SYLLABUS

1. Data about the program of study

1.1	Institution	The Technical University of Cluj-Napoca
1.2	Faculty	Faculty of Automation and Computer Science
1.3	Department	Computer Science
1.4	Field of study	Computer Science and Information Technology
1.5	Cycle of study	Bachelor of Science
1.6	Program of study/Qualification	Artificial Intelligence and Vision
1.7	Form of education	Full time
1.8	Subject code	8.1

2. Data about the subject

2.1	Subject name			***Interactive Systems***			
2.2	Subject area						
2.2	Course responsible/lecturer			Prof.dr.eng. Dorian Gorgan, dorian.gorgan@cs.utcluj.ro			
2.3	Teachers in charge of seminars			Prof.dr.eng. Dorian Gorgan, dorian.gorgan@cs.utcluj.ro			
2.4 Year of study	I	2.5 Semester	2	2.6 Assessment	E–**e**xam, C–**c**olloq., V-**v**erif.		E
2.7 Subject category	Formative category: DA – advanced, DS – speciality, DC – complementary						DA
	Optionality: DI – imposed, DO – optional (alternative), DF – optional (free choice)						DO

3. Estimated total time

3.1 Number of hours per week	3	of which	3.2 Course	2	3.3 Seminar	1	3.3 Laborator	-	3.3 Project	-
3.4 Total hours in the curriculum	42	of which	3.5 Course	28	3.6 Seminar	14	3.6 Laborator	-	3.6 Project	-

3.7 Individual study:	
(a) Manual, lecture material and notes, bibliography	20
(b) Supplementary study in the library, online and in the field	10
(c) Preparation for seminars/laboratory works, homework, reports, portfolios, essays	21
(d) Tutoring	4
(e) Exams and tests	3
(f) Other activities	-

3.8 Total hours of individual study (summ (3.7(a)…3.7(f)))	58
3.9 Total hours per semester (3.4+3.8)	100
3.10 Number of credit points	4

4. Pre-requisites (where appropriate)

4.1	Curriculum	Object oriented programming language (e.g., C++, C#)
4.2	Competence	Software application development methodology

5. Requirements (where appropriate)

5.1	For the course	Minimum 75% course attendance for admission to the final exam
5.2	For the seminar	100% attendance is mandatory for admission to the final exam

6. Specific competences

6.1 Professional competences	C1 - Operation with mathematical methods and models, techniques and advanced specific engineering and IT technologies • C1.1 - Knowledge and demonstration of advanced theoretical and practical concepts and principles in the field of software systems. • C1.2 - The use of specific theories and tools (algorithms, schemes, models, tools, etc.) to explain the structure and mode of operation of the latest software technologies, environments and program systems reported in the specialized scientific literature. • C1.3 - The use of specific models and methods for the identification of viable software components and solutions under conditions of partial specification. • C1.4 - Formal and comparative evaluation of the characteristics of software development methods, techniques, and models, as well as complex software systems. • C1.5 - The theoretical foundation of the characteristics of complex software systems, based on modern theoretical and practical trends used in all stages of the software development cycle (specification, analysis, design, implementation, testing and integration, validation). C3 - Specification, analysis, modeling, design, verification, testing, validation, and maintenance of advanced software systems and software components, using domain-specific tools • C3.1 - Demonstrating knowledge of technologies, programming environments, software development CASE tools, and complex program system concepts. • C3.2 - Analysis and explanation of the role, interactions and mode of operation of software components developed on the basis of the latest methodologies for the development of complex software systems proposed in the scientific literature. • C3.3 - Analysis, modeling and innovative design of computing systems and computer applications, related hardware, and software components. • C3.4 - Comparative, synthetic, including experimental evaluation of solution alternatives for performance optimization, based on usability criteria. • C3.5 - Develop and implement original software solutions for domain-specific problems, starting from a set of informally specified requirements.
6.2 Cross competences	N/A

7. Discipline objectives (as results from the *key competences gained*)

7.1	General objective	The main objective of the discipline is the development of interactive applications by acquiring the techniques of analysis, specification, design, implementation, and evaluation of the components that ensure user interaction. Software engineering concepts and techniques specific to user-oriented methodologies are highlighted.
7.2	Specific objectives	To achieve these general objectives, students will learn to: • Design the architecture of interactive software systems. • Use software tools for the development of interactive applications. • Carry out bibliographic and experimental research, the results of which are written up in a scientific paper. • Carry out a scientific synthesis and analysis and, to support an oral presentation of a scientific topic. • Carry out a project according to the methodology of development and evaluation of user-oriented interactive applications. • Work individually or within a team.

8. Contents

8.1. Lecture (syllabus)	Number of hours	Teaching methods	Notes
Introduction. History	2	Multimedia teaching techniques are used in the	Consultations are planned during the
Concepts of user interface development	2		
Inputs and outputs communication concepts	2		

User-oriented design	2	course. The course is interactive with demonstrations, to exemplify the methodology of developing interactive applications.	semester and before each exam.
User interface design methodology	2		
Usability in graphical user interfaces	2		
Defining user requirements. Task description and analysis	2		
User interface prototyping. User interface evaluation	2		
Techniques and styles of interaction	2		
Conceptual architecture of a computer game	2		
Game engines. The architecture of a game engine	2		
Computer games development methodology	2		
Strategies in computer games	2		
The components of a strategy. Strategic options	2		

Bibliography (minimum bibliography of the discipline containing at least one bibliographic reference work of the discipline, which is available to students in a corresponding number of copies)

In the UTC-N library

1. B. Shneiderman, C. Plaisant, M.S. Cohen, S.M. Jacobs, Designing the User Interface. Strategies for Effective Human Computer Interaction, Addison-Wesley (5th Edition).
2. A. Watt, F. Policarpo, 3D Games: Real-time Rendering and Software Technology, Addison-Wesley, 2000.

In virtual library

1. Curs Sisteme Interactive, https://moodle.cs.utcluj.ro/
2. Resurse curs Sisteme Interactive, https://moodle.cs.utcluj.ro/

8.2. Seminars	Number of hours	Teaching methods	Notes
Computer game development methodology.	1	Case studies on topics in the field of interactive applications, examples using software tools and specialized technologies, presentations on the board, additional explanations, discussions.	Each student develops a project based on initial requirements.
Computer game project proposal.	1		
Game analysis and specification. Task description and analysis. Low fidelity prototyping. Game scenarios.	1		
Computer game design. Scene of objects. Game strategy. Interaction techniques.	1		
Technology and Implementation Considerations. Unity technology. Game implementation using Unity technology.	1		
Game rating. Evaluation of functional correctness. Usability evaluation. Heuristic evaluation. Evaluators' reports.	1		
Development of the final version of the game. Conclusions and final presentation of the project. Presentation of game execution. Presentation of the project report.	1		

Bibliography (minimum bibliography of the discipline containing at least one bibliographic reference work of the discipline, which is available to students in a corresponding number of copies)

In the UTC-N library

1. A. Watt, F. Policarpo, 3D Games: Real-time Rendering and Software Technology, Volume 1, Addison-Wesley.
2. A. Watt, F. Policarpo, 3D Games, Volume 2: Animation and Advanced Real-time Rendering, Addison-Wesley.

9. Bridging course contents with the expectations of the representatives of the community, professional associations and employers in the field

Unlike established software engineering, the course presents the user-oriented methodology widely used in the development of interactive applications. Techniques specific to this methodology based on the concept of usability, user scenarios, prototyping, metaphors, cognitive evaluation, heuristic evaluation, multimodal interaction, etc. are studied and exemplified. The content of the discipline was discussed with important actors in this field, both from academia and industry, from Romania or other countries. The discipline has been evaluated by ARACIS, along with other master's study programs.

10. Evaluation

Activity type	10.1 Assessment criteria	10.2 Assessment methods	10.3 Weight in the final grade
10.4 Course	The written exam (E) assesses the knowledge taught in the course. Course activity (CA) reflects active participation in scientific presentations and debates in the course.	Written examination. Assessment during the semester. Online tests using distance education platforms.	30% (E) 10% (CA)
10.5 Seminars	The scientific paper (L) demonstrates the ability to carry out a scientific study and the elaboration of a scientific paper or report. The project (P) demonstrates the ability to use the interactive application development methodology.	Scientific paper, Project. Project presentation onsite or online using distance education platform.	30% (L) 30% (P)
10.6 Minimum standard of performance Final mark: M = 0,3*E + 0,3*L + 0,3*P + 0.1*CA Condition for obtaining credits: M≥5; E ≥ 5; L ≥ 5; P ≥ 5; CA ≥ 5.			

Date of filling in:		**Title Surname Name**	**Signature**
	Lecturer	Prof. dr. eng. Gorgan Dorian	
	Teachers in charge of application	Prof. dr. eng. Gorgan Dorian	

Date of approval in the department 20.02.2024	Head of department Prof.dr.eng. Rodica Potolea
Date of approval in the faculty council 22.02.2024	Dean Prof.dr.eng. Mihaela Dinsoreanu