

**Федеральное государственное автономное образовательное
учреждение высшего образования
«Московский физико-технический институт
(национальный исследовательский университет)»**

УТВЕРЖДЕНО

**Директор высшей школы
программной инженерии
А.В. Малеев**

	Рабочая программа дисциплины (модуля)
по дисциплине:	Разработка компьютерных игр
по направлению:	Программная инженерия
профиль подготовки:	Разработка программно-информационных систем высшая школа программной инженерии высшая школа программной инженерии МФТИ - Яндекс
курс:	4
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 8 (весенний) - Дифференцированный зачет

Аудиторных часов: 60 всего, в том числе:

лекции: 30 час.

семинары: 30 час.

лабораторные занятия: 0 час.

Самостоятельная работа: 48 час.

Всего часов: 108, всего зач. ед.: 3

Количество контрольных работ, заданий: 1

Программу составил: А.В. Созыкин, канд. техн. наук, доцент

Программа обсуждена на заседании высшей школы программной инженерии МФТИ - Яндекс 28.04.2023

Аннотация

Курс содержит основные теоретические и практические аспекты разработки видеоигр с использованием движка Unity. В теоретической части изучаются основы языка C#, проектирование архитектуры игр, алгоритмы для разработки игр. В практической части студент проходит все этапы создания игры: использование движка Unity для разработки игр, проектирование интерфейсов, создание контента для игры: 3D-моделей и анимаций.

1. Цели и задачи

Цель дисциплины

- формирование базовых знаний и навыков в области разработки игр под любые существующие платформы.

Задачи дисциплины

- формирование у обучающихся базовых знаний о процессе создания цифровых продуктов игровой направленности;
- овладение навыками работы с игровым движком Unity;
- освоение методов и алгоритмов проектирования и разработки игровых проектов на движке Unity;
- овладение базовыми знаниями о 3D моделинге;
- освоение методов проектирования и создания игровых интерфейсов.

2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
УК-3 Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде	УК-3.2 Взаимодействует с другими членами команды для достижения поставленной задачи
ОПК-4 Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью	ОПК-4.1 Знает основные правила оформления технической документации на различных стадиях жизненного цикла информационной системы
	ОПК-4.2 Владеет на практике методологией составления научно-технических отчетов (проектов)
ОПК-7 Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой	ОПК-7.2 Понимает принципы работы баз данных и умеет проектировать структуру данных для эффективного хранения информации
ПК-4 Способен разрабатывать тесты, подготавливать тестовые данные, проводить тестирование, разрабатывать документы для тестирования	ПК-4.1 Обладает навыками проведения необходимых видов тестирования в соответствии с планом тестирования
	ПК-4.2 Умеет оценивать важность различных тестов на основе приоритетов пользователя, проектных задач и рисков возникновения ошибки
	ПК-4.4 Умеет выполнять анализ полученных результатов тестирования и оформлять их в соответствии с требуемым форматом
	ПК-4.3 Имеет практический опыт работы с тестовыми средами и системами управления тестированием в своей профессиональной деятельности

3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны знать:

- основы разработки на C#;
- паттерны проектирования игровых проектов в Unity;
- теорию игрового дизайна;
- теорию цвета, композиции, теорию типографик.

уметь:

- создавать игровые проекты в Unity;
- создавать гейм-дизайн-документ;
- проектировать, верстать и анимировать игровые интерфейсы в Figma;
- создавать трехмерные ассеты персонажей и окружения, создавать для них анимации.

владеть:

- методами прототипирования игровых механик;
- методами настройки игрового баланса;
- методами тестирования игровых прототипов на целевой аудитории.

4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Основы языка программирования C#.	5	5		10
2	Разработка игр на Unity.	15	15		18
3	Проектирование игровых интерфейсов.	5	5		10
4	3D моделирование.	5	5		10
Итого часов		30	30		48
Подготовка к экзамену		0 час.			
Общая трудоёмкость		108 час., 3 зач.ед.			

4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 8 (Весенний)

1. Основы языка программирования C#.

Введение в C#. Система типов. Сборка мусора. Сериализация. Асинхронность. Обработка исключений.

2. Разработка игр на Unity.

Введение в C# и Unity3D. Физика и рендеринг. Шейдеры. Анимации и Particle System. Методы оптимизации и профайлинг. Паттерны проектирования. Искусственный интеллект. Мультиплеер.

3. Проектирование игровых интерфейсов.

Теория цвета. Теория композиции. Типографика. Сетки, адаптивный дизайн. Токены и дизайн системы. Прототипирование, отрисовка, анимирование и тестирование игровых интерфейсов. Основы работы в Figma.

4. 3D моделирование.

3D-моделинг, текстурирование и анимирование игровых персонажей и окружения. Экспорт в игровой движок. Основы работы в Blender.

5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

учебная аудитория, оснащенная медиапроектором и экраном. Персональные компьютеры с предустановленным ПО: Unity 3D, Figma, Blender.

