National Technical University of Ukraine
“Igor Sikorsky Kyiv Polytechnic Institute”

Department of Computer Engineering

BASICS OF COMPUTER GAMES DEVELOPMENT

Working program educational discipline (Syllabus)

Requisites of the Course

Level of higher education	*First (bachelor’s)*
Field of Study	*12 Informational technologies*
Specialty	*121 Software Engineering*
Education Program	*Computer Systems Software Engineering*
Type of Course	*Normative*
Mode of Studies	*Full-time*
Year of studies, semester	*4th year, autumn semester*
ECTS workload	*4 credits, 120 hours. Lectures 36 hours, Laboratory 18 hours, Self-study 66 hours.*
Testing and assessment	*Final test , MCW*
Course Schedule	*http://roz.kpi.ua/*
Language of Instruction	*English*
Course Instructors	**Lectures and laboratory works:** Senior lecturer of the Department of CE Aleshchenko O.V. alexey _ aleshchenko @ gmail . com
Field of Study	https ://comsys.kpi.ua

Outline of the Course

1. **Description educational discipline, her goal, subject learning and the results teaching**

The goal of teaching the discipline "Fundamentals of computer game development" is for students to master the method and means of programming problems of arbitrary complexity using the game creation methodology. Knowledge of the basics of computer game development is necessary for creating software for computer systems, real-time systems, Internet applications, and mobile devices.

According to the educational program, the discipline "Basics of Computer Games Development" provides the following professional competences (PC) and program learning outcomes (PLO):

- Ability to algorithmic and logical thinking (PC14);
- Ability to develop and use network technologies (PC15);
- Ability to develop and use methods and algorithms of high-performance computing (PC17);
- Ability to develop and use software for high-performance computer systems (PC18);
- Ability to develop and use artificial intelligence systems (PC19);
- Know the methods and algorithms of high-performance computing (PLO27)

In result study academic discipline "Basics of Computer Games Development" the student has

Know:

- methods and algorithms of high-performance computing;
- methods and algorithms of computer graphics;
- basics of object modeling and their interaction

Be able:

- develop, test, release and support computer games;
- choose programming tools and technologies;

- create your own and/or customize existing game engines;
- develop content and game mechanics of games

2. Pre-requisites and post-requisites of the discipline (place in the structural and logical scheme of training according tocorresponding educational program)

Necessary disciplines: "Programming Fundamental", "Algorithms and data structures", "Computer discrete mathematics", "Object-oriented programming", "Networks and network information technologies", "High Performance Systems Software".

For an in-depth study of the aspects of computer game development, optional disciplines can also be useful, in particular, those devoted to computer graphics, geometric modeling, the use of high-performance graphics processors, etc. It can also be useful for a developer of computer games to get acquainted with the discipline "Artificial Intelligence Technologies"

components, which are based on results teaching given disciplines: "Pre-diploma practice", "Diploma design"

3. Content educational disciplines

Chapter 1. Introduction. Roles and positions in development
Chapter 2. Stages of development
Topic 2.1. Designing
Topic 2.2. Production
Topic 2.2.1. Game engine
Topic 2.2.2. Content development
Topic 2.2.3. Development of game mechanics
Topic 2.2.4. Testing
Topic 2.3. Output
Topic 2.4. Support

4. Educational materials and resources

Basic :

1. Design and creation of game applications: a study guide for students of specialties 122 "Computer science" and 151 "Automation and computer-integrated technologies", specialization "System engineering" / I.V. Grebennik, E.V. Gubarenko, O.V. Khryapkin; Ministry of Education and Science of Ukraine, Kharkiv National University of Radio Electronics. - Kharkiv: Khnure, 2018. - 116 p. https://opac.kpi.ua/F/TGBM1YMEAMP6IG62PPLJB5C8I4KHMAMGRS9N597H4GFGXX3RCX-04524?func=full-set-set&set_number=008290&set_entry=000002&format=999
2. Software production and game modeling methods : collective monograph / V.B. Kyselov, V.I. Domnich, M.H. Medvediev, O.M. Muliava ; V.I. Vernadsky Taurida National University. - Lviv :Toruń : Liha-Pres, 2019. - 176 c. https://opac.kpi.ua/F/TGBM1YMEAMP6IG62PPLJB5C8I4KHMAMGRS9N597H4GFGXX3RCX-04402?func=full-set-set&set_number=008290&set_entry=000001&format=999

Additional:

3. Designing computer games for learning: a textbook / T.A. Lugova, O.A. please - Odesa: FOP "Pobuta". - 2018. - 212 p.
4. Computer game development technologies: module guide. / V.C. Breslavets.- X.: "Madrid Printing House", 2018. - 162 p.
5. Basics of computer game development: an electronic study guide for training students at the first (bachelor) level of higher education, field of knowledge 12 "Information technologies", specialty 121 "Software engineering" / Compiler: V.G. Wool – Kherson: V.S. Vyshemyrskyi publishing house, 2018. – 210 p.

