



**УТВЕРЖДЕНО:**  
Ученым советом факультета сервиса  
Протокол № 3 от «10» октября 2022 г.

**РАБОЧАЯ ПРОГРАММА  
ДИСЦИПЛИНЫ**

***Б1.В.1 Трехмерное моделирование и визуализация пространственных объектов***  
основной профессиональной образовательной программы высшего образования –  
программы  
*магистра*  
по направлению подготовки: *43.03.01 Сервис*  
направленность (профиль): *Геоинформационный сервис*  
Квалификация: *магистр*

**Разработчики:**

должность	ученая степень и звание, ФИО
<i>Доцент Высшей школы сервиса</i>	<i>к.т.н., доцент Минитаева А.М.</i>

**Рабочая программа согласована и одобрена директором ОПОП:**

должность	ученая степень и звание, ФИО
<i>Доцент Высшей школы сервиса</i>	<i>к.т.н. Митрофанов Е.М.</i>



## 1. Аннотация рабочей программы дисциплины (модуля)

Дисциплина Б1.В.1 «Трехмерное моделирование и визуализация пространственных объектов» является частью 1 блока программы магистратуры направления 43.04.01 Сервис, и относится к части, формируемой участниками образовательных отношений.

Дисциплина направлена на формирование следующих компетенций выпускника:

ПК-2 - Способен проектировать трехмерные модели и выполнять визуализацию пространственных объектов; в части индикаторов достижения компетенции ПК-2.1. (Разрабатывает виртуальные среды в интерактивных приложениях Unity, Blender), ПК-2.2. (Выполняет трехмерное моделирование объектов профессиональной деятельности).

Содержание дисциплины охватывает изучение профессиональных навыков использования информационных технологий, которые относятся к сфере геоинформационных технологий для создания виртуальных сред и интерактивных 3D-приложений объектов (ГИО).

Изучению подлежат также подлежат симуляторы продуктов и товаров, эргономического тестирования, маркетинга и рекламы, научных и образовательных приложений которые связаны с информационно-вычислительными объектами, геоинформационных систем и геоинформационных технологий, и их подсистемы.

Изучаются вопросы создания виртуальных сред и интерактивных приложений, освоения современных технологии виртуальной реальности и 3D-визуализации, этапы создания виртуального проекта, принципы создания виртуальных проектов в сочетании с подвижной графикой. Создания виртуального приложения: 3D-графика, фото-, видео-, аудиоматериалы с использованием различных информационных технологий, использования различного средства и инструменты программного трехмерного моделирования, Unity 3D, Blender, для создания интерактивных приложений

Преподавание дисциплины предусматривает проведение учебных занятий следующих видов: лекции, в том числе лекции-традиционная лекция с презентацией, практические занятия в форме практических работ, самостоятельная работа обучающихся, групповые и индивидуальные консультации.

Общая трудоемкость освоения дисциплины составляет 8 зачетных единиц, 288 часов.

Преподавание дисциплины ведется на 2 курсе, 4 семестр, и предусматривает проведение учебных занятий следующих видов: лекции (6 ч), в том числе, традиционные лекции с презентацией, практические занятия в форме выполнения практических работ (12 ч.), самостоятельная работа обучающихся (266 ч.), групповые и индивидуальные консультации (2 ч), промежуточная аттестация (2 ч).

Программой предусмотрены следующие виды контроля:

На заочной форме обучения текущий контроль успеваемости в форме защиты практических работ, тестирования; промежуточная аттестация в форме экзамена в 4 семестре



## 2. Перечень планируемых результатов обучения по дисциплине, соотнесенные с планируемыми результатами освоения образовательной программы

№ пп	Индекс компетенции, индикатора	Планируемые результаты обучения (компетенции, индикатора)
1	ПК-2	Способен проектировать трехмерные модели и выполнять визуализацию пространственных объектов ПК-2.1. Разрабатывает виртуальные среды в интерактивных приложениях Unity, Blender ПК-2.2. Выполняет трехмерное моделирование объектов профессиональной деятельности

## 3. Место дисциплины (модуля) в структуре ОПОП:

Дисциплина «Трехмерное моделирование и визуализация пространственных объектов» входит в часть, формируемую участниками образовательных отношений блока Б1.В.1 по направлению 43.04.01 Сервис профиль «Геоинформационный сервис».

Формирование компетенции ПК-2 способен проектировать трехмерные модели и выполнять визуализацию пространственных объектов, *начинается* в дисциплине «Трехмерное моделирование и визуализация пространственных объектов» во 2 семестре.

Компетенция ПК-2 *заканчивает* формироваться при изучении дисциплин: в 4 семестре и при выполнении ВКР.

## 4. Объем дисциплины (модуля) в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу обучающихся

Общая трудоемкость дисциплины составляет 8/288 зачетных единиц/ акад.часов. <sup>1</sup>

### Для заочной формы обучения:

Виды учебной деятельности	Всего	Семестр
		4
<b>Контактная работа обучающихся с преподавателем</b>	<b>22</b>	<b>22</b>
в том числе:		
Лекции	6	6
Практические занятия	12	12
Консультации	2	2
Промежуточная аттестация	2	2
<b>Самостоятельная работа</b>	<b>266</b>	<b>266</b>
<b>Форма промежуточной аттестации</b>		<b>Экз.</b>
<b>Общая трудоемкость</b>	<b>час</b>	<b>288</b>
	<b>з.е.</b>	<b>8</b>

<sup>1</sup> (1 зачетная единица соответствует 36 академическим часам)

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

**Для заочной формы обучения:**

Номер недели семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения					
			Лекции, акад. Часов	Форма проведения лекции	Практические занятия, акад. часов	Форма проведения практического занятия	СРС, акад. часов	Форма проведения СРС
1/4	1. Построение 3D моделей на Unity 3D, с выводом на голографический стол.	Введение трехмерное моделирование Unity 3D, Blender	0.5	Традиционная с презентацией			66	Изучение лекционного материала. Самостоятельное изучение отдельных тем блока. Подготовка к практическим занятиям
4/4		Создание изображений методом послойной визуализации	0.5					
5,7/4		Создание виртуального 3D тура из серии виртуальных фотопанорам	0.5					
1-7/4		ПЗ 1. Создание сцены в Unity 3D			2	Практическая работа		
8/4		Тестирование. (К.т.№1)			1	тестирование (К.т.№1)		
8/4		Тестирование. (К.т.№2)			1	Тестирование. (К.т.№2)		
9/4		2. Работа в среде Blender	Проанализировать технологию создания виртуальных панорам.	0.5	Традиционная с презентацией			
11/4	Создание материалов виртуального проекта использованием информационных технологий.		0.5					

Номер недели семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения					
			Лекции, акад. Часов	Форма проведения лекции	Практические занятия, акад. часов	Форма проведения практического занятия	СРС, акад. часов	Форма проведения СРС
13-16/4		Создание фотореалистичных изображений методом послойной визуализации	0.5					
9-16/4		ПЗ 2. Создание объектов сцены в Unity 3D			2	Практическая работа		
1,2/4		3. Основные принципы создания виртуальных сред и интерактивных приложений Unity, Blender		0.5	Традиционная с презентацией			66
3/4	Подготовить панораму для возможности размещения ее на сайте.		0.5					
5/4	Осуществление взаимодействия пользователя с программным продуктом		1					
7/4								
9 – 11/4								
13/4								
1-10/4	ПЗ 3. Создание подвижных объектов в Unity 3D			2	Практическая работа			

Номер недели семестра	Наименование раздела	Наименование тем лекций, практических работ, лабораторных работ, семинаров, СРС	Виды учебных занятий и формы их проведения					
			Лекции, акад. Часов	Форма проведения лекции	Практические занятия, акад. часов	Форма проведения практического занятия	СРС, акад. часов	Форма проведения СРС
12-14/4	4. Работа в среде Unity, Blender.		0.5	Традиционная с презентацией			66	Изучение лекционного материала. Самостоятельное изучение отдельных тем блока. Подготовка к практическим занятиям
15-16/2		Тестирование программного продукта.	0,5					
12-16/4		ПЗ 4 Создание скриптов в Unity 3D.			2	Практическая работа		
17/4		Тестирование. (К.т.№3,4)			2	тестирование		
<b>Консультация – 2 часа</b>								
<b>Промежуточная аттестация – экзамен – 2 часа</b>								



**6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)**

№ п/п	Тема, трудоемкость в акад.ч.	Учебно-методическое обеспечение
1	Построение 3D моделей на Unity 3D, с выводом на голографический стол. Создание виртуального 3D тура из серии виртуальных фотопанорам Заочная форма – 66	<b>Основная литература</b> 1. Пушкарева, Т. П. Компьютерный дизайн : учебное пособие / Т. П. Пушкарева, С. А. Титова. - Красноярск : Сиб. федер. ун-т, 2020. - 192 с. - ISBN 978-5-7638-4194-7. - Текст : электронный. - URL: <a href="https://znanium.com/catalog/product/1819273">https://znanium.com/catalog/product/1819273</a> (дата обращения: 16.05.2023). – Режим доступа: по подписке.
2	Работа в среде Blender. Создание материалов виртуального проекта использованием информационных технологий Заочная форма – 68	2. Представление и визуализация результатов научных исследований : учебник / О. С. Логунова, П. Ю. Романов, Л. Г. Егорова, Е. А. Ильина ; под ред. О. С. Логуновой. — Москва : ИНФРА-М, 2020. — 156 с. — (Высшее образование: Аспирантура). - ISBN 978-5-16-014111-4. - Текст : электронный. - URL: <a href="https://znanium.com/catalog/product/1056236">https://znanium.com/catalog/product/1056236</a> 3. Представление и визуализация результатов научных исследований : учебник / О.С. Логунова, П.Ю. Романов, Л.Г. Егорова, Е.А. Ильина ; под ред. О.С. Логуновой. — Москва : ИНФРА-М, 2020. — 156 с. + Доп. материалы [Электронный ресурс]. — (Высшее образование: Аспирантура). — DOI 10.12737/textbook_5c178eb6cf1e63.57981471. - ISBN 978-5-16-014111-4. - Текст : электронный. - URL: <a href="https://znanium.com/catalog/product/1056237">https://znanium.com/catalog/product/1056237</a> (дата обращения: 16.05.2023). – Режим доступа: по подписке.
3	Основные принципы создания виртуальных сред и интерактивных приложений Unity, Blender, Осуществление взаимодействия пользователя с программным продуктом Заочная форма – 66	<b>Дополнительная литература</b> 1. Искусственная световая панорама как выражение планировочной структуры города [Architecture and Modern Information Technologies = Архитектура и современные информационные технологии, №1 (18), 2012, стр. -] - Режим доступа: <a href="http://znanium.com/catalog/product/441151">http://znanium.com/catalog/product/441151</a> Карпенко, В. Е. Формирование световой панорамы прибрежного города (на примере Владивостока) [Электронный ресурс] : диссертация / В. Е. Карпенко. - Владивосток, 2013. - 208 с. - Режим доступа: <a href="https://znanium.com/catalog/document?id=4925">https://znanium.com/catalog/document?id=4925</a>
4	Работа в среде Unity, Blender, Тестирование программного продукта Заочная форма – 66	



## 7. Фонд оценочных средств для проведения текущей и промежуточной аттестации обучающихся по дисциплине (модулю)

### 7.1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

№ п/п	Индекс компетенции, индикатора	Содержание компетенции, индикатора	Раздел дисциплины, обеспечивающий формирование компетенции, индикатора	В результате изучения раздела дисциплины, обеспечивающего формирование компетенции, индикатора обучающийся должен:		
				знать	уметь	владеть
1	ПК-2.	Способен проектировать трехмерные модели и выполнять визуализацию пространственных объектов	Все разделы	Знает теоретические основы трехмерного моделирования в приложениях Unity 3D, Blender	Умеет создавать изображения методом послойной визуализации	Разрабатывает 3D модели на основе серии виртуальных фотопанорам
		ПК-2.1. Разрабатывает виртуальные среды в интерактивных приложениях Unity, Blender				





## 7.2. Описание показателей и критериев оценивания компетенций на разных этапах их формирования, описание шкал оценивания.

Результат обучения по дисциплине	Показатель оценивания	Критерий оценивания	Этап освоения компетенции
Знать теоретические основы трехмерного моделирования в приложениях Unity 3D, Blender. Уметь создавать изображения методом послойной визуализации. Владеть навыками разработки 3D моделей на основе серии виртуальных фотопанорам.	тестирование	Студент демонстрирует теоретические знания основ трехмерного моделирования в приложениях Unity 3D, Blender. Студент демонстрирует умение создавать изображения методом послойной визуализации. Студент демонстрирует владение навыками разработки 3D моделей на основе серии виртуальных фотопанорам.	Способен проектировать трехмерные модели и выполнять визуализацию пространственных объектов

### Критерии и шкала оценивания освоения этапов компетенций на промежуточной аттестации

Порядок, критерии и шкала оценивания освоения этапов компетенций на промежуточной аттестации определяется в соответствии с Положением о проведении текущего контроля успеваемости и промежуточной аттестации обучающихся по образовательным программам высшего образования – программам бакалавриата, программам магистратуры, реализуемым по федеральным государственным образовательным стандартам в ФГБОУ ВО «РГУТИС».

### Виды средств оценивания, применяемых при проведении текущего контроля и шкалы оценки уровня знаний, умений и навыков при выполнении отдельных форм текущего контроля

*Средство оценивания – тестирование*

#### Шкала оценки уровня знаний, умений и навыков при решении тестовых заданий

Критерии оценки	оценка
выполнено верно заданий	«5», если (90 – 100)% правильных ответов
	«4», если (70 – 89)% правильных ответов
	«3», если (50 – 69)% правильных ответов
	«2», если менее 50% правильных ответов

### Виды средств оценивания, применяемых при проведении промежуточной аттестации и шкалы оценки уровня знаний, умений и навыков при их выполнении



Устный опрос

Шкала оценки уровня знаний, умений и навыков при устном ответе

оценка	Критерии оценивания	Показатели оценивания
«5»	<ul style="list-style-type: none"><li>– полно раскрыто содержание материала;</li><li>– материал изложен грамотно, в определенной логической последовательности;</li><li>– продемонстрировано системное и глубокое знание программного материала;</li><li>– точно используется терминология;</li><li>– показано умение иллюстрировать теоретические положения конкретными примерами, применять их в новой ситуации;</li><li>– продемонстрировано усвоение ранее изученных сопутствующих вопросов, сформированность и устойчивость компетенций, умений и навыков;</li><li>– ответ прозвучал самостоятельно, без наводящих вопросов;</li><li>– продемонстрирована способность творчески применять знание теории к решению профессиональных задач;</li><li>– продемонстрировано знание современной учебной и научной литературы;</li><li>– допущены одна – две неточности при освещении второстепенных вопросов, которые исправляются по замечанию</li></ul>	<ul style="list-style-type: none"><li>– Обучающийся показывает все-сторонние и глубокие знания программного материала,</li><li>– знание основной и дополнительной литературы;</li><li>– последовательно и четко отвечает на вопросы билета и дополнительные вопросы;</li><li>– уверенно ориентируется в проблемных ситуациях;</li><li>– демонстрирует способность применять теоретические знания для анализа практических ситуаций, делать правильные выводы, проявляет творческие способности в понимании, изложении и использовании программного материала;</li><li>– подтверждает полное освоение компетенций, предусмотренных программой</li></ul>
«4»	<ul style="list-style-type: none"><li>– вопросы излагаются систематизировано и последовательно;</li><li>– продемонстрировано умение анализировать материал, однако не все выводы носят аргументированный и доказательный характер;</li><li>– продемонстрировано усвоение основной литературы.</li><li>– ответ удовлетворяет в основном требованиям на оценку «5», но при этом имеет один из недостатков:</li><li>– а) в изложении допущены небольшие пробелы, не исказившие со-</li></ul>	<ul style="list-style-type: none"><li>– обучающийся показывает полное знание</li><li>– программного материала, основной и</li><li>– дополнительной литературы;</li><li>– дает полные ответы на теоретические вопросы билета и дополнительные вопросы, допуская некоторые неточности;</li><li>– правильно применяет теоретические положения к оценке практических ситуаций;</li></ul>



