

Сертифицированный тестировщик Тестирование игр (СТ-GaMe) Программа обучения

Версия 1.0.1 RU

International Software Testing Qualifications Board



Разработана
Российской Коллегией Экспертов по Качеству ПО
Russian Software Testing Qualifications Board (RSTQB)



Уведомление об авторских правах

Уведомление об авторских правах © International Software Testing Qualifications Board (далее просто ISTQB®) ISTQB является зарегистрированной торговой маркой International Software Testing Qualifications Board.

Авторские права © 2022 авторы перевода: Андрей Конушин (руководитель группы), Александр Александров (редактор), Оксана Быкова, Вадим Луковатый, Александр Торговкин, Павел Шариков.

Авторские права © 2022 авторы: Андрей Конушин (руководитель группы), Александр Александров, Ярослав Верещагин, Евгений Глушкин, Любовь Журавлева, Дмитрий Карасев, Елена Карасева, Елизавета Кручинина, Вадим Луковатый, Дмитрий Мелишев, Максим Николаев, Александр Прохоров, Антон Савватеев, Айрат Сайфуллов, Артем Стукалов, Никита Сысуев, Татьяна Тепаева, Александр Торговкин, Маргарита Трофимова, Павел Шариков, Кирилл Шевелев, Светлана Юшина.

Все права защищены. Авторы передают свои права International Software Testing Qualifications Board (далее ISTQB®). Авторы (владельцы авторских прав в данный момент) и ISTQB® (как будущий владелец авторских прав) договорились о следующих условиях использования:

Выдержки из этого документа для некоммерческого использования могут быть скопированы, если указан источник. Любая аккредитованная обучающая компания может использовать эту программу обучения в качестве основы для учебного курса, если авторы и ISTQB® указаны как источник и владельцы авторских прав программы обучения и при условии, что в любой рекламе таких курсов данная программа обучения может быть упомянута только после письменного уведомления об аккредитации материалов тренингов коллегиями, признанными ISTQB®.

Любое частное лицо или группа частных лиц может использовать программу как основу для статей, книг или других производных письменных материалов, если авторы и ISTQB® упомянуты как источник и владельцы авторских прав программы.

Любое другое использование этой программы запрещено без предварительного письменного одобрения ISTQB®.

Любая коллегия, признанная ISTQB®, может переводить эту программу обучения при условии, что она воспроизводит вышеупомянутое Уведомление об авторских правах в переведенной версии программы.

История изменений

Версия	Дата	Комментарии
v1.0.1	21 октября 2022	Выпуск GA
v1.0.1 RU	21 октября 2022	Перевод на русский язык

Оглавление

Уведомление об авторских правах	2
История изменений	3
Оглавление	4
Благодарности	7
0. Предисловие к программе обучения	8
0.1 Цель этого документа	8
0.2 Сертифицированный тестировщик игр	8
0.3 Карьерный путь тестировщиков	8
0.4 Бизнес-результаты	8
0.5 Проверяемые цели обучения и когнитивные уровни знаний	9
0.6 Экзамен «Сертифицированный тестировщик игр»	9
0.7 Аккредитация	10
0.8 Поддерживаемые стандарты	10
0.9 Актуальность программы обучения	10
0.10 Уровень детализации	10
0.11 Как организована эта программа обучения	10
1. Специфика тестирования игр – 75 минут (К2)	12
1.1 Основы тестирования игр	12
1.1.1 Специфика тестирования игр	12
1.1.2 Риски игровых продуктов	12
1.1.3 Дефекты, специфичные для тестирования игр	13
1.1.4 Как тестирование снижает риски видеоигр	14
1.1.5 Различие между тестированием и «игрой»	15
1.2 Типовые роли команды разработки игрового ПО	16
1.3 Мероприятия тестирования в жизненном цикле разработки игрового ПО	16
2. Тестирование игровых механик - 180 минут (К3)	19
2.1 Игровые механики	19
2.1.1 Типы игровых механик	19
2.1.2 Разница между тестированием игровых и неигровых механик	20
2.1.3 Разница между тестированием ключевых механик и мета-механик	20
2.1.4 Разница между тестированием клиентской, серверной и клиент-серверной механик	21
2.1.5 Примеры дефектов игровой механики и возможные причины их возникновения	22
2.2 Подходы к тестированию игровых механик	23
2.2.1 Процедуры и подходы к тестированию игровых механик на протяжении всего жизненного цикла разработки программного обеспечения игрового продукта	23
2.2.2 Важность тестирования игровых механик	24
2.2.3 Важность проверки игровых механик	24
2.2.4 Тестирование игрового состояния после возобновления сеанса и при неактивности пользователя	25
3. Тестирование графики - 165 минут (К3)	29
3.1 Принципы и концепции игровой графики	29
3.1.1 Особенности графического содержания игрового продукта	29

3.1.2	Типы дефектов графического содержания	33
3.2	Подходы к тестированию графики в игровых продуктах	37
3.2.1	Художественное тестирование	37
3.2.2	Техническое тестирование	38
3.2.3	Тестирование во время игры (игровое тестирование)	39
3.3	Выполнение тестирования графики	39
3.3.1	Выполнение тестов графики на разных этапах производства объекта 39	
3.3.2	Тестирование графики на историческую точность	41
3.4	Инструментальная поддержка тестирования графики	42
4.	Тестирование звука - 190 минут (К3)	43
4.1	Особенности звукового содержания игрового продукта	43
4.1.1	Типы звуков	44
4.1.2	Звуковые эффекты и технологии	45
4.1.3	Звуковые зоны	45
4.2	Типы дефектов звукового содержания	46
4.3	Подходы к тестированию звукового содержания в игровых продуктах	47
4.3.1	Тестирование звукового содержания	48
4.3.2	Тестирование сочетания музыки и звуков игры	48
4.4	Тестирование звука	49
4.4.1	Уровни тестирования аудио-музыкального содержания в жизненном цикле разработки игрового программного обеспечения	49
4.4.2	Интеграция звуков в игру	50
4.4.3	Зоны ответственности вовлеченных лиц	50
4.4.4	Процедуры и подходы к тестированию звуковых объектов	51
4.5	Инструментальная поддержка тестирования звука	51
5.	Тестирование игровых уровней - 65 минут (К2)	53
5.1	Принципы и концепции проектирования уровней видеоигр	53
5.1.1	Термин «уровень» и его специфика в зависимости от жанра игрового проекта	53
5.1.2	Понимание типов дефектов в проектировании уровней видеоигры ..	54
5.2	Этапы и выполнение тестирования уровней видеоигр	56
5.2.1	Основные этапы проектирования и тестирования уровней видеоигр	56
5.2.2	Области ответственности вовлеченных лиц	58
5.3	Инструментальная поддержка тестирования уровней видеоигры	59
6.	Тестирование игровых контроллеров - 95 минут (К2)	60
6.1	Принципы и концепции игровых контроллеров	60
6.1.1	Типы игровых контроллеров	60
6.1.2	Дефекты, связанные со спецификой игровых контроллеров	62
6.2	Подходы к тестированию контроллеров в игровых продуктах	63
6.3	Инструментальная поддержка тестирования игровых контроллеров	64
7.	Тестирование локализации - 155 минут (К3)	65
7.1	Принципы и концепции тестирования локализации	65
7.1.1	Локализация и интернационализация	65
7.1.2	Разница между локализацией игрового продукта и прикладного ПО	67
7.1.3	Этапы тестирования локализации	68
7.2	Типы дефектов локализации и их причины	71
7.2.1	Возможные причины дефектов локализации игровых продуктов	71
7.2.2	Дефекты и риски локализации	72