Educational content

5. Methodology

Names of sections and topics	Number of hours			
	In total	including		
		Lectures	Laboratory work	Self-study
Chapter 1. Introduction. Roles and positions in development	8	2	-	6
Chapter 2. Stages of development	9	2	-	7
Topic 2.1. Designing	17	6	4	7
Topic 2.2.1. Game engine	2 3	8	6	9
Topic 2.2.2. Content development	1 6	6	4	6
Topic 2.2.3. Development of game mechanics	1 6	6	4	6
Topic 2.2.4. Testing	9	2	-	7
Topic 2.3. Output	9	2	-	7
Topic 2. 4 . Support	9	2	-	7
MCW	2	-	-	2
Test	2	-	-	2
Total in the semester:	1 2 0	36	18	6 6

Lectures

lecture no	The name of the topic of the lecture and the list of main tasks for the self-study
1	**"Introduction. Roles and positions in development".** History of games. The process of creating a video game. Video game developer. T rivalry and cost of development. The complexity of the project. Commercial video game development. Arcade machines and home game consoles. Game designer. Manager. Programmer. Artist. Soundman. Composer. Sound engineer. Screenwriter. Actor. Voice actor. Level designer. PR manager. Tester. **Self-study:** Learn in detail about the roles, positions and history of computer game development.
2	**"Stages of development".** General project. Design document. Description of the universe, gameplay and plot. Graphic concepts of levels and characters. C typology of the game. Landmarks for artists and modelers. Development or customization of a game engine for the needs of the game. Creation of graphic, three-dimensional, and audio content of the game. Implementation and testing of game mechanics (game design). Alpha and Beta versions. Open testing with the participation of players (open alpha/beta tests). Game support. **Self-study:** Learn in detail the stages of video game development. Find examples of design documents for games and game engines.
3	**"Designing".** P re-production. The idea of creating a new game. Project manager. Moscow storm. Attracting players' attention. **Self-study:** Partially complete the first lab of this credit module: come up with an idea for a new game, brainstorm and come up with ways your game could potentially attract players' attention.
4	**"Design (continued)".** Choice of genre. Images of characters and game world. The name of the game. Development team. The timelines for various types of work and the approximate release date of the game. **Self-study:** To improve the performance of the first laboratory work of this credit module: choose the genre of the game, describe the images of the characters and the game world, come up with a name for the game, roughly plan the deadlines for the completion of various types of work.
5	**"Design (continued)".** Concept document. Design document. Prototype. Game process.

