

**Федеральное государственное автономное образовательное учреждение
высшего образования
"Национальный исследовательский университет
"Высшая школа экономики"**

Факультет компьютерных наук
Департамент программной инженерии

**Рабочая программа дисциплины
«Разработка игр» –
на английском языке: «Games Development»**

для образовательной программы «Программная инженерия»
направления подготовки 09.03.04 «Программная инженерия»
уровень - бакалавр

Разработчик программы
Максименкова О.В., старший преподаватель, omaksimenkova@hse.ru

Одобрена на заседании департамента программной инженерии «___»_____ 2017 г.
Руководитель департамента Авдошин С.М. _____

Утверждена Академическим советом образовательной программы
«___»_____ 2017 г., № протокола _____

Академический руководитель образовательной программы
Шилов В.В. _____

Москва, 2017

*Настоящая программа не может быть использована другими подразделениями
университета и другими вузами без разрешения подразделения-разработчика программы.*

1. Course Description

Nowadays, actual bank of digital entertainment technologies is very big. It does not only contain virtual and augmented reality (VR/AR), but also deals with collaboration and socialization and artificial intelligence (AI). This course introduces the games development and entrepreneurship. It includes a systematization of games' genres, a review of approaches to games development, a classification of games' engines. We will study following primary topics: the specifics of games applications lifecycle management (ALM), the determining and modelling of a game user, the transition from a game idea to human-computer interaction design (UX), the developing games for the most popular modern platforms.

A bachelor of software engineering, who works in the field of game development should be familiar with as technological as management features of the gaming business. This course addresses the following questions:

- How we can use electronic devices for entertainment?
- What are the features and constraints of modern devices from the gamer point of view?
- How to use modern development kits in game development?
- Is there a difference between games, which earned millions, and the projects, which were not compensated even their development cost?
- How to evaluate an overall user's satisfaction and to suggest him the best experience on the selected platforms?
- How to integrate various development, ranking, feedback tools in game construction?

2. Title of a Course

Games Development

a. Pre-requisites

(M.2) Applications lifecycle management

(M.2) Object-oriented analysis and programming

(M.2) Algorithms and data structures basics

Java/C#/C++ programming experience is eligible.

b. Course Type

Elective

c. Abstract

This course introduces the games development and elements of software engineering of games. The includes a review of games development approaches and their applications. We will concern on three main topics: the determining and modelling of a game user, software engineering of games for modern platforms, and game development and programming.

3. Learning Objectives

By the end of this course students:

- a) will *know* distinctive properties of gaming and games software construction;
- b) will be able to *define* a gamer profile;
- c) will be able to *define* a software requirement specification (SRS) to a game;
- d) will be able to *select* reasonably a platform and a framework to a game development;
- e) will *know* about the main components of game mechanics and user interaction;
- f) will be able to *develop* a game agreed to SRS;

4. Learning Outcomes

Upon successful completion of this course, students will demonstrate:

- a) an ability to design a game within realistic constraints such as economics, environmental, social, political, ethical, health and safety, manufacturability, and sustainability (ИК-4, ИК-6, ИК-7, ИК-12, ИК-14, ИК-16);
- b) an ability to use development techniques, skills and tools necessary for games development practice (ИК-17);
- c) an ability to design, validate, implement, and maintain games (ИК-9, ИК-17,);
- d) an ability to communicate effectively (СК-Б8, СК-Б9Б СЛК-Б4);
- e) an understanding of professional and ethical responsibility (СЛК-Б1, СЛК-Б2, СЛК-Б9).

5. Course plan

Lectures: 32 academic hours.

Seminars: 32 academic hours.

Self-training: 190 academic hours.