	<p>– держание ответа;</p> <p>– б) допущены один – два недочета при освещении основного содержания ответа, исправленные по замечанию преподавателя;</p> <p>– в) допущены ошибка или более двух недочетов при освещении второстепенных вопросов, которые легко исправляются по замечанию преподавателя</p>	<p>– демонстрирует хороший уровень освоения материала и в целом подтверждает освоение компетенций, предусмотренных программой</p>
«3»	<p>– неполно или непоследовательно раскрыто содержание материала, но показано общее понимание вопроса и продемонстрированы умения, достаточные для дальнейшего усвоения материала;</p> <p>– усвоены основные категории по рассматриваемому и дополнительным вопросам;</p> <p>– имелись затруднения или допущены ошибки в определении понятий, использовании терминологии, исправленные после нескольких наводящих вопросов;</p> <p>– при неполном знании теоретического материала выявлена недостаточная сформированность компетенций, умений и навыков, студент не может применить теорию в новой ситуации;</p> <p>– продемонстрировано усвоение основной литературы</p>	<p>– обучающийся показывает знание основного материала в объеме, необходимом для предстоящей основной профессиональной образовательной программы;</p> <p>– при ответе на вопросы билета и дополнительные вопросы не допускает грубых ошибок, но испытывает затруднения в последовательности их изложения;</p> <p>– не в полной мере демонстрирует способность применять теоретические знания для анализа практических ситуаций;</p> <p>– подтверждает освоение компетенций, предусмотренных программой на минимально допустимом уровне</p>
«2»	<p>– не раскрыто основное содержание учебного материала;</p> <p>– обнаружено незнание или непонимание большей или наиболее важной части учебного материала;</p> <p>– допущены ошибки в определении понятий, при использовании терминологии, которые не исправлены после нескольких наводящих вопросов.</p> <p>– не сформированы компетенции, умения и навыки.</p>	<p>– обучающийся имеет существенные пробелы в знаниях основного учебного материала по дисциплине;</p> <p>– не способен аргументировано и последовательно его излагать, допускает грубые ошибки в ответах, неправильно отвечает на задаваемые вопросы или затрудняется с ответом;</p> <p>– не подтверждает освоение компетенций, предусмотренных программой</p>

**7.3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы.**



Номер недели семестра	Раздел дисциплины, обеспечивающий формирование компетенции (или ее части)	Вид и содержание контрольного задания	Требования к выполнению контрольного задания и срокам сдачи
1	Блок 1. контрольные точки 1,2	Тест на выявление уровня освоения теоретических знаний по блоку «Построение 3D моделей на Unity 3D, с выводом на голографический стол.	Контрольная работа. 10 вариантов тестовых заданий. В каждом задании – 5 вопросов, с 5 вариантами ответа, правильный ответ один
2	Блок 2. Контрольные точки 3,4	Тест на выявление уровня освоения теоретических знаний по блоку «Виртуальные туры, панорамы с выводом на шлемы виртуальной реальности»	10 вариантов тестовых заданий В каждом задании – 5 вопросов, с 5 вариантами ответа, правильный ответ один
3	Блок 3. Контрольные точки 1,2	Тест на выявление уровня освоения теоретических знаний по блоку «Работа в среде Blender»	10 вариантов тестовых заданий В каждом задании – 5 вопросов, с 5 вариантами ответа, правильный ответ один
4	Блок 4. контрольные точки 3,4	Тест на выявление уровня освоения теоретических знаний по блоку «Основные принципы создания виртуальных сред и интерактивных приложений Unity, Blender. Работа в среде Unity, Blender»	10 вариантов тестовых заданий В каждом задании – 5 вопросов, с 5 вариантами ответа, правильный ответ один

### Блок первый «Работа в среде и построение 3D моделей на Unity 3D»

1 контрольная точка: Вид контрольного задания - тесты

1. Какие существуют основные компоненты Unity?
  1. значения (Values)
  2. ссылки (References)
  3. Копирование (Copy)

2. Раздел «Анимации» в Importer позволяет различными способами интерпретировать анимации, созданные в вашем приложении для моделирования. В раскрывающемся меню Поколение вы можете выбрать следующие методы?

ответы:

1. Не импортировать: установите модель без анимации



2. Сохранить в исходных корнях: установите модель для отображения анимации на отдельных родительских объектах, поскольку родительские или корневые объекты могут импортироваться по-разному в Unity.
3. Хранить в узлах: установите модель для отображения анимации на отдельных дочерних объектах по всей модели, предоставляя больший контроль сценариев для анимации каждой части.
4. Хранить в корне: установите для модели функцию анимации только на родительском объекте всей группы. 1.
5. Все вышеперечисленное

3. В сети Unity есть \_\_ API высокого уровня?

1. Сценарии
2. язык
3. Инструмент
4. SDK

4. Что из следующего является утверждением ИСТИНА о 2D и 3D объектах?

ответы:

1. 2D и 3D объект не находится в сцене
2. 2D объект находится в сцене и выбирается
3. 2D объект не может быть выбран (в сцене)
4. 3D объект в сцене

5. Используя следующий код, вы можете создать пользовательский интерфейс?

1. 

```
void OnGUI () {
GUI.TextField (новый Rect (Screen.width / 2-150, 500, 600, 600), »текст»);
if (GUI.Button (новый Rect (Screen.width / 2-50, 200, 200, 30), »текст кнопки»)) {
// делать что-то
}
}
```
2. 

```
void OnGUI () {
GUI.BackField (новый Rect (Screen.width / 2-150, 500, 600, 600), »текст»);
if (GUI.Button (новый Rect (Screen.width / 2-50, 200, 200, 30), »текст кнопки»)) {
// делать что-то
}
}
```
3. 

```
void OnGUI () {
GUI.ValueField (новый Rect (Screen.width / 2-150, 500, 600, 600), »текст»);
if (GUI.Button (новый Rect (Screen.width / 2-50, 200, 200, 30), »текст кнопки»)) {
// делать что-то
}
}
```

4. Все вышеперечисленное

6. Как мы можем в Unity изменить размер окна игры?

1. Выберите «Добавить»> «Настройки проекта»> «Игрок 2». Измените ширину и высоту по умолчанию 3. Перейдите в раскрывающееся меню «Вид игры» (как вы делали ранее) и измените его на пользовательский размер.
2. Перейдите в «Просмотр»> «Настройки проекта»> «Игрок 2». Измените ширину и высоту по умолчанию 3. Перейдите в раскрывающееся меню «Вид игры» (как вы делали ранее) и измените его на пользовательский размер.
3. Перейдите в «Редактирование»> «Настройки проекта»> «Игрок 2». Измените ширину и высоту по умолчанию 3. Перейдите в раскрывающееся меню «Вид игры» (как вы это делали ранее) и измените его на произвольный размер.



4. Все вышеперечисленное
7. Класс C # может наследовать только от класса \_\_\_\_\_?
1. One
  2. Base
  3. Parent
  4. Multiple
8. Доступ к данным пользователей из игровых сервисов Google play в Unity?
1. PlayGamesPlatform.Instance.RealTime.GetSelf().Displayname
  2. PlayGames.Instance.RealTime.GetSelf().Displayname
  3. PlayGames.Instance.RealTime.GetSelf().Displayname.GetBlank()
  4. Все вышеперечисленное
9. Unity имеет сложный и мощный \_\_\_\_ для создания и объединения частиц, чтобы дать вам всевозможные спецэффекты?
1. Framework
  2. **System**
  3. Script
  4. Все вышеперечисленное
10. Тени в реальном времени смешиваются с тенями карты света в \_\_\_\_ метрах от основной камеры?
1. 20
  2. 35
  3. 40
  4. 45

2 контрольная точка: Вид контрольного задания - тесты

1. Чему равняется относительная точность d?

1.  $p\left(\frac{|\bar{\Theta} - M[\Theta]|}{M[\Theta]} < d\right)$
2.  $p\left(\frac{|\bar{\Theta} - M[\Theta]|}{\varepsilon} < \varepsilon\right)$
3.  $\overline{M[\Theta]}$

2. Формула  $\Phi(t_\alpha)$  для определения оценки матожидания обозначает:

1. матожидание случайной величины
2. интеграл Лапласа интеграл вероятности

3. Разность N-(n+1) между числом наблюдений N и числом коэффициентов регрессии (n+1) называется:

4. регрессионным анализом
5. корреляционным анализом
6. степенью свободы

4. При каком условии гипотеза об однородности вариантов фактора подтверждается?

1.  $F = F_2$
2.  $F \leq F_2$
3.  $F > F_2$

5. Каким свойством не обладает оценка дисперсии случайной величины  $S^2 = \frac{1}{N} \sum_{i=1}^N (a_i - \bar{a})^2$

1. свойством состоятельности



2. свойством эффективности
3. свойством несмещенности
6. Какая ошибка не относится к ошибкам гипотезы первого и второго рода?
  1. забракованность проверяемой гипотезы, если она верна
  2. принятие проверяемой гипотезы, когда она не верна
  3. отказ от проверки гипотезы
7. Предположение о законе распределения вероятностей случайных величин называется:
  1. гипотезой
  2. матожиданием
  3. критерием согласия
8. Функция  $F = \frac{U}{\sqrt{\frac{k_1}{k_2}}}$  является функцией:
  1. гамма-функцией
  2. хи-квадрата
  3. Фишера
9. Какое условие не соответствует гипотезе о вариантах фактора?
  1.  $F = F_2$
  2.  $F > F_2$
  3.  $F \leq F_2$
10. Оценка  $\bar{P} = \frac{m}{N}$  имеет характеристику:
  1. вероятность события P
  2. дисперсию
  3. коэффициент корреляции
  4. среднее квадратическое отклонение

**Блок второй «Виртуальные туры, панорамы с выводом на шлемы виртуальной реальности»**

3 контрольная точка: Вид контрольного задания - тесты.

Тесты:

1. Используя следующее, вы можете применить установочную позицию дочернего преобразования относительно родительского в Unity?
  1. `hudCamera.transform.parent = hudSelectedObject.transform;`  
`hudCamera.transform.position = new Vector3(0, 0, -50);`
  2. `hudCamera.transform.SetParent(hudSelectedObject.transform);`  
`hudCamera.transform.localScale = Vector3.one;`  
`hudCamera.transform.localPosition = Vector3.zero;`
  3. Both
  4. None of the above
2. Карты нормалей являются средством создания виртуальной геометрии для \_\_\_\_\_ вычислений?
  1. Shader
  2. Rendering
  3. Lighting
  4. Polygons
3. Какие из следующих свойств Light? Примечание: может быть более одного правильного ответа.
  1. Shadow



2. Intensity
3. Range
4. Deferred

4. \_\_ общая концепция для всех 3D-приложений, поскольку они предоставляют средства для настройки визуального вида 3D-модели. От основных цветов до отражающих оснований на изображении поверхностей?

1. Physics
2. Navigation
3. Materials
4. Audio

5. Какое из следующих свойств Axe's для Inputs?

1. Gravity shadow
2. Sensitivity
3. Snap
4. Invert

6. Используя следующий код, вы можете назначить материал второму слоту материала?

1. `CurMaterial = LightObject.GetComponent().materials;`  
`CurMaterial[1] = LightOnMat;`  
`LightObject.GetComponent().materials = CurMaterial;`
2. `LightObject.GetComponent().materials[1] = LightOnMat;`
3. `CurMaterial[1] = LightOnMat;`  
`LightObject.GetComponent().materials[1] = LightOnMat;`
4. All of the above

7. Как мне перечислить требование UnityEvent в интерфейсе C #?

1. 

```
public interface IPushNotificationService
{
    UnityEvent pushNotificationEvent;
    void Initialize();
    void PushNotificationHandler(string message, Dictionary<string, object> additionalData, bool isActive);
}
```
2. 

```
public interface IPushNotificationService
{
    UnityEvent pushNotificationEvent;
    void Initialize();
    void PushNotificationHandler(string message, Dictionary<string, object> additionalData, bool isActive);
}
```
3. 

```
public interface IBlah {
    UnityEvent thisEventHasToBeImplemented { get; }
}
public class Blah : IBlah {
    UnityEvent _event = new UnityEvent();
    UnityEvent thisEventHasToBeImplemented {
        get { return _event; }
    }
}
```





}

}

4. All of the above

8. Используя следующий код, вы будете воспроизводить Animation A после завершения Animation B?

1. `gameObject.animation.Play(«Once»);`

`AnimationPlayed = true;`

2. `if (AnimationPlayed && !gameObject.animation.isPlaying(«Once»){`

`gameObject.animation.Play(«Loop»);`

`AnimationPlayed = false;`

`}`

3. `addObject.animation.Play(«Once»);`

`AnimationPlayed = true;`

4. All of the above

9. Эта опция удаляет все сохраненные в настоящее время карты освещения, позволяя вам видеть только динамическое освещение и исходную яркость материалов и текстуры объекта?

1. Clear

2. Bake Selected

3. Bake

4. All of the above

10. Какие из следующих типов касательных обеспечивают гладкость?

1. Free Smooth

2. Flater

3. Linear

4. Constance

4 контрольная точка: Вид контрольного задания - тесты.

Тесты:

1. Какое моделирование основано на применении моделей, представляющих собой реальные технические конструкции?

5. имитационное

6. материальное

7. абстрактное

2. Какие модели отображают процессы, в которых отсутствуют случайные воздействия?

1. детерминированные

2. дискретно-непрерывные

3. абстрактные

3. На каком этапе моделирования идет уяснение целей моделирования?

1. на третьем

2. на втором

3. на первом

4. Какие цели, из ниже перечисленных относятся к целям моделирования?

1. подбор сочетания и значений факторов

2. прогноз поведения объекта при новых режимах

3. проверка различного рода гипотез



5. Моделирование — это:

1. замещения одного объекта другим с целью получения информации о важнейших свойствах объекта-оригинала
2. материальный объект той или иной природы по отношению к оригиналу
3. создание определено новой модели для тестирования какого-либо объекта

6. Какое моделирование выполняет процесс построения и изучения математических моделей?

1. математическое
2. аналитическое
3. имитационное

7. Какое моделирование предполагает представление модели в виде некоторого алгоритма - компьютерной программы?

1. аналитическое
2. смешанное
3. имитационное

8. Промежуточный объект между процессом моделирования и оригиналом называется:

1. материальным объектом
2. объект-оригинал
3. моделью

9. На каком этапе моделирования идет выбор языка программирования или моделирования?

1. на третьем
2. на втором
3. на четвертом

10. К каким признакам классификации не относятся абстрактные модели?

1. характер моделируемой стороны объекта
2. характер процессов, протекающих в объекте
3. способ реализации

11. Выберите не верное утверждение:

1. однородные заявки имеют разные права на начало обслуживания
2. число уравнений в системе равно числу состояний
3. одно из свойств простейшего потока - это отсутствие последствий

12. К какой системе массового обслуживания относится следующая задача?

В порту имеется один причал для разгрузки судов. Интенсивность потока судов равна 0,4 (судов в сутки). Среднее время разгрузки одного судна составляет 2 суток. Найти показатели эффективности работы причала, если известно, что приходящее судно покидает причал, если в очереди на разгрузку стоит более 3 судов.

1. многоканальная СМО с ограниченной очередью
2. одноканальная СМО с ожиданием
3. одноканальная СМО с неограниченной очередью

### Блок третий «Работа в среде Blender 3D»

Вид контрольного задания - тесты.

Тесты:

1. Освещение: в 3D основное non-GI выполняется на основе \_\_\_\_\_ ?
  1. Vertex
  2. Invert
  3. GUI
  4. Angle



2. Blender – это

1. пакет для создания трёхмерной компьютерной графики, анимации и интерактивных приложений
2. графический редактор
3. текстовый редактор
4. программная среда для объектно-ориентированного программирования

3. Окно blender состоит из трёх дочерних окон:

1. меню, окно 3D вида, панель кнопок
2. строка заголовка, панель инструментов, рабочая область
3. меню, панель инструментов, рабочая область
4. окно запуска программы, строка состояния, окно задач

4. Объекты сцены:

1. квадрат, лупа, курсор
2. куб, лампа, камера
3. куб, шар, цилиндр
4. окно, лампа, камера

5. Рендер является

1. графическим редактором
2. графическим отображением 3D сцены или объекта
3. источником света
4. отображением осей координат

6. Лампа является

1. графическим редактором
2. графическим отображением 3D сцены или объекта
3. источником света
4. отображением осей координат

7. Клавиша F12 служит для

1. рендеринга
2. вида сверху
3. поворота сцены
4. изменения масштаба

8. Клавиша 7 (NumPad) служит для

1. рендеринга
2. вида сверху
3. поворота сцены
4. изменения масштаба

9. Клавиша 5 (NumPad) служит для

1. рендеринга
2. перспективы
3. текстурирования
4. масштабирования

Клавиша 1 (NumPad) служит для

1. вида спереди
2. вида сверху



3. поворота сцены
4. изменения масштаба

2 контрольная точка: Вид контрольного задания - тесты

Тесты:

1. Что означает  $P_j(k)$  в рекуррентной зависимости:  $P_j(k)$ ?