	Self-study: Improve the implementation of the first laboratory work of this credit module: develop concept and design documents, prepare a game prototype that would roughly demonstrate the gameplay.
6	**"Game engine".** History and overview of game engines . Unification and systematization of the internal structure of the game. Hardware abstraction. Licenses. Game designer. **Self-study:** Try to choose a game engine or game designer for your own project or decide on the development of your own engine as part of the second laboratory work of this credit module.
7	**"Game mover (continued)".** Development and adjustment of the engine. Multi-platform. Graphical, physical and sound engine. **Self-study:** Partially complete the second laboratory work of this credit module: develop your own or customize an existing game engine, decide on the platforms that will be supported by the engine, select and use graphics, physics and sound engines if necessary.
8	**"Game mover (continued)".** Script system, animation. Game artificial intelligence. **Self-study:** Improve the performance of the second laboratory work of this credit module: integrate the use of the scripting system and game artificial intelligence as necessary.
9	**"Game mover (continued)".** Scene graph. B agate flux. Network code. Memory management. Reusing the engine. **Self-study:** Improve the implementation of the second lab of this credit module: integrate the scene graph, multithreading, network code and memory management as needed. Think about the possibility of reusing the engine in other games.
10	**"Content development".** Creation of graphics, sounds and texts. Character models, items and scenery. **Self-study:** Partially complete the third laboratory work of this credit module: develop or improve graphics (models of characters, objects and scenery) , sounds and texts within the framework of your own game project .
11	**"Content development (continued)".** Animation, sets of movements. sprites Motion capture. Visual effects. Realism. Character states and actions. **Self-study:** Improve the implementation of the third laboratory work of this credit module: add animation and visual effects if necessary.
12	**"Content development (continued)".** Graphic and sound engines. Writing music. Voicing of characters. Sound effects and their libraries. Video inserts. Plot and dialogues. **Self-study:** Improve the implementation of the third laboratory work of this credit module: add music, character voices and sound effects if necessary.
13	**"Development of game mechanics".** Saturation of the gameplay. Rules of the game. And game objects. **Self-study:** Partially complete the fourth laboratory work of this credit module: apply the rules of the game within the framework of your own game project, if necessary .
14	**"Development of game mechanics (continued)".** Management. User interface. Interaction with the game world. Levels (locations). **Self-study:** Improve the implementation of the fourth laboratory work of this credit module: implement or improve management, user interface, interaction with the game world and levels (locations).
15	**"Development of game mechanics (continued)".** Physical driver. Laws of inertia, gravity, behavior of liquids, properties of objects. Artificial Intelligence. Scenarios, scripts. **Self-study:** Improve the implementation of the fourth laboratory work of this credit module: improve physical interaction if necessary, using the laws of inertia, gravity, fluid behavior, properties of objects, and also improve artificial intelligence and scripting scenarios if necessary.
16	**"Testing".** Alpha version. Identifying problems. Testers. Advertising trailer. Game process. Beta version. The correctness of the interaction of objects in the game world. Game balance. Character abilities. **Self-study:** Try to test your own game project, check the correctness of the interaction of

	objects in the game world and game balance.
17	**" Issue ".** Localization of the game. Adaptation to the laws and culture of different countries. Digital distribution. **Self-study:** Try to localize your own game project, familiarize yourself with digital distribution services and the laws and cultural features of other countries that may affect the release of computer games.
18	**"Support".** Difficulty of the game. P patches. Improve. Additions and development of the plot. **Self-study:** Try to add difficulty settings to your own game project and prepare a patch with improvements and additions, development of the plot.

Laboratory classes

No. z/p	The name of the laboratory work	Number of aud. hours
1	Projecting (Topic 2.1)	4
2	Game engine (Topic 2.2.1)	6
3	Content development (Topic 2.2.2)	4
4	Development of game mechanics (Topic 2.2.3)	4
	Total:	18

6. Self-study

No. z/p	The name of the topic submitted for self-study processing	Number of hours
1	**"Introduction. Roles and positions in development".** Familiarize yourself with the roles, positions and history of computer game development in detail.	6
2	**"Stages of development".** Learn more about the stages of video game development. Find examples of design documents for games and game engines.	7
3	**"Designing".** Partially complete the first laboratory work of this credit module: come up with an idea for creating a new game, brainstorm and think about how your game can potentially attract the attention of players.	3
4	**"Design (continued)".** Improve the performance of the first laboratory work of this credit module: choose a game genre, describe the images of the characters and the game world, come up with a name for the game, roughly plan the deadlines for the completion of various types of work.	2
5	**"Design (continued)".** To improve the implementation of the first laboratory work of this credit module: to develop concept and design documents, to prepare a game prototype that would roughly demonstrate the gameplay.	2
6	**"Game engine".** Try to choose a game engine or game designer for your own project or decide on the development of your own engine as part of the second laboratory work of this credit module.	3
7	**"Game mover (continued)".** Partially complete the second laboratory work of this credit module: develop your own or customize an existing game engine, decide on the platforms that will be supported by the engine, select and use graphic, physical and sound engines if necessary.	2
8	**"Game mover (continued)".** Improve the implementation of the second laboratory work of this credit module: integrate the use of the scripting system and game artificial intelligence if necessary.	2
9	**"Game mover (continued)".** Improve the implementation of the second laboratory work of this credit module: integrate the scene graph, multithreading, network code and memory management as necessary. Think about the possibility of reusing the engine in other games.	2