1. Introduction
 - 1.1. History of computer games.
 - 1.2. Gaming industry.
 - 1.3. Modern games platforms.
2. Determining and modelling of a mobile game user
 - 2.1. Game users and their classifications.
 - 2.2. Human psychophysiological characteristics. Perception and cognitive biases.
 - 2.3. Goal-setting and motivation. Gamification, progress and achievements.
 - 2.4. Symbolic systems and applied semiotics. Internationalization and localization problems.
 - 2.5. Behavior and interaction patterns.
 - 2.6. Education and re-education.
 - 2.7. User-oriented design principles.
3. Human computer interaction in games.
 - 3.1. UI/UX. Interaction patterns.
 - 3.2. User centered design.
4. Software engineering of games.
 - 4.1. Games classification.
 - 4.2. Software engineering of games. Games lifecycle management.
 - 4.3. Developing games for modern platforms. Game prototyping.
 - 4.4. Games architectural/programming patterns.
 - 4.5. Game difficulty and balance, in-games metrics.
 - 4.6. Game testing.
 - 4.7. Games analytics.
5. Game engines
 - 5.1. Game engine components.
 - 5.2. Popular frameworks for games development overview. 2D and 3D engines, multiplayer support. Unreal engine, Unity, Corona SDK, Marmalade, Autodesk, ARM, LibGDX.
 - 5.3. Unreal Engine (UE)
 - 5.3.1. UE introduction.
 - 5.3.2. Blueprints visual scripting, workflow, game programming.
 - 5.3.3. Games design and programming using UE.
 - 5.3.4. 2D/3D games programming.

- 5.3.5. VR/AR games programming.
- 6. Game mechanic implementation. Artificial intelligence for single player and multiplayer modes.
- 7. Mobile games development.
- 8. Modern trends in game development.
 - 8.1. Virtual reality, augmented reality, mixed reality.
- 9. Cloud in games programming.

6. Reading list

a. Required

1. Information Resources Management Association (IRMA), *Human-Computer Interaction: Concepts, Methodologies, Tools, and Applications Vol. I-IV*. IGI Global, 2016.
2. Cooper, Kendra and Scacchi, Walt. *Computer games and software engineering*. Taylor & Francis, 2015. Print.
3. Richards, M. *Software Architecture Patterns*, O'Reilly, 2015. Print.
4. Charles P. Schultz & Robert Denton Bryant. *Game Testing: All in One, Third Edition*, Mercury Learning, 2017. Print.
5. Magy Seif El-Nasr, Anders Drachen, Alessandro Canossa. *Game Analytics: Maximizing the Value of Player Data*, IGI Global, 2017. Print.
6. Prototyping Augmented Reality
7. Nystrom, R. *Game programming patterns*. Web. <http://gameprogrammingpatterns.com>
8. Gregory, James. *Game engine architecture*. Web. <http://www.gameenginebook.com>
9. Goodwin, Steven. *Polished Game Development: From First Steps to Final Release*, Apress, 2016. Print.
10. UBM. "Gamasutra. *The Art & Business of Making Games*." Web
11. Berne, Eric. *Games People Play – The Basic Hand Book of Transactional Analysis*. Ballantine, 1964. Print.
12. Kushner, David. *Masters of Doom: How Two Guys Created an Empire and Transformed Pop Culture*. Random House Trade Paperbacks, 2004.
13. Weston, Mark. *The Mobile App Masterplan*. 2014.
14. Habgood, Jacob and Mark Overmars. *The Game Maker's Apprentice: Game Development for Beginners*. Delmar Cengage Learning, 2011.
15. Adams, Ernest. *Fundamentals of Game Design*. 3rd ed. New Riders, 2013.
16. MVA. "Building Blocks: Game Development." Web. https://mva.microsoft.com/en-US/training-courses/building-blocks-game-development-16063?l=OOAAoLZDC_9606218949
17. ESRB. "Entertainment software rating board." Web. <http://www.esrb.org>.
18. IARS. "International age rating coalition". Web. <https://www.globalratings.com>.
19. CTIA. "App Content Classification and Rating Guidelines.". Web. <http://www.ctia.org/policy-initiatives/voluntary-guidelines/app-content-classification-and-ratings-guidelines>

b. Optional

1. Chi (Harold) Liu, Yan Zhang, *Cyber Physical Systems: Architectures, Protocols, and Applications*, CRC Press, 2016.
2. Gregory Kipper, Joseph Rampolla, *Augmented Reality: An Emerging Technologies Guide to AR*. Syngress Publishing, 2013.