1. вероятность j-го состояния системы после k-го шага переходные вероятности
2. вероятность i-го состояния системы после (k-1)-го шага

3. В формуле  $\lambda_{ij} = \lim_{\Delta t \rightarrow 0} \frac{P_{ij}(\Delta t)}{\Delta t}$  плотностью вероятностей переходов будет:

1.  $\lim_{\Delta t \rightarrow 0}$
2.  $\lambda_{ij}$
3.  $P_{ij}(\Delta t)$

4. Какую зависимость, из ниже перечисленных, применяют для нахождения вероятностей состояния неоднородной марковской цепи?

1.  $P_j(k) = \sum_{i=1}^n p_i(k-1) \cdot p_{ij}^{(k)}$
2.  $P_j(k) = \sum_{i=1}^n p_i(k-1) \cdot p_{ij}$
3.  $P_i(k, j) = \sum_{j=1}^k p_i(k-1) \cdot p_i$

5. Какая теорема, из ниже приведенных, является истиной теоремой Маркова?

1. если для однородного дискретного марковского процесса с конечным или счетным состояний все  $P_{ij} > 0$ , то предельные значения  $P_j(k)$  существуют и их значения не зависят от выбранного начального состояния системы
2. если для однородного дискретного марковского процесса с бесконечным числом состояний все  $P_{ij} > 0$ , то предельные значения  $P_j(k)$  существуют и их значения зависят от выбранного начального состояния системы.
3. если для неоднородного детерминированного марковского процесса с конечным или счетным состояний все  $P_{ij} > 0$ , то предельные значения  $P_j(k)$  существуют и их значения зависят от выбранного начального состояния системы.

6. Случайный процесс, при котором смена дискретных состояний происходит в случайные моменты времени называют:

1. дискретно-непрерывным марковским процессом
2. непрерывным марковским процессом
3. детерминированным марковским процессом

7. Если переходные вероятности не зависят от времени, то это:

1. однородная марковская цепь
2. стохастическая марковская цепь
3. непрерывная марковская цепь

8. Случайный процесс, при котором смена дискретных состояний происходит в определенные моменты времени называют:

1. динамической марковской цепью
2. дискретной марковской цепью
3. стохастической марковской цепью

9. Система будет многоканальной, если:

1.  $n > 0$
2.  $n = 1$
3.  $n < 1$

10. Выберите верное утверждение:



1. марковская цепь называется стохастической, если переходные вероятности зависят от времени
  2. вероятность «перескока» системы из одного состояния в другое точно в момент времени  $t$  равна 1
  3. любой случайный процесс может быть сведен к марковскому
11. СМО будет с потерями, если:
1.  $m > 1$
  2.  $m = 0$
  3.  $n > 0$
12. Случайный процесс — это:
1. функция времени  $x_i(t)$ , описывающая течение процесса в некотором  $i$ -м опыте
  2. случайная величина  $x(t_j)$ , являющаяся значением случайного процесса в фиксированный момент времени
  3. случайная величина  $X(t)$ , зависящая от одного неслучайного вещественного аргумента  $t$ .
13. При каком способе, из ниже перечисленных, случайные числа формируются специальным устройством?
1. табличный способ
  2. аппаратный способ
  3. алгоритмический способ
14. Реализация случайного процесса — это:
1. сечения случайного процесса неоднородны в вероятностном смысле
  2. функция времени  $x_i(t)$ , описывающая течение процесса в некотором  $i$ -м опыте
  3. случайная величина  $x(t_j)$ , являющаяся значением случайного процесса в фиксированный момент времени
15. Какой модуль, из ниже перечисленных, представляет собой программы, имитирующие действия соответствующих активных событий и связанных с ними пассивных событий?
1. модуль формирования результата
  2. модуль реакции
  3. модуль продвижения модельного времени
16. Целью имитационного моделирования является:
1. определение показателей эффективности различных операций
  2. определение непрерывно равномерно распределенной случайной величины
  3. реализация случайного процесса
17. Если элемент системы может находиться во многих несовместных состояниях, то такие события называют:
1. определением исходов по жребию
  2. полной группой несовместных событий
  3. равномерным распределением

Блок четвертый «Основные принципы создания виртуальных сред и интерактивных приложений Unity, Blender»

1. Основной 3D меш-объект
  1. куб
  2. икосаэдр
  3. тор
  4. сфера



2.К меш-объектам относятся

1. куб, сфера, окружность, плоскость
2. цилиндр, кольцо, отрезок, вектор
3. цилиндр, конус, додекаэдр, параллелограмм
4. куб, сфера, прямоугольник, плоскость

3.Окно редактор нодов (свойств объектов)

1. служит для настройки применяемых эффектов при рендеринге
2. появляется автоматически при сохранении файла или картинки
3. служит для отображения конечного изображения
4. используется для просмотра и работы с моделями

4.Обозреватель Файлов/Картинок

1. служит для настройки применяемых эффектов при рендеринге
2. появляется автоматически при сохранении файла или картинки
3. служит для отображения конечного изображения
4. используется для просмотра и работы с моделями

5.Правая кнопка используется для

1. выбора объектов (или вершин в режиме Редактирования)
2. перемещение трехмерного курсора
3. выбора инструмента заливки
4. включения Лампы в режиме Объекта

6. Изображение рендеринга сохраняется

1. в формате объекта blender
2. в формате изображения jpeg
3. объектный программный код
4. в формате текстового файла

7. В Unity, Как вы можете изменить настройку модели форпоста?

1. Under Meshes—Scale Factor is set to 1.5, and Generate Colliders and Generate Lightmap UVs are selected
2. Under Materials—Generation is set to Per Texture
3. Under Animations—Split Animations is selected
4. All of the above

8. Что из перечисленного является конвейером рендеринга Unity?

```
1. void Update () {  
  if ((Input.GetButtonUp(«M»))) {  
    GameObject go = EventSystem.current.currentSelectedGameObject;  
    InputField inputField = null; //creating dummy, null, InputField component  
    if (go!= null) {  
      inputField = go.GetComponent();  
    }  
    if (inputField==null) {Globals.lockmouse = !Globals.lockmouse;}  
  }  
}
```



- ```
2. public interface IBlah {
    UnityEvent thisEventHasToBeImplemented { get; }
}
public class Blah : IBlah {
    UnityEvent _event = new UnityEvent();
    UnityEvent thisEventHasToBeImplemented {
    get { return _event; }
}
}
```
3. Both  
4. None of the above

9. Система пользовательского интерфейса позволяет создавать \_\_\_\_\_ быстро и интуитивно. Это введение в основные функции системы пользовательского интерфейса Unity?

1. user interfaces
2. audio interfaces
3. Users interfaces
4. All of the above

10. Аудиоклип с именем \_\_, который будет воспроизведен игроком во время сбора

ячейки в Book Assets | Sounds folder?

1. cell\_collected
2. hud\_charge
3. power\_cell
4. All of the above

4 контрольная точка: Вид контрольного задания - тесты:

1. Какой модуль, из ниже перечисленных, устанавливает значения разного рода констант?

2. модуль формирования результата
3. модуль реакции
4. модуль установки начальных условий

2. Сечение случайного процесса  $X(t)$  — это:

1. случайная величина  $x(t_j)$ , являющаяся значением случайного процесса в фиксированный момент времени
2. функция времени  $x_i(t)$ , описывающая течение процесса в некотором  $i$ -м опыте.
3. сечения случайного процесса неоднородны в вероятностном смысле

3. Если совместные независимые события сводятся к одному сложному событию, то это:

1. моделирование активных событий
2. моделирование пассивных событий
3. моделирование совместных независимых событий

4. Какие языки программирования, из ниже перечисленных, не являются языками моделирования?

1. симпас
2. модула
3. C++
4. GPSS

5. Достижение заданной точности  $P(|\bar{\theta} - M[\theta]| < \epsilon)$  называется:



1. достоверностью оценки
  2. результативностью оценки
  3. характеристикой оценки
6. Объекты: выходные переменные, входные переменные и уровни факторов являются объектами:
1. комплекс по планированию
  2. стратегического планирования
  3. тактического планирования
7. В какой формуле, из ниже перечисленных, высчитывается оценка дисперсии  $\Theta$ ?
1.  $S_{\Theta}^2 = \frac{1}{n^0 - 1} \sum_{k=1}^{n^0} (\Theta_k - \bar{\Theta})^2$
  2.  $S_q^2 = \frac{1}{n^0 - 1} \sum_{k=1}^{n^0} (q_k - \bar{q})^2$
  3.  $r_{\Theta, q} = \frac{1}{n^* - 1} \sum_{k=1}^{n^*} (\Theta_k - \bar{\Theta})(q_k - \bar{q})$
8. Если необходимое число циклов определено и выполняется неравенство  $n = n_{\text{зад}}$ :
1. то моделирование заканчивается и при необходимости выдается оценка достигнутой точности
  2. то моделирование продолжается до достижения  $n$  циклов
  3. признак конца моделирования
9. Какой вид, из ниже перечисленных, будет иметь достоверность оценки?
1.  $p\left(\frac{|\bar{\Theta} - M[\Theta]|}{M[\Theta]} < d\right)$
  2.  $p(|\bar{\Theta} - M[\Theta]| < \varepsilon)$
  3.  $pq^k$
10. Эксперимент это:
1. процесс изучения, эксперимента, концептуализации и проверки теории, связанный с получением научных знаний
  2. метод исследования некоторого явления в управляемых условиях
  3. исследовательский метод, заключающийся в целенаправленном и организованном восприятии и регистрации поведения изучаемого объекта
11. Что означает совместимость факторов?
1. должны быть осуществимы все комбинации значений факторов
  2. возможность установки и поддержания значения фактора постоянным или изменяющимся в соответствии с планом эксперимента
  3. определяет возможность установления значения фактора на любом уровне
12. Что означает оператор  $f(x)$  в кибернетическом представлении эксперимента?
1. входные переменные
  2. выходная переменная
  3. оператор моделирующий действия реальной системы
13. Если необходимое число циклов определено и выполняется неравенство  $n \leq n^*$ :
1. признак конца моделирования
  2. то моделирование продолжается до достижения  $n$  циклов
  3. то моделирование заканчивается и при необходимости выдается оценка достигнутой точности

### Контрольные вопросы к экзамену

1. Основные задачи компьютерной графики.
2. Виды графических систем. Основные достоинства и недостатки.
3. Графические системы с векторным сканированием.





4. Растровые графические системы. Основные характеристики растра.
5. Растровые графические системы. Построчная и чересстрочная развертки растра.
6. Форматы графических файлов.
7. Векторные форматы графических файлов. Основные достоинства и недостатки.
8. Растровые форматы графических файлов. Основные достоинства и недостатки.
9. Методы сжатия растровых файлов.
10. Векторные и растровые прикладные графические редакторы. Области применения.
11. Аддитивная цветовая модель RGB.
12. Субтрактивная цветовая модель CMY, CMYK.
13. Цветовая модель HSB.
14. Базовые растровые алгоритмы. Основные решаемые задачи. Понятие связности.
15. Растровое представление отрезка. Алгоритм Брезенхэма.
16. Устранение ступенчатого эффекта в растровых изображениях.
17. Заполнение области (закрашивание).
18. Закрашивание многоугольников, заданных своими вершинами.
19. Отсечение многоугольников относительно видимого окна.
20. Аффинные преобразования на плоскости. Основные частные случаи. Применение однородных координат для матричной формы записи уравнений аффинных преобразований.
21. Аффинные преобразования в пространстве. Основные частные случаи. Композиция преобразований.
22. Проецирование. Виды плоских геометрических проекций.
23. Виды параллельных проекций. Искажения объекта при параллельном проецировании.
24. Ортографическая проекция.
25. Аксонометрические проекции.
26. Косоугольные проекции.
27. Перспективные (центральные) проекции.
28. В чем заключается сущность корреляционного анализа?
29. Каким свойством обладает оценка математического ожидания случайной величины  $\bar{a} = \frac{\sum_{i=1}^N a_i}{N}$ ?
30. В блоке SEIZE A операнд A применяется для:
31. Выберите не верные утверждения:
32. В блоке ADVANCE A,[B] операнд B применяется:
33. Какой режим блока TRANSFER является параметрическим?
34. Каким блоком проверяется состояние МКУ?
35. Какое действие произойдет при выполнении записи ADVANCE Ring, FN\$Exper?
36. Какие блоки, из ниже перечисленных, служат для организации списка пользователя?
37. Список транзактов, ожидающих занятия ОКУ в порядке приоритета называют:
38. Какие операторы связаны с объектами аппаратной категории и используются для определения состояния этих объектов?
39. В какой список помещаются транзакты, которые не удовлетворяют всем заданным условиям входа в блок?
40. В какой список попадает список транзактов в порядке приоритета, ожидающих освобождения элементов памяти?
41. Совокупность методов построения и исследования регрессионной зависимости между величинами по статистическим данным называется:
42. Выберите не верные определения:



43. В основе какого моделирования элемент системы может находиться во многих несовместных состояниях?
44. В основе какого моделирования лежит метод обратной функции?

### Тестовые вопросы к экзамену

#### *Задания закрытого типа*

- 1. Какое из перечисленных преимуществ лазерного сканирования вытекает из того факта, что это активная система дистанционного зондирования?**
1. Можно работать в условиях низкой освещенности.
  2. Высокая точность угловых измерений.
  3. Возможность писать в процессе съемки трек движения.
- 2. Какая из перечисленных компонент оборудованного панорамной цифровой фотокамерой стационарного лазерного сканера хуже всего будет работать в условиях низкой освещенности?**
1. Цифровая фотокамера.
  2. Инерциальная система.
  3. Сканирующая головка.
- 3. Мобильный лазерный сканер в общем случае имеет преимущества перед стационарным наземным лазерным сканером при работе на лесных объектах потому, что ...**
1. Упрощает процесс съемки объекта и более эффективен при работе в сложных условиях.
  2. Дает большую точность измерений.
  3. Не имеет преимуществ.
- 4. При использовании мобильного лазерного сканера в сложных условиях съемки без явных контуров при уравнивании следует отдавать предпочтение методам, определяющим положение прибора в пространстве, на основе ...**
1. Внутренней инерциальной системы сканера.
  2. Обратной фотограмметрической засечки.
  3. Метода наименьших квадратов.
- 5. Стационарный лазерный сканер имеет преимущества перед мобильным в ситуации, когда нужно выполнить съемку фасада за счет того, что ....**
1. Дает большую точность определения геометрических характеристик объекта.
  2. Позволяет выполнить работу быстрее.
  3. Имеет панорамную камеру.
- 6. Авиационный лазерный сканер имеет преимущества перед мобильным лазерным сканером при съемке рельефа по причине ...**
1. Высокой производительности.
  2. Большей точности угловых измерений.
  3. Простоте применения.
- 7. Для какой задачи используется проверка FPS при работе с трехмерными измерительными моделями различно степени сложности?**



1. Определите степени эффективности визуализации объекта в процессе обработки.
2. Для увеличения качества математической обработки сцены.
3. В качестве «бенчмарка» для ЭВМ.

**8. В каких ситуациях при работе с трехмерными измерительными сценами удобно выполнять их клонирование?**

1. Обработка требует эмпирического подбора параметров обработки на данной стадии.
2. Требуется сравнить результат операции с исходным набором данных.
3. Оба варианта верны.

**9. Если мы работаем с трехмерной измерительной сценой с целью построения рельефа местности, оправданно ли выполнять прореживание всего плотного облака?**

1. Да, если это не сказывается на геометрии поверхности земли и ускоряет процесс обработки.
2. Нет.
3. Да, обязательно.

**10. Если мы работаем с трехмерной измерительной сценой с целью построения точной высотной модели древостоя, оправданно ли выполнять прореживание всего плотного облака?**

1. Нет – будет потеряна информация на верхушках отдельных деревьев.
2. Да, любым алгоритмом.
3. Да, но только алгоритмом Random.

**11. При создании эталонной трехмерной модели дерева, какую его часть можно чистить с использованием автоматизированных фильтров в общем случае?**

1. Ствол.
2. Крона.
3. Точка комутовки.

