-dependent processes i.e. process offloading etc.) and apply the different types of application models/architectures used to develop native and non-native Object Oriented libraries

4. design and develop device-specific, native software and middleware mechanisms using specifications by utilizing local and global abstraction features of the mobile platform
5. research and apply various Software Development Kits (SDKs), frameworks and toolkits supporting specified mobile platforms
6. collaborate in a group for the development of Mobile Application project

Course Contents:

1. Introduction to mobile programming, best practices and the mobile standards W3C Mobile group and the Open Mobile Alliance/ Open Handset Alliance.
2. Mobile platforms, their properties and execution environments (AppFurnace, Application Craft, appMobi, iOS, Android, HTML5 Web Apps, HTML5 Hybrid Apps, BlackBerry Dev., Convertigo mob., etc.).
3. Cross-Platform Development Tools and Native Development Tools (Android SDK and Eclipse Android Development Tools (ADT)/C# environment and needed components Objective-C language syntax).
4. Language syntax and memory management, Static and dynamic code analysis/Native and non-native/custom classes/Designing for code re-use. Limitations and requirements.
5. Different Mobile application life-cycles and source code management (Git/SDK). Compatibility issues.
6. Building Cross Platform Applications: best practices for developing mobile applications with Xamarin-existing Layouts to develop essential process-oriented widgets.
7. Manipulating software design principles: mobile apps design with the HTML 5 container, Java and other compatible language features.
8. Next generation services: REST Web Services, AsyncTask, HttpURLConnection, Processing JSON Responses, JSONObject and JSONArray, platforms/open source and open-module cross-platforms paradigms/Trade-off between paradigms for different scenarios using the existing platforms.
9. Social apps and orientation: Data Stores, Network Services, and supportive cross platform APIs, Managing and Accessing Local and Remote Databases, Leveraging Java and C# Libraries.

10. Model-View-Controller design patterns, Handling Activity Lifecycle Events, Handling user generated actions and passive ones. Threads' handling and Rooting.
11. Web service integration and Services according to user-interface and interaction patterns/Alerting users via notifications. Requesting and Requiring Permissions, Accessing Location-Based Services Mapping with MapView and MapActivity, Handling interruptions (Telephone Calls) and parallel external actions.
12. Hardware module interactions and power diversities as a first-order design constraint/ Development of integrated *Mobile/Tablet/Sensor* project.

Learning Activities and Teaching Methods:

Online Lectures, Virtual Lab Presentations, Theoretical Exercises and Assignments.

** Every week a Virtual Laboratory Session will take place where student would be expected to carry out hands-on practice and gain experience.*

Assessment Methods:

Final Exam	60%
Projects (3 Programming Individual and 2 Programming Group)	25%
Assignments (2), Quizzes (3)	15%

Required Textbooks/Reading:

Authors	Title	Publisher	Year	ISBN
Paul Deitel, Harvey Deitel, Alexander Wald	Android™ 6 for programmers an app-driven approach	Addison-Wesley Professional	2016	ISBN-13: 978-0-13-428936-6, ISBN-10: 0-13-428936-6
John M. Wargo	PhoneGap Essentials: Building Cross-Platform Mobile Applications	Addison-Wesley Professional	2015	ISBN-10: 0321814290 •

				ISBN-13: 9780321814296
--	--	--	--	---------------------------

Moreover, students will be provided with a prepared material/notes by the lecturer on specific topics covered by the course contents.

Recommended Textbooks/Reading:

Authors	Title	Publisher	Year	ISBN
Crestani, Fabio, Mizzaro, Stefano, Scagnetto, Ivan	Mobile Information Retrieval	Springer International Publishing AG	2017	ISBN: 978-3-319-60777-1
Abhishek Mishra	Amazon Web Services for Mobile Developers: Building Apps with AWS	Wiley	2017	ISBN: 978-1-119-37785-6
Rob Napier, Mugunth Kumar	iOS 7 Programming Pushing the Limits: Develop Advance Applications for Apple iPhone, iPad, and iPod Touch	Wiley	2017	ISBN: 978-1-118-81834-3

Note: Published articles from leading Journals and Conference Proceedings will also be considered as a part of the required reading material.

Weekly Schedule:

Please see the course/module Study Guide.



UNIVERSITY *of* NICOSIA

SCHOOL OF SCIENCES AND ENGINEERING

MASTER OF SCIENCE IN COMPUTER SCIENCE

Study Guide

COMP-528 Mobile Platforms and Software Development

Dr. Constandinos X. Mavromoustakis

Nicosia

2017

Introductory note

This Study Guide is a basic supplement for the distance learning course “COMP-528 Mobile Platforms and Software Development”, which is offered by the distance learning MSc in Computer Science programme. The aim of this Guide is to direct the students and help them into making systematic use of the educational material on which the teaching of the course is based.

The Guide must be used in parallel with the Course Outline and with the educational material (course textbooks, extensive notes of the lecturer, powerpoint presentations, articles and book chapters indicated for each lecture, instructions for the composition of essays and for references etc.) which has been uploaded to the interactive internet-based platform of the course (Moodle). The student must start his/her study by using the extended notes of the lecturer, which correspond to the powerpoint presentation of each lecture, and subsequently he/she takes advice from the present Guide, which aims to extend his/her knowledge by making use of the rest of the educational material of each lecture.

The Guide includes 12 sections. Each one of these sections is composed of the following parts: objectives, expected learning outcomes, key words, bibliography, additional bibliography, and activities. Conducting the latter is essential, since you will be helped into absorbing the material of the courses as well as into developing critical thought on them. The activities and the proposed bibliography will be enriched during the semester.

Dr. Constandinos X. Mavromoustakis, Professor of Computer Science, School of Sciences and Engineering

Section 1. Introduction to Mobile Programming, review of best practices and the mobile standards

Objectives

The aim of this section is to introduce the students into the Mobile Computing Environments in contrast to the different development tools and programming technologies and the required infrastructures that are involved in the development process. The section presents the mobile programming issues and existing popular Development Environments as well as an overview of Mobile Devices and Developing Mobile Applications. The latter embraces different tools and add-ons to existing frameworks aiming to show the different diversities and characteristics of the devices and their capabilities (e.g. Mobile phones, Wearables, Sensor devices etc. with different Operating Systems).

