Approved in 37th BoA Meeting (29-10-2020)

Course Name: Computer Graphics and Geometric Design

Course Number: CS550

Credits: 2-0-2-3

Prerequisites: IC111 Linear Algebra or similar course, IC250 Data Structure and Algorithms or similar course.

Intended for: B.Tech. (CSE, EE and ME) II/III/IV year/ MS/M.Tech /PhD (any branch)

Distribution: Elective for B.Tech. II/III/IV year/ MS/M.Tech/PhD

Preamble: This course is intended to introduce the graphics processing pipeline found in modern day computers. This pipeline has three main phases. Depending on the application domain, the first phase creates 2D/3D modeling data which is in either discrete or continuous format. The second phase converts this data into simpler discrete primitive forms such as points, line-segments and triangles. The final phase maps this data onto the pixels of the computer display. Topics from all the three phases will be discussed. The course will include assignments about using popular graphics software suites as well as programming assignments.

Learning outcome: After taking this course, students will

1. be familiar with the graphics processing pipeline.
2. understand in depth, the set of fundamental algorithms which drive modern graphics engines.
3. be able to choose between alternative graphics/geometry data-structures which best suit the target application.
4. get hands-on experience with graphics programming APIs such as OpenGL.
5. get familiarity with using 3D graphics software packages such as Blender.
6. be able to create realistic scenes and environments.
7. be able to choose between alternative light sources and object materials in order to achieve the desired visual effect.
8. be able to use design kernels such as Open Cascade for solid modeling.
9. be able to design simple video games.

Course modules:

1. *Introduction:* (2 hours)
   Motivation; Use of computer graphics in different domains; Digital display technologies; Color; Graphics processing pipeline; OpenGL.

2. *Geometry representation:* (2 hours)
   Lines, triangles, polygons, meshes, point-clouds, polynomials, B-splines; Parametric v/s. implicit representation, boundary representation, volumetric representation.

3. *Rasterization:* (6 hours)
   World and camera transformations; Orthogonal and perspective projections; Hidden-

surface removal, Z-buffering; Bresenham's line drawing algorithm; Triangle drawing primitive; Flood-fill algorithm.

4. *Lighting, shaders and texture mapping:* (6 hours)
Radiometric concepts such as BRDF, BTDF, BSSRDF; Types of light sources such as point, spot, parallel; Texture mapping; Shading schemes such as Flat, Gouraud, Phong; Flattening of 3D surfaces; Bump maps, Normal maps, Displacement maps.

5. *Ray tracing and ray casting:* (3 hours)
Mechanism of tracing the path of light in order to generate realistic images from 3D scenes.

6. *Modeling with curves:* (5 hours)
Differential geometry of planar and space curves; Definition and basic properties of Bezier and B-spline curves; Weierstrass approximation theorem.

7. *Modeling with surfaces:* (4hours)
Differential geometry of surfaces; Definition and basic properties of tensor-product Bezier surfaces.

Lab exercises:
Lab to be conducted on a 2-hour slot. It will be conducted in tandem with the theory course so the topics for problems given in the lab are already initiated in the theory class. The topics taught in the theory course shall be appropriately sequenced for synchronization with the laboratory.
Lab1: Installation of Blender, familiarization of the GUI, simple exercises involving creation of primitive shapes. Installation of OpenGL, writing a 'Hello World' application. Draw a few parametric as well as implicitly defined curves.
Lab2-4: Implementation of Bresenham's line drawing algorithm. Implementation of flood-fill algorithm.
Lab5-7: Create a 3D scene in Blender and apply various models of light-interaction such as BRDF, BTDF, BSSRDF. Insert different types of light sources into the scene. Apply textures to 3D objects. Lighting and shading in OpenGL, Raytracing
Lab8-10: Implementation of ray-tracing.
Lab11-12: Installation of Open Cascade. Construct and plot Bezier curves. Plot the tangents and osculating circles at a few points. Plot the Frenet frame at a few sampled points. Construct and plot B-spline curves.
Lab13-14: Construct and plot tensor-product Bezier surfaces. Plot tangent-plane and curvature at sampled points. Extract iso-curves.

Textbooks:
1. Donald D. Hearn *et al., Computer Graphics with OpenGL,* 3rd edition, Pearson, 2013

Reference Books:

1. Steve Marschner et al., *Fundamentals of Computer Graphics,* 4th Edition, A K Peters/CRC Press, 2015
2. Gerald Farin, *Curves and Surfaces for CAGD,* 5th Edition, 2001
3. Dave Shreiner, OpenGL Programming Guide: The Official Guide to Learning OpenGL, Versions 3.0 and 3.1 (7th Edition)

Content Similarity Declaration with Existing Courses: N/A

Sr #	Course code	Similarity content	Approx. % of content
1	CS451	Linear transforms, RGB space	5%

Justification for new course proposal if cumulative similarity content is > 30%: N.A.