7.3	Подходы к локализации и их выполнение	73
7.3.1	Разница между полным и частичным тестированием локализации ..	73
7.3.2	Процедуры и подходы в тестировании локализации в жизненном цикле разработки игрового продукта	73
7.3.3	Типы тестирования локализации	75
7.3.4	Области ответственности вовлеченных специалистов.....	76
7.4	Инструментальная поддержка тестирования локализации	77
8.	Ссылки.....	78
8.1	Стандарты	78
8.2	Документы ISTQB.....	78
8.3	Книги.....	78
8.4	Ссылки (Интернет)	79
9.	Приложение А – Цели обучения/Когнитивные уровни знаний	80
10.	Приложение Б – Матрица трассируемости бизнес-результатов и целей обучения	82
11.	Приложение В – Описание изменений.....	85
12.	Приложение Г – Термины, специфичные для тестирования игр, и прочие термины	86
13.	Приложение Д – Предметный указатель	91

Благодарности

Перевод версии документа 1.0 выполнен рабочей группой АНО «Коллегия экспертов по качеству программного обеспечения» (Russian Software Testing Qualifications Board, RSTQB): Андрей Конушин (руководитель группы), Александр Александров (редактор), Оксана Быкова, Вадим Луковатый, Александр Торговкин, Павел Шариков.

Этот документ был подготовлен основной командой АНО «Коллегия экспертов по качеству программного обеспечения» (Russian Software Testing Qualifications Board, RSTQB) в составе:

Андрей Конушин (председатель коллегии)
Маргарита Трофимова (вице-председатель коллегии)
Александр Александров (редактор)
Вадим Луковатый
Максим Николаев
Александр Торговкин
Павел Шариков
Светлана Юшина

Основная команда благодарит группу проверки и все национальные коллегии за их предложения и вклад. АНО «Коллегия экспертов по качеству программного обеспечения» (RSTQB) благодарит ООО «Байтэкс» за их вклад в разработку данной программы обучения.

Кроме того, основная команда выражает признательность и благодарность лидерам и членам рабочих групп за их раннее и постоянное руководство: Galit Zucker (генеральный секретарь), Graham Bath (председатель рабочей группы Специалист), Gary Mogyorodi (эксперт рабочей группы Глоссария) и Klaus Skafte (председатель Экзаменационной рабочей группы).

Следующие специалисты принимали участие в рецензировании, комментировании или обсуждении данной программы обучения или ее предшественников:

Иван Пчелкин	Наталья Стерхова	Ирина Яковлева
Matthias Hamburg	Chunhui Li	Blair Mo
Tal Pe'er	Wim Decoutere	Claude Zhang
Meile Posthuma	Nishan Portoyan	Francisca Cano Ortiz
Wojciech Becla	Jana Gierloff	Paul Weymouth
Nitzan Goldenberg	Jürgen Beniermann	Erwin Engelsma
Georg Haupt	Gary Mogyorodi	Péter Sótér
Bíró Ádám	Darvay Tamás Béla	Laura Albert
Gergely Ágnesz	Graham Bath	Mike Smith

Документ был официально выпущен Генеральной Ассамблеей ISTQB 21 октября 2022 года.

0. Предисловие к программе обучения

0.1 Цель этого документа

Эта программа представляет основу международной сертификации на квалификацию «Сертифицированный тестировщик игр ISTQB®». ISTQB® предоставляет эту программу:

1. Национальным коллегиям для перевода на национальный язык и аккредитации организаторов обучения. Национальные коллегии могут адаптировать программу обучения к особенностям местных языков и определить ссылки для адаптации к местным публикациям.
2. Экзаменационным комиссиям для формирования экзаменационных вопросов на местном языке, адаптированные к целям обучения каждого курса.
3. Организаторам обучения для разработки программы обучения и определения соответствующих методов обучения.
4. Кандидатам на получение сертификатов для подготовки к экзамену (в рамках программы обучения или независимо).
5. Международному сообществу разработки ПО и систем для повышения уровня профессионализма при тестировании ПО и систем, и использования как основы для книг и статей.

ISTQB® может разрешить другим лицам использовать эту программу обучения в своих целях при условии, что они обратятся за письменным разрешением и получат его.

0.2 Сертифицированный тестировщик игр

Квалификация тестировщика игр предназначена для всех, кто занимается тестированием программного обеспечения, кто хочет расширить свои знания в области тестирования игр, или для тех, кто хочет начать карьеру специалиста в области тестирования игр. Квалификация также предназначена для всех, кто занимается разработкой игрового программного обеспечения и хочет лучше понять тестирование игр.

В программе рассматриваются следующие основные аспекты тестирования игр:

- Технические аспекты,
- Методические аспекты,
- Организационные аспекты.

0.3 Карьерный путь тестировщиков

Основываясь на Базовом уровне, тестирование игр поддерживает определение карьерных путей профессиональных тестировщиков. Сертифицированный специалист «Тестировщик игр» будет иметь расширенные знания в тестировании, полученные в модуле «Базового уровня», что позволит ему эффективно работать в качестве профессионального тестировщика в игровом проекте.

Обладатели сертификата тестировщика игр могут использовать аббревиатуру CT-GaMe.

Пожалуйста, посетите [ISTQB-Web] для получения актуального обзора карьерного пути ISTQB®.

0.4 Бизнес-результаты

В этом разделе перечислены бизнес-результаты, ожидаемые от кандидата, имеющего сертификат «Тестировщика игр».

GaMe-1 Описывать базовые концепции игр и тестирования игрового программного обеспечения.

