Ucayali State University (UNU)
School of Computer Science
Sillabus 2023-I

1. COURSE
CS251. Computer graphics (Elective)

2. GENERAL INFORMATION

2.1 Credits	:	4
2.2 Theory Hours	:	2 (Weekly)
2.3 Practice Hours	:	2 (Weekly)
2.4 Duration of the period	:	16 weeks
2.5 Type of course	:	Elective
2.6 Modality	:	Blended
2.7 Prerrequisites	:	• CS312. Advanced Data Structures . (6^{th} Sem) • MA307. Mathematics applied to computing. (6^{th} Sem)

3. PROFESSORS

Meetings after coordination with the professor

4. INTRODUCTION TO THE COURSE
It offers an introduction to the area of Computer Graphics, which is an important part of Computer Science. The purpose of this course is to investigate the fundamental principles, techniques and tools for this area.

5. GOALS

- Bring students to concepts and techniques used in complex 3-D graphics applications.
- Give the student the necessary tools to determine which graphics software and which platform are best suited to develop a specific application.

6. COMPETENCES

a) An ability to apply knowledge of mathematics, science. (**Usage**)

b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)

i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)

j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Usage**)

7. TOPICS

<table>
<tr><td colspan="2">Unit 1: Fundamental Concepts (6)</td></tr>
<tr><td colspan="2">Competences Expected: a,b</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

Media applications including user interfaces, audio and video editing, game engines, cad, visualization, virtual reality
Tradeoffs between storing data and re-computing data as embodied by vector and raster representations of images
Additive and subtractive color models (CMYK and RGB) and why these provide a range of colors
Animation as a sequence of still images

</td><td>

Explain in general terms how analog signals can be reasonably represented by discrete samples, for example, how images can be represented by pixels [Familiarity]
Describe color models and their use in graphics display devices [Familiarity]
Describe the tradeoffs between storing information vs storing enough information to reproduce the information, as in the difference between vector and raster rendering [Familiarity]
Describe the basic process of producing continuous motion from a sequence of discrete frames (sometimes called “flicker fusion”) [Familiarity]

</td></tr>
<tr><td colspan="2">Readings : [HB90]</td></tr>
</table>

Unit 2: Basic Rendering (12)	
Competences Expected: a,b,i	
Topics	**Learning Outcomes**
• Rendering in nature, e.g., the emission and scattering of light and its relation to numerical integration • Forward and backward rendering (i.e., ray-casting and rasterization) • Basic radiometry, similar triangles, and projection model • Affine and coordinate system transformations • Ray tracing • Visibility and occlusion, including solutions to this problem such as depth buffering, Painter's algorithm, and ray tracing • Simple triangle rasterization • Rendering with a shader-based API • Application of spatial data structures to rendering • Sampling and anti-aliasing • Forward and backward rendering (i.e., ray-casting and rasterization)	• Discuss the light transport problem and its relation to numerical integration ie, light is emitted, scatters around the scene, and is measured by the eye [Familiarity] • Describe the basic graphics pipeline and how forward and backward rendering factor in this [Familiarity] • Create a program to display 3D models of simple graphics images [Usage] • Obtain 2-dimensional and 3-dimensional points by applying affine transformations [Usage] • Apply 3-dimensional coordinate system and the changes required to extend 2D transformation operations to handle transformations in 3D [Usage] • Contrast forward and backward rendering [Assessment] • Explain the concept and applications of texture mapping, sampling, and anti-aliasing [Familiarity] • Explain the ray tracing/rasterization duality for the visibility problem [Familiarity] • Implement a simple real-time renderer using a rasterization API (eg, OpenGL) using vertex buffers and shaders [Usage] • Compute space requirements based on resolution and color coding [Assessment] • Compute time requirements based on refresh rates, rasterization techniques [Assessment]
Readings : [HB90], [Hug+13], [Wol11], [Shr+13]	

<table>
<tr><td colspan="2">Unit 3: Programming Interactive Systems (2)</td></tr>
<tr><td colspan="2">Competences Expected: a,b</td></tr>
<tr><td>Topics</td><td>Learning Outcomes</td></tr>
<tr><td>

Event management and user interaction
Approaches to design, implementation and evaluation of non-mouse interaction

Touch and multi-touch interfaces
Shared, embodied, and large interfaces
New input modalities (such as sensor and location data)
New Windows, e.g., iPhone, Android
Speech recognition and natural language processing
Wearable and tangible interfaces
Persuasive interaction and emotion
Ubiquitous and context-aware interaction technologies (Ubicomp)
Bayesian inference (e.g. predictive text, guided pointing)
Ambient/peripheral display and interaction

