Subject code: G.2(1)	**Subject name:** Computer Game Programming Basics		
Study load: 4 ECTS	**Load of contact hours:** 60	**Study semester:** Autumn	**Assessment:** Exam
Objectives:	The purpose of the discipline is to study the basic language tools and capabilities of the game application programming tool system, generalize a wide range of practical experience in the field of computer game programming and consolidate the knowledge gained in solving specific problems.		
Course outline:	Topics covered: 1. Introduction to high-performance computing 2. Introduction to video game development tools 3. Introduction to the basics of computer game design 4. Introduction to game engine architecture 5. Introduction to simulation modeling 6. Analysis and development of algorithms 7. Concurrency different view 8. Introduction to the development of applied game artificial intelligence		
Learning Outcomes:	By the end of the course students (in the terms of knowledge, skills, and attitudes) should be able to: 1 – write medium-level game applications; 2 – critically evaluate the architecture of modern game engines and virtual reality systems; 3 – implement applied game artificial intelligence; 4 – critically evaluate computer game design and simulation; 5 – critically evaluate fundamentals of game application architecture design.		
Assessment Methods:	The assessment of knowledge, skills and abilities that characterize the stages of competence formation in the course of studying the discipline is carried out during the current and intermediate attestations. Current certification is conducted in the form of a written and oral survey (individual). Intermediate certification includes theoretical questions that allow you to assess the level of knowledge obtained and the defense of the control work, which allows you to assess the degree of formation of skills. The assessment uses qualitative assessment scales.		
Teacher(s):	Vyacheslav Tarasov		

Prerequisite subject(s):	1. Modern programming technologies. 2. Parallel programming. 3. Algorithms and data structures.Computer Architecture 4. Operating Systems
Compulsory Literature:	1. Jason Gregory, Game Engine Architecture, 2018, A K Peters/CRC Press 2. Georgios N. Yannakakis, Julian Togelius, Artificial Intelligence and Games, 2018, CRC Press
Replacement Literature:	1. John M. Quick. Learn to implement games with code, 2017, CRC Press 2. Colleen Macklin, John Sharp Games, Design, and Play, 2016, Addison-Wesley Professional
Participation requirements:	None.
Independent work:	1. Multithreading and parallel computing. 2. Processing and analysis of data. 3. Smart content generation. 4. Sorting. 5. Algorithms for graph traversal, finding ways. 6. Differential equations in partial derivatives.
Grading criteria scale or the minimal level necessary for passing the subject:	Points distribution: **Excellent** – Sufficient skills: correct and specific answers without major mistakes, several inaccuracies allowed; **Good** – Sufficient skills: correct and specific answers without major mistakes, two or three minor mistakes; **Satisfactory** – General understanding of the subject, several mistakes; **Unsatisfactory** – Insufficient understanding of the subject: wrong answer.
Information about the course:	Room ___, on ___ at ___
1) Date 1	**Lecture 1** Classroom presentation: CPU/GPU development Classroom presentation: multithreaded computing in games, game designer responsibilities

	Homework: Overview of GPU companies
2) Date 2	**Game Programming Workshop 1** Students presentations: Overview of GPU companies Classroom test: CPU/GPU development, multithreaded computing in games (3 points)
3) Date 3	**Lecture 2** Classroom presentation: The graphics rendering Classroom presentation: The design of the game camera Homework: Errors in the design of the game camera
4) Date 4	**Game Programming Workshop 2** Students presentation: Errors in the design of the game camera Classroom test: The graphics rendering, The design of the game camera (3 points)
5) Date 5	**Lecture 3** Classroom presentation: Introduction to video game development tools. Classroom presentation: Modern game engines. Homework: Open source game engines (5 points)
6) Date 6	**Game Programming Workshop 3** Group classroom task: Programming simple games Classroom test: Open source game engines, introduction to video game development tools. (3 points)
7) Date 7	**Lecture 4** Classroom presentation: Introduction to the basics of computer game design and gameCraft Homework: Iterative game design
8) Date 8	**Game Programming Workshop 4** Students presentations: GameCraft Analyses Report (10 points)
9) Date 9	**Lecture 5** Classroom presentation: Prototyping Homework: Prototypes for group projects (5 points)
10) Date 10	**Game Programming Workshop 5** Students presentations: Prototypes for group projects Classroom test: Prototyping methods (3 points)
11) Date 11	**Lecture 6** Classroom presentation: Level design in strategies, shooters, immersive sims Classroom presentation: Level analyses in genre specific games Homework: Level design in stealth games, rouge-like games, adventure games (5 points)
12) Date 12	**Game Programming Workshop 6** Students presentations: Level design in stealth games, rouge-like games, adventure games Classroom test: Level design basics (3 points)
13) Date 13	**Lecture 7** Classroom presentation: Introduction to game engine architecture. Homework: State of parallel execution and data synchronization

14) Date 14	**Game Programming Workshop 7** Classroom test: Game engine architecture (3 points) Students presentations: Unity3d and UnrealEngine
15) Date 15	**Lecture 8** Classroom presentation: Implementing a multithreaded game engine architecture
16) Date 16	**Game Programming Workshop 8** Classroom test: Multithreaded game engine architecture (7 points) Group classroom task: Getting experience working with the game engine
17) Date 17	**Lecture 9** Classroom presentation: Physics, movement, mechanics in game engine Classroom presentation: UI, Coroutines, Animations Homework: applying knowledge in student projects
18) Date 18	**Game Programming Workshop 9** Classroom test: The basics of the game engine Students presentations: Prototyping results
19) Date 19	**Lecture 10** Classroom presentation: Introduction to simulation Homework: Physics modelling
20) Date 20	**Game Programming Workshop 10** Classroom test: Introduction to simulation Students presentations: Physics modelling results
21) Date 21	**Lecture 11** Classroom presentation: 3d models and prefabs Homework: Create simple models and setting behaviour
22) Date 22	**Game Programming Workshop 11** Individual task: Implementation of the collision and destruction model (5 points) Students presentations: Simple models result
23) Date 23	**Lecture 12** Classroom presentation: Light modeling in games, shaders Homework: Implementation of light reflection
24) Date 24	**Game Programming Workshop 12** Students presentations: Light reflection result (5 points)
25) Date 25	**Lecture 13** Classroom presentation: Introduction to the development of applied game artificial intelligence
26) Date 26	**Game Design Workshop 13** Group classroom task: Implementation of artificial intelligence in the game engine. Homework: The main tasks of artificial intelligence in games (10 points)
27) Date 27	**Lecture 14** Classroom presentation: Pathfinder algorithms
28) Date 28	**Game Design Workshop 14**

	Classroom individual task: Algorithm A*
29) Date 29	**Lecture 15** Classroom presentation: Game Build and Publishing Details
30) Date 30	**Game Design Workshop 15** Students presentations: Group projects demonstration (10 points)