Expected learning outcomes

After completion of the section students are expected to be able to:

- have a good realization of how a Mobile Computing Device works and the existing Mobile platforms and their potential in different domains.
- provide students with deep knowledge for installing, using and developing applications based on the usability principles and on modern technologies.
- provide a good realization and clear identification of the popular Development Environments as well as an overview of Mobile Devices and Developing Mobile Applications.
- introduce state-of-the art tools and the existing add-ons (assistive software/middleware) of modern frameworks aiming to show the different diversities and characteristics of the devices and their capabilities.

Keywords

Popular Integrated Development Environments (P-IDE), Mobile Programming, Android Development Tools (ADT) plugin for Eclipse, Setup Development Environment, Mobile platforms, mobile standards W3C Mobile group and the Open Mobile Alliance/ Open Handset Alliance.

Required Textbooks

Hermes D., Xamarin Mobile Application Development: Cross-Platform C# and Xamarin, Publisher: Apress, 2015, ISBN-10: 1484202155 ISBN-13: 978-1484202159 [Chapter 2 and Chapter 3].

Additional Textbooks/Reading:

Access the <http://developer.android.com/sdk/installing.html> for installing Android or PhoneGap at <http://phonegap.com/> of Adobe, for a Cross-platform Environment.

Learning Activities

1. Be able to make a distinction between a device with many 'capabilities' and a device with limited 'capabilities' and think on how devices may be optimized with respect to the mobile device's computational and physical elements.
2. What are the basic differences between the existing Mobile Platforms and how the Android Open Source Project (AOSP) and the Open Handset Alliance are standing in terms of simplicity and performance?
3. Examples for the existing Mobile platforms and their capabilities in developing modern applications.
4. Look at many of the tools provided with the Android SDK and Review the Android SDK packages. Experiment with the sample applications available with the SDKs and PhoneGap APIs and or online platforms.

Summary

The trend today is to be able to program and customize with flexibility any handheld and wireless device. This section assist you by providing you taxonomy of different platforms in contrast to the design and implementation of applications for mobile devices. Developing an app from scratch, assumes a basic knowledge of an Object Oriented Language (such as Java), and this section guides you on how to select a platform for your 'hands', tailored for your knowledge, based on different platforms and Operating Systems. The basic platforms for Mobile Programming will be analyzed with their specific architectures and their pros and cons for developing today's Mobile Applications.

Section 2. Mobile platforms, their properties and execution environments

Objectives

This section aims at providing to the students a comprehensive demonstration of the Mobile platforms, their properties and the different diversified characteristics that these platforms expose in their respective execution environments. It navigates through the different environments including the PhoneGap, Appcelerator, Xamarin, Sencha Touch, Monocross, Codename One, Kony Mobile Platform, Nativescript as well as the appMobi, iOS, Android, HTML5 Web Apps, HTML5 Hybrid Apps, BlackBerry Dev., Convertigo mob., etc. and presents their APIs and communication interfaces.

Expected learning outcomes

After completion of the course students are expected to be able to:

- provide a good realization and clear identification of the Write Once Run Anywhere (WORA) which refers to a piece of code that's written once on one platform but can run on multiple platforms.
- illustrate the cross-platform benefits of the Java language and the way that a code can be developed on any device, compiled into a standard bytecode and be expected to run on any device equipped with a virtual machine.
- demonstrate the variability in today's mobile platforms supporting WORA.
- introduce and provide students with deep knowledge for the Top Cross-Platform Mobile Development Tools and their supporting Application Programming Interfaces (APIs).

Keywords

Mobile platforms, Application Programming Interfaces, WORA, Mobile platforms, PhoneGap, multi-platform apps with code reuse.

Required Textbooks/Reading

Panhale M., Beginning Hybrid Mobile Application Development, DOI 10.1007/978-1-4842-1314-8 Print ISBN 978-1-4842-1315-5, 2014, Publisher: Apress [Chapters 1-2].

Additional Textbooks/Reading:

Ghatol R. and Patel Y, Beginning PhoneGap, Mobile Web Framework for JavaScript and HTML5, ISBN: 978-1-4302-3903-1 (Print) 978-1-4302-3904-8, [Chapters 1-2].

Ghatol R. and Patel Y., Understanding Cross-Platform Mobile Application Development, Book Title: Beginning PhoneGap, Book Subtitle: Mobile Web Framework for JavaScript and HTML5, Pages pp 1-16, 2012, DOI 10.1007/978-1-4302-3904-8_1, Print ISBN 978-1-4302-3903-1, Apress.

Learning Activities

1. Install -after reviewing the top 3 Mobile platforms- the tools that better fits to your knowledge for developing cross platform mobile applications.
2. Demonstrate the variability in today's mobile platforms supporting WORA with substantiated responses towards the demands of today's mobile ecosystem.
3. Choose and implement the best Android and iOS testing tools and establish your optimal mobile test and launch strategy and create a test app.
4. Define the Mobile Release Cycles in contrast to a developed application for testing.

Summary

As variety of mobile platforms increase, developers are finding it harder to provide their app in consumable form as users may be on very different platforms. This section provides with clear demonstrations the students with comprehensive taxonomy of the Mobile platforms, their properties in their respective execution environments. It navigates through the different environments and tests 'a first program' whereas, it demonstrates the top existing platforms that have already penetrated in the mobile industry including the PhoneGap, Appcelerator, Xamarin, Sencha Touch, Kony Mobile Platform etc. with their supported interfaces.

Section 3. Cross-Platform Development Tools and Native Development Tools: Cordova, PhoneGap & Ionic

Objectives

This section aims at providing to the students a comprehensive demonstration and the essential knowledge of Cordova, PhoneGap and Ionic. The different characteristics that Cordova enables the programmer to write code only once and deploy the app on just about any platform out there while still using the same code base. It aims to show to students the pathways to create projects, preview them in the platform we selected, and easily deploy our apps. In addition, this section allows students to gain experience with code by writing native code for each platform experiencing the advantages. The ability to re-use the source code and deploy it on multiple devices and stores can reduce the development effort and resources for a project.