- GaMe-2 Определять риски, цели и требования к игровому ПО для нужд и ожиданий заинтересованных сторон.
- GaMe-3 Концептуально проектировать, реализовывать и выполнять базовые тесты игрового программного обеспечения.
- GaMe-4 Знать подходы к тестированию игрового ПО и их назначение.
- GaMe-5 Знать инструменты, поддерживающие тестирование игр.
- GaMe-6 Определять, как мероприятия по тестированию согласуются с жизненным циклом разработки программного обеспечения, и снижение стоимости разработки и публикации видеоигр.

0.5 Проверяемые цели обучения и когнитивные уровни знаний

Цели обучения поддерживают достижение бизнес-результатов, а также используются для разработки сертификационных экзаменов по тестированию игр. Каждая цель обучения соотносится с когнитивным уровнем (К-уровнем).

К-уровень, или когнитивный уровень, используется для классификации целей обучения в соответствии с пересмотренной таксономией Блума [Anderson01]. ISTQB® использует эту таксономию для разработки экзаменов по программам обучения (дополнительную информацию см. в Приложении А).

В этой программе рассматриваются три разных К-уровня (от К1 до К3):

К-уровень	Ключевое слово	Описание
1	Запомнить	Кандидат должен помнить или распознавать термин или понятие.
2	Понять	Кандидат должен выбрать объяснение утверждения, относящегося к теме вопроса.
3	Применить	Кандидат должен выбрать правильное применение концепции или метода и применить их к данному контексту.

В целом все части данной программы обучения содержат цели обучения для уровня К1. Другими словами, кандидат на сертификат должен распознать, запомнить и воспроизвести термин или понятие. Цели обучения для уровней К2 и К3 указаны в начале глав, которые их содержат.

0.6 Экзамен «Сертифицированный тестировщик игр»

Экзамен «Сертифицированный тестировщик игр» будет основан на этой программе обучения. Для ответов на вопросы экзамена может потребоваться использование материала, основанного на более чем одном разделе этой программы. Все разделы программы подлежат изучению, кроме введения и приложений. Стандарты и книги включены в качестве ссылок, но их содержание не подлежит изучению, за исключением того, что перечислено в самой программе из таких стандартов и книг.

См. [ISTQB_EXAM_S&R] для ознакомления со структурой и правилами экзамена «Сертифицированный тестировщик игр».

До сдачи сертификационного экзамена «Сертифицированный тестировщик игр» кандидатом должна быть получена сертификация ISTQB® Certified Tester Foundation Level [ISTQB_FL_SYL]. Кандидатам также настоятельно рекомендуется:

- иметь хотя бы минимальный опыт разработки или тестирования игрового программного обеспечения, например, 1 год опыта работы в качестве приемочного тестировщика игрового программного обеспечения.
- пройти курс, аккредитованный в соответствии со стандартами ISTQB® (одной из коллегий-членов, признанных ISTQB®).

0.7 Аккредитация

Коллегия-член ISTQB® может аккредитовать обучающие организации, учебные материалы которых следуют этой программе обучения. Обучающие организации должны получить руководство по аккредитации от коллегии-члена или органа, который проводит аккредитацию. Аккредитованный курс признается соответствующим этой программе, и в рамках курса разрешается сдавать экзамен ISTQB®.

Руководство по аккредитации для этого учебного плана соответствует общему Руководству по аккредитации, опубликованному Рабочей группой по управлению процессами и соответствию ISTQB®.

0.8 Поддерживаемые стандарты

В программе обучения «Тестирование игр» для сертифицированных тестировщиков есть ссылки на стандарты (например, IEEE, ISO и т.д.). Цель этих ссылок — предоставить основу (как в ссылках на ISO 25010 в отношении характеристик качества) или предоставить источник дополнительной информации, если читатель пожелает. Обратите внимание, что программа обучения использует стандарты в качестве справочных документов. Стандарты не предназначены для изучения. Дополнительную информацию о стандартах см. в разделе 8. Ссылки.

0.9 Актуальность программы обучения

Индустрия программного обеспечения быстро меняется. Чтобы справиться с этими изменениями и предоставить заинтересованным сторонам доступ к релевантной и актуальной информации, рабочие группы ISTQB® разместили ссылки на [ISTQB-Web], которые относятся к вспомогательным документам, изменениям в стандартах и новым событиям в отрасли. Эта информация не подлежит проверке в рамках этой программы обучения.

0.10 Уровень детализации

Уровень детализации этой программы обучения позволяет проводить согласованные на международном уровне курсы и экзамены. Для достижения этой цели учебная программа состоит из:

- Общих учебных целях, описывающих намерения сертифицированного тестировщика игр.
- Списка терминов, которые кандидаты должны запомнить.
- Целей обучения для каждой области знаний с описанием когнитивных результатов обучения, которые должны быть достигнуты.
- Описание ключевых понятий, включая ссылки на такие источники, как литература или стандарты.

Содержание программы не является описанием всей области знаний по тестированию игр; оно отражает уровень детализации, который необходимо охватить в учебных курсах по тестированию игр. Основное внимание уделяется областям тестирования и методам, которые можно применить к большинству игровых проектов.

0.11 Как организована эта программа обучения

Программа обучения сертифицированного тестировщика игр содержит семь глав, охватывающих знания, необходимые для того, чтобы стать специалистом по тестированию игр.

Заголовок верхнего уровня для каждой главы указывает минимальное время для этой главы; ниже уровня главы время не указано. Для аккредитованных учебных курсов программа требует, как минимум 15 часов и 25 минут обучения, распределенных по семи главам следующим образом:

- Глава 1: 75 минут Специфика тестирования игр

- Тестировщик изучает основные принципы, связанные с тестированием игрового программного обеспечения, причины, по которым требуется тестирование игр, каковы цели тестирования, а также разницу между тестированием игр и самой игрой.
- Тестировщик понимает процесс тестирования игры, основные действия и рабочие продукты.
- Глава 2: 180 минут Тестирование игровых механик
 - Тестировщик изучает различные игровые механики и подходы к их тестированию.
 - Тестировщик изучает как динамические, так и статические методы, используемые для тестирования игровых механик.
- Глава 3: 165 минут Тестирование графики
 - Тестировщик изучает особенности и типы графического содержания игрового продукта, а также лучшие практики тестирования графики, в том числе их инструментальную поддержку.
 - Тестировщик изучает зоны ответственности задействованных лиц.
- Глава 4: 190 минут Тестирование звука
 - Тестировщик изучает особенности и типы звукового содержания игрового продукта, а также лучшие практики тестирования звука, в том числе их инструментальную поддержку.
 - Тестировщик изучает зоны ответственности задействованных лиц.
- Глава 5: 65 минут Тестирование игровых уровней
 - Тестировщик изучает зависимость проектирования уровней и подходов к тестированию от жанра игры и их инструментальную поддержку.
 - Тестировщик изучает зоны ответственности вовлеченных лиц
- Глава 6: 95 минут Тестирование игровых контроллеров
 - Тестировщик изучает типы игровых контроллеров, включая связанные с ними дефекты и их возможные причины.
 - Тестировщик изучает подходы к тестированию контроллеров и их инструментальную поддержку.
- Глава 7: 155 минут Тестирование локализации
 - Тестировщик изучает основы и различия локализации и интернационализации, их риски, подходы к тестированию и инструментальную поддержку.
 - Тестировщик изучает зоны ответственности задействованных лиц.