6. Перечень рекомендуемой литературы

Основная литература

1. Технологии разработки программного обеспечения, учебник для вузов, стандарт третьего поколения / С. А. Орлов, Б. Я. Цилькер. — Санкт-Петербург, Питер, 2012.— URL: <https://ibooks.ru/bookshelf/28460/reading> (дата обращения: 25.11.2020). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)

Литература, рекомендованная для самостоятельного изучения:

- The Indie Game Developer Handbook - Richard Hill-Whittall
- Level Design: Concept, Theory, and Practice - Rudolf Kremers
- Паттерны программирования игр - Роберт Нистрем

Дополнительная литература

Литература, рекомендованная для самостоятельного изучения:

- Video Game Law: Everything you need to know about Legal and Business Issues in the Game Industry - S. Gregory Boyd, Brian Pye, Sean F. Kane
- Unreal Engine 4 для дизайна и визуализации - Том Шэннон
- Cross Platform Game Programming - Steven Goodwin

7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

Официальный сайт онлайн-курсов по Unity <https://unity.com/ru/learn>

8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)

На лекционных занятиях используются мультимедийные технологии, включая демонстрацию презентаций.

В процессе самостоятельной работы обучающихся предполагается использование таких программных средств, как Figma, Unity, JetBrains Rider, Blender 2.8 и другие.

9. Методические указания для обучающихся по освоению дисциплины (модуля)

успешное освоение курса требует напряжённой самостоятельной работы студента. В программе курса приведено минимально необходимое время для работы студента над темой.

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)

по направлению:	Программная инженерия
профиль подготовки:	Разработка программно-информационных систем высшая школа программной инженерии МФТИ - Яндекс высшая школа программной инженерии
курс:	4
квалификация:	бакалавр

Семестр, формы промежуточной аттестации: 8 (весенний) - Дифференцированный зачет

Разработчик: А.В. Созыкин, канд. техн. наук, доцент

1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
УК-3 Способен осуществлять социальное взаимодействие и реализовывать свою роль в команде	УК-3.2 Взаимодействует с другими членами команды для достижения поставленной задачи
ОПК-4 Способен участвовать в разработке стандартов, норм и правил, а также технической документации, связанной с профессиональной деятельностью	ОПК-4.1 Знает основные правила оформления технической документации на различных стадиях жизненного цикла информационной системы
	ОПК-4.2 Владеет на практике методологией составления научно-технических отчетов (проектов)
ОПК-7 Способен применять в практической деятельности основные концепции, принципы, теории и факты, связанные с информатикой	ОПК-7.2 Понимает принципы работы баз данных и умеет проектировать структуру данных для эффективного хранения информации
ПК-4 Способен разрабатывать тесты, подготавливать тестовые данные, проводить тестирование, разрабатывать документы для тестирования	ПК-4.1 Обладает навыками проведения необходимых видов тестирования в соответствии с планом тестирования
	ПК-4.2 Умеет оценивать важность различных тестов на основе приоритетов пользователя, проектных задач и рисков возникновения ошибки
	ПК-4.4 Умеет выполнять анализ полученных результатов тестирования и оформлять их в соответствии с требуемым форматом
	ПК-4.3 Имеет практический опыт работы с тестовыми средами и системами управления тестированием в своей профессиональной деятельности

2. Показатели оценивания компетенций

В результате изучения дисциплины «Разработка компьютерных игр» обучающийся должен:

знать:

- основы разработки на C#;
- паттерны проектирования игровых проектов в Unity;
- теорию игрового дизайна;
- теорию цвета, композиции, теорию типографик.

уметь:

- создавать игровые проекты в Unity;
- создавать гейм-дизайн-документ;
- проектировать, верстать и анимировать игровые интерфейсы в Figma;
- создавать трехмерные ассеты персонажей и окружения, создавать для них анимации.

владеть:

- методами прототипирования игровых механик;
- методами настройки игрового баланса;
- методами тестирования игровых прототипов на целевой аудитории.

3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

1. CLR, .NET Framework, Mono. Определение и связь с C# и Unity.
2. Проектирование типов: System.Object, приведение типов, пространство имен, примитивные, ссылочные и значимые типы. Ключевые слова dynamic и var.
3. Индексаторы. Расширения.
4. Свойства. События. Обобщения.
5. Делегаты и лямбда-выражения.
6. Настраиваемые атрибуты.