3. Fei Hu, *Cyber-Physical Systems: Integrated Computing and Engineering Design*. Auerbach Publications, 2014.
4. Mullen, Tony, *Prototyping Augmented Reality*. Sybex, 2011.
5. Schell, Jesse. *The Art of Game Design: A Deck of Lenses – Cards*. Schell Games, 2008.
6. Horton, John. *Learning Java by Building Android Games – Explore Java Through Mobile Game Development*. Packt Publishing, 2015.
7. Nardone, Massimo, and Vladimir Silva. *Pro Android Games: L Edition*. 3rd ed. Apress, 2015. Print.
8. DeQuadros, Miguel. *Mastering iOS Game Development*. Packt Publishing, 2015. Print.
9. Garney, Chuck. *iOS 9 Game Development Essentials*. Packt Publishing, 2015. Print.
10. Misra, Nitish. *Learning Unreal Engine Android Game Development*. Packt Publishing, 2015. Print.
11. Cordeiro, Jacob. *Gamestar Mechanic for Dummies*, John Wiley & Sons, 2014. Print.
12. High, John, and Jeannie Novak. *Game Development Essentials: Game Project Management*. Course Technology, 2007. Print.
13. Unger, Kimberly, and Jeannie Novak. *Game development essentials: Mobile Game Development*. Course Technology, 2011. Print.
14. Habgood, Jacob, and Nana Nielsen. *The Game Maker's Companion (Technology in Action)*. Apress, 2010. Print.
15. Dunniway, Troy and Jeannie Novak. *Game development essentials: Gameplay Mechanics*. Delmar Cengage Learning, 2008. Print.
16. Juul, Jesper. *Half-Real: Video Games between Real Rules and Fictional Worlds*. The MIT Press, 2005. Print.
17. Salen, Katie and Eric Zimmerman, *Rules of Play: Game Design Fundamentals*, The MIT Press, 2003. Print.
18. Rouse, Richard III. *Game Design: Theory and Practice*. 2nd ed. Jones & Bartlett Learning, 2004. Print.
19. Epic Games. Inc. "Unreal engine. Overview" Web. <<https://www.unrealengine.com/what-is-unreal-engine-4>>
20. Corona Labs. "Corona SDK" Web. <<https://coronalabs.com/products/corona-sdk/>>
21. Marmalade Technologies Ltd. "Marmalade. The developer's platform." Web. <<https://www.madewithmarmalade.com>>
22. Unity Technologies. "Tutorials." Web. <<https://unity3d.com/ru/learn/tutorials>>
23. Autodesk Inc. "Autodesk University." Web. <<http://au.autodesk.com/au-online/overview>>

7. Grading system

Students' final grades are based on the following activities: *home assignments* at seminars, *in-class assignment*, *group projects*.

Ongoing assessment is delivered as *home assignment* (HA), *in-class assignments* (ICA) and *group-projects* (GP). Grades, which are gained by students while ongoing assessment, are parts of the *cumulative grade* (CG). Final grade (FG) is calculated as follows: $FG = CG$.

$$FG = 0,1*HA_{11} + 0,1*HA_{12} + 0,2*ICA_1 + 0,2*GP_1 + 0,1*HA_{21} + 0,3*GP_2$$

Only rounded grades take place in calculations. The arithmetic rules are used. Example, 4,5 -> 5; 4,49 -> 4.

Attendance is graded according to 10-point scale applied in HSE.

10-point scale	Russian grading framework	ECTS grading scheme
----------------	---------------------------	---------------------

10	<i>Excellent</i>	A+	<i>Excellent</i>
9	<i>Excellent</i>	A	<i>Very good</i>
8	<i>Excellent</i>	A–	<i>Very good</i>
7	<i>Good</i>	B+	<i>Good</i>
6	<i>Good</i>	B–	<i>Good</i>
5	<i>Satisfactory</i>	C+	<i>Satisfactory</i>
4	<i>Satisfactory</i>	C–	<i>Satisfactory</i>
3	<i>Fail</i>	F	<i>Fail</i>
2	<i>Fail</i>	F	<i>Fail</i>
1	<i>Fail</i>	F	<i>Fail</i>

8. Guidelines for Knowledge Assessment

Ongoing assessment

Home assignment is an individual programming assignment which is targeted to force students to familiarize with Unreal Engine 4 and help them to shape their skills in the UE4 editor. A student prepares home assignment for 2-3 weeks (this depends on the difficulty of an assignment) and presents the results of his/her work in-class.

