



DISCIPLINE ACADEMIC SHEET
ACADEMIC YEAR 2025 – 2026

1. PROGRAMME DATA

1.1. Higher Education Institution	UNIVERSITY OF CRAIOVA
1.2. Faculty	Automation, Computers and Electronics
1.3. Department	Computers and Information Technology
1.4. Field of Study	Computers and Information Technology
1.5. Study level	L(License / Undergraduate)
1.6. Organizational form	IF
1.7. Study Program (name/code) /Qualification	Computers / L2060101010

2. Discipline data

2.1 Discipline Name	Graphic Processing Systems						
2.2 Course Activities Holder	Lect. PhD. Constantin-Renato IVANESCU						
2.3. Practical Activities Holder	Lect. PhD. Constantin-Renato IVANESCU						
2.4. Study Year	4	2.5. Semester	8	2.6. Evaluation type	E	2.7. Discipline regime	DOP

3. Estimated total time (hours per semester of teaching activities)

3.1. Number of hours per week	4	in which: 3.2 course	2	3.3. sem./lab./pr.	0/2/0
3.4. Total hours of curriculum	40	in which: 3.5 course	20	3.6. sem./lab./pr.	0/20/0
Time distribution - hours/week					
Study after manual, course support, bibliography and notes					20
Additional documentation in library, on specialized electronic platforms and field					15
Training seminars / labs, homework, portfolios and essays					20
Tutorial					2
Examinations					3
Other activities					0
3.7. Total hours per individual activities					60
3.8. Total hours per semester					100
3.9. Number of credits					4

4. Preconditions (where appropriate)

4.1. of curriculum	The student must possess fundamental and specialized knowledge acquired during previous courses in the following subjects: • Computer Graphics • Mathematical Analysis • Linear Algebra and Geometry • Numerical Methods • Computer Programming
--------------------	---

	• Programming Techniques • Algorithms and Data Structures • Object-Oriented Programming • Computer Architecture • Databases • Software Engineering
4.2. of competence	The student is at an advantage if they already possess the competencies listed below (however, studying the subject without these competencies is not penalizing): • Working with scientific, engineering, and computer science fundamentals • Demonstrating initiative and action in updating professional, economic, and organizational culture knowledge

5. Conditions (where appropriate)

5.1. of the course	The course is taught using a video projector, computer, and the Internet. For more detailed explanations of concepts illustrated in the course and to answer questions from students in the room, a chalkboard is used. Students are encouraged to practice note-taking. Course materials are provided in electronic format, and access to updated documentation is ensured. The teaching process is structured as follows: • 80% theoretical presentation, based on course materials; • 20% interactive activity (practical demonstrations and dialogue with students).
5.2. of laboratory	In the laboratory, the chalkboard is used to demonstrate fundamental algorithms and mathematical models, along with a network of computers with characteristics adapted to the requirements of the field of computer graphics. These are used together with integrated programming environments to illustrate the concepts presented in the course and to solve specific types of problems assigned to students in the lab, organized in the form of two applications/projects. Experimentation with various techniques and commonly used graphic libraries (DirectX, OpenGL) will be encouraged.

6. COURSE OBJECTIVES – expected learning outcomes that are supported by completing and passing the course

Knowledge	Student/Graduate: 1. Understands the structure, functioning, and components of graphic processing systems, including processing, display, and input subsystems. 2. Explains fundamental and advanced algorithms for computer graphics, such as rasterization, lighting, texture mapping, and hidden surface removal. 3. Describes the mathematical and computational models used for realistic rendering and the development of interactive graphic applications.
Skills (Abilities)	Student/Graduate: 1. Applies mathematical and algorithmic concepts to design and implement 2D and 3D graphical applications using specialized libraries (OpenGL, Direct3D). 2. Uses modern development environments and tools to create interactive, real-time visual systems. 3. Analyzes, evaluates, and optimizes the performance and visual quality of graphical applications through experimentation and testing. 4. Integrates hardware and software components to build functional graphic subsystems based on specific requirements.

Responsibility and autonomy	Student/Graduate: 1. Works independently or collaboratively to complete laboratory assignments and graphical projects. 2. Demonstrates responsibility, ethical conduct, and professionalism in developing, documenting, and presenting graphic applications. 3. Takes initiative in updating and extending knowledge in the field of computer graphics and related technologies. 4. Assumes responsibility for meeting project deadlines and ensuring the quality of implemented solutions.
------------------------------------	---