Expected learning outcomes

After completion of the course students are expected to be able to:

- provide a good realization and clear identification of the requirements of the different platforms (Cordova, PhoneGap and Ionic).
- gain experience with real-code cross-platform writing and realize the way that a code can be developed on any device, compiled into a standard bytecode and be expected to run on any device.
- Realize and use the Basic Device Information (Device API): cordova-plugin-device and the Network Connection Events: cordova-plugin-network-info.
- presents the iOS Simulator in Cordova, PhoneGap and Ionic and develop a project on iOS, by installing the “ios-x” project.
- configure WebStorm IDE with the PhoneGap project and configure over this an Ionic project.

Keywords

Cordova, PhoneGap, supported platforms, WebStorm IDE, configuration, Ionic project.

Required Textbooks/Reading

Elrom E., Pro MEAN Stack Development, ISBN: 978-1-4842-2043-6, 2016, Publisher: Apress [Chapter 4].

Additional Textbooks/Reading:

Ghatol R. and Patel Y., Understanding Cross-Platform Mobile Application Development, Book Title: Beginning PhoneGap, Book Subtitle: Mobile Web Framework for JavaScript and HTML5, Pages pp 1-16, 2012, DOI 10.1007/978-1-4302-3904-8_1, Print ISBN 978-1-4302-3903-1, Apress.

Ghatol R. and Patel Y., Beginning PhoneGap, Mobile Web Framework for JavaScript and HTML5, ISBN: 978-1-4302-3903-1 (Print) 978-1-4302-3904-8, [Chapters 1-2].

Learning Activities

1. Install-writing native code for Cordova, PhoneGap & Ionic.
2. Experiment if the written code can run on any other terminal and extract screenshots for the extracted outcomes.
3. Create projects, preview them in the platform, and deploy sample apps.
4. Gain experience in installing native plugins and learn about the distribution process.
Apply a real-time solution to a specific problem by developing a source-code for an app.

Summary

Writing native code for each platform has its own advantages. In this section, students are expected to gain knowledge and experience Apache Cordova, created a Cordova “First_App” sample app, and experience how to run this app. It presents the different platform deployments that Cordova supports and also it demonstrates the different Cordova plugins available and where to locate them. The section covered several of the Cordova distributions, such as PhoneGap and Ionic as well as presents the installed PhoneGap and the PhoneGap Mobile App engine and created a PhoneGap sample project. Knowledge for the configured WebStorm IDE with the PhoneGap project is gained, while configuration of an Ionic project, preview of some of the features and commands that can help students to use Ionic to its fullest potential, and deploying their app in the browser and on iPhone, are of the main presented components.

Section 4. PhoneGap Platform and memory management, Static and dynamic code analysis/Native and non-native/custom classes/Design for code re-use. Limitations and requirements

Objectives

The aim of this section is to provide the knowledge for configuring the PhoneGap environment to be setup in the following two manners: Local development environment on a machine; Cloud build environment on PhoneGap Build. The local development environment setup includes the developer setting up environments for each mobile platform that the developer wants to launch a PhoneGap application on. This section covers the local environment setup in detail and hopes that the audience won't require any other documentation to run a PhoneGap application on each of the platform emulators.

Expected learning outcomes

After completion of the course students are expected to be able to:

- provide a good practice for real-code cross-platform writing and realize the interpretation and compilation of native machines by a standard bytecode and/or HTML 5.0+ enabled scripts.
- realize and use the Architecture for Device Information using native Device APIs and plugins.
- configure the PhoneGap environment to be setup in the following two manners: Local development environment on a machine and onto a Cloud build environment on PhoneGap Build in contrast to the role of the memory.
- extract the developer certificate from Mac's keychain access by opening a keychain access and located the developer certificate and export it.

Keywords

PhoneGap environment, Local, Cloud build environment, memory manipulation.

Bibliography

Ghatol R. and Patel Y., Understanding Cross-Platform Mobile Application Development, Book Title: Beginning PhoneGap, Book Subtitle: Mobile Web Framework for JavaScript and HTML5, Pages pp 1-16, 2012, DOI 10.1007/978-1-4302-3904-8_1, Print ISBN 978-1-4302-3903-1, Apress [Chapter 3].

Elad Elrom, Pro MEAN Stack Development, ISBN: 978-1-4842-2043-6, 2016, Publisher: Apress [Chapter 4].

Murphy L. M., The Busy Coder's Guide to Android Development Version 7.4, Paperback: 3893 pages, Publisher: CommonsWare, LLC (June 13, 2016), ISBN-10: 0981678009, ISBN-13: 978-0981678009. [pages included 467-503]

Additional Textbooks/Reading:

Ghatol R. and Patel Y, Beginning PhoneGap, Mobile Web Framework for JavaScript and HTML5, ISBN: 978-1-4302-3903-1 (Print) 978-1-4302-3904-8, [Chapters 1-2].

Learning Activities

1. Write native code in PhoneGap Build and access your source code, as it's a build tool. This means you will need to share your source code with PhoneGap Build. Provide these steps in a documentation.
2. experiment if the written code can run on any other terminal and extract screenshots for the extracted outcomes.
3. Set-up iOS Build Environment and apply all the prerequisites for iOS build on PhoneGap. Be able to provide iOS Keys to PhoneGap Build.

Summary

PhoneGap Build requires access to your source code, as it's a build tool. This means you will need to share your source code with PhoneGap Build. This is like we want to build PhoneGap sample code provided by PhoneGap in one of their GIT-hub repositories. This section provides a complete guidance on developing applications with special features (local and global dependencies) and help students to apply the different types of application models/architectures used to develop native and non-native Object Oriented libraries.