1. Специфика тестирования игр – 75 минут (K2)

Ключевые термины тестирования

Тестирование во время игры, функциональное тестирование

Ключевые термины, специфичные для игр

Игра, многоплатформенность, предварительный этап, проектировщик видеоигр, этап концепта, этап постпроизводства, этап производства

Цели обучения для Главы 1

1.1 Основы тестирования игр

GaMe-1.1.1 (K1) Узнать цели и специфику тестирования игр

GaMe-1.1.2 (K2) Привести примеры продуктовых рисков в игровом ПО

GaMe-1.1.3 (K2) Привести примеры дефектов, специфичных для игрового ПО

GaMe-1.1.4 (K2) Обобщить, как могут быть снижены риски тестирования игр

GaMe-1.1.5 (K2) Сравнить мероприятия по тестированию игр и самой игры

1.2 Типовые роли команды разработки игрового ПО

GaMe-1.2.1 (K1) Узнать роль и задачи команды разработки игрового ПО

1.3 Мероприятия тестирования в жизненном цикле разработки игрового ПО

GaMe-1.3.1 (K1) Запомнить мероприятия тестирования в жизненном цикле разработки игрового ПО

1.1 Основы тестирования игр

1.1.1 Специфика тестирования игр

В тестировании программного обеспечения принято различать разные специализации. Одни тестировщики занимаются тестированием производительности приложений, другие сосредоточены на тестировании мобильных приложений, третьи ищут уязвимости в системах безопасности, а четвертые ищут дефекты в компьютерных играх.

Игры могут различаться по жанру и быть предназначены для одного или нескольких игроков. Однако подходы к тестированию игр можно считать независимыми от этих характеристик. Глава 2.1 этой программы обучения подробно описывает функциональные и нефункциональные подходы к тестированию игровых механик.

Важно отметить, что термины «игра», «компьютерная игра» и «видеоигра» рассматриваются в этой программе как синонимы. Это делается для того, чтобы текст был более удобочитаемым и соответствовал программе обучения. Игра может рассматриваться только как программное обеспечение, и ее нельзя путать со спортивными играми или азартными играми.

1.1.2 Риски игровых продуктов

Взрывное развитие индустрии видеоигр привело к их распространению на множестве различных платформ. Игры стали объемнее, сложнее и требовать больше технических ресурсов. При этом аудитория игроков значительно увеличилась, став более искушенной и требовательной к качеству игры.

Общие риски программных продуктов и проектов относятся как к видеоиграм, так и к разработке игр. В результате игры, как и другое программное обеспечение, подлежат тестированию. Некоторые риски специфичны для игрового программного обеспечения, и им следует уделять больше внимания. Такие риски включают, но не ограничиваются:

- Получение несправедливого преимущества путем жульничества.
- Зависимость успеха на рынке от субъективного мнения игроков.

- Проблемы многоплатформенности.
- Сложность прогнозирования эффективности игровой механики (см. главу 2.1).

Тестировщики могут найти дефекты в следующих областях:

- В архитектуре игрового программного обеспечения на ранних этапах жизненного цикла разработки.
- В графическом и звуковом оформлении.
- В тексте уже готовой к выпуску видеоигры.

1.1.3 Дефекты, специфичные для тестирования игр

Существует ряд дефектов, которые особенно часто встречаются в видеоиграх. Они включают:

- проблемы с графикой и анимацией,
- нарушение физических законов в игровом мире,
- дефекты искусственного интеллекта видеоигры,
- неточности в построении сюжета видеоигры,
- некорректную работу некоторых игровых объектов (например, персонажи видеоигр, появляющиеся из воздуха, не атакующие врагов, и парящие автомобили).

Хотя такие дефекты могут быть легко обнаружены, определить точные шаги для их воспроизведения часто бывает сложно. Это усложняет тестирование игр. Хороший тестировщик должен проектировать тесты в соответствии с предполагаемым игровым сценарием, проверять его соответствие спецификации, анализировать возможные отклонения от разработанного сценария и сообщать о последствиях этих отклонений.

Негативное тестирование

При тестировании игр, как и любого программного обеспечения, важно выявлять явные дефекты и стремиться найти неявные дефекты, возникающие в результате нестандартных действий пользователя. Например, вместо того, чтобы сражаться с противниками и искать ключ к запертой двери, пользователь может собирать ящики со всего уровня, делать из них лестницу и с их помощью перелезть через забор, тем самым избегая жесткой схватки, что не было задумано сценарием игры.

Иные дефекты могут возникнуть, если взаимодействие с другими объектами настроено неверно, например, для трехмерной модели здания, используемого в игре. Этот тип дефекта может позволить персонажу, управляемому игроком, пройти сквозь стену, что может испортить общий пользовательский опыт и дать игрокам незаконные игровые преимущества. Это особенно критично в многопользовательских играх. Спрятавшись за такой стеной, игрок будет скрыт от противников, но сможет по ним стрелять.

Подобные риски повышают важность негативного тестирования игровых проектов. Может оказаться большое количество игроков, которые не захотят играть так, как задумал разработчик, а попытаются «сломать» или обойти систему, чтобы легче выиграть и затратить меньше усилий. Для этого не всегда нужно использовать стороннее ПО или специальные команды. Может быть достаточно найти «правильный» дефект. Такие игроки намеренно ищут дефекты в логике игры и используют их в личных целях, например, для получения доступа к бесконечным ресурсам или мощному оружию. В монетизированных играх это также может привести к уменьшению прибыли владельца игры.

Зависимость от мнения игроков

Игровое программное обеспечение, как никакое другое, зависит от субъективных и часто эмоциональных суждений игроков. Для программного обеспечения, предназначенного для решения бизнес-задач, его функциональность является наиболее важным фактором. Для игрового

программного обеспечения наиболее важным фактором является уровень заинтересованности пользователя и его впечатления от игровой сессии. Игроки могут «закрыть глаза» на какие-то функциональные дефекты, но, если игра окажется скучной, однообразной или использует устаревшую графику, пользователи могут просто перестать ею пользоваться. Значительная часть аудитории видеоигр принимает решение о покупке или использовании продукта, основываясь в первую очередь на отзывах игровых критиков, рецензентов и других лидеров общественного мнения.