7. Content

7.1. Course	Teaching methods	Allocated time (hours)
Course 01. Graphics-System Architecture. Raster and vector graphics; CPU-GPU organization; processing, display and input subsystems; color spaces.	The course is taught using a chalkboard, computer, internet, and video projector. The practical demonstration of the concepts presented and problem-solving is done directly in the visual presentation environment. The course is structured as follows: 80% theoretical presentation, based on the course materials; 20% interactive activity (dialogue with students). The necessary materials are provided to students in electronic and/or printed format. The course materials are made available to students in electronic format, and the bibliography is available at the faculty library. The teaching activities are conducted face-to-face, in a physical format.	2
Course 02. Display and Output Technologies. Raster displays, LCD/LED/OLED technologies, resolution, color depth, refresh rate and printing technologies.		2
Course 03. Input and Interaction. Input devices, event processing, interactive graphics loop and application interaction models.		2
Course 04. Graphics Data and the Programmable Pipeline. Meshes, scene graphs, vertex/index buffers, coordinate systems and programmable GPU stages.		2
Course 05. Geometric Transformations and Camera Models. Hierarchical transformations, view and projection matrices, clipping and viewport mapping.		2
Course 06. Rasterization. Line and triangle rasterization, interpolation, polygon filling, sampling and anti-aliasing.		2
Course 07. Texture and Surface Mapping. Texture sampling, filtering, mipmapping, transparency, normal and displacement mapping.		2
Course 08. Visibility and Hidden-Surface Removal. Depth buffering, culling, occlusion and transparency ordering.		2
Course 09. Local Illumination and Materials. Lambert, Phong and Blinn-Phong models; physically based material concepts and shader implementation.		2
Course 10. Global Illumination and Ray Tracing. Shadows, reflections, ray tracing, acceleration structures and integration with raster rendering.		2
Bibliography:		
1. Akenine-Moller, T., Haines, E., and Hoffman, N. Real-Time Rendering, 4th ed., CRC Press, 2018. 2. Marschner, S., and Shirley, P. Fundamentals of Computer Graphics, 5th ed., CRC Press, 2021. 3. Hughes, J. F. et al. Computer Graphics: Principles and Practice, 3rd ed., Addison-Wesley, 2013. 4. Khronos Group. OpenGL 4.6 Core Profile Specification and OpenGL Shading Language Specification, current online versions. 5. Microsoft Learn. Direct3D 12 graphics documentation, current online version. 6. Dogaru, D. Grafica pe calculator, vol. I, Editura Didactica si Pedagogica, 1995 (supplementary historical reference).		

7.2. Laboratory	Teaching methods	Allocated time (hours)
Laboratory 01 Technical specifications for graphic devices: characteristics, performance, types of interfaces.	Laboratory work is performed individually, using practical examples, online documentation, and the development environment, as well as the applications available on the workstations in the laboratory network. For the creation of a 3D real-time interactive	2
Laboratory 02 Building a specification for the graphic system (subsystem) based on application requirements. Examples.		2

Laboratory 03. Modern graphics programming: buffers, shaders, texturing, transparency and lighting using OpenGL or Direct3D.	graphic application, working in teams of two students is allowed (but not mandatory). The teaching activities are conducted face-to-face, in a physical format. Scheduled activities: • 50%-75% performing the laboratory work • 50%-25% tracking the progress of assigned tasks and dialogue with students.	2
Laboratory 04. Hierarchical modeling and animation: transformations, camera control, skeletal hierarchies and movement constraints.		2
Laboratory 05. Materials and advanced surface effects: physically based material parameters, normal mapping and particle systems.		2
Laboratory 06 Creating a sequential (batch) graphic application using the introduced concepts. Assignment of tasks, case analysis, application architecture.		2
Laboratory 07 Creating a sequential (batch) graphic application using the introduced concepts. Implementation, progress check.		2
Laboratory 08 Creating a 3D interactive graphic application – movement and observation within a closed environment, with static actors and actors in programmed movement. Application description, requirements, application architecture.		2
Laboratory 09 Creating a 3D interactive graphic application – movement and observation within a closed environment, with static actors and actors in programmed movement. Implementation, progress check.		2
Laboratory 10 Presentation of the completed applications.		2
Bibliography:		
1. Akenine-Moller, T., Haines, E., and Hoffman, N. Real-Time Rendering, 4th ed., CRC Press, 2018. 2. Marschner, S., and Shirley, P. Fundamentals of Computer Graphics, 5th ed., CRC Press, 2021. 3. Khronos Group. OpenGL and GLSL specifications and reference pages, current online versions. 4. Microsoft Learn. Direct3D 12 graphics documentation, current online version.		

8. Correlating the content of the discipline with the expectations of representatives from the epistemic community, professional associations and representative employers in the field related to the program

The course content is in accordance with the recommendations of representatives from the scientific community, international professional associations (IEEE, ACM SIGGRAPH), and is aligned with certain requirements formulated during discussions with representatives of key employers.

9. Evaluation

Activity Type	9.1. Evaluation criteria	9.2. Evaluation methods	9.3. Final mark weight
9.4. Course	• Understanding the theoretical foundations covered in the course topics; • Ability to solve the most common classes of problems in the field using existing technologies; • Capacity for analysis and synthesis in concrete situations;	Exam: written test. Exam supervision: 2 internal examiners. Condition for exam participation: completion of all laboratory work and submission of all homework.	60%

	Ability to make connections between the taught concepts.	The exam is held in the room assigned by the dean's office, and the laboratory test is generally conducted in the last week of the semester.	
9.5. Laboratory	Laboratory: - Individual work; - Capacity for analysis, identifying the optimal solution, and solving the proposed problems; - Correct completion of the assigned homework; - Acquisition of programming skills and proficiency with specific applications necessary for completing the laboratory work;	Ongoing evaluation of the student's individual performance and completion of all homework assignments. Laboratory grade = $50\% \times \text{grade for the presentation of laboratory program assignments} + 50\% \times \text{grade for individual work during laboratory sessions}.$	40% The final grade is calculated by rounding the final score to an integer value between 1 and 10.
9.6. Minimum standard of performance			
A minimum of 50% of the total score from ongoing assessments and the final exam must be achieved. The final grade is calculated by rounding the final score to the nearest whole number.			

Date of Completion	Signature of the Course Activities Holder,	Signature of the Practical Activities Holder,
01.10.2025	Lect. PhD. Constantin-Renato IVANESCU	

Date of Approval in the Department	Signature of the Department Director,
03.10.2025	Ș.l.dr.ing. Nicolae ENESCU

Date of Approval in the Faculty	Signature of the Dean,
	Prof.dr.ing. Dorin ȘENDRESCU