Deadline is soft and a student may introduce his/her home assignment for two weeks after the soft deadline. The home assignment evaluates with 50% penalty. There is a *hard deadline* after these two weeks. There is no possibility to gain a mark after the hard deadline.

Not working programming code evaluates as a fail (2).

Group project is implemented as a semester project, which is prepared by a small collaborative team (2 persons).

Each collaborative team makes a short in-class presentation and introduces a game and a report at the end of every stage of the home assignment. The report contains a game idea, a software requirements specification (SRS), roles of team's members with clear definition of personal achievement and the argumentation of project solutions and the results of a stage.

Deadlines are hard.

A teacher evaluates reports and provides each group with a *formative feedback*. Every member of a team gains the same grade.

Members of group, who missed a public presentation does not gain a mark.

Not working programming code evaluates as a fail (2).

In-class assignment is individual assignment. The assignment is implemented in a WebQuest form. The most preferred categories of WebQuests are *design tasks* and *analytical tasks* (for details see WebQuest taxonomy of tasks: <http://webquest.org/sdsu/taskonomy.html>). Students are given the various of WebQuests and in-class prepare a report with project solutions and their justification.

WebQuest Variant Example

You are going to design, develop and promote an educational mobile game for three- and four-year-old children.

You have an obligation to develop your game using *GameSalad for Education*.

WebQuest activities:

1. Study your targeted audience via visiting Academia web site and studying Child-Centered Design of Preschoolers Mobile Educational Games at: https://www.academia.edu/4325529/Child-Centered_Design_of_Preschoolers_Mobile_Educational_Games.
2. Meet your developer's framework. Visit The GameSalad website at: <http://gamesalad.com>.
3. From these sited define a set of keywords and prepare a body of knowledge (links, document) for your project.
4. Prepare a MS Word document with a report about your project design solutions.

Report requirements:

1. *Project name.* Make up a name for your game.
2. *The idea.* In a few words describe the idea of the game.
3. *Targeted audience.* Describe the audience of your application, constraint possible use cases and formulate informal requirements.
4. *First chapters of Software Requirement Specification.* List some functional and nonfunctional requirements.
5. *Project solutions.* List the solutions, which follows from your requirements.
6. *Development tools.* Describe the developer's tools set you are going to use. Describe the features of GameSalad significant to your project.
7. *Game mechanics.* Give a brief description of the solutions about game mechanics you are going to implement in your application.
8. *Project solutions.* List the solutions, which follows from sections 6 and 7.

Assessment schedule

Week	Assessment type
<i>Module 3</i>	
Week 2	HA ₁₁ deadline
Week 4	HA ₁₂ deadline
Week 6-7	In-class assignment (ICA ₁)
Week 10-11	Group project deadline and presentation (GP ₁)
<i>Module 4</i>	
Week 7	HA ₂₁ deadline
Week 10	Group project deadline and presentation (GP ₂)

9. Methods of Instruction

The course combines direct and indirect teaching techniques and contains: lectures, collaborative learning, problem based learning, blended learning, reports, feedback and formative assessment.

Direct teaching is carried out by a lecture method. The preferred informal lectures (lectures with discussions), where students play an active role. The main purpose of lectures is to introduce course's topics, to overview basic information and to discuss and form the directions of further course activities.

Collaborative and problem based learning is implemented through *home assignments* and *group projects*.

Home assignment means the individual work of students on programming assignment in Unreal Engine 4.

In-class activity (seminars) consists of short programming and project assignments, discussions. Students work in pairs (*think share pare technique*) or in small randomly generated groups.

Blended learning is implemented at first through the *formative assessment* in a WebQuest form. The course supposes an *in-class assessment* in a WebQuest form. Moreover, being a powerful constructivists' educational technique, the WebQuest plays a *formative* role. As second, group projects mean as offline students activities and collaboration through collaborative spaces as in-class presentations and feedback from a teacher and the other students.

10. Special Equipment and Software Support

Equipment for in-class and home activity:

1. Classroom multimedia projector;

Software recommended for in-class and home activity:

1. Microsoft Visual Studio 2013 or upper (with mobile device emulators).
2. Epic Games Unreal Engine 4
3. Web-browser with HTML5+JavaScript support.
4. Microsoft Power Point 2010 or upper.

The course is supported by 2017-2018 Microsoft Azure University Grant program.