Многоплатформенность

Важной особенностью игрового программного обеспечения является то, что оно часто используется на нескольких платформах. Стремясь охватить как можно большую аудиторию, игровые студии/издатели выпускают ее на самых разных платформах, включая различные конфигурации персональных компьютеров (ПК), интернет, мобильные устройства и консоли.

Каждое обновление должно быть протестировано для всех доступных платформ. Это порождает серьезные риски и значительно увеличивает требуемое время тестирования.

Игра может хорошо работать на компьютере со средними характеристиками, но иметь множество различных дефектов на консолях последнего поколения. Дефекты также возможны из-за неэффективной связи между различными технологиями или неполных требований к портированию игры с платформы, для которой она изначально разрабатывалась, на другую платформу, принадлежащую сторонней (аутсорсинговой) организации.

Чтобы снизить эти риски, рекомендуется уделять больше времени разработке и тестированию игрового программного обеспечения, тестированию функциональности и производительности игры на множестве различных аппаратных конфигураций, а также проведению тестов с учетом особенностей каждой платформы.

Отдельно следует учитывать тестирование видеоигр на игровых консолях. Игровая приставка — это устройство, предназначенное исключительно для игр и не содержащее никакого другого программного обеспечения, как мобильные устройства или компьютеры.

Существует очень мало различий между платформами при тестировании видеоигр с использованием методов тестирования черного ящика. Однако у каждого производителя игровых консолей могут быть свои особые требования, которым должна соответствовать видеоигра, прежде чем ее можно будет опубликовать. Эти требования являются проприетарными документами, предоставляемыми разработчикам и издателям в соответствии с соглашениями о конфиденциальности. Требования могут состоять из нескольких категорий, и игра должна им соответствовать, чтобы не быть отклоненной производителем консоли [URL1], [URL2].

Поэтому тестировщики видеоигр на консолях должны проводить тесты на соответствие этим требованиям в дополнение к стандартным методам тестирования программного обеспечения.

Может потребоваться использование специального оборудования для обнаружения, выявления и понимания причин проявления дефектов. Это оборудование по сути является той же консолью, но предоставляет дополнительные режимы, помогающие в разработке и тестировании игр. Консоль регистрируется на сайте для авторизованных разработчиков и тестировщиков, после чего активируется и предоставляется доступ к специальным режимам.

1.1.4 Как тестирование снижает риски видеоигр

Большинство видеоигр, особенно с открытым миром, настолько сложны, что невозможно протестировать все возможные комбинации игровых объектов, событий и факторов. Даже проведение одиночных тестов каждой комбинации в течение всего жизненного цикла разработки программного обеспечения может быть длительным, а подтверждающее тестирование после изменений в следующей версии игры требует дополнительных усилий.

Управление рисками используется для эффективного распределения ресурсов и оценки возможных дефектов на основе вероятности их возникновения и влияния на игру.

Для каждого тестового сценария задаются два вопроса для определения приоритета теста:

1. Насколько вероятно, что игроки столкнутся здесь с дефектом?
2. Если они обнаружат дефект, как это повлияет на игроков и компанию, владеющую игрой?

Например, если игра имеет линейную структуру, и пользователь шаг за шагом открывает доступ к ее содержанию, то дефекты, возникающие во время первой игровой сессии, скорее всего, будут иметь более высокий приоритет (но не всегда критичность) для исправления, чем дефекты, которые происходят на пятидесятом часу игры. Если лояльный пользователь уже провел в игре много времени, он, скорее всего, более спокойно воспримет возникший дефект, и на общее впечатление пользователя от игры это не сильно повлияет. Другое дело, если такой дефект затрагивает всех новых игроков во время начального обучения. В этом случае многие игроки могут отказаться от игрового продукта.

Для снижения рисков с помощью тестирования важно выявлять группы дефектов, а не искать отдельные дефекты в игре. Это позволяет разработчикам одновременно решить проблему в коде или архитектуре игры и обеспечить повышение качества конечного продукта.

Группы дефектов представляют собой скопления дефектов в критических областях. В игровых продуктах есть области, в которых с наибольшей вероятностью могут быть дефекты, такие как графика, звук, локализация, клиент-серверное взаимодействие и аппаратное обеспечение [Nystrom14].

Важно завоевать лояльность игроков, когда они учатся играть или проходят обучение в игре. На этом этапе игроку доступно огромное количество игровых функций, чтобы он мог оценить продукт. Как результат, этот ранний этап обычно объявляется критически важным.

1.1.5 Различие между тестированием и «игрой»

Может сложиться впечатление, что суть работы тестировщиков компьютерных игр состоит в том, чтобы целыми днями играть онлайн с коллегами. Но это утверждение далеко от истины.

Тестировщикам редко удается просто поиграть, разве что во время знакомства с новым проектом или при тестировании во время игры. Большая часть их работы связана с проведением различных видов тестирования, которые во многом перекликаются с видами тестирования, применяемыми для другого программного обеспечения.

Обычный пользователь запускает игру, чтобы пройти ее, победить соперников или хорошо провести время, а тестировщик проверяет, соответствует ли игра требованиям, зафиксированным в спецификации. Более того, личные предпочтения тестировщика могут не соответствовать предполагаемой целевой аудитории самой игры. Тем не менее, тестировщик будет проходить один и тот же уровень снова и снова, чтобы проверить, например, правильно ли отображаются игровые объекты. Монотонность при выполнении этих повторяющихся действий, компенсируется удовлетворением, которое он получает от нахождения дефектов, способных нарушить задуманный разработчиком игровой процесс.

Связь с другими программами обучения

Подходы, обсуждаемые в этой программе обучения, применимы исключительно к тестированию игр. Они применяются как к тестированию программного обеспечения (например, к тестированию игровых механик), так и к тестированию оборудования (например, к тестированию контроллеров).

Некоторые аспекты тестирования игры не так уж отличаются от тестирования другого программного обеспечения, особенно при выполнении таких типов тестов, как регрессионное тестирование, тестирование производительности или тестирование практичности. Основные концепции

тестирования, методы нефункционального тестирования и особенности тестирования программного обеспечения на мобильных платформах подробно рассматриваются в других программах ISTQB®.

1.2 Типовые роли команды разработки игрового ПО

В большинстве случаев команда разработки игрового ПО включает в себя роли (но не ограничивается ими), которые принципиально не отличаются от тех, что встречаются в других областях разработки программного обеспечения. К ним относятся владельцы бизнеса, руководители проектов, аналитики, разработчики, проектировщики игр и тестировщики. В небольших командах один человек может выполнять несколько ролей одновременно. Ниже приводится краткое изложение основных функций, которые могут выполнять типовые роли в проекте игрового программного обеспечения. Дополнительные роли, которые могут быть включены в команду для специальных областей разработки игр, описаны в соответствующих главах этой программы обучения.

Роль	Ответственность
Владелец бизнеса	<ul style="list-style-type: none">• Развивает бизнес• Определяет маркетинговую политику• Запускает новые продукты• Может принимать участие в построении команды
Руководитель проекта	<ul style="list-style-type: none">• Ставит задачи и контролирует их выполнение• Отслеживает этапы жизненного цикла разработки программного обеспечения
Аналитик	<ul style="list-style-type: none">• Выявляет требования к продукту• Работает с оригинальной спецификацией• Обновляет спецификации, поддерживает их в актуальном состоянии
Разработчик	<ul style="list-style-type: none">• Разрабатывает и отлаживает код• Исправляет дефекты• Разрабатывает компонентные тесты
Проектировщик игр	<ul style="list-style-type: none">• Проектирует игровые механики, баланс, сюжет, уровни, экономику игры• Создает документацию игры• Продумывает и разрабатывает содержание игры
Тестировщик	<ul style="list-style-type: none">• Разрабатывает тестовую модель и стратегию тестирования• Проектирует и подготавливает тестовые данные• Рецензирует спецификации• Выполняет различные виды тестирования• Находит и анализирует дефекты

1.3 Мероприятия тестирования в жизненном цикле разработки игрового ПО

Этап концепта

Основная цель этого этапа – достижение единого видения будущего продукта между командой разработки и заказчиком. Идеи оформляются в виде документов, в которых кратко описывается общее видение игры, ее жанр, предустановки, суть игрового процесса, механики и основные особенности.

Артефакты, которые могут подвергаться рецензированию на данном этапе, включают:

- Видение,
- Концепт,
- Продуктовые и технические ограничения.

Тестовое обеспечение, которое создается на данном этапе, включает:

- Стратегию тестирования,
- Проект плана тестирования,
- Высокоуровневую оценку затрат на тестирование,
- Тестовое окружение.

Предварительный этап

Основная цель этого этапа — создание прототипа будущей игры, который является начальной рабочей версией продукта. В прототипе реализован основной игровой процесс в черновом качестве. На этом этапе проверяются основные игровые механики и исходные гипотезы (см. главу 2), а также оцениваются технические возможности команды. В тестировании прототипа могут принимать участие тестировщики, проектировщики игры и другие члены команды разработки.

Команда разработки рецензирует предварительные требования, чтобы оценить возможность реализации, а команда тестирования рецензирует их с целью выявить взаимодействие механик друг с другом и функциональность собранного прототипа. Кроме того, тестировщики участвуют в создании плана тестирования и документов графика тестирования.

По мере создания документации к игреразрабатывается и обновляется план тестирования, а документация рецензируется. Эти меры снижают технические риски продукта на предварительном этапе.

Артефакты, которые могут подвергаться рецензированию на данном этапе, включают:

- Спецификации требований,
- Прототип игры,
- Тестовые артефакты, созданные на предыдущем этапе.

Тестовое обеспечение, которое создается и обновляется на данном этапе, включает:

- Стратегию тестирования,
- План тестирования,
- График тестирования,
- Тестовое окружение.

Этап производства

На этапе производства создается игровой продукт. Это самый долгий этап, который обычно разбивается на отдельные части для альфа- и бета-версий игры.

На этом этапе тестировщики дорабатывают чек-листы и наборы тестовых условий, а также выполняют функциональное и нефункциональное тестирование реализованных компонентов продукта. Выполняется тестирование интеграции компонентов, системное и приемочное тестирование (см. [ISTQB_FL_SYL]). На этом этапе выявляется наибольшее количество дефектов, в результате чего тестировщики проводят подтверждающее и регрессионное тестирование.

Тестовое обеспечение, которое создается и обновляется на данном этапе, включает:

- План тестирования,
- График тестирования,

- Тестовые сценарии, тестовые условия, чек-листы, наборы тестов, автоматизированные сценарии тестирования,
- Отчеты о дефектах,
- Отчетность тестирования.

Этап постпроизводства

Этап постпроизводства для тестирования игр включает в себя общие мероприятия по тестированию в период сопровождения программного обеспечения. Обычно он включает тестирование обновлений и регрессионное тестирование, как описано в [ISTQB_FL_SYL].

2. Тестирование игровых механик - 180 минут (К3)

Ключевые термины

Свободное тестирование, функциональное тестирование

Ключевые термины, специфичные для игр

Игровое состояние, игровые механики, клиентские механики, ключевые механики, мета-механики, предварительный этап, проектировщик видеоигр, серверные механики, этап концепта, этап постпроизводства

Цели обучения главы 2

2.1 Игровые механики

GaMe-2.1.1 (K2) Классифицировать виды игровых механик

GaMe-2.1.2 (K2) Различить тестирование игровой и неигровой механики

GaMe-2.1.3 (K2) Различить тестирование ключевой механики и мета-механики

GaMe-2.1.4 (K2) Различить тестирование клиентской, серверной и клиент-серверной механик

GaMe-2.1.5 (K2) Привести примеры дефектов игровой механики

2.2 Подходы к тестированию игровой механики

GaMe-2.2.1 (K2) Обобщить основные подходы и объекты тестирования на разных этапах создания игрового продукта

GaMe-2.2.2 (K2) Выделить важность тестирования игровых механик

GaMe-2.2.3 (K2) Выделить важность рецензирования документации, описывающей игровые механики

GaMe-2.2.4 (K3) Применить фундаментальные подходы тестирования игровых механик

2.1 Игровые механики

2.1.1 Типы игровых механик

В главе 1 упоминалось, что для игр одной из важных составляющих является стремление вызвать интерес пользователей, с целью удержания и увеличения аудитории. Игроки оценивают графику игры, сам игровой процесс и механики, составляющие игровой процесс.

Игровые механики лежат в основе любой игры. Они являются средством взаимодействия между игрой и пользователем. Механики учитывают влияние и обратную связь между игрой и пользователем, а также изменения игрового состояния в рамках заданных требований. Игровые механики можно разделить на разные типы в зависимости от различных аспектов и целей игры.

В зависимости от типа взаимодействия игрока с механикой:

- Игровые механики используются, когда пользователь сознательно взаимодействует с игровой системой, основывая свои действия на доступной информации. Пользователь может влиять на изменение игрового состояния, при этом четко видя результат своих действий.
- Неигровые механики используются, когда пользователь не может или может лишь частично влиять на состояние игровой среды своими действиями. Пользователь не осознает своего влияния, потому что от него скрыта часть игровой механики.

В зависимости от влияния на основной игровой процесс:

- Ключевые механики составляют основу игры и определяют действия, которые пользователь осуществляет на протяжении всей игры. Они направлены на то, чтобы пользователь приобрел конкретные впечатления, получение которых было заложено в игру ее создателями.
- Мета-механики находятся вне основного игрового процесса, но могут влиять на него. Они направлены на то, чтобы заставить игрока делать то, что хочет проектировщик видеоигр

(например: вернуться в игру, продолжить играть или совершить покупки). Эти механики можно комбинировать с ключевыми механиками, чтобы сделать игру более интересной.

В зависимости от архитектуры строения игры:

- Клиентские механики связаны с обработкой действий пользователя, происходящих исключительно на устройстве пользователя.
- К серверным механикам относятся механики, которые обрабатываются только на игровом сервере.
- К клиент-серверным механикам относятся механики, которые обрабатываются как на устройстве пользователя, так и на сервере. При этом происходит постоянный обмен данными между сервером и клиентом.

2.1.2 Разница между тестированием игровых и неигровых механик

Пользователь не взаимодействует напрямую со всеми механиками, реализованными в игре.

Игровые механики видны и понятны пользователю. Часто они напрямую влияют на достижение цели игрового процесса. Например, игровой персонаж должен собрать монеты в течение заданного времени. Функционал, отвечающий за передвижение и сбор монет, а также увеличение количества монет, является примером игровой механики.

Неигровые механики обычно скрыты от пользователя, хотя могут быть запущены действием самого пользователя. Например, когда пользователь совершает покупку какого-либо товара в игре, запускается поток неигровых механик, например:

1. Делается запрос к базе данных для получения состояния игрового аккаунта.
2. Совершается транзакция по списанию средств.
3. Покупка протоколируется, то есть информация о совершенном действии записывается в специальный текстовый файл.
4. Электронное письмо с деталями покупки отправляется пользователю.

Дополнительно можно зафиксировать дату покупки и тип приобретаемого товара, чтобы отправлялась автоматическая рассылка о скидке на аналогичный товар. Неигровые механики должны быть указаны как явные требования, поскольку они не очевидны.

Тестирование игровой и неигровой механик является функциональным тестированием, как описано в [ISTQB_FL_SYL].

2.1.3 Разница между тестированием ключевых механик и мета-механик

Разница в подходах к тестированию ключевых механик и мета-механик зависит от их воздействия на игровой процесс.

Удаление ключевой механики из игры полностью меняет игровой процесс. Например, если из гоночного симулятора убрать движение автомобиля, данную игру уже нельзя будет отнести к разряду «гонок».

Примерами мета-механики могут быть: выбор модели автомобиля с определенными параметрами в зависимости от типа гоночной трассы, покупка или экипировка персонажа игрока новой броней перед вступлением в бой. Удаление мета-механики из игры редко меняет суть всей игры. В дополнение к функциональным требованиям ключевые механики и мета-механики могут иметь и нефункциональные требования, которые напрямую влияют на то, как их следует тестировать.

На примере изменения внешности игрового персонажа видно, что для мета-механики очень важно, чтобы данный функционал работал корректно. Для игр, в которых не так важно быстрое действие, работа механики с небольшой задержкой может не считаться дефектом. Однако для ключевых механик персонажа, прыгающего или использующего способности в бою, такая задержка была бы критической, потому что могла бы помешать игроку выиграть. Этот пример показывает, что тестирование производительности также важно для ключевых механик.

Суть ключевых и мета-механик также влияет на их тестирование. Как правило, их проверка обычно начинается на разных этапах разработки игры. Ключевые механики, такие как перемещение игрового персонажа, получение награды и т.д., появляются в ранних версиях продукта. При этом начинается проверка функциональной корректности и уровня заинтересованности целевой аудитории.

Мета-механики обычно добавляются в игру и начинают тестироваться позже, на этапе производства. На этапе бета-тестирования происходит оценка их влияния на среднюю продолжительность игровой сессии, количество приглашенных друзей, размер среднего платежа, наиболее популярные игровые продукты и т.д. Также для определения наиболее эффективной механики часто используются Альфа/Бета тестирование.

2.1.4 Разница между тестированием клиентской, серверной и клиент-серверной механик

Необходимость тестирования клиентской, серверной и клиент-серверной механик зависит от архитектуры самой игры. Каждый из этих типов механик требует разного подхода к тестированию в зависимости от специфики их реализации.

Клиентские механики обрабатываются на устройствах самих игроков, поэтому пользователь может получить доступ ко всем данным, хранящимся на клиенте. Некоторые игры могут полностью запускаться на клиенте и вообще не использовать сервер, то есть вся механика выполняется на стороне клиента. Такие игры обычно однопользовательские и не требуют подключения к Интернету.

Игровой клиент содержит код, который выполняется на устройстве игрока, а также все параметры, формулы расчета урона, игровые уровни, модели персонажей и другое. В результате игрок по желанию может изменить все данные в игровом клиенте. Модифицируя код клиента, пользователь может, например, увеличить скорость своей техники, значение здоровья персонажа, величину награды за выполнение задания и другое.

Такие изменения клиента могут дать игроку преимущество в игре, но в одиночных играх это обычно не считается серьезным недостатком. Например, в однопользовательских играх, может быть реализована механика фехтования с компьютерным противником. В этом случае параметры оружия не будут проверяться на сервере. Поэтому, если конкретный игрок сделает изменения в игровом клиенте и увеличит урон от своего меча, это изменит только его личный игровой процесс. Это не затронет интересы других независимых игроков, которые играют в ту же игру на своих устройствах.

Клиентские механики также могут присутствовать в многопользовательских играх, но они все равно не повлияют на других игроков. Например, механика просмотра карты уровня, открытие инвентаря персонажа и другое обрабатывается только на стороне клиента и проверяется функциональными тестами без участия сервера.

Тестирование клиентской механики обычно проводится методом черного ящика. При этом тестировщику может потребоваться проверить пользовательский интерфейс. В этом случае сам пользовательский интерфейс можно использовать при тестировании игры для получения реальных результатов.

Серверные механики реализованы исключительно на стороне сервера. Благодаря этой особенности логика механики защищена от пользовательского вмешательства. Игроки не могут напрямую влиять на процессы, которые выполняются на удаленном сервере, поэтому разработчики переносят туда самые важные части расчетов игровой логики.

Тестирование серверных механик наиболее важно для многопользовательских онлайн-игр. Например, когда игрок начинает командное соревнование в режиме игрок против игрока (PvP), на сервере проводится подбор союзников и противников с примерно равными игровыми рейтингами и уровнями персонажей. В противном случае соревнование может оказаться неинтересным и неудобным для всех участников.

Подход к тестированию функционала, реализованного на сервере, отличается от тестирования клиентских механик. Здесь, тестировщику обычно не нужен пользовательский интерфейс. Тестирование происходит в серверных консолях или с помощью специальных инструментов.

Тестировщик тратит большую часть своего времени на анализ логов или взаимодействие с базой данных.

Для серверных механик наиболее важными видами тестирования являются функциональное тестирование, тестирование производительности и тестирование безопасности.

Клиент-серверные механики включают как клиентскую, так и серверную стороны, и сочетают черты двух ранее упомянутых типов механик. Одна часть логики обрабатывается на устройствах игроков, другая - на удаленном сервере. Части таких механик постоянно общаются друг с другом, обмениваясь данными. В многопользовательской игре, сервер получает данные от клиента, а затем рассылает результат действий всем игрокам сервера, на которых эти действия влияют.

Данные, поступающие на сервер, проходят обязательную проверку. Например, если пользователь хочет что-то купить в игровом магазине, то сервер должен проверить наличие достаточного количества игровой валюты, доступной для покупки, которая влияет на прогресс игрока. Даже если игрок попытается модифицировать клиентскую часть на своем устройстве и отправить на сервер ложные параметры, чтобы получить игровое преимущество, сервер заменит их правильными данными.

Когда тестировщики тестируют клиент-серверные механики, они создают один или несколько тестовых сценариев таким образом, чтобы покрыть тестовыми сценариями максимальное количество разных механик.

Примером такого теста может быть последовательность действий, во время которых клиент заходит на игровой сервер и убивает противника, которым управляет компьютер. Несмотря на небольшое количество необходимых действий, составляющих этот тест, здесь используются практически все механики клиент-серверного взаимодействия, а именно:

- Отправка запроса от клиента на сервер, который отвечает за авторизацию,
- Выполнение запроса с сервера авторизации к базе данных, выполнение авторизации или восстановление предыдущего игрового состояния,
- Выбор уже загруженных на сервер карт, создание новых игровых механик на сервере,
- Подключение клиента к серверу игровой механики, на котором находится карта,
- Обеспечение просмотра игрового персонажа и игровых объектов в непосредственной близости от местонахождения персонажа, во время создания сервером объектов и отправка команды отображения местонахождения персонажа клиенту,
- Перемещение персонажа по карте при отображении новых игровых объектов,
- Нахождение мобильного объекта (моба) и его атаки, отображение атаки на клиенте,
- Получение команды "атака" от сервера,
- Проверка обеспечения возможности атаки таким образом, чтобы персонаж и моб находились на приемлемом для атаки расстоянии, и чтобы между персонажем и мобом не было объектов, препятствующих проведению атаки,
- Когда атака выполнена, сервер отправляет сообщение мобу о получении урона моб
- Подсчет на сервере количества урона, получаемого мобом и уменьшение значения параметра, отвечающего за здоровье моба.
- Отображение урона, нанесенного игроку мобом.

2.1.5 Примеры дефектов игровой механики и возможные причины их возникновения

Игровые механики включают воздействия на игровые объекты и обратную связь, сигнализирующую о результате этих действий. В совокупности это создает уникальный характер игры с соответствующей игровой динамикой. Чаще всего в играх используется широкий спектр воздействий и элементов обратной связи.

Следующие виды дефектов связаны с игровыми механиками:

- Функциональные дефекты в механиках,
- Дефекты проявления механик,
- Дефекты эффективности механики, проявляющиеся в конкретной игровой среде.

Функциональные дефекты игры упоминаются чаще всего, когда игроки говорят о дефектах игровой механики. Например, оружие не перезаряжается, добыча не собирается, изображение не приближается при использовании бинокля. Такие дефекты обычно возникают из-за ошибок, допущенных разработчиком в коде игры. Они довольно заметны и их легко обнаружить и исправить.

Дефекты проявления механик связаны с получением реакции от игры при использовании той или иной механики. Некорректна ситуация, например, когда игрок не понимает, почему игра закончилась. Поэтому игровые механики часто сопровождаются визуальными и звуковыми эффектами. Это могут быть визуальные эффекты взрыва, звук счетчика времени или даже текстовое сообщение. Проверяется само наличие реакции при использовании механики и оценивается обоснованность ее наличия или отсутствия. Работа и корректность эффекта проверяется другими видами тестирования (графическое или звуковое).

Третий тип дефектов связан с эффективностью и областью действия механики. Механика может быть эффективной сама по себе, но может перестать работать в игровом процессе. Чтобы избежать такой ситуации, механика тестируется совместно с другими механиками, объектами и прочим необходимым содержанием на уровне игры. Поэтому интеграционное тестирование проводится вместе с тестированием удобства использования, чтобы проверить эффективность механики в игровом процессе.

Например, проектировщик видеоигр может добавить в игру двух мобов с разным поведением и характеристиками. Первый – быстрый и прыгучий, но наносит небольшое количество урона, а второй – медленный, неповоротливый, но бьет точно и сильно. Оба противника имеют свои преимущества и недостатки. У каждого может быть свой алгоритм поведения и по идее они должны представлять примерно одинаковую опасность для игрока. Однако, если большая часть игрового уровня состоит из различных холмов, на которые может забираться игровой персонаж, ситуация меняется. Находясь на высоте, недоступной для медленного врага, игрок может смело атаковать противника. В этом случае пользователю будет сложно противостоять другому противнику, который также может использовать холмы, но передвигается с большой скоростью и высоко прыгает. Решением этой проблемы может быть как изменение самой механики, так и изменение расположения объектов на уровне.

Сложность подобных тестов связана в первую очередь с трудностью прогнозирования подобных ситуаций и того, как в них будет работать механика. Поэтому поиск проблем взаимодействия механики и объектов окружения часто осуществляется в рамках свободного тестирования на этапе бета-тестирования с большой группой игроков.

2.2 Подходы к тестированию игровых механик

2.2.1 Процедуры и подходы к тестированию игровых механик на протяжении всего жизненного цикла разработки программного обеспечения игрового продукта.

Тестирование механик в играх может проводиться на разных этапах разработки программного обеспечения с использованием различных подходов к тестированию.

На этапе создания прототипа игры реализуются только ключевые механики. Здесь основными задачами тестировщика являются проверки игровых механик по следующим характеристикам:

- Функциональная правильность,
- Уровень заинтересованности игроков,
- Привлекательность для игроков.

Все механики и правила их работы пересматриваются, чтобы убедиться, что они полностью и однозначно описаны в документации игры. Должны быть проработаны все допустимые варианты взаимодействия пользователя с механикой, а также определено влияние каждой механики на весь игровой процесс.

На этапе производства, внедренные механики подвергаются функциональному тестированию, чтобы убедиться, что они соответствуют заявленным требованиям. Как правило, тестирование здесь происходит с использованием наборов тