University of Macau
Faculty of Science and Technology
Department of Computer and Information Science
CISB355 Computer Graphics
Syllabus
2nd Semester 2014/2015
Part A – Course Outline

Elective course in Computer Science

Catalog description:
(2-2) 3 credits. Study and implement computer graphics techniques. Topics include graphics systems and interactive techniques, graphics primitives, 2D/3D geometric and viewing transformations, clipping, projections, illumination models, and shading methods.

Course type:
Theoretical with substantial laboratory/practice content

Prerequisites:
- CISB110
- MATB220

Textbook(s) and other required material:
- Donald D. Hearn, M. Pauline Baker, Warren Carithers, *Computer Graphics with OpenGL*, 4th Edition, Pearson Prentice Hall, 2010.

References:
- OpenGL Architecture Review Board, Dave Shreiner, Mason Woo, Jackie Neider, Tom Davis. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.1*, 8th Edition, Addison-Wesley, 2012.
- Edward Angel, Dave Shreiner, *Interactive Computer Graphics: A Top-Down Approach with Shader-Based OpenGL*, 6th Edition, Addison-Wesley, 2011.

Major prerequisites by topic:
- Analytic geometry and linear algebra.
- C/C++ programming.

Course objectives:
- To make students understand the basic concepts in computer graphics. [a]
- To teach students the basic techniques to model 3D objects. [a]
- To teach students the graphics rendering pipeline and methods to create the 3D virtual scene. [a, c]
- To teach students the illumination models and rendering techniques. [a]
- Learning to implement these techniques by OpenGL. [a, c]

Topics covered:
- **Graphics hardware and software, graphics applications (4 hours):** Introduce the basic concept and vocabulary of computer graphics, along with the overview of the hardware and software components of graphics systems. Introduce a plenty of computer graphics techniques and applications nowadays.
- **Graphics primitives and their attributes (5 hours):** Introduce the graphics output primitives and their attributes. Discuss the scan-conversion algorithms for displaying output primitives.
- **2D/3D geometric and modeling transformations (5 hours):** Study the 2D/3D transformations in matrix representations. Learn how to apply transformation operations to geometric models to reposition or resize them.
- **Graphics pipeline and 2D/3D viewing transformations (6 hours):** Introduce the concept of graphics pipeline. Discuss the procedures for displaying views of 2D and 3D scenes.
- **Illumination models and surface-rendering methods (5 hours):** Discuss the illumination models and methods for applying lighting conditions to a scene.
- **Texture mapping (2 hours):** Introduce the concept of texture mapping and related techniques.

- **OpenGL and GLUT programming (8 hours):** Study how to use OpenGL and GLUT to implement all these graphics concepts introduced.

Class/laboratory schedule:

Timetabled work in hours per week			No of teaching weeks	Total hours	Total credits	No/Duration of exam papers
Lecture	Tutorial	Practice				
2.5	0.5	0.5	14	49	3	1 / 2 hours

Student study effort required:

Class contact:	
Lecture	35 hours
Tutorial and practice	14 hours
Other study effort	
Self-study	14 hours
Homework assignment	14 hours
Programming practice	14 hours
Total student study effort	91 hours

Student assessment:

Final assessment will be determined on the basis of:

Assignments	15%
Programming practice	20%
Mid-term exam	30%
Final exam	35%

Course assessment:

The assessment of course objectives will be determined on the basis of:

- Assignments, programming practice and exams
- Course evaluation

Course outline:

Weeks	Topic	Course work
1	**Introduction of Computer Graphics** Computer graphics concept; Computer graphics applications	
2	**Graphics Hardware and Software Systems** Video display devices; Input devices; Graphics software and OpenGL	Assignment#1
3-4	**Graphics Output Primitives** Coordinate reference frames; Line-Drawing Algorithms; Points, lines, and polygons; OpenGL implementation and related functions	Assginment#2, Programming Practice#1
5	**OpenGL and GLUT Programming** OpenGL; GLUT	Programming Practice#2
6	**Attributes of Graphics Primitives** Color and gray scale; Points, lines, and fill-area attributes; OpenGL implementation and related functions	Programming Practice#3
7-8	**Geometric Transformations** Translation, rotation, scaling, reflection and shear; Matrix representation and homogeneous coordinates; Inverse and composite transformation; OpenGL implementation and related functions	Assignment#3, Programming Practice#4
9	**Midterm Exam**	
10-11	**2D Viewing** 2D viewing pipeline; Clipping window; Normalization and viewport transformations; Clipping algorithms; OpenGL implementation and related functions	Programming Practice#5
12	**3D Viewing**	Assignment#4,

Weeks	Topic	Course work
	3D viewing concepts and viewing pipeline; Projection transformations; Viewport transformation; OpenGL implementation and related functions	Programming Practice#6
13-14	**Illumination Models and Surface Rendering Methods** Light sources; Basic illumination models; Global illumination models; Texture mapping; OpenGL implementation and related functions **Review**	Assignment#5

Contribution of course to meet the professional component:
This course provides students the fundamental knowledge of computer graphics, which are necessary for their professional career in this field.

Relationship to CS program objectives and outcomes:
This course primarily contributes to the Computer Science program outcomes that develop student abilities to:
(a) An ability to apply knowledge of computing and mathematics appropriate to the programme outcomes and to the discipline;
(c) An ability to analyse a problem, and identify and define the computing requirements appropriate to its solution;

Relationship to CS program criteria:

Criterion	DS	PF	AL	AR	OS	NC	PL	HC	GV	IS	IM	SP	SE	CN
Scale: 1 (highest) to 4 (lowest)							2		1					

Discrete Structures (DS), Programming Fundamentals (PF), Algorithms and Complexity (AL), Architecture and Organization (AR), Operating Systems (OS), Net-Centric Computing (NC), Programming Languages (PL), Human-Computer Interaction (HC), Graphics and Visual Computing (GV), Intelligent Systems (IS), Information Management (IM), Social and Professional Issues (SP), Software Engineering (SE), Computational Science (CN).

Course content distribution:

Percentage content for			
Mathematics	Science and engineering subjects	Complementary electives	**Total**
0%	100%	0%	100%

Persons who prepared this description:
Dr. Wen Wu, Prof. En Hua Wu

Part B – General Course Information and Policies

2nd Semester 2014/2015

Instructor:	Dr. Wen Wu	Office:	E11-4019
Office hour:	by appointment	Phone:	8822 4477
Email:	wenwu@umac.mo		

Time/Venue: Thursday 11:00 – 13:00 / E11-1012 (Lecture)
Monday 11:00 – 13:00 / E6-2095 (Tutorial/Lab)

Grading distribution:

Percentage Grade	Final Grade	Percentage Grade	Final Grade
100 - 93	A	92 - 88	A–
87 - 83	B+	82 - 78	B
77 - 73	B–	72 - 68	C+
67 - 63	C	62 - 58	C–
57 - 53	D+	52 - 50	D
below 50	F		

Comment:
This course is to provide students with a comprehensive introduction to 2D and 3D computer graphics techniques. These techniques are taught in learning by doing rather than theoretical. Students will write programs to test various concepts and techniques, and gain practical experience by OpenGL. A good knowledge of C/C++ programming is the prerequisite.

Homework and programming practice policy:
Learning by doing is an essential way to grasp the principles of the course.
- There will be approximately 5 homework assignments and 6 programming practice.
- No late homework and programming practice is accepted.
- No marks will be given in case of copying from others.

Exam:
A mid-term exam will be held during the semester. Both the mid-term and final exams are closed book examinations.

Note:
- Reading the textbook is imperative.
- Check UMMoodle (https://ummoodle.umac.mo/) for announcement, lecture notes, homework and related resources.
- Cheating is absolutely prohibited by the university. Issues regarding the final exam follow the university rules and policies.

Student Disabilities Support Service:

The University of Macau is committed to providing an equal opportunity in education to persons with disabilities. If you are a student with a physical, visual, hearing, speech, learning or psychological impairment(s) which substantially limit your learning and/or activities of daily living, you are encouraged to communicate with your instructors about your impairment(s) and the accommodations you need in your studies. You are also encouraged to contact the Student Disability Support Service of the Student Counselling and Development Section (SCD), which provides appropriate resources and accommodations to allow each student with a disability to have an equal opportunity in education, university life activities and services at the University of Macau. To learn more about the service, please contact SCD at scd.disability@umac.mo, or 8822 4901 or visit the following website:
http://www.umac.mo/sao/scd/sds/aboutus/en/scd_mission.php

Appendix - Measurement Dimensions and Rubric for Program Outcome (a) and (c)

(a) An ability to apply knowledge of computing and mathematics appropriate to the programme outcomes and to the discipline

Measurement Dimension	Excellent (80-100%)	Average (60-79%)	Poor (<60%)
1. An ability to apply knowledge of computing to the solution of complex computing problems.	Students understand the computing principles, and their limitations in the respective applications. Use the computing principles to formulate and solve complex computing problems.	Students understand the computing principles, and their limitations in the respective applications. But they have trouble in applying these computing principles to formulate and solve complex computing problems.	Students do not understand the computing principles, and their limitations in the respective applications. Do not know how to apply the appropriate computing principles to formulate and solve complex computing problems.
2. **An ability to apply knowledge of mathematics to the solution of complex computing problems.**	Students understand the mathematical principles, e.g., calculus, linear algebra, probability and statistics, relevant to computer science, and their limitations in the respective applications. Use mathematical principles to formulate and solve complex computing problems.	Students understand the theoretical background and know how to choose mathematical principles relevant to computer science. But they have trouble in applying these mathematical principles to formulate and solve complex computing problems.	Students do not understand the mathematical principles and do not know how to formulate and solve complex computing problems.

(c) An ability to analyse a problem, and identify and define the computing requirements appropriate to its solution

Measurement Dimension	Excellent (80-100%)	Average (60-79%)	Poor (<60%)
1. An ability to understand problem and identify the fundamental formulation	Students understand problem correctly and can identify the fundamental formulation	Student understand problem correctly, but have trouble in identifying the fundamental formulation	Students cannot understand problem correctly, and they do not know how to identify the fundamental formulation
2. An ability to choose and properly apply the correct techniques	Students know how to choose and properly apply the correct techniques to solve problem.	Students can choose correct techniques but have trouble in applying these techniques to solve problem.	Students have trouble in choosing the correct techniques to solve problem.