Section 5. Different Mobile application life-cycles and source code management (Git/SDK)

Objectives

This section aims at providing to the students a comprehensive demonstration and the essential knowledge of Different Mobile application life-cycles including Mobile Development (Software Development Lifecycle-SDLC, lifecycle of mobile development for web or desktop applications). These are figured in the 5 major portions of the Mobile application life-cycle: Inception of an idea; Design; Development; Stabilization and Deployment. The different characteristics during this process enables the programmer to write code, assess the life-cycle throughout the run-out of the program and gain experience with real native code.

Expected learning outcomes

After completion of the course students are expected to be able to:

- introduce effective Software Development Lifecycle (SDLC) with specific examples.
- examine all phases of the SDLC with respect to mobile application development, including: Inspiration, Design, Development, Stabilization, Deployment, and Maintenance.
- design Mobile Applications once the features and functionality of the app have been determined, and start trying to solve the User Experience or UX issues.
- create UX Mockups to consider the Interface Guidelines for the various platforms that the app will target following official design guidelines for each platform such as: Apple-Human Interface; Android-Design Guidelines; Windows Phone-Design library for Windows Phone.

Keywords

Mobile Applications features, Mobile user requirements, UX issues, Software Development Lifecycle (SDLC), UX Mockups, Interfaces for UXs.

Bibliography

Ghatol R. and Patel Y., Understanding Cross-Platform Mobile Application Development, Book Title: Beginning PhoneGap, Book Subtitle: Mobile Web Framework for JavaScript and HTML5,

Pages pp 1-16, 2012, DOI 10.1007/978-1-4302-3904-8_1, Print ISBN 978-1-4302-3903-1, Apress [Chapter 3].

Elad Elrom, Pro MEAN Stack Development, ISBN: 978-1-4842-2043-6, 2016, Publisher: Apress [Chapter 4].

Additional Textbooks/Reading:

Ghatol R. and Patel Y., Beginning PhoneGap, Mobile Web Framework for JavaScript and HTML5, ISBN: 978-1-4302-3903-1 (Print) 978-1-4302-3904-8, [Chapters 1-2].

Learning Activities

1. Assess specific source code paradigms with the respective output the effective Software Development Lifecycle (SDLC) with specific examples.
2. Provide the diversities of each one of the phases of the SDLC with respect to mobile application development for: Stabilization, Deployment, and Maintenance and experiment if the written code can run on any other terminal and extract screenshots for the extracted outcomes.
3. Create UX Mockups to consider the Interface restrictions and guidelines for the various platforms that the specific/provided app will target.

Summary

While developing mobile applications isn't fundamentally different than traditional web/desktop development in terms of process or architecture, there are some considerations to be aware of. There are significant challenges to multitasking (having multiple applications running at once) on a mobile device that should be faced with a precise consideration of the Software Development Lifecycle (SDLC). The section presents and provides guidance with examples on how to critically extract the diversities of each one of the phases of the SDLC with respect to mobile application development for allowing Stabilization, Deployment, and Maintenance whereas the overall UX is assessed for different platforms.

Section 6. Building Cross Platform Applications: practices for developing mobile applications with advanced features

Objectives

The aim of this section is to provide a solid experience in creating Android Virtual Devices (AVDs), detect through source code when the user touches the screen, moves a finger across the screen and removes a finger from the screen or for example would like to use the camera. Advance features will be included like processing multiple touches so the user can draw with multiple fingers at once, use a SensorManager and the accelerometer to detect motion events, use a Paint object to specify the color and width of a line, use Path objects to store each line's data and use a Canvas to draw each line into a Bitmap and create a menu and display menu items on the app bar. The section also explains how to use the printing framework and the Android Support Library's PrintHelper class to enable the user to print a drawing. Students will be provided with solid knowledge on how to use Android 6.0's new permissions model to request permission for saving an image to external storage and how to add libraries to an app with the Gradle build system. Students will also gain experience with the Android SDK's Android emulator which will allow the programmers to test apps on your computer rather than on an Android device-no need to have any Android devices.

Expected learning outcomes

After completion of the course students are expected to be able to:

- create and develop Android Virtual Device (AVD) to detect through source code action events like when the user touches the screen and/or initiates an action etc.
- apply source code with advance features like processing multiple touches so the user, use a SensorManager and accelerometer to detect motion events etc.
- process Multiple Touch Events and Storing Lines in Paths as well as saving to the Device, Printing and the Android Support Library's PrintHelper Class, New Android 6.0 (Marshmallow) and Nugat (7.0) Permissions Model and Adding Dependencies Using the Gradle Build SystemDoodleView.

Keywords

Android Virtual Devices, Multiple Touch Events, Paths, Library's PrintHelper Class, New Android 6.0 (Marshmallow) and Nugat (7.0) Permissions Model, Dependencies.

Bibliography

Deitel P., Deitel H. and Wald A., Android™ 6 for programmers an app-driven approach 3rd Edition ISBN-13: 978-0-13-428936-6, ISBN-10: 0-13-428936-6, 2016, pp. 400. [Chapter 5]

Ghatol R. and Patel Y., Understanding Cross-Platform Mobile Application Development, Book Title: Beginning PhoneGap, Book Subtitle: Mobile Web Framework for JavaScript and HTML5, Pages pp 1-16, 2012, DOI 10.1007/978-1-4302-3904-8_1, Print ISBN 978-1-4302-3903-1, Apress [Chapter 3].

Elad Elrom, Pro MEAN Stack Development, ISBN: 978-1-4842-2043-6, 2016, Publisher: Apress [Chapter 7].

Additional Textbooks/Reading:

Android Reference: "Service": <http://d.android.com/reference/android/app/Service.html>

Android API Guides: "Services": <http://d.android.com/guide/components/services.html>

Android API Guides: "Bound Services": <http://d.android.com/guide/components/bound-services.html>

Android API Guides: "Android Interface Definition Language (AIDL)":

<http://d.android.com/guide/components/aidl.html>

Android API Guides: "Processes and Threads":

<http://d.android.com/guide/components/processes-and-threads.html>

Android API Guides: "Android Application Framework FAQ":

<http://d.android.com/guide/faq/framework.html>

Learning Activities

1. Detect when the user touches the screen, moves a finger across the screen and removes a finger from the screen.
2. Process multiple touches so the user can draw with multiple fingers at once.
3. Use a SensorManager and the accelerometer to detect motion events.
4. Use a Paint object to specify the color and width of a line and use Path objects to store

each line's data and use a Canvas to draw each line into a Bitmap.

5. Create a menu and display menu items on the app bar as well as use the printing framework and the Android Support.
6. Library's PrintHelper class to enable the user to print a drawing.
7. Use Android 6.0's new permissions model to request permission for saving an image to external storage and add libraries to an app with the Gradle build system.

Summary

When a programmer or a user is allowed to use a feature and how this is done is a subject covered in this section. With several apps as examples this section introduces Android 6.0's new permissions mechanism. For example, Android requires the user's permission to allow an app to save files/data on a device. In Android 6.0, rather than prompting the user at installation time with a complete list of permissions the app requires, the app requests each permission individually, only when the permission is required to perform a given task for the first time. All the features are being processed and are "allowed" by the low-level system mechanisms that this section presents. Finally, it demonstrates the Activity and Fragment Lifecycle Methods based in parent activity features.

Section 7. Manipulating software design principles: mobile apps design with the HTML 5 container, Java and other compatible language features

Objectives

The aim of this section is to provide a solid experience in creating Manual Frame-By-Frame Animation, Graphics, Sound, Threading, SurfaceView and SurfaceHolder, Immersive Mode and screen features. In this section students, will create a simple game app that's easy to code and fun to play by creating a custom SurfaceView subclass for displaying the game's graphics from a separate thread of execution and by drawing graphics using Paints and a Canvas. The section demonstrates how to override View's onTouchEvent method to fire a cannonball when the user touches the screen and student may add sound to an app using a SoundPool and addons like an AudioManager. In addition, the section presents the immersive mode to enable the game to occupy the entire screen, but still allow the user to access the system bars.

Expected learning outcomes

After completion of the course students are expected to be able to:

- introduce the manual Frame-By-Frame Animation, Graphics, Sound, Threading, SurfaceView and SurfaceHolder, Immersive Mode and Full-Screen.
- introduce and apply using real-time source code writing on building the GUI and the associated resource files Blocker Subclass of GameElement as well as the Target Subclass of GameElement, MainActivityFragment Subclass of Fragment, and Adjusting the Theme to Remove the App Title and App Bar.
- introduce and apply using real-time source code writing on the Overriding View Method onSizeChanged under action confirmed (passive and/or active) orders by overriding View Method onTouchEvent.

Keywords

Manual Frame-By-Frame Animation, Graphics, Sound, Threading, SurfaceView and SurfaceHolder, Immersive Mode, Full-Screen, Overriding View Method.

Bibliography

Deitel P., Deitel H. and Wald A., Android™ 6 for programmers an app-driven approach 3rd Edition ISBN-13: 978-0-13-428936-6, ISBN-10: 0-13-428936-6, 2016, pp. 400. [Chapter 6]

Murphy L. M., The Busy Coder's Guide to Android Development Version 7.4, Paperback:3893 pages, Publisher: CommonsWare, LLC (June 13, 2016), ISBN-10: 0981678009, ISBN-13: 978-0981678009. [pages included 381-408]

Additional Textbooks/Reading:

AndEngine—<http://www.andengine.org>

Cocos2D—<http://code.google.com/p/cocos2d-android>

GameMaker—<http://www.yoyogames.com/studio>

libgdx—<https://libgdx.badlogicgames.com>

Unity—<http://www.unity3d.com>

Unreal Engine—<http://www.unrealengine.com>

Learning Activities

1. Create a simple game app that's easy to code and to apply specific features (i.e. Cannon Game app).
2. Create a custom SurfaceView subclass for displaying the game's graphics from a separate thread of execution.
3. Draw graphics using Paints and a Canvas and override View's onTouchEvent method to fire a cannonball when the user touches the screen.
4. Perform simple collision detection. Add sound to your app using a SoundPool and the AudioManager. Override Fragment lifecycle method onDestroy and then use immersive mode to enable the game to occupy the entire screen, but still allow the user to access the system bars.

Summary

In this section, the mystery on how to create a Cannon Game app, which challenges the player to destroy nine targets before a 10-second time limit expires is presented as a case-study example with real-source code. The user aims and fires the cannon by touching the screen. To draw on the screen from a separate thread, you created a custom view by extending class SurfaceView. The section presents that custom component class names must be fully qualified in the XML layout element that represents the component with additional Fragment lifecycle methods. The

developed app manually performs its animations by updating the game elements on a SurfaceView from a separate thread of execution.

Section 8. REST Web Services, AsyncTask, HttpURLConnection, Processing JSON Responses, JSONObject and JSONArray

Objectives

This section discusses the architectural style for implementing web services known as Representational State Transfer (REST). It demonstrates that apps use web standards, such as HyperText Transfer Protocol (HTTP), to invoke RESTful web services and receive their responses. In addition, it presents the OpenWeatherMap.org web service used in this app which returns the forecast as a String in JavaScript Object Notation (JSON) format demonstrating that JSON as a textbased format (as objects are represented as collections of name/value pairs). The classes JSONObject and JSONArray from the org.json package to process the JSON data are presented and the section shows the methodologies and steps that are required to invoke web services. Finally it demonstrates how to perform long-running operations outside the GUI thread and receive their results in the GUI thread by using AsyncTask objects.

Expected learning outcomes

After the completion of this section, the students are expected to:

- introduce and apply using real-time source code writing on REST web services, AsyncTask, HttpURLConnection, JSON response using package org.json classes, JSONObject, JSONArray etc.
- introduce the steps that are required to invoke the web service.
- understand constraints of web services invocation and the RESTful web services.
- demonstrate and analyze the several material-design features from the Android Design Support Library (i.e. hint on the screen even after the user began entering text, a FloatingActionButton to enable the user to submit input and a Snackbar to display an informational message to the user).

Keywords

REST web services, AsyncTask, HttpURLConnection, JSON response using package org.json classes, JSONObject, JSONArray.

Bibliography

Deitel P., Deitel H. and Wald A., Android™ 6 for programmers an app-driven approach 3rd Edition ISBN-13: 978-0-13-428936-6, ISBN-10: 0-13-428936-6, 2016, pp. 400. [Chapter 7]

Additional Textbooks/Reading:

<http://openweathermap.org/forecast16>

<http://openweathermap.org/api>

Learning Activities

1. Use the free OpenWeatherMap.org REST web services to get a 16-day weather forecast for a city specified by the user.
2. Use an AsyncTask and an HttpURLConnection to invoke a REST web service or to download an image in a separate thread and deliver results to the GUI thread.
3. Process a JSON response using package org.json classes JSONObject and JSONArray.
4. Define an ArrayAdapter that specifies the data to display in a ListView. Use the ViewHolder pattern to reuse views that scroll off the screen in a ListView, rather than creating new views as well as use the material design components TextInputLayout, Snackbar and FloatingActionButton from the Android Design Support Library.

Summary

In this section, the architectural style for implementing web services known as Representational State Transfer (REST) are demonstrated through a WeatherViewer app. The app is able to obtain, based on a specified scenario, a city's 16-day weather forecast from web services provided by OpenWeatherMap.org and displayed the forecast in a ListView. It demonstrates how the apps use web standards, such as HyperText Transfer Protocol (HTTP), to invoke RESTful web services and receive their responses. The OpenWeatherMap.org web service used in this app returned the forecast as a String in JavaScript Object Notation (JSON) format. Finally JSON as a textbased format in which objects are represented as collections of name/value pairs and the classes JSONObject and JSONArray from the org.json package to process the JSON data. All the above are performed using a methodology-based scenario for implementing JSON-oriented web services.

Section 9. Social apps and orientation: Data Stores, Network Services, and supportive cross platform APIs

Objectives

The main objectives of the section are to provide a good realization and clear identification of the basic mechanisms for developing Social apps and their respective APIs and orientation in the context of Data Stores, Network Services, and supportive cross platform APIs. The Twitter Searches app is demonstrated and students will gain knowledge for the mechanisms that allow users to save favorite searches/queries with easy-to-remember short tag names that are displayed as a scrollable list. The mechanisms for the Twitter Searches app are thoroughly assessed with respect to the mechanisms for SharedPreferences file to store and manipulate key-value pairs representing the user's saved Twitter searches by the introduced RecyclerView (from package support.v7.widget)—as a flexible, customizable view that enables you to control how an app displays a scrolling list of data. Finally the section analyzes through case studies the operational characteristics and utilized mechanisms towards enabling social orientation in a mobile application.

Expected learning outcomes

After the completion of this section, the students are expected to:

- apply using real-time source code writing on social web services, SharedPreferences, RecyclerView, RecyclerView.Adapter, RecyclerView.ItemDecoration, AlertDialog.Builder, AlertDialog etc.
- demonstrate and analyze the several material-design features from the Android Design Support as well as C# visual design operations.
- introduce the steps that are required to invoke the social web service.
- understand constraints of social web services invocation and be able to develop jQueryMobile or Sencha Touch, along with the Facebook (for example PhoneGap plugin), to develop an attractive operational social application.

Keywords

Android Design Support, C# visual design operations, SharedPreferences, RecyclerView, RecyclerView.ItemDecoration, social web services and utilized mechanisms.

Bibliography

Deitel P., Deitel H. and Wald A., Android™ 6 for programmers an app-driven approach 3rd Edition ISBN-13: 978-0-13-428936-6, ISBN-10: 0-13-428936-6, 2016, pp. 400. [Chapter 8]

Additional Textbooks/Reading:

1. <https://about.twitter.com/company>.
2. <http://www.businessinsider.com/twitter-monthly-active-users-2015-7?r=UK&IR=T>.

Learning Activities

1. Use SharedPreferences to store key–value pairs of data associated with an app using a real-time app.
2. Use an implicit Intent to open a website in a browser and use an implicit Intent to display an intent chooser containing a list of apps that can share text.
3. Display a scrolling list of items in a RecyclerView and use a subclass of RecyclerView.Adapter to specify a RecyclerView’s data.
4. Use in an Object Oriented notation/way to subclass/derive a class from RecyclerView.ViewHolder to implement the view-holder pattern for a RecyclerView. In addition, use a subclass of RecyclerView.ItemDecoration to display lines between a RecyclerView’s items.
5. Demonstrate in a real-time source code the AlertDialog.Builder object to create an AlertDialog that displays a list of options.

Summary

How the social app engine works in an application and the basic mechanisms for developing Social apps and their respective APIs and orientation, are presented in this section. The social app engine in the context of Data Stores, Network Services, and supportive cross platform APIs are demonstrated through real-time source code writing and more specifically the Twitter Searches app. The mechanisms for the Twitter Searches app are thoroughly assessed with respect to the mechanisms for SharedPreferences file to store and manipulate key–value pairs representing the user’s saved Twitter searches via RecyclerView and analysis that is provided through case studies for the operational characteristics and the utilized mechanisms within this social context.

Section 10. Model-View-Controller design patterns, Handling Activity Lifecycle Events, Handling user generated actions and passive ones. Threads' handling and Rooting.

Objectives

The aim of this section is to provide a good realization and experience on the IDE of a utilized tool (ie. Android or any other) and the features that are used to build a specific application. The section presents the various Android classes to create objects call methods on classes and objects, use inheritance to create a class that defines the sample Calculator's functionality and use event handling, anonymous inner classes and interfaces to process the user's GUI interactions. It focuses also on the Model-View-Controller design patterns, and on the issues for handing the activity such as the Activity_Lifecycle_Events, and demonstrates the user generated actions in an active and passive mode using parallel processing like Threads' handling policies. Finally, it demonstrates how to create a root policy management by creating a single mechanism to root a mobile device for any platform.

Expected learning outcomes

After the completion of this section, students are expected to:

- introduce and apply using real-time source code writing on GUI interactions, active_Activity, paused_Activity, stopped_activity and override the onCreate method in every activity etc.
- understand constraints of Activity_Lifecycle_Events.
- demonstrate and analyze the several material-design features from the AppCompatActivity Library and Class AppCompatActivity Design Support as well as C# visual design operations.
- introduce the steps that are required to how to use a Configuration object to determine whether the app was running on a tablet in landscape orientation.
- demonstrate how to manage a large number of image resources using subfolders in the app's assets folder and how to access those resources via an AssetManager, InputStream, and an ImageView.

Keywords

GUI interactions, Activity_Lifecycle_Events, root policy management, AssetManager, InputStream, ImageView, thread management app engine.

Bibliography

Deitel P., Deitel H. and Wald A., Android™ 6 for programmers an app-driven approach 3rd Edition ISBN-13: 978-0-13-428936-6, ISBN-10: 0-13-428936-6, 2016, pp. 400. [Chapters 3 and 4]

Additional Textbooks/Reading:

The Android design guidelines indicate that text displayed in your GUI should be brief, simple and friendly with the important words first. For details on the recommended writing style, see <http://developer.android.com/design/style/writing.html>.

<http://developer.android.com/tools/support-library>

<http://developer.android.com/reference/android/widget/packagesummary>.

Html

<http://developer.android.com/tools/support-library/index.html>

<http://www.google.com/design/spec/style/color.html#color-themes>

<http://developer.android.com/training/material/theme.html>

Learning Activities

1. Use in real source code program the IDE's Component Tree window to add views to a GridLayout. Use TextViews, an EditText and a SeekBar and by utilizing Java object-oriented programming capabilities, including classes, objects, interfaces, anonymous inner classes and inheritance to add functionality to an app.
2. Use in real source code program event handling to respond to user interactions with an EditText and a SeekBar.
3. Use in 'real source code program' a PreferenceFragment to automatically manage and persist an app's user preferences, i.e. SharedPreferences.Editor to modify key-value pairs of data associated with an app and use then the app's assets subfolders to organize image resources and manipulate them with an AssetManager. Finally in your program define an animation and apply it to a View.
4. Use real source code to create a program consisting of a Handler to schedule a future

task to perform on the GUI thread and utilize Android's logging mechanism to log error messages.

Summary

The section provides answers to the questions on how the programmers can manage the different objects that call methods, and how to use inheritance to create a class that defines the sample mechanism and use event handling, in order to process the user's GUI interactions. It demonstrated the Model-View-Controller design patterns, and the issues for handling the activity such as the Activity_Lifecycle_Events, and presented how to create a root policy management by creating a single mechanism to root a mobile device for any platform.

Section 11. Web service integration and Services according to user-interface and interaction patterns

Objectives

The aim of this section is to introduce and provide a thorough demonstration of existing technologies supporting the web service integration and services according to user-interface and interaction patterns including the context awareness interaction of the users. This means that based on the mobility users can receive Alerts via notifications based on the context and on the ambient environment. The section includes mechanisms for Requesting and Requiring Permissions, Accessing Location-Based Services Mapping with MapView and MapActivity, Handling interruptions (Telephone Calls) and parallel external actions. In addition, it presents the FragmentTransactions and the Fragment Back Stack, SQLite, SQLiteDatabase, SQLiteOpenHelper, ContentProvider, ContentResolver, Loader, LoaderManager etc.

Expected learning outcomes

After the completion of this section, students are expected to:

- introduce and apply using real-time source code writing on mechanisms that trigger the alerts for users, requesting and requiring permissions, Accessing Location-Based Services Mapping with MapView and MapActivity, Handling interruptions etc.
- demonstrate and analyze the several process-oriented issues and material-design features from the supported platforms.
- be able to provide real source code and a complete error-free program with the associated steps that are required to invoke the transaction services and demonstrate the transaction through mechanisms provided by FragmentTransactions, SQLiteOpenHelper, ContentProvider and SQLiteDatabase object.

Keywords

Location-Based Services Mapping MapView, MapActivity Handling, FragmentTransactions, SQLiteOpenHelper, ContentProvider, SQLiteDatabase object.

Bibliography

Deitel P., Deitel H. and Wald A., Android™ 6 for programmers an app-driven approach 3rd Edition ISBN-13: 978-0-13-428936-6, ISBN-10: 0-13-428936-6, 2016, pp. 400. [Chapter 9]

Additional Textbooks/Reading:

Nakamura M. et al., Programming Android, 2nd Edition, 2016.

A series of examples on the LBS and Location-Based Services Mapping with MapView and MapActivity found at www.androidbook.com/proandroid4/projects

<https://developer.android.com/training/building-location.html>

Learning Activities

1. Use in real source code program the FragmentTransactions and the back stack to dynamically attach Fragments to and detach Fragments from the GUI.
2. Be able to provide a RecyclerView to display data from a database. To this end you should be able to create and open databases with SQLiteOpenHelper, use a ContentProvider and a SQLiteDatabase object to interact with data in a SQLite database, use a ContentResolver to invoke methods of a ContentProvider to perform tasks with a database and use a LoaderManager and Loaders to perform database access asynchronously outside the GUI thread.
3. Use in real source code program to manage on-demand Location-Based Services Mapping with a simple MapView and MapActivity textual reports on the screen (i.e. by using widgets).

Summary

Being in a location with a mobile device means more for the programmer that it seems in reality. Mobile users can receive Alerts via notifications based on the context and ambient environment and this section serves as the 'knowledge bridge' with the existing technologies encompassing mechanisms for Requesting and Requiring Permissions, Accessing Location-Based Services Mapping with MapView and MapActivity, Handling interruptions and parallel external actions. This section also includes the DB components for such implementations like the FragmentTransactions and the Fragment Back Stack, SQLite, SQLiteDatabase etc.

Section 12 Hardware module interactions and power diversities as a first-order design constraint

Objectives

This section introduces the background and architectural concepts for implementing with real-source code how to monitor significant changes in Battery Level and interact with the low level processes of the system. This means that based on the level of usage, users can conserve energy notifications based on the context and level of draining apps that the mobile device hosts. The section includes mechanisms for ACTION_BATTERY_CHANGED Context.registerReceiver() Intent.ACTION_BATTERY_CHANGED and how to use the Battery Manager API with the BroadcastReceiver, batteryInfoReceiver, IntentFilter etc.

Expected learning outcomes

After completion of the course students are expected to be able to:

- understand the power constraints and the limitations of a mobile device and apply effective code solution to overcome energy consumption problems.
- demonstrate and analyze the various components and be able to use them in real-time source code writing in pilot apps.
- write real-time source code for implementing solution of power-oriented resource-constraint problems in integrated Mobile/Tablet/Sensor device.
- identify the Hardware module interactions and power diversities of a mobile device.

Keywords

Battery Manager API, BroadcastReceiver, IntentFilter, [ACTION CHARGING](#), [ACTION DISCHARGING](#), batteryInfoReceiver, Context.registerReceiver(), Intent.ACTION_BATTERY_CHANGED, modules for actions for manipulating batteries and charging for a device.

Bibliography

Manas E. L. and Grancini D., Android High Performance Programming, Paperback: 412 pages, Publisher: Packt Publishing - ebooks Account (August 29, 2016), ISBN-10: 1785288954, ISBN-13: 978-1785288951. [Chapters 6 and 7]

Deitel P., Deitel H. and Wald A., Android™ 6 for programmers an app-driven approach 3rd Edition ISBN-13: 978-0-13-428936-6, ISBN-10: 0-13-428936-6, 2016, pp. 400. [Chapter 10]

Additional Textbooks/Reading:

Manas E.L, Grancini D., Android High Performance Programming [Chapter 8. Optimizing Battery Consumption].

Optimizing for Doze and App Standby policies found at
https://developer.android.com/training/monitoring-device-state/doze-standby.html#understand_app_standby

<https://developer.android.com/training/advanced.html>

Learning Activities

1. Use in real source code in a complete error-free program how to manage the levels of consumption and store them in history in the local DB of the device.
2. Use in real source code in a complete error-free program the various components for low-level monitoring of the processes of a mobile device.
3. Provide a solution for a power resource constrained device by using the Battery Manager API with the BroadcastReceiver, batteryInfoReceiver, IntentFilter etc.
4. Show that you can manipulate the Hardware components and modules and allow an efficient interaction among them for a Mobile device/Tablet/Sensor mote or any other programmable mobile OS-oriented device.

Summary

The recent advancements in smart devices' technologies are bringing forth new and exciting opportunities for the mobile computing application scenarios and are enhancing with more challenges the field for programmers. This section shows dynamically how low level processes can be reached and manipulated in a mobile device and demonstrates with real-source code how to monitor significant changes in Battery Levels and how to interact with the low level processes of the system. Manipulations for the Hardware components and modules of a Mobile device/Tablet/Sensor mote or any other programmable mobile OS-oriented device are being shown with example -focus on- scenarios and by using real-time examples by utilizing two different tools (PhoneGap IDE with multi-platform capabilities and Android Studio).

Appendix D

Below is a list of recent books to be added to each course, where it applies. The course outlines have been updated accordingly.

COMP-512

- IEEE Computer Society, Pierre Bourque (Ed.) and Richard E. Fairley (Ed.), “Guide to the Software Engineering Body of Knowledge (SWEBOK(R)): Version 3.0”, IEEE Computer Society Press; 3rd edition, 2014.

COMP-513

- H. Song, D. B. Rawat, S. Jeschke and C. Brecher, “Cyber-Physical Systems: Foundations, Principles and Applications (Intelligent Data-Centric Systems: Sensor Collected Intelligence)”, Academic Press, 2016.
- S. Shahrestani, “Internet of Things and Smart Environments”, Springer International Publishing AG, 2017.
- C. Dobre et al., “Ambient Assisted Living and Enhanced Living Environments: Principles, Technologies and Control”, Elsevier, 2017.

COMP-515

- M. Van Steen and A. S. Tanenbaum, “Distributed Systems”, 3rd Edition. CreateSpace Independent Publishing Platform, 2017.

COMP-522

- I. H. Witten, E. Frank, M. A. Hall, C. J. Pal, Data Mining: Practical Machine Learning Tools and Techniques (Morgan Kaufmann Series in Data Management Systems) 4th Edition, 2016.

COMP-523

- K. C. Finney, “3D Game Programming: All in One”, 3rd Ed, Course Technology, 2013.

COMP-524

- J. Farrell, “Visual C# 2015”, 6th Ed, Cengage Learning, 2016.

COMP-525

- M. Migliardi, A. Merlo, S. Al-HajBaddar, “Adaptive Mobile Computing”, 1st ed., Elsevier, 2017.
- C. X. Mavromoustakis, G. Mastorakis, E. Pallis, “Resource Management in Mobile Computing Environments”, Springer International Publishing AG, series in Modeling and Optimization in Science and Technologies, 2015.
- Loke, Seng, “Crowd-Powered Mobile Computing and Smart Things”, Springer International Publishing AG, 2017.
- C. Dobre, C. Mavromoustakis, N. Garcia, R. Goleva, and G. Mastorakis, “Ambient Assisted Living and Enhanced Living Environments: Principles, Technologies and Control”, Elsevier, 2017.

COMP-528

- P. Deitel, H. Deitel, A. Wald, “Android™ 6 for programmers an app-driven approach”, Addison-Wesley Professional, 2016.
- J. M. Wargo, “PhoneGap Essentials: Building Cross-Platform Mobile Applications”, Addison-Wesley Profession, 2015.
- Crestani, Fabio, Mizzaro, Stefano, Scagnetto, Ivan, “Mobile Information Retrieval”, Springer International Publishing AG, 2017.
- A. Mishra, “Amazon Web Services for Mobile Developers: Building Apps with AWS”, Wiley, 2017.

COMP-538

- Tapscott, D. and Tapscott, A., “The Blockchain Revolution”, Portfolio, 2016.