7. Сериализация. Асинхронность.
8. Сборка мусора. IDisposable, Dispose. Финализаторы.
9. Стандартные компоненты Unity. MonoBehaviour.
10. Устройство нативных пользовательских интерфейсов в движке. Обзор возможностей Canvas. Обработка событий инпутов.
11. UnityEditor.
12. 2D и 3D физика и работа с ней: Colliders, Rigidbodies, Physics Materials, Triggers, Raycasting.
13. Character Controller. Raycasts and overlaps.
14. Освещение, рендеринг, тени в Unity. Постобработка. Рендер пайплайн.
15. Шейдеры. ShaderLab и его синтаксис. ShaderGraph.
16. Возможности анимации в движке и дополнительные плагины: Animator, DOTween, Anima2D.
17. Методы оптимизации Unity. Профайлинг, оптимизация рендеринга.
18. Паттерны проектирования игровой разработки. Update-метод. Component. Object Pool. Model–View–Controller.
19. Паттерны проектирования игровой разработки. State Machine. Entity component system.
20. ИИ. Логика и поведение NPC. Pathfinding, NavMesh.
21. Локализация.
22. Аналитика.

4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

Примерный перечень вопросов к дифференцированному зачету:

1. Проектирование типов: System.Object, приведение типов, пространство имен, примитивные, ссылочные и значимые типы. Ключевые слова dynamic и var.
2. Правила юзабилити. Подходы к проектированию
3. CLR, .NET Framework, Mono. Определение и связь с C# и Unity.
4. Цвет, шрифт, сетка. Основные элементы интерфейса.
5. Члены типа, видимость типа, дружественные сборки, доступ к членам типа, статические классы, структуры.
6. Классификации игр.
7. Константы и поля. Конструкторы, перегрузка операторов, расширения.
8. Типы игроков.
9. Свойства и события. Обобщения и интерфейсы.
10. Изначальная мотивация играть.
11. Основные типы данных. Null-совместимые значимые типы.
12. Мотивация продолжать играть.
13. Исключения и управление состояниями.
14. Этапы разработки проекта.
15. Сборщик мусора.
16. Userstory.
17. Сериализация.
18. Этапы создания концепта, фильтрация идей.
19. Асинхронность.
20. Подходы к созданию концепта.
21. Стандартные компоненты Unity. MonoBehaviour.
22. Дизайн систем и подсистем.
23. Устройство нативных пользовательских интерфейсов в движке. Обзор возможностей Canvas. Обработка событий инпутов.
24. Модели монетизации.
25. UnityEditor.
26. Дизайн-документация.
27. 2D и 3D физика и работа с ней: Colliders, Rigidbodies, Physics Materials, Triggers, Raycasting. Постобработка.

28. Сбор и анализ отзывов.
29. Освещение, рендеринг, тени в Unity.
30. Pre-production.
31. Scriptable Rendering Pipeline. LWRP и HDRP.
32. Production.
33. Шейдеры. ShaderLab и его синтаксис. ShaderGraph.
34. Оперирование существующего проекта.
35. Возможности анимации в движке и дополнительные плагины: Animator, State Machines, IK-анимации, Mecanim, DOTween, Anima2D. Particle System.
36. Паттерны проектирования игровой разработки. State Machine. Entity component system.
37. Паттерны проектирования игровой разработки. Update-метод. Component. Object Pool. Model–View–Controller.
38. Работа с ресурсной системой, AssetBundles, атласы. Gameplay Debugging.
39. III. Логика и поведение NPC. Pathfinding, NavMesh.
40. Методы оптимизации U.

Критерии оценивания

отлично (10) - выставляется студенту, показавшему всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

отлично (9) - выставляется студенту, показавшему свободное оперирование знаниями учебной программы дисциплины, выполнение заданий творческого характера.

отлично (8) - выставляется студенту, показавшему владение программным учебным материалом с наличием несущественных ошибок в действиях, самостоятельно исправляемых учащимся.

хорошо (7) - выставляется студенту, если он твердо знает материал, грамотно и по существу излагает его, умеет применять полученные знания на практике, но допускается в ответе или в решении задач некоторые неточности.

хорошо (6) - выставляется студенту если он осознает воспроизведение программного учебного материала, в том числе и различной степени сложности, с несущественными ошибками, затруднения в применении отдельных навыков.

хорошо (5) - выставляется студенту если теоретическое содержание освоено не полностью, некоторые практические навыки сформированы недостаточно, в некоторых случаях были допущены ошибки.

удовлетворительно (4) - выставляется студенту, показавшему фрагментарный, разрозненный характер знаний, недостаточно правильные формулировки базовых понятий, нарушения логической последовательности в изложении программного материала, но при этом он владеет основными разделами учебной программы, необходимыми для дальнейшего обучения и может применять полученные знания по образцу в стандартной ситуации.

удовлетворительно (3) - выставляется студенту в случае большого количества недочетов и неправильных ответов, а также пассивной работе в ходе занятий, многие учебные задания не выполнены.

неудовлетворительно (2) - выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

неудовлетворительно (1) - выставляется студенту, который не освоил теоретическое и практическое содержание курса, все выполненные учебные задания содержат грубые ошибки.

5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Дифференцированный зачет проводится по итогам текущей успеваемости и сдачи заданий и других видов работ, предусмотренных программой дисциплины и (или) путем организации специального опроса, а также выполнением самостоятельной работы и контрольного тестирования.