**12. Если в процессе привязки плотного облака точек мы пользуемся точками, координаты которых были определены на местности заранее то это ...**

1. Привязка по контрольным точкам.
2. Привязка по связующим точкам.
3. Привязка с использованием метода подобия.

**13. Если в процессе привязки одного плотного облака к другому-референтному точек мы пользуемся точки, которые можно определить на обоих облаках, то это..**

1. Привязка по контрольным точкам.
2. Привязка по связующим точкам.
3. Привязка с использованием метода подобия.

**14. Цифровая модель рельефа, выраженная в виде плотного облака точек, по своей сути является ...**

1. Дискретной.
2. Интегральной.
3. Фрактальной.

**15. Цифровая модель рельефа, выраженная в виде MESH-поверхности точек, является ...**



1. Дискретной.
2. Интегральной.
3. Фрактальной.

### *Задания открытого типа*

1. Какое обязательное условие нужно соблюдать для точного измерений высот отдельных деревьев по плотному облаку, полученному в результате авиационного лазерного сканирования?
2. Какое необходимое условие необходимо соблюсти, если мы хотим записать в каждую точку плотного облака о цвете, которой она имеет в видимом диапазоне?
3. Какой тип лазерного сканера (стационарный, мобильный наземный, авиационный) наиболее эффективно применять, когда перед инженерами стоит задача съемки больших лесных площадей?
4. Какой тип лазерного сканера (стационарный наземный, мобильный наземный, авиационный) наиболее эффективно применять, когда перед инженерами стоит задача съемки фасада здания с высокой точностью для оценки его структуры?
5. Какой тип лазерного сканера (стационарный наземный, мобильный наземный, авиационный) наиболее эффективно применять, когда перед инженерами стоит задача съемки извилистого туннеля?
6. Какой тип лазерного сканера (стационарный, мобильный наземный, авиационный) наиболее эффективно применять, когда перед инженерами съемки трубопровода длиной 5 километров более?
7. Какой тип лазерного сканера (стационарный наземный, мобильный наземный, авиационный) наиболее эффективно применять, когда перед инженерами стоит высокоточного обмера земельного участка с постройками для решения кадастровых задач?
8. Какой тип лазерного сканера (стационарный наземный, мобильный наземный, авиационный) наиболее эффективно применять, когда перед инженерами стоит задача обмера радиальной лесной площади с целью последующей валидации результатов государственной инвентаризации леса?
9. Какой формат данных, в котором представляется плотное облако точек, на текущий момент является основным стандартным при их обработке в специальном программном обеспечении и геоинформационных системах?
10. Какое свойство наземного лазерного сканирования позволяет говорить о полной автоматизации процесса сбора информации об объекте
11. Какой тип лазерного сканера (стационарный наземный, мобильный наземный, авиационный) наиболее требователен к процессу рекогносцировки на снимаемом участке?
12. Какой тип лазерного сканера (стационарный наземный, мобильный наземный, авиационный) наименее требователен к процессу рекогносцировки на снимаемом участке?



13. Как называется функция в Cloud Compare, которая позволяет создавать объекты правильной геометрической формы заданного размера для того чтобы использовать их в качестве референтных при визуальном анализе?
14. Как называется функция в Cloud Compare, которая позволяет выполнять срезы плотного облака точек плоскостями для построения профилей ?
15. Зачем при использовании стационарного сканера рекомендуется расклеивать изображения- марки на плоские поверхности?
16. Как называется функция в Cloud Compare, которая позволяет вырезать из него отдельные подмножества по произвольному контуру или векторному файлу?
17. Какой фильтр рациональнее всего использовать в ситуации, когда сцена изобилует неявными контурами.
18. Какой фильтр рациональнее всего использовать в ситуации, четкими контурами, которые можно аппроксимировать, которые можно аппроксимировать плоскостями?
19. Можно ли установить стационарный лазерный сканер на БВС?
20. Можно ли установить мобильный лазерный сканер на БВС?
21. Обработывая трёхмерную измерительную видео сцену можно ли экспортировать в скалярное поле плотного облака координаты в плане и по высоте?
22. Какую задачу визуального анализа можно решить, включив отображение точек плотного облака по интенсивности отраженного сигнала?
23. Как в Cloud Compare называется функция прореживания плотного облака точек?
24. Процесс растеризации плотного облака точек представляет из себя ...
25. Можно ли создать из растрового изображения плотное облако точек?
26. Можно ли построить горизонтали по плотному облаку точек?
27. В каких единицах рассчитывается RMS внутренней сходимости модели в процессе привязки?
28. В каких случаях оправданна чистка облака в ручном режиме?
29. Как называется группа алгоритмов и методик, которая позволяет изучать изменения по разновременным трёхмерным
30. Какая команда в Cloud Compare позволяет рассчитывать объемы насыпей/штабелей?



31. Цифровая модель рельефа выраженная виде горизонталей в формате .shp является относится к типу (растровая, векторная, фрактальная) ...
32. Цифровая модель рельефа выраженная виде горизонталей в формате .tiff является относится к типу (растровая, векторная, фрактальная) ...
33. В общем виде объясните принцип работы CFS-фильтра для определения точек рельефа местности.
34. Какие параметры древостоя можно определять автоматизированным методом по результатам кластеризации в СПО на основе наземного лазерного сканирования?
35. Какие программные пакеты для работы с плотными облаками точек вам известны?

36. Сопоставьте данные

| № п/п | Вопрос                                                                                        | Ответ                                                                                                                      |
|-------|-----------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| 1     | Дайте определение термину Моделирование:                                                      | 1. Науке и промышленности, компьютерных играх, медицине                                                                    |
| 2     | Что такое рендеринг?                                                                          | 2. Blender Foundation Blender, Side Effects Software Houdini                                                               |
| 3     | Где применяют трехмерную графику (изображение)?                                               | 3. Формула корней квадратного уравнения                                                                                    |
| 4     | Модель человека в виде манекена в витрине магазина используют с целью:                        | 4. Формализацией                                                                                                           |
| 5     | Рисунки, карты, чертежи, диаграммы, схемы, графики представляют собой модели следующего вида: | 5. Совокупность записанных на языке математики формул, отражающих те или иные свойства объекта-оригинала или его поведение |
| 6     | Программные обеспечения, позволяющие создавать трёхмерную графику это...                      | 6. Создание трёхмерной математической модели сцены и объектов в ней                                                        |
| 7     | К числу математических моделей относится:                                                     | 7. Математические                                                                                                          |
| 8     | Процесс построения информационных моделей с помощью формальных языков называется:             | 8. Пять                                                                                                                    |
| 9     | Математическая модель объекта:                                                                | 9. Развлечения                                                                                                             |
| 10    | Сколько существует основных этапов разработки и исследование моделей на компьютере:           | 10. Построение проекции в соответствии с выбранной физической моделью                                                      |

37. 3D моделирование — это

38. Задача 3D-моделирования:

39. В трёхмерной системе координат 3D-3-dimensional оси обозначаются как



40. Местоположение объектов, выраженных по отношению к системе координат XYZ, называется
41. Выделяют основные типы моделирования::
42. Методы построения полигональных моделей:
43. Визуализация – это
44. В современных ГИС встречаются два альтернативных подхода к визуализации векторных данных:
45. Для визуализации линейных объектов используются следующие основные способы:
46. При визуализации площадных объектов графический стиль имеет две компоненты:
47. Тематические карты – это карты,
48. Пространственные данные — это
49. Визуализация информации — это процесс
50. На сегодняшний день известно три вида визуализации:
51. Среди моделей визуализации карт существует классификация:
52. Векторные карты — это
53. Растровые карты — это
54. Программа Blender – это
55. Какие основные операции можно выполнять над объектом в программе Blender:
56. Как называется изображение, облегающее форму модели:
57. Основу геоинформационного моделирования как специализированной ГИС-технологии составляют
58. Объектами геоинформационного моделирования являются
59. 3D графика – это
60. Процесс создания трехмерной модели включает три этапа:
61. Пространственная модель местности –
62. Виртуальная модель –



63. Виртуальная сцена –
64. Под пространственной моделью местности понимается
65. Существует 2 основных принципа создания трехмерных моделей:
66. Наглядность – это
67. Информативность –
68. Общее цифровое описание пространственного объекта (feature) включает:
69. Чтобы слои совместились на карте, они должны иметь
70. Геометрическое представление пространственных объектов базируется на следующих принципах:

### Ситуационные задачи к экзаменам

#### Задание 1 3D-моделирование

##### *Цели работы:*

- 3D-моделирование позволяет развивать пространственное мышление у студентов.
- Необходимость развития у учащихся новейших навыков обращения с инновационными технологиями.
- Необходимость воспитания технического творчества и инженерной мысли у студентов.
- Повышение мотивации студентов на занятиях посредством использования 3D-моделирования на занятиях.

##### *Задание:*

Ситуационная задача с применением 3D-моделей

Необходимо создать виртуальную модель технических средств при помощи 3D технологий с последующим их натурным воспроизведением.

Создать элективный курс с использованием 3D-моделирования.

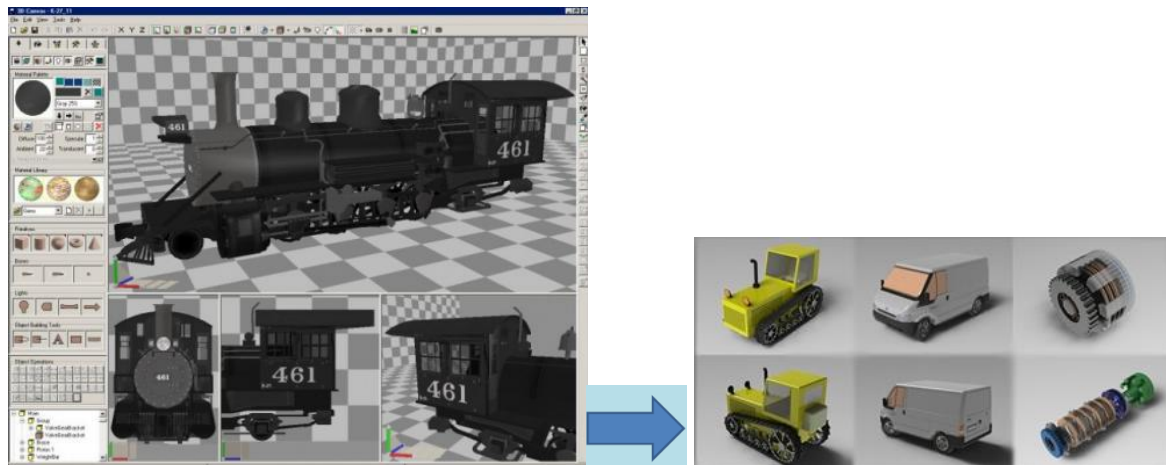
Разработка ситуационных задач, для решения которых необходимо 3D-моделирование устройств.

Для проверки результативности используются:

Проектная деятельность по созданию 3D-моделей новых устройств.

Расчет моделей при помощи физических формул.





## Задание 2 Трёхмерные модели как системы отображения пространственной информации и их практическое использование

### Цели работы:

- Проектирование сельскохозяйственных ферм в Африке по программе Международного развития и банка реконструкции.
- Создание топографической карты и 3D-модели местности на территорию фермы в Ваку Кунго.

### Задание.

1. Заказ и планирование новой цветной космической съемки со спутника GeoEye-1 с пространственным разрешением 0,5 м;
2. фотограмметрическая обработка и ортотрансформирование взятых изображений с применением коэффициентов рациональных функций (RPC-полиномы);
3. создание на базе ортотрансформированных изображений фотокарты и цифровой топографической карты;
4. создание 3D-модели фермы на базе одиночного снимка GeoEye-1 и подготовленной топографической карты с точностью до трех метров по высоте.
5. Оформите отчёт по проделанной работе.

## Задание 3 Электронная карта сельского туризма

### Цели работы:

- Интерактивный модуль размещения 3D моделей на географической карте.
- Анализ картографических сервисов и браузерных технологий взаимодействия с 3D моделями
- использование нескольких инструментов для представления объектов.
- Анализ существующих решений и формирование требований к системе подбора
- Анализ существующих реализаций систем навигации внутри зданий и кампусов



- Программная система обеспечения навигации внутри здания

*Задание:*

1. Сценарий проекта.
2. На экране карта МО с нанесенными объектами.
3. После указания на объект появляется:
4. Маршрут движения из данной точки в объект.
5. Описание объекта
6. 3-х мерное изображение
7. Панорама объекта (внутренняя или внешняя).
8. Трехмерные модели объектов должны содержать информацию о геометрии и текстурах и могут быть использованы в виде файлов одного из стандартных форматов 3Dмоделей.

В окружении леса и реки находится гостевой дом в деревне Михалёвская. Комнаты и мебель дома отделаны натуральным деревом. Дом отлично подходит для отдыха большим семьям и компаниям. Можно арендовать дом полностью или поселиться в одном из отдельных блоков дома. Оборудование базы позволяет готовить еду самостоятельно или заказывать питание у повара. Есть услуга приготовления заказных блюд, из предоставляемых вами продуктов.

Около деревни есть река и озеро, где можно порыбачить самостоятельно. При желании, вам предоставят снасти и ледоруб. В реке и озере обитают щуки, налимы, окуни, хариус и др. рыба.

Вокруг деревни расположены вековые леса, а так же сосновый молодняк. Так же недалеко от неё находится один из маршрутов миграции дикого гуся. Таким образом, это место отлично подойдет для охотников на различную дичь. Администрация базы поможет организовать не только сезонные, но и трофейные охоты в частном охотхозяйстве.

Из зимних видов развлечений гостевой дом предлагает беговые лыжи, а так же катание с горки на «ватрушках». Есть возможность организации поездки в горнолыжный комплекс «Малиновка»

#### **Задание 4 Создание модуля "Уникальные села России"**

*Цели работы:*

- определить требования, предъявляемые к трехмерной цифровой модели;
- рассмотреть различные методы и технологии создания трехмерных моделей;
- проанализировать технологии публикации трехмерных моделей на геопорталах;
- рассмотреть состав и расположение объекта.
- разработать методику геомоделирования трехмерных объектов на примере выбранного объекта.
- создать и представить в графическом виде.

*Задание:*

1. Зарегистрированное доменное имя
2. Созданный и отлаженный интернет-портал
3. Интегрированный модуль 3Д - визуализации объектов с включением объектов (порядка

25 объектов).

4. Инструкция по работе с порталом и созданию 3Д-визуализации.

#### 7.4. Содержание занятий семинарского типа.

##### Практические занятия.

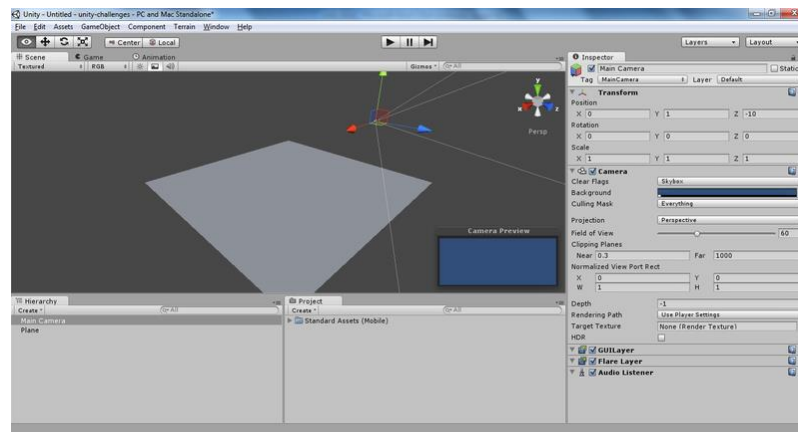
### ПРАКТИЧЕСКАЯ РАБОТА №1. Создание сцены в Unity 3D

Вид практического занятия: Создать сцену, в которой при нажатии на пробел в стенку кидается ящик, и после удара стенка исчезает.

Тема и содержание занятия Работа в Unity 3D

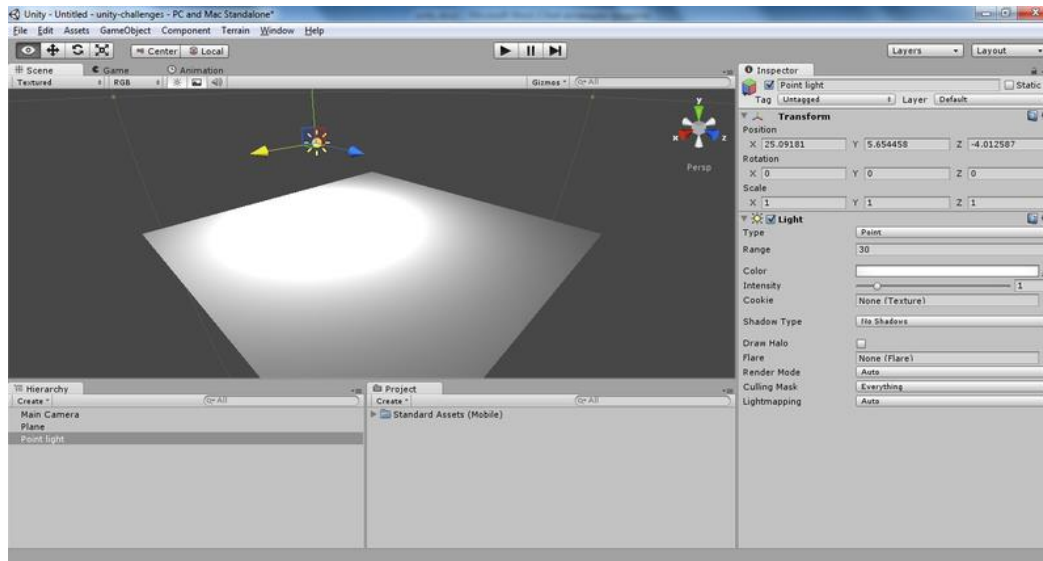
Цель. Работа со сценой.

Сначала создадим пустой новый проект, сразу появится сцена с камерой. Добавляем в сцену плоскость - это будет наш «пол». Слегка приблизим точку просмотра к плоскости (колёсиком мыши) для достижения состояния «не сильно мелко, не сильно крупно».

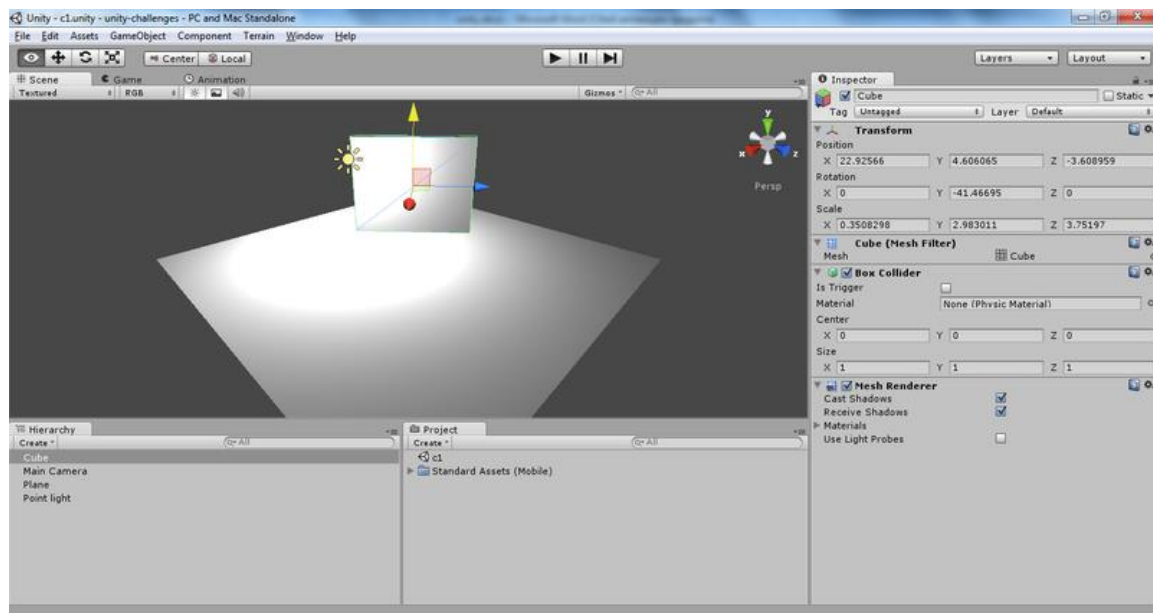


На скриншоте выделена камера.

Заметим, что камера (объект MainCamera на панели Hierarchy) развёрнута так, что только что добавленной плоскости не видно (см. скриншот выше). Надо это исправить. Не зря же мы «прицеливались» к плоскости! Теперь можно выделить камеру (в панели Hierarchy или непосредственно на сцене) и нажать GameObject->Align With View (или нажать Ctrl+Shift+F). Это перенесёт камеру в ту точку пространства, из которой мы и смотрели на плоскость. Камера на месте, теперь поправим освещение. Добавим на сцену источник света Point Light и разместим где-нибудь так, чтобы сцена стала поприятнее визуально:



Теперь добавим в сцену куб (назовём его wall) и растянем его до состояния стенки. Повернём/подвинем стенку так, чтобы она стояла «лицом» к камере. Эти манипуляции вызваны тем, что стрелять ящиком мы будем из точки расположения камеры «вперёд» (в привычках, так как это достаточно относительное понятие; наш «вперёд» будет смотреть туда же, куда и камера).



Со сценой почти закончили. Точнее, с видимой её частью. Ещё нам понадобится куб-префаб, которым будем стрелять. Добавим в сцену куб (можно его слегка уменьшить, если хочется), а в проект — префаб (назовём его, например, cube). Перетащим куб из панели иерархии на префаб и удалим куб из сцены — он нам больше не понадобится. Выберем в панели иерархии наш cube и добавим ему компонент Rigidbody. Теперь можно приступать к написанию двух скриптов — скрипт SelfDestroyer будет удалять объект при уда-



ре о него другого объекта, а скрипт Shooter будет, собственно, стрелять кубом в направлении стенки.

Создадим новый C# скрипт, назовём его Selfdestroyer. Удалим методы Update и Start из его текста – нам нужно лишь удалить объект при ударе. Описываем метод OnCollisionEnter — он будет прост до безобразия (предполагается, что ничто другое кроме ящиков нашу стенку ударять не будет):

```
void OnCollisionEnter()
{
    Destroy(gameObject);
}
```

Здесь gameObject – это объект типа GameObject, связанный с тем объектом на сцене, к которому прикреплен скрипт.

Заметим, что можно было описать функцию с прототипом void OnCollisionEnter(Collision obj) и в ней использовать информацию об объекте, ударившем объект, к которому прикреплен скрипт. Но нам это не нужно, так что оставим список формальных параметров пустым.

Перетащим из панели Project этот скрипт на объект wall в панели иерархии. Убедимся, что пока при запуске сцены ничего не происходит.

Теперь займёмся вторым скриптом — Shooter. Заведём переменную типа GameObject, которая и будет содержать тот объект, которым мы впоследствии будем стрелять в стенку:

```
public GameObject cubePrefab;
```

Здесь нам не понадобится метод Start(), а вот Update оставим и дополним так, чтобы при нажатии на пробел (это клавиша прыжка при стандартных настройках) из точки расположения объекта, к которому прикреплен скрипт (в нашем случае это будет камера) вперёд с некоторой силой бросался куб. В итоге файл скрипта будет выглядеть так:

```
using UnityEngine;
using System.Collections;
```

```
public class Shooter : MonoBehaviour
{
    public GameObject cubePrefab;
    // Update is called once per frame
    void Update ()
    {
        if (Input.GetButtonDown("Jump"))
        {
            GameObject cube = Instantiate(cubePrefab, transform.position,
transform.rotation) as GameObject;
            cube.rigidbody.AddForce(transform.forward * 1000);
        }
    }
}
```

Прикрепим скрипт к объекту MainCamera. Запустим сцену и убедимся, что ... при нажатии на пробел в консоль выдаётся ошибка:

```
! UnassignedReferenceException: The variable cubePrefab of 'Shooter' has not been assigned.
```

Она сообщает нам, что в скрипте Shooter переменной cubePrefab не присвоено никакого

значения. Исправим этот недочёт — выберем в панели иерархии объект камеры, найдем в панели Inspector раздел с названием “Shooter (Script)”. И правда, в поле “Cube Prefab” стоит значение None. Перетащим в это поле cube из панели Project. Теперь запускаем и стреляем кубами, пока не надоест.

Итак, написав 5 содержательных строчек кода, мы имеем: сцену, реагирующую на пользовательский ввод и объект (стенку), реагирующий на «внешние раздражители» в виде соударений.

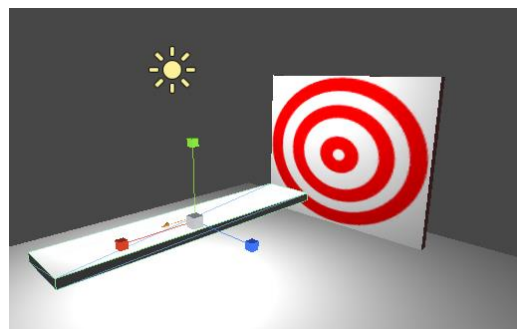
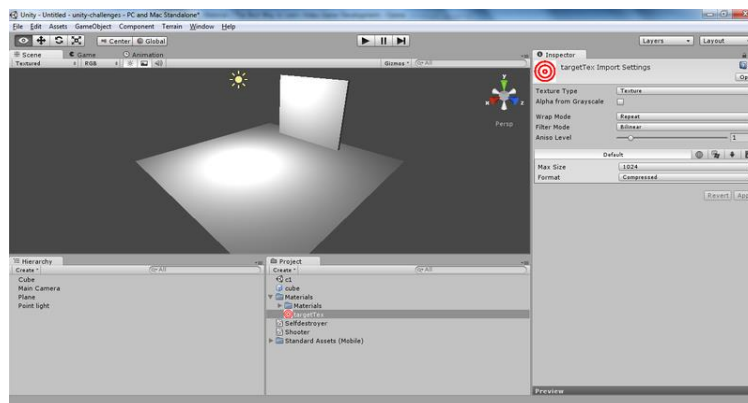
## ПРАКТИЧЕСКИЕ РАБОТЫ № 2. Создание объектов сцены в Unity 3D

Вид практического занятия: Работа 3D UNITY.

Тема и содержание занятия: Сцена.

Цель занятия: нужно создать сцену, в которой по нажатию пробела в поле зрения игрока начнёт падать ящик, в итоге запуская другой ящик в сторону мишени. После попадания мишень должна исчезнуть с каким-нибудь звуком. Запуск ящика – на основе рычага: один ящик уже лежит с одной стороны, ещё один падает на другую сторону рычага.

Сначала стандартно создадим плоскость, источник света, и расположим камеру так, чтобы было «всё хорошо видно» (об этом есть в разборе предыдущей задачи). Теперь добавим мишень: создадим куб и растянем его до размеров, подходящих порядочной мишени. Пододвинем мишень так, чтобы она была перпендикулярна оси OX (направления осей можно посмотреть справа сверху во вкладке Scene). Сразу же набросаем в любом редакторе простенькую текстуру, отдалённо напоминающую мишень (сделаем её квадратной) и сохраним в папку панели Project (предварительно создав эту папку):



Теперь добавим в проект (опять же в папку Materials) материал targetMaterial и перетащим в его поле Texture (в панели Inspector) только что добавленную текстуру. Материал перетащим на мишень (назовём её объект в панели Hierarchy “target”). Теперь зай-

мёмся рычагом-катапульты. Добавим в сцену куб, назовём его plank1 и растянем его «в доску»:

Теперь добавим к plank1 компонент Hinge Joint (Component->Physics->Hinge Joint) – в переводе означает «шарнир». На скриншоте выше он уже добавлен – появился оранжевый вектор. Он показывает, вокруг какой оси будет вращаться объект. Сейчас это направление совпадает с осью OX. Можно для проверки слегка повернуть доску относительно OX и запустить сцену.

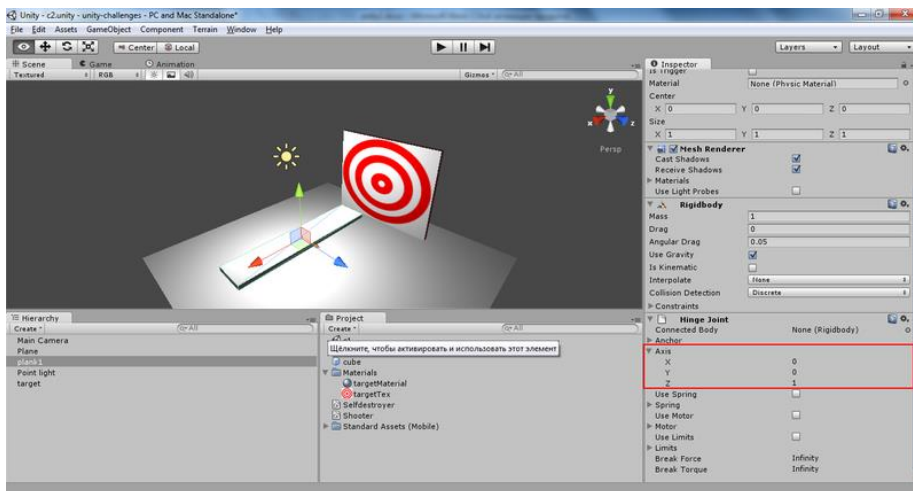
Итак, нас не устраивает ось вращения – ведь нужно, чтобы при резком повороте доски ящик с одного её края полетел в направлении мишени. В панели Inspector находим компонент Hinge Joint для plank1 и находим группу полей Axis. Это и есть ось вращения. По умолчанию установлены следующие значения:

X:1

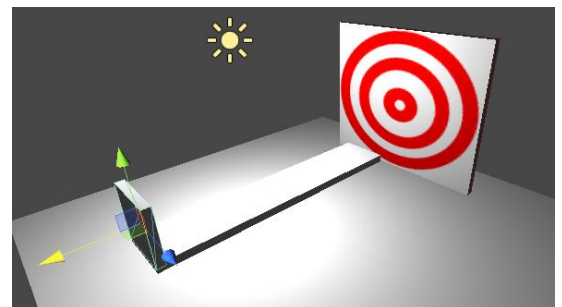
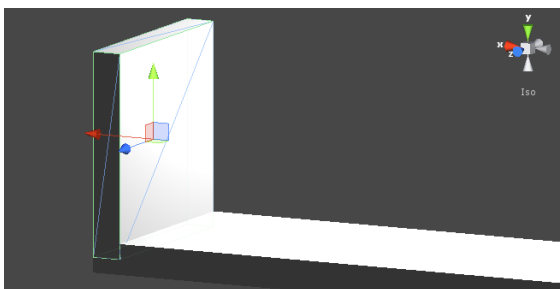
Y:0

Z:0

Это и показывает сонаправленность оси OX. Нам подойдёт ось, параллельная OZ. Установим в поле Z единицу, в остальные – нули. Заметим, как поменял направление оранжевый вектор из центра plank1 на сцене.

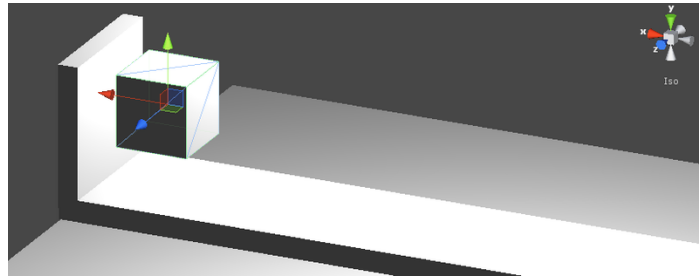


С рычагом почти закончили. Сейчас проблема в том, что не хватает упора для бросаемого кубика. Добавим его: добавляем куб (назовём его plank2), сжимаем (см. скриншот ниже) и передвигаем так, чтобы он слегка пересекался с plank1, образуя сцепление (пока что лишь виртуальное):



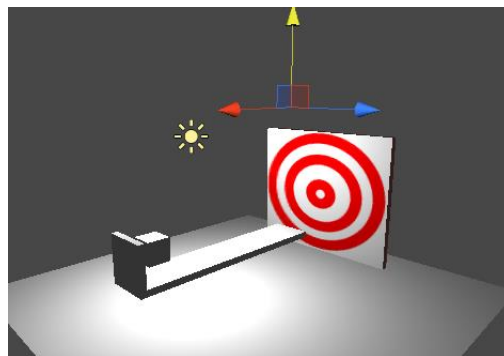
Добавим plank2 компонент Rigidbody (к plank1 этот компонент уже добавился автомати-

чески при добавлении Hinge Joint). «Сообщим» физическому движку о том, что plank1 и plank2 скреплены. Объекту plank1 добавим компонент Fixed Joint (Component->Physics->Fixed Joint). В поле Connected Body в панели Inspector сейчас стоит значение None (Rigidbody). Перетащим на его место plank2 из панели иерархии. Теперь запустив сцену, мы увидим, как наш рычаг опускается одним краем вниз. Это уже радует. Добавим в сцену куб и поместим его так, чтобы он находился чуть выше рычага у его левого (ближнего к камере) конца. При этом уменьшим все размеры куба в 2 раза:



Добавим ему компонент Rigidbody. Если запустить сцену сейчас, когда рычаг опускается, куб падает на его конец (опускающийся). Конструкция готова к «выстрелу», осталось подготовить противовес, падение которого будет отправлять в полёт только что добавленный куб.

Добавим в проект префаб, назовём его dropcube. Создадим новый куб в сцене и перетянем его из панели иерархии на префаб. Добавим префабу компонент Rigidbody, а сам куб из сцены удалим. Массу префаба установим равной 30 (для расположения объектов на созданной мной сцене это значение позволяет запускаемому кубу долететь до мишени). Теперь добавим в сцену пустой объект (Game Object->Create Empty) с названием dropper. Поместим его над дальним от камеры концом рычага в поле зрения камеры (это можно проверить, выбрав объект Main Camera и выбрав Game Object -> Align View to Selected) – оттуда позже будут появляться кубы (такие же, как dropcube):



## Часть 2: Скрипты.

Займёмся скриптовой частью. Во-первых, мишень должна исчезнуть после попадания в неё куба. Из прошлой задачи (ССЫЛКА) возьмём скрипт Selfdestroyer (я для всех задач буду использовать один проект, так что даже добавлять новый скрипт не пришлось – всё уже есть в панели Project) и прикрепим его к объекту мишени. Вот полный код этого скрипта (совсем небольшой):

```
using UnityEngine;  
using System.Collections;  
public class Selfdestroyer : MonoBehaviour  
{
```





```
void OnCollisionEnter(Collision obj)
{
    Destroy(gameObject);
}
}
```

Теперь добавим в проект C#-скрипт `Voxdropper`. Он должен по нажатию на пробел создавать в точке нахождения объекта, к которому прикреплен, куб (точнее, создавать Instance префаба `dropcube`). Раз требуется только реакция на нажатие клавиш – оставляем метод `Update`, метод `Start` (если он был автоматически сгенерирован) удаляем. Для осуществления задуманного нам, как минимум, потребуется передать в скрипт префаб. Добавим `public` переменную типа `GameObject` в класс:

```
public GameObject cubePrefab;
```

Изменим метод `Update`:

```
void Update ()
{
    if (Input.GetButtonDown("Jump"))
    {
        GameObject.Instantiate(cubePrefab, transform.position, transform.rotation);
    }
}
```

Пара замечаний по поводу этого кода:

во-первых, вместо `Input.GetButtonDown("Jump")` можно использовать `Input.GetKeyDown(KeyCode.Space)`. Второй вариант привязан строго к коду клавиши пробел, а первый — к клавише, которая в настройках управления игры исполняет роль клавиши прыжка (по умолчанию – пробел). Эти настройки можно увидеть, собрав проект и запустив получившийся исполняемый файл (если это окажется непонятно, могу расписать процесс поподробнее).

также стоит заметить, что в отличие от скрипта в первом задании, при вызове `GameObject.Instantiate` результат функции никуда не сохраняется потому, что он нам попросту не нужен. Если в прошлый раз мы после создания объекта дополнительно действовали на него силой, то здесь это не требуется.

здесь вызывается перегруженная функция `Instantiate`, имеющая 3 параметра, тогда как есть вариант и с одним: `Instantiate(Object original)`. Последняя будет задавать параметры `position` и `rotation` (второй и третий параметр соответственно в вызове `Instantiate` с тремя параметрами) такими же, как у переданного фактическим параметром объекта. Важно помнить, что эти параметры берутся не из соответствующих значений для объекта, к которому прикреплен скрипт!

Теперь перетащим скрипт `Voxdropper` из панели `Project` на объект `dropper` в панели иерархии. Что ж, задание почти выполнено. Если запустить сцену и нажать на пробел, упадет куб и маленький куб (который уже лежал на плече рычага) полетит в мишень. Осталась одна незначительная нестыковка с формулировкой: нет звука при уничтожении мишени. В первую очередь нам нужен сам звук, то есть файл с каким-нибудь звуком. Добавляем звук в проект (у меня он назван `onDestroySound`). Небольшое лирическое отступление о звуке в `Unity`. Звук здесь бывает моно и 3D (`3D Sound`). Фактически это показывает 1 или 2 канала в аудио. Для эффектов, имеющих достаточно четкое пространственное положение (взрыв, звук удара и т.п.) рекомендуется использовать моно-звук, так как для его реальная

громкость (слышимая игроком) будет зависеть от удалённости источника от слушателя (определяется наличием компонента Audio Listener, по умолчанию он есть у объекта Main Camera).

Если ваш звук оказался 3D (стоит галочка “3D Sound” в панели Inspector), не отчаивайтесь. Уберите «галочку» с пункта “3D Sound”, выберете в той же панели пункт “Force to mono” и нажмите “Apply”. Теперь модифицируем скрипт SelfDestroyer. Нам понадобится public переменная для звука, который будем проигрывать:

```
public AudioClip onDestroy;
```

Теперь в методе OnCollisionEnter помимо вызова метода Destroy добавим проигрывание звука:

```
void OnCollisionEnter(Collision obj)
```

```
{
```

```
    AudioSource.PlayClipAtPoint(onDestroy, transform.position);
```

```
    Destroy(gameObject);
```

```
}
```

Такой вызов метода AudioSource.PlayClipAtPoint проиграет звук onDestroy так, будто источник звука находится в точке расположения объекта, к которому этот скрипт прикреплен (transform.position).

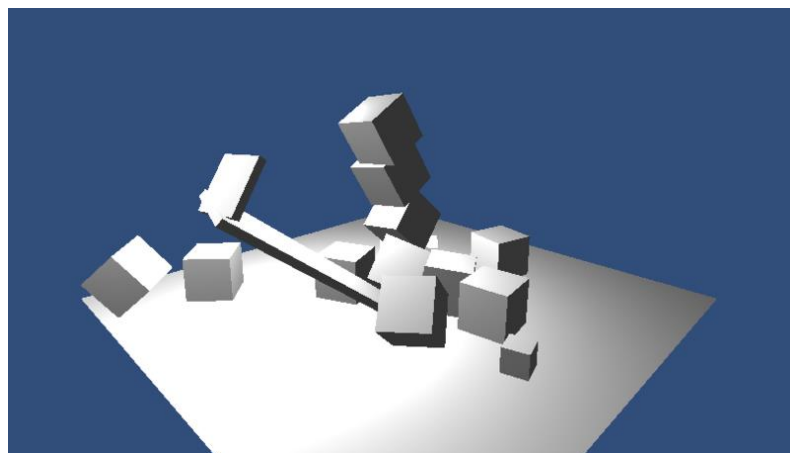
Осталось выбрать объект мишени на сцене или в панели иерархии и перетащить из проекта звуковой файл (onDestroySound) в поле “On Destroy (Audio Clip)” панели Inspector. Замечание: можно добавить к любому объекту компонент Audio Source и после этого обращаться к нему как «audio» из скриптов. Есть даже специальный метод, который проигрывает переданный звук в точке расположения объекта:

```
audio.PlayOneShot(AudioClip clip);
```

Однако в нашем случае проигрывание звука происходит непосредственно перед уничтожением объекта, и если использовать audio.PlayOneShot, то после удаления звук перестанет проигрываться. Это означает, что мы его и не услышим. Можете проверить сами (это даже полезно). Итак, задание выполнено.

Часть 3. Эстетическая.

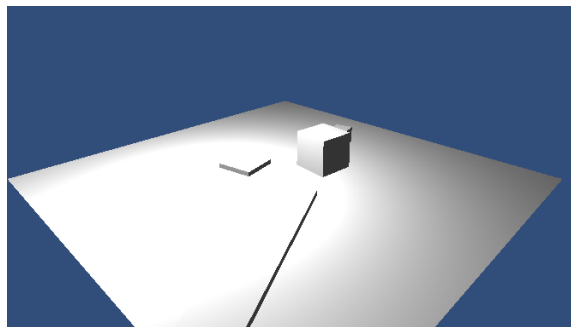
Казалось бы, чего ещё? Еще одна деталь: если бросать по несколько кубиков за раз (нажимать пробел несколько раз подряд), две «доски», образующие рычаг (plank1 и plank2) начинают смещаться друг относительно друга. Видно, например, на этом скриншоте (здесь в сцену добавлен ещё один источник света):



Тогда о самом простом решении: раз соединение plank1 и plank2 нам нужно всего на один раз, то почему бы от него не избавиться после того, как оно уже сыграло свою роль? Просто уничтожить plank2? Некрасиво. Снова выбираем plank1 в панели иерархии и находим компонент Fixed Joint в панели Inspector. Замечаем любопытное поле: Break Force: Infinity. То, что нужно! В моём случае подошло значение силы, требуемой для разлома соединений, в 10 (надеюсь, что ньютонов). Под «подошло» я имею в виду, что в момент отрыва метаемого кубика от поверхности рычага, plank2 отрывается от plank1 и отправляется в полёт:



Как оказалось, той же проблеме подвержен и Hinge Joint – странное поведение plank1 снова наблюдается при большом количестве падающих кубов. Решим проблему аналогично – установим в поле Break Force значение 30. В результате вся конструкция рычага после первого запуска разрушается, причём достаточно «физично». Так выглядит сцена после первого запуска:



P.S. Для пущей красоты можно добавить материалы всем объектам в сцене: рычагу, всем кубам (большим и малым), полу. Как это делается, описано в начале статьи. Заключение. В этот раз кода чуть побольше (хотя всё так же меньше 10 «смысловых» строк, если не считать фигурные скобки и код, генерируемый автоматически), да и задание более содержательное. Уже появилась некоторая контролируемая нами физика (в виде соединений) и звук, включающийся при определённых событиях в сцене.

### **ПРАКТИЧЕСКАЯ РАБОТА № 3. Создание подвижных объектов Unity 3D**

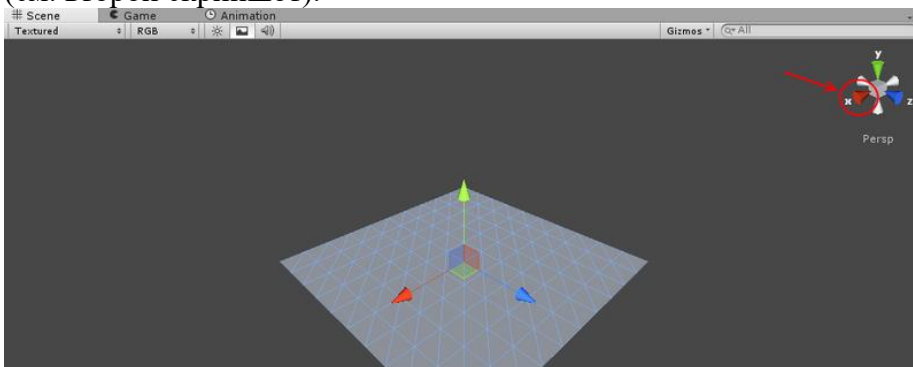
Вид практического занятия: расчетная работа.

Тема и содержание: Создание подвижных объектов Unity 3D

Цель работы: игрок должен уметь перемещаться вправо/влево и стрелять по трём мишеням. Мишени при попадании должны падать, а игроку за каждую сбитую мишень начисляется очко. По достижении трёх очков показывается экран окончания игры.

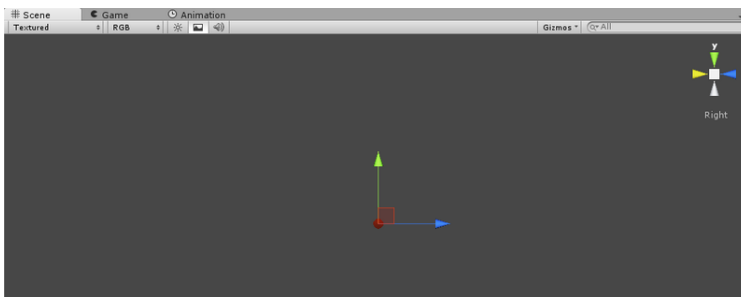
Добавим на сцену плоскость – пол, на который будут падать кубы-мишени, её размер пока изменять не будем. Предположим, что игрок будет стрелять из точки положения камеры в том же направлении, куда она смотрит. Для простоты предположим, что стрелять будем вдоль оси OX (напоминаю, направления осей показано в правом верхнем углу вкладки Scene).

Необходимо в соответствии с этим расположить камеру. Выберем на сцене плоскость. Если нажать на ось OX (на отвечающий ей конус-кнопку на первом скриншоте ниже), то мы «посмотрим» на выбранный объект (то есть плоскость) в направлении, параллельном оси OX (см. второй скриншот).



Скриншот

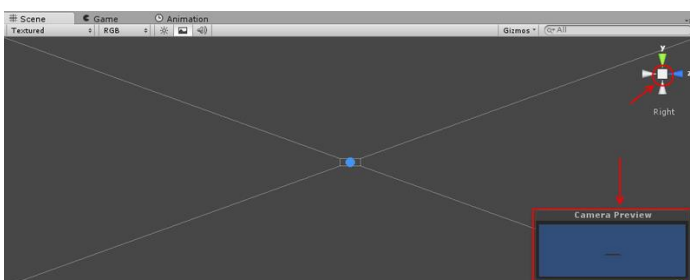
1



Скриншот

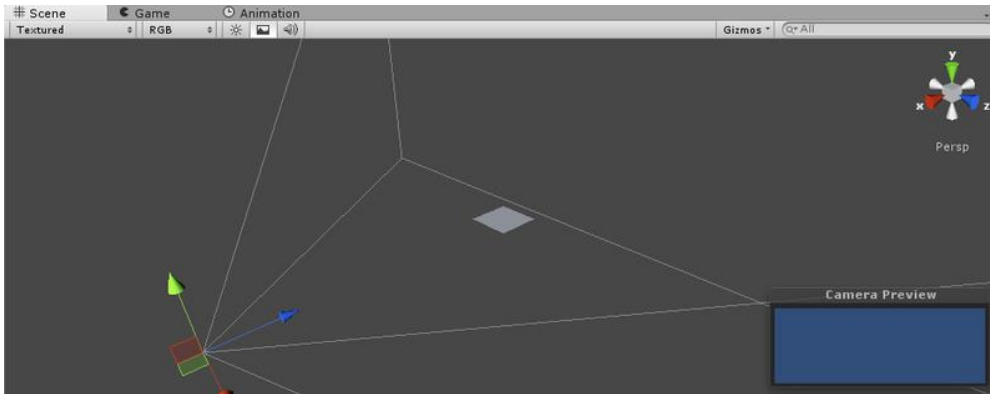
2

Не удивляйтесь, что перестала быть видна сама плоскость, так и должно быть. Теперь поставим камеру в ту же точку, из которой смотрим: выбираем в панели иерархии объект Main Camera, нажимаем GameObject -> Align With View (или нажмите Ctrl+Shift+F). Теперь из камеры видно плоскость в нужном ракурсе, но далековато, да и камера слишком низко расположена (при выбранном объекте камеры в правом нижнем углу вкладки Scene – предпросмотр вида камеры). На обведённое кругом пока не обращайте внимания.