10	**"Content development".** Partially complete the third laboratory work of this credit module: develop or improve graphics (models of characters, objects and scenery) , sounds and texts within the framework of your own game project .	2
11	**"Content development (continued)".** Improve the implementation of the third laboratory work of this credit module: add animation and visual effects if necessary.	2
12	**"Content development (continued)".** Improve the performance of the third laboratory work of this credit module: add music, character voices and sound effects if necessary.	2
13	**"Development of game mechanics".** Partially complete the fourth laboratory work of this credit module: apply the rules of the game within the framework of your own game project, if necessary .	2
14	**"Development of game mechanics (continued)".** Improve the implementation of the fourth laboratory work of this credit module: implement or improve management, user interface, interaction with the game world and levels (locations).	2
15	**"Development of game mechanics (continued)".** Improve the implementation of the fourth laboratory work of this credit module: improve physical interaction if necessary, using the laws of inertia, gravity, fluid behavior, properties of objects, and also improve artificial intelligence and scripting scenarios if necessary.	2
16	**"Testing".** Try to test your own game project, check the correctness of the interaction of objects in the game world and game balance.	7
17	**" Issue ".** Try to localize your own game project, familiarize yourself with digital distribution services and the laws and cultural features of other countries that may affect the release of computer games.	7
18	**"Support".** Try to add difficulty settings to your own game project and prepare a patch with improvements and additions, development of the plot.	7
19	**Preparation for MCW**	2
20	**Preparation for the final test**	2
	Total:	6 6

Policy and CONTROL

7. Policy educational disciplines (educational component)

Rules for the protection of laboratory work:

- the student provides the program code for laboratory work,
- the student defends the provided software code through an interview;

Policy of deadlines and rescheduling:

- the student has the right to submit and resubmit laboratory work up to and including the day of assessment,
- there is no limit on the number of folds;

Academic Integrity Policy:

- the student has the right to protect the program code that was written not only by him, but in this case the student must understand "what" and "how" this code performs, be able to use it as a basis for making modifications that may be suggested by the teacher.

8. Kinds control and rating system assessment results teaching (RSO)

Types of control from the educational discipline "Fundamentals of computer game development" include:

Laboratory works:

Independent performance of six laboratory works is planned. The topics of laboratory works are coordinated in time and content with the topics of lectures. Carrying out laboratory work in full allows you to acquire practical skills in the development of computer games.

Current control:

It is planned to carry out control work (MCW)

Semester control:

Assessment is conducted in the form of an interview with the student to objectively determine the level of knowledge, skills and practical skills acquired during the semester

The student's semester rating consists of the points he receives for the types of work in accordance with Table 1.

Table 1

Assessment of individual types of student academic work (in points)

Type of educational work	**Total by type of work**
Performance and protection of laboratory work No. 1	2 0
Performance and protection of laboratory work No. 2	2 0
Performance and protection of laboratory work No. 3	2 0
Performance and protection of laboratory work No. 4	2 0
MCW **R k**	2 0
Student's semester rating **Rp**	**100**
Credit (R3) as desired	**20**

Student's individual semester rating (**Rp)** consists of the points he gets for performing laboratory work (**Rl**) and MCW (**Rk**). During the semester, students perform 4 laboratory works. The maximum number of points for each laboratory work is 20. Points are awarded for:

- theoretical component – 10 points,
- practical component - 10 points.

The maximum possible score for laboratory work is 20 points. The maximum number of points for all laboratory work is 4 x 20 = 80 points.

Calculation of the scale size (R) of the rating

The sum of the weighted points of control measures during the semester is:

Rп = R л + Rк , where Rп is the semester rating of the student (MCW, laboratory work).

A necessary condition for a student's admission to credit is his individual semester rating (Rp), not less than 40 points, and the absence of debt from laboratory work and M KR. If the mentioned requirements are not met, the student will not be admitted to the credit.

Table of correspondence of rating points to grades on the university scale

Rating	Grades
100-95	Excellent
94-85	Very good
84-75	Good
74-65	Satisfactorily
64-60	Sufficient
Less than 60	Fail
Admission conditions not met	Not allowed

9. Additional information with disciplines (educational component)

As part of the study of the discipline "Basics of Computer Games Development", crediting of points obtained as a result of distance courses is allowed, provided that the program of this course is agreed with the teacher.

Working program of the academic discipline (syllabus):

Designed by a senior lecturer at the Department of Computer Engineering Oleksiy Vadimovych Aleshchenko

Adopted by the Department of Computer Engineering (protocol No. 10 dated 05/25/2022)

Approved by the Methodical Commission of the faculty (protocol No. 10 dated 06/09/2022)