</td><td>

Discuss the advantages (and disadvantages) of non-mouse interfaces [Assessment]

</td></tr>
<tr><td colspan="2">Readings : [HB90]</td></tr>
</table>

Unit 4: Geometric Modeling (15)	
Competences Expected: a,b,i,j	
Topics	**Learning Outcomes**
• Basic geometric operations such as intersection calculation and proximity tests • Volumes, voxels, and point-based representations • Parametric polynomial curves and surfaces • Implicit representation of curves and surfaces • Approximation techniques such as polynomial curves, Bezier curves, spline curves and surfaces, and nonuniform rational basis (NURB) spines, and level set method • Surface representation techniques including tessellation, mesh representation, mesh fairing, and mesh generation techniques such as Delaunay triangulation, marching cubes • Spatial subdivision techniques • Procedural models such as fractals, generative modeling, and L-systems • Elastically deformable and freeform deformable models • Subdivision surfaces • Multiresolution modeling • Reconstruction • Constructive Solid Geometry (CSG) representation	• Represent curves and surfaces using both implicit and parametric forms [Usage] • Create simple polyhedral models by surface tessellation [Usage] • Generate a mesh representation from an implicit surface [Usage] • Generate a mesh from data points acquired with a laser scanner [Usage] • Construct CSG models from simple primitives, such as cubes and quadric surfaces [Usage] • Contrast modeling approaches with respect to space and time complexity and quality of image [Assessment]
Readings : [HB90], [Shr+13]	

Unit 5: Advanced Rendering (6)	
Competences Expected: a,b,i	
Topics	**Learning Outcomes**
• Time (motion blur), lens position (focus), and continuous frequency (color) and their impact on rendering • Shadow mapping • Occlusion culling • Subsurface scattering • Non-photorealistic rendering • GPU architecture • Human visual systems including adaptation to light, sensitivity to noise, and flicker fusion	• Demonstrate how an algorithm estimates a solution to the rendering equation [Assessment] • Prove the properties of a rendering algorithm, eg, complete, consistent, and unbiased [Assessment] • Implement a non-trivial shading algorithm (eg, toon shading, cascaded shadow maps) under a rasterization API [Usage] • Discuss how a particular artistic technique might be implemented in a renderer [Familiarity] • Explain how to recognize the graphics techniques used to create a particular image [Familiarity]
Readings : [HB90], [Hug+13], [Wol11], [Shr+13]	

Unit 6: Computer Animation (4)	
Competences Expected: a,b,i,j	
Topics	**Learning Outcomes**
• Forward and inverse kinematics • Collision detection and response • Procedural animation using noise, rules (boids/crowds), and particle systems • Skinning algorithms • Physics based motions including rigid body dynamics, physical particle systems, mass-spring networks for cloth and flesh and hair • Key-frame animation • Splines • Data structures for rotations, such as quaternions • Camera animation • Motion capture	• Compute the location and orientation of model parts using an forward kinematic approach [Usage] • Implement the spline interpolation method for producing in-between positions and orientations [Usage] • Implement algorithms for physical modeling of particle dynamics using simple Newtonian mechanics, for example Witkin & Kass, snakes and worms, symplectic Euler, Stormer/Verlet, or midpoint Euler methods [Usage] • Discuss the basic ideas behind some methods for fluid dynamics for modeling ballistic trajectories, for example for splashes, dust, fire, or smoke [Familiarity] • Use common animation software to construct simple organic forms using metaball and skeleton [Usage]
Readings : [HB90], [Shr+13]	

8. WORKPLAN

8.1 Methodology

Individual and team participation is encouraged to present their ideas, motivating them with additional points in the different stages of the course evaluation.

8.2 Theory Sessions

The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

8.3 Practical Sessions

The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

9. EVALUATION SYSTEM

********* EVALUATION MISSING ********

10. BASIC BIBLIOGRAPHY

[HB90] Donald Hearn and Pauline Baker. *Computer Graphics in C.* Prentice Hall, 1990.

[Hug+13] John F. Hughes et al. *Computer Graphics - Principles and Practice 3rd Edition.* Addison-Wesley, 2013.

[Shr+13] Dave Shreiner et al. *OpenGL, Programming Guide, Eighth Edition.* Addison-Wesley, 2013.

[Wol11] David Wolff. *OpenGL 4.0 Shading Language Cookbook.* Packt Publishing, 2011.