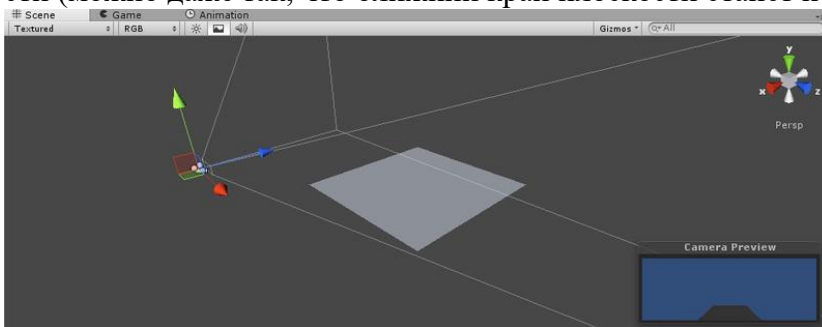


Вернёмся к «нормальному» виду на сцену: для этого нужно выделить на сцене плоскость и нажать на кубик между стрелками осей в верхнем правом углу (сейчас он виден как квад-

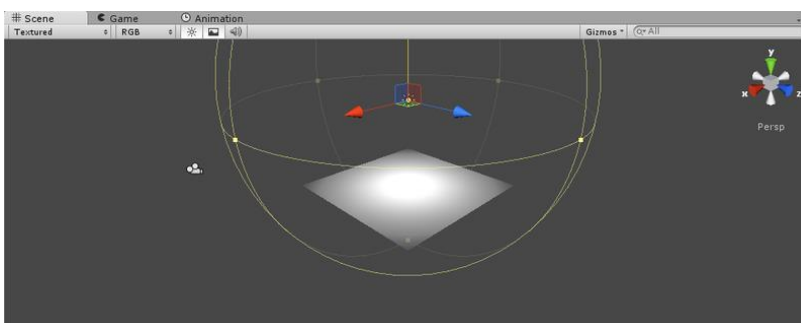
рат) – он обведён красным кругом на скриншоте выше. Получим что-то вроде этого (может быть нужно будет удалиться от плоскости, покрутив колёсико мыши, чтобы камера попала в поле зрения):



На скриншоте выделена камера. Теперь поднимем камеру повыше и пододвинем к плоскости (можно даже так, что ближний край плоскости станет не виден):

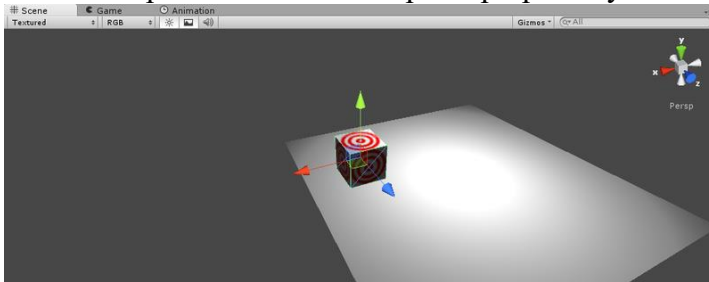


С камерой пока всё. Добавим в сцену освещение – точечный источник света (можно и любой другой, здесь цель добавления освещения – просто сделать сцену «посветлее», не более того; для красивого освещения, возможно, придётся использовать другие источники света) и приподнимем его:

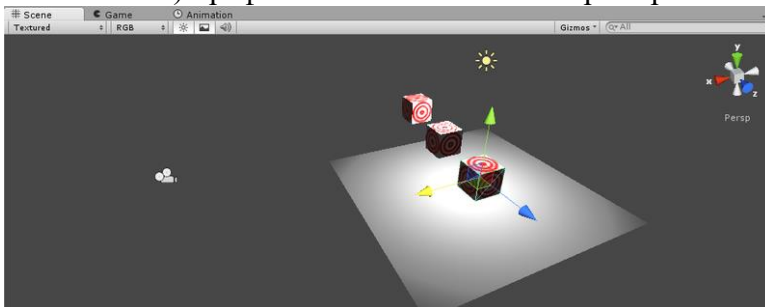


Замечу, что у точечных источников света есть несколько редактируемых параметров. В частности, нас интересуют Intensity и Range. Первый – интенсивность света – отвечает за то, как ярко светит источник. Второй – дальность действия. То есть источник света может светить очень ярко, но лишь в небольшом объёме. И сколько бы мы не меняли параметр Intensity, дальше заданного радиуса свет от источника распространяться не будет. По умолчанию выставляются параметры Range = 10 и Intensity = 1. Если выделить объект источника света, на сцене вокруг него обрисовывается шар – это и есть объём, в котором «светит» источник. В моём случае плоскость лежала внутри этого объёма, так что Range

можно не менять, но плоскость была как-то тускло освещена, поэтому я увеличил Intensity до 4 (на скриншоте выше это уже сделано). Создадим теперь префаб для мишени в панели Project, назовём его target (можно, конечно, обойтись и тремя отдельными объектами, но это неудобно, да и вдруг понадобится не три, а сто таких мишеней?). Добавим в сцену куб. Слегка разукрасим его, добавив текстуру. В прошлом разборе мы создавали материал с некоторой текстурой, почему бы не использовать его ещё раз? Если не помните, как создать материал – загляните в разбор предыдущей задачи.



Теперь из панели иерархии перетащим этот куб на префаб в панели Project, а сам куб удалим. Создадим три копии (или «образца»; по-английски, насколько я понимаю, это называется instance) префаба. Разместим их все примерно на той же высоте, что и камера:



Положение мишеней можно (и, может быть, придётся) подкорректировать и уже после того, как игрок научится стрелять (если невозможно будет попасть по мишеням). Основная сцена готова, экран завершения игры опять же будет позже.

#### **ПРАКТИЧЕСКАЯ РАБОТА №4. Создание скриптов в Unity 3D**

Вид практического занятия: Часть 2: Скрипты. Unity 3D

Тема и содержание: работа в среде Unity 3D

Цель занятия: работа со скриптом Playerscript.

Практические навыки:

Сначала определимся, какие скрипты нам нужны.

Во-первых, нужно из точки расположения камеры стрелять по мишеням, в том же скрипте (он будет «повешен» на саму камеру) будем двигать её вправо/влево. Этот скрипт назовём Playerscript.

Ещё к префабу мишени нужно прикрепить скрипт, который будет при соударении добавлять компонент Rigidbody и засчитывать игроку очко. Это будет Rigidbodytizer. Но кто будет вести счёт? Можно было бы делать это в скрипте для камеры, но как-то некрасиво получается. Добавим в сцену пустой объект (GameObject->Create Empty) и назовём его GameManager – он будет считать набранные очки и при нужном количестве переходить на другую сцену – экран окончания игры. Этому объекту тоже нужен свой скрипт, пусть называется так же: GameManager. Сначала разберёмся с тем, что нужно в первую очередь: скрипт GameManager. По сути нам нужны только метод Update для проверки счёта и пара переменных для хранения текущего счёта и количества очков, которые нужно



набрать для победы (можно, конечно, обойтись использованием числа 3, но если захочется модифицировать игру, всплывут некоторые неудобства).

```
using UnityEngine;
```

```
using System.Collections;
```

```
public class GameManager : MonoBehaviour
```

```
{
```

```
    public uint neededScore = 3;
```

```
    public static uint score = 0;
```

```
    // Update is called once per frame
```

```
    void Update ()
```

```
    {
```

```
        if (score == neededScore)
```

```
        {
```

```
            // будет добавлено позже
```

```
        }
```

```
    }
```

```
}
```

Переменная `score` описана как `public static`, так как в этом случае к ней можно будет обращаться из любого скрипта как `GameManager.score`. Конечно, не очень хорошо, что переменная, в которой хранится значение счёта, является `public`-членом, но для упрощения скриптовой логики оставим так. Не забудьте прикрепить только что написанный скрипт к объекту `GameManager`.

Теперь займёмся скриптом `Rigidbodytizer`. Методы `Update` или `Start` нам не понадобятся, добавим метод `OnCollisionEnter()`:

```
void OnCollisionEnter()
```

```
{
```

```
    if (rigidbody == null)
```

```
    {
```

```
        gameObject.AddComponent("Rigidbody");
```

```
        GameManager.score++;
```

```
    }
```

```
}
```

Условие проверяет, добавлен ли уже компонент `Rigidbody` к объекту, и если нет, добавляет его и увеличивает число засчитанных очков. Обратите внимание, увеличение числа очков – тоже внутри условия! Иначе очко засчитается даже за соударение мишени с полом. Теперь прикрепим `Rigidbodytizer` к префабу `target`. Осталось добавить стрелка – скрипт `PlayerScript`. Но для начала нужен префаб «пули» — им и будем стрелять. Добавляем в проект префаб “bullet”, а в сцену – сферу. Последнюю сожмём до размеров, например, 0.4 по всем измерениям: в `Inspector` у компонента `Transform` изменим все значения поля `Scale` на 0.4. Можно добавить материал и сфере, это опционально. Перетащим сферу из панели `Hierarchy` на префаб `bullet` в панели `Project` и удалим сферу со сцены. Как мы будем двигать камеру? Скорее, вопрос не «как» (поменять `transform.Position` – не проблема), а «в соответствии с чем?». У класса `Input` есть замечательный статический метод `GetAxis`, который по названию оси выдаёт некое `float`-значение. В том числе для строк “Horizontal” и “Vertical” мы получим значение от -1 до 1, показывающее степень отклонения от, соответственно, вертикальной и горизонтальной осей. Это сработает для стрелок и WASD на клавиатуре и для джойстика. При этом если для джойстика фраза про «отклонение от оси»



имеет смысл, то для клавиатуры, как я понял, большую часть времени при нажатой клавише возвращается +1 или -1 (в зависимости от направления), хотя и есть некоторая инертность при нажатии и отпуске клавиши (значение, возвращаемое методом Input.GetAxis, проходит через промежуточное значение).

С помощью полученной информации заставим камеру двигаться:

```
public class Playerscript : MonoBehaviour
{
    public const float speed = 7;
    public GameObject bulletPrefab;
    public float force = 1500;

    // Update is called once per frame
    void Update ()
    {
        float translation = Input.GetAxis("Horizontal") * Time.deltaTime * speed;

        transform.position += new Vector3(0, 0, translation);
    }
}
```

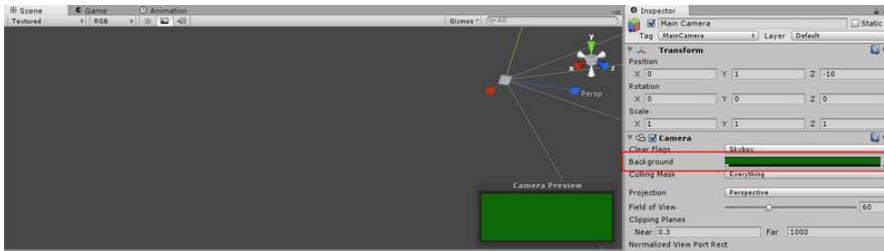
Public переменная speed отвечает за максимальную скорость движения камеры. Значение, возвращаемое Input.GetAxis умножаем на Time.deltaTime для того, чтобы камера двигалась не быстрее speed метров в секунду, а не «метров за кадр». А bulletPrefab – префаб пули, которым будем стрелять. force – модуль силы, с которой будем эту самую пулю «кидать». Прикрепим скрипт к камере, запустим сцену и попробуем нажимать стрелки вправо/влево. Всё отлично, пора и пострелять!

Добавим выстрел при нажатии на клавишу прыжка (пробел по умолчанию) в метод Update:

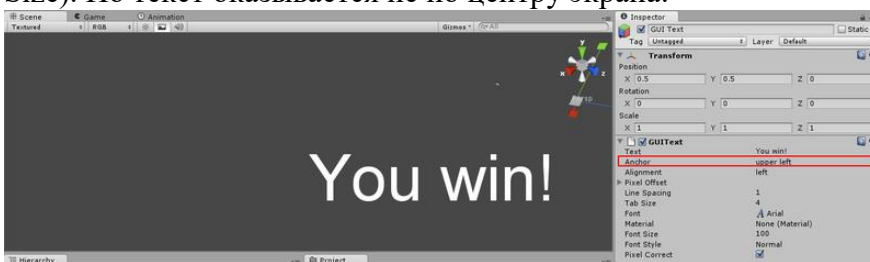
```
if (Input.GetButtonDown("Jump"))
{
    GameObject bullet = GameObject.Instantiate(bulletPrefab, transform.position,
transform.rotation) as GameObject;
    bullet.rigidbody.AddForce(transform.forward * force);
}
```

Не забудем выбрать объект MainCamera и в панели Inspector в поле Bullet Prefab компонента Palyerscript (Script) перетащить префаб bullet из панели Project. Можно запустить сцену – игра почти готова. Осталось сделать переход на другую сцену по окончании игры. Для этого нужно сначала создать новую сцену обязательно включить её в список сцен для сборки. Добавляем в проект новую сцену, сразу сохраним её под именем “win”. Выберем File -> Build Settings. Потом нужно нажать кнопку Add Current в появившемся окне (можно и перетащить сцену из панели Project в список Scenes In Build). Поменяем цвет фона у объекта Main Camera на какой-нибудь более оптимистичный:





Теперь нужно добавить текст (например, “You win!”). Нажимаем GameObject -> Create Other -> GUI Text. Появится новый объект с текстом по умолчанию “Gui Text”. Изменим поле Text компонента GUI Text на нужный, также поменяем размер шрифта (поле Font Size). Но текст оказывается не по центру экрана:



На картинке выделена и причина такого поведения текста. Свойство Anchor показывает, где у текста «центр». Не физический центр, а тот центр, которым определяется положение текста. Сейчас выбрано значение “Upper left”, то есть текст «подвешен» за верхний левый угол. Как можно заметить, этот угол и в самом деле находится в центре экрана. Выберем значение «Middle center», и текст переместится в положенное место по середине экрана. Почему именно середина экрана, и что делать, если нужно сместить текст? У GUI Text есть ещё и группа полей Pixel Offset с полями X и Y. Задавая им нужные значения можно сместить текст на указанное количество пикселей относительно центра экрана (точнее, центра окна с игрой). Добавим переход на эту сцену в нужный момент времени. Вернёмся на основную сцену и подправим метод Update в скрипте GameManager:

```
void Update ()
{
    if (score == neededScore)
    {
        Application.LoadLevel("win");
    }
}
```

Это загрузит недавно созданную сцену при достижении нужного счёта. На этом основная часть заканчивается, получилось вполне играбельно. Стоило бы, конечно, добавить надпись с текущим результатом в основную сцену, но пусть это будет небольшим заданием :) Замечу, что доступ к тексту объекта, имеющего компонент GUI Text осуществляется так:

```
guiText.text = "Hello world!";
```

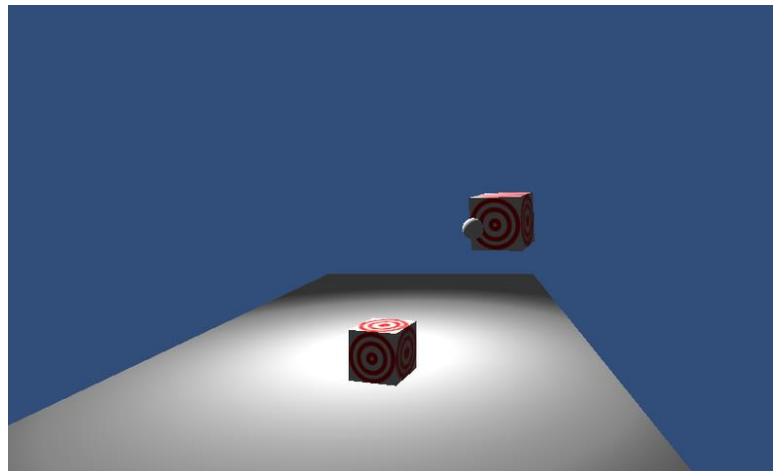
Часть 3: Эстетическая.

И снова ощущение, что в сцене что-то не так! Что сделает порядочный кубик, если в него

врезается порядочная пуля (хотя у нас это, скорее, ядро)? Правильно, отлетит! Но почему же не отлетает сейчас? Ответ вполне логичен. Когда в методе `OnCollisionEnter` мы добавляем мишени компонент `Rigidbody`, ядро уже столкнулось с мишенью без `Rigidbody`. Но нет же метода, вызывающегося, когда «ещё чуть-чуть, и будет столкновение»! Поэтому проблема решается проще. Что по сути должно происходить? Мишень висит в воздухе. Как такое возможно? Если она не имеет массы? Начнём с того, что так не бывает. А зачем нам отсутствие массы? Чтобы на тело не действовала сила притяжения к земле. Так может это выключается? Прошлый абзац – от начала и до конца лирическое отступление, на деле обошлось без этих рассуждений. Я просто добавил префабу мишени компонент `Rigidbody` и случайно заметил там поле `Use Gravity`. Вот и нашёлся тот самый выключатель силы тяжести. Кстати, если запустить игру сейчас (продлав только описанное абзацем выше), получится забавный эффект: мишени разлетаются, но не падают вниз под действием силы тяжести.

Пришла пора изменить скрипт `Rigidbodytizer` (теперь он уже, правда, не оправдывает своего названия).

```
void OnCollisionEnter()
{
    if (!rigidbody.useGravity)
    {
        rigidbody.useGravity = true;
        GameManager.score++;
    }
}
```



Теперь игра выгладит как-то так:

Второй момент, который хотелось бы затронуть: управление камерой. Движение камеры только вдоль одной прямой накладывает значительные ограничения на положение кубиков (чтобы их ещё и сбить можно было). По аналогии с движением вправо/влево добавьте движение вверх/вниз (придётся использовать `Input.GetAxis("Vertical")`)

). Тогда можно будет размещать кубики-мишени где угодно в пределах видимости, и в них можно будет попасть. Напоследок отмечу, что если размещать мишени только над плоскостью, логично было бы ограничить движение вправо/влево границами плоскости, внося небольшие изменения в `PlayerScript`. Добавим парочку переменных и метод `Start` (пояснения в коде):

```
float zBoundMin, zBoundMax; // ограничения на перемещение вдоль оси Z
```



```
void Start()
```

```
{
```

```
    // ограничивающий параллелепипед объекта с названием "Plane"  
    Bounds planeBounds = GameObject.Find("Plane").collider.bounds;  
    zBoundMin = planeBounds.min.z; // минимальное значение Z  
    zBoundMax = planeBounds.max.z; //максимальное значение Z
```

```
}
```

Конечно, можно перевычислять ограничения на координату Z каждый кадр, но это неэффективно, а в случае неподвижной плоскости ещё и бессмысленно. Теперь изменим метод Update: после изменения вектора transform.position добавим ещё строчку: transform.position = new Vector3(transform.position.x, transform.position.y, Mathf.Clamp(transform.position.z, zBoundMin, zBoundMax));

К сожалению, отдельно поменять координату Z вектора transform.position нельзя. Функция Mathf.Clamp(x, a, b) возвращает значение x, если зажать его между значениями a и b (то есть если, например, x меньше a, то функция вернёт значение a). Если вы добавили перемещение вверх/вниз, убедитесь схожим образом, что камера не уходит ниже плоскости и не поднимается слишком высоко.

### Тесты к экзамену (3 семестр)

1. Метод обратной трассировки рассчитывает:

1. вторичное диффузное освещение
2. **диффузную освещенность трехмерных поверхностей только от прямых источников света**
3. диффузную освещенность трехмерных поверхностей от прямых источников света и вторичное диффузное освещение

2. Укажите блок в архитектуре графического процессора, в состав которого входит интерполятор:

1. блок растеризации
2. ultra-threaded dispatch processor
3. потоковый процессор SPU
4. **setup engine**

3. Рабочие регистры вершинного шейдера содержат:

1. положение источника света
2. **промежуточные результаты расчета**
3. исходные координаты местоположения вершин, цвета и координаты текстуры

4. Укажите функции, которые в графическом процессоре выполняет блок растеризации: накладывает на объекты сцены различные текстуры

1. **записывает рассчитанные пиксели в кадровый буфер, производит операции их смешивания и обработки**
2. считывает исходные данные из видеопамяти, производит первичную обработку и отправляет их в блоки предварительной подготовки данных к вычислениям
3. определяет расчеты, которые необходимо выполнить, готовит данные в соответствии с форматом и отправляет их в вычислительные элементы

5. Укажите название минимальной единицы текстуры трёхмерного объекта:

1. **тексел**
2. воксел
3. пиксел

6. На каком этапе графического конвейера происходит исполнение вершинных шейдеров?



1. рендеринг
  2. triangle setup
  3. **transform&lighting**
  4. декомпозиция
7. При перемещении камеры результаты расчета методом Radiosity:
1. изменяются всегда
  2. изменяются в некоторых случаях
  3. **не изменяются**
8. Укажите недостаток метода Radiosity:
1. неприменим при подвижной камере
  2. рассчитывает диффузную освещенность трехмерных поверхностей только от прямых источников света
  3. **не работает с эффектами зеркального отражения и прозрачности**
9. При каком методе визуализации результаты расчета при изменении месторасположения камеры изменяются?
1. метод обратной трассировки и метод Radiosity
  2. **метод обратной трассировки**
  3. метод Radiosity
10. В каком процессоре планирование исполнения потока параллельных команд является статическим, то есть на стадии компиляции?
1. **процессор с VLIW-архитектурой**
  2. любой суперскалярный процессор
  3. суперскалярный процессор, не имеющий VLIW-архитектуры
11. Укажите количество регистров констант у вершинного шейдера версии 2.0:
- 128
- 96**
- 64
12. Какой блок в архитектуре графического процессора считывает исходные данные из видеопамати, производит первичную обработку и отправляет их в блоки предварительной подготовки данных к вычислениям?
1. блок растеризации
  2. ultra-threaded dispatch processor
  3. текстурный процессор TMU
  4. **setup engine**
  5. потоковый процессор SPU
13. Обработать не только одну вершину, но и целый примитив, способен:  
только вершинный шейдер  
как вершинный шейдер, так и геометрический шейдер  
**только геометрический шейдер**
14. Укажите метод визуализации, который учитывает вторичное диффузное освещение:  
метод обратной трассировки  
**метод Radiosity**  
метод обратной трассировки и метод Radiosity
15. Для самосветящихся объектов используется:
1. субтрактивное формирование оттенков
  2. субтрактивное и аддитивное формирование оттенков
  3. **аддитивное формирование оттенков**
16. При диффузном отражении света точечного источника от идеального рассеивателя:
1. угол между направлениями наблюдения и отражения всегда не равен нулю



2. угол между направлениями наблюдения и отражения всегда равен нулю
  3. **падающий свет рассеивается во все стороны с одинаковой интенсивностью**
17. Хроматическая диаграмма XYZ описывает:
1. текущее положение объекта в пространстве
  2. **цветовую модель**
  3. прозрачность объекта
18. Какие регистры вершинного шейдера содержат информацию о цвете и итоговых координатах текстур?
1. **выходные регистры**
  2. входные затуманивание пикселя регистры
  3. рабочие регистры
  4. регистры констант
19. Формирование оттенков называется аддитивным, когда:
1. происходит вычитание из падающего света определенных длин волн
  2. **требуемый цвет формируется за счет смешения трех основных оттенков цветов**
  3. происходит вычитание из падающего света определенных длин волн и требуемый цвет формируется за счет смешения трех основных оттенков цветов
20. Результатом работы вершинного шейдера является:
1. затуманивание пикселя
  2. **итоговые координаты текстур**
  3. цвет пикселя
21. При подповерхностном рассеивании свет ...
1. не выходит из объекта
  2. **выходит из объекта в выходной точке, отличной от точки вхождения в объект**
  3. выходит из объекта в выходной точке, являющейся точкой его вхождения в объект
22. Сколько существует типов колбочек в глазе человека, отличающихся фоточувствительным пигментом?
- 3  
5  
2  
4
23. Если значение параметра `glBegin` в OpenGL равно `gl_lines`, то:
1. каждые три вызова `glVertex` задают треугольник
  2. **каждые два вызова `glVertex` задают отрезок**
  3. каждые четыре вызова `glVertex` задают четырехугольник
  4. каждый вызов `glVertex` задает отдельную точку

### Контрольные вопросы к экзамену

1. Системы координат в компьютерной графике. Переход от мировых к экранным координатам.
2. Основные геометрические модели трехмерных объектов.
3. Каркасная и граневая геометрические модели трехмерных объектов. Достоинства и недостатки, область применения.
4. Граневая геометрическая модель трехмерных объектов. Полигональная сетка, параметрические бикубические куски.
5. Объемно-параметрическая геометрическая модель трехмерных объектов.
6. Кинематическая геометрическая модель трехмерных объектов.



7. Способы визуализации трехмерных изображений.
8. Способы задания полигональной сетки. Основные достоинства и недостатки.
9. Основные способы математического описания кривых и поверхностей. Достоинства параметрического способа описания.
10. Форма Эрмита для задания параметрической кубической кривой. Основные достоинства и недостатки. Условия непрерывности.
11. Форма Безье для задания параметрической кубической кривой. Область применения. Условия непрерывности.
12. Форма В-сплайнов для задания параметрической кубической кривой. Область применения.
13. Форма Эрмита для задания параметрической бикубической поверхности. Условия непрерывности.
14. Форма Безье для задания параметрической бикубической поверхности. Условия непрерывности.
15. Форма В-сплайнов для задания параметрической бикубической поверхности. Область применения.
16. Аффинные преобразования параметрических кривых и поверхностей.
17. Удаление скрытых ребер и поверхностей. Сортировка граней по глубине.
18. Удаление скрытых ребер и поверхностей. Метод плавающего горизонта.
19. Удаление скрытых ребер и поверхностей. Метод z-буфера.
20. Удаление скрытых ребер и поверхностей. Методы оптимизации. Метод порталов.
21. Модели отражения и преломления цвета. Определение цвета закрашивания.
22. Методы закрашивания поверхностей: плоское закрашивание, закрашивание методами Гуро, Фонга. Тени.
23. Методы закрашивания поверхностей: трассировка лучей, метод анализа излучательности.
24. Детализация поверхностей цветом и фактурой.
25. Текстуры.
26. Моделирование микрорельефа поверхности.
27. Применение эффекта «затуманивания» для передачи глубины пространства.
28. Построение трехмерных сцен. Граф сцены. Форматы файлов трехмерной графики.
29. Понятие интерфейса прикладного программирования (API).

**8. Перечень основной и дополнительной учебной литературы; перечень ресурсов информационно-телекоммуникационной сети «Интернет», перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень программного обеспечения и информационных справочных систем (при необходимости)**

### **8.1 Основная литература**

4. Пушкарева, Т. П. Компьютерный дизайн : учебное пособие / Т. П. Пушкарева, С. А. Титова. - Красноярск : Сиб. федер. ун-т, 2020. - 192 с. - ISBN 978-5-7638-4194-7. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1819273> (дата обращения: 16.05.2023). – Режим доступа: по подписке.
5. Представление и визуализация результатов научных исследований : учебник / О. С. Логунова, П. Ю. Романов, Л. Г. Егорова, Е. А. Ильина ; под ред. О. С. Логуновой. —



Москва : ИНФРА-М, 2020. — 156 с. — (Высшее образование: Аспирантура). - ISBN 978-5-16-014111-4. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1056236>

6. Представление и визуализация результатов научных исследований : учебник / О.С. Логунова, П.Ю. Романов, Л.Г. Егорова, Е.А. Ильина ; под ред. О.С. Логуновой. — Москва : ИНФРА-М, 2020. — 156 с. + Доп. материалы [Электронный ресурс]. — (Высшее образование: Аспирантура). — DOI 10.12737/textbook\_5c178eb6cf1e63.57981471. - ISBN 978-5-16-014111-4. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1056237> (дата обращения: 16.05.2023). – Режим доступа: по подписке.

## 8.2 Дополнительная литература

2. Искусственная световая панорама как выражение планировочной структуры города [Architecture and Modern Information Technologies = Архитектура и современные информационные технологии, №1 (18), 2012, стр. -] - Режим доступа: <http://znanium.com/catalog/product/441151>

Карпенко, В. Е. Формирование световой панорамы прибрежного города (на примере Владивостока) [Электронный ресурс] : диссертация / В. Е. Карпенко. - Владивосток, 2013. - 208 с. - Режим доступа: <https://znanium.com/catalog/document?id=4925>

## 8.3. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

1. Электронно-библиотечная система ZNANIUM. Режим доступа: <https://znanium.com/> Круглосуточный доступ из любой точки посредством сети Интернет в многопользовательском режиме.
2. Электронно-библиотечная система BOOK.ru. Режим доступа: <https://book.ru/> Круглосуточный доступ из любой точки посредством сети Интернет в многопользовательском режиме.
3. Научная электронная библиотека E-library (информационно-справочная система). Режим доступа: <http://www.e-library.ru/> Доступ свободный

## 8.4. Перечень программного обеспечения, современных профессиональных баз данных и информационных справочных системам

1. Microsoft Windows
2. Microsoft Office
3. Построение пространственных моделей территорий и объектов (РЕКОД-Модель)
4. Свободная географическая информационная система с открытым кодом QGIS 2.18

## 9. Методические указания для обучающихся по освоению дисциплины

Процесс изучения дисциплины предусматривает контактную работу с преподавателем (работа на лекциях и практических занятиях) и самостоятельную (самоподготовка к лекциям и практическим занятиям) работу обучающегося.

В качестве основных форм организации учебного процесса по дисциплине «Трёхмерное моделирование и визуализация пространственных объектов» по предлагаемой методике обучения выступают лекционные и практические занятия (с использованием интерактивных технологий обучения), а также самостоятельная работа обучающихся.



Теоретические занятия (лекции) организуются по потокам. На лекциях излагаются темы дисциплины, предусмотренные рабочей программой, акцентируется внимание на наиболее принципиальных и сложных вопросах дисциплины, устанавливаются вопросы для самостоятельной проработки. При проведении лекций планируется использование интерактивных форм изложения материала в виде проблемных лекций с использованием мультимедийных технологий в виде презентаций. Конспект лекций является базой при подготовке к практическим занятиям, к экзаменам, а также самостоятельной научной деятельности.

- *Традиционная лекция с презентацией* - подразумевает традиционное изложение учебного материала посредством акцентуации основных смысловых доминант; лекция сопровождается презентацией;

Практические занятия по дисциплине «Трехмерное моделирование и визуализация пространственных объектов» проводятся в форме выполнения практических работ с целью приобретения практических навыков в решении задач по стандартизации и управлению качеством в сфере государственного муниципального управления.

Практические занятия способствуют более глубокому пониманию теоретического материала учебного курса, а также развитию, формированию и становлению различных уровней, составляющих основную профессиональную образовательную программы компетентности студентов.

Целью самостоятельной (внеаудиторной) работы обучающихся является обучение навыкам работы с научно-теоретической, периодической, научно-технической литературой и технической документацией, необходимыми для углубленного изучения дисциплины «Трехмерное моделирование и визуализация пространственных объектов», а также развитие у них устойчивых способностей к самостоятельному изучению и изложению полученной информации.

Основными задачами самостоятельной работы обучающихся являются:

- овладение фундаментальными знаниями;
- наработка профессиональных навыков;
- приобретение опыта творческой и исследовательской деятельности;
- развитие творческой инициативы, самостоятельности и ответственности студентов.

Самостоятельная работа студентов по дисциплине «Трехмерное моделирование и визуализация пространственных объектов» обеспечивает:

- закрепление знаний, полученных студентами в процессе лекционных и практических занятий;
- формирование навыков работы с периодической, научно-технической литературой и технической документацией;

Самостоятельная работа является обязательной для каждого обучающегося.

#### **10. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине (модулю):**

Учебные занятия по дисциплине «Трехмерное моделирование и визуализация пространственных объектов» проводятся в следующих оборудованных учебных кабинетах:

| Вид учебных занятий по дисциплине           | Наименование оборудованных учебных кабинетов, объектов для проведения практических занятий с перечнем основного оборудования |
|---------------------------------------------|------------------------------------------------------------------------------------------------------------------------------|
| Занятия лекционного типа, групповые и инди- | учебная аудитория, специализированная учебная мебель<br>ТСО: видеопроекторное оборудование/переносное видеопр-               |





|                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| видуальные консультации, текущий контроль, промежуточная аттестация | екционное оборудование<br>доска                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Занятия семинарского типа                                           | Инновационно- образовательный центр космических услуг<br>Специализированная учебная мебель<br>ТСО: Видеопроекционное оборудование<br>Интерактивный стол Creority Touch для использования с программным комплексом РЕКОД-МОДЕЛЬ (разработчик - ОАО "Научно-производственная корпорация "Рекод"), рабочие станции, РЕКОД-Модель - построение пространственных моделей территорий и объектов<br>Лицензионное программное обеспечение: в соответствии с рабочей программой                                                      |
| Самостоятельная работа обучающихся                                  | помещение для самостоятельной работы, специализированная учебная мебель, ТСО: видеопроекционное оборудование, автоматизированные рабочие места студентов с возможностью выхода в информационно-телекоммуникационную сеть "Интернет", доска;<br>Помещение для самостоятельной работы в читальном зале Научно-технической библиотеки университета, специализированная учебная мебель автоматизированные рабочие места студентов с возможностью выхода информационно-телекоммуникационную сеть «Интернет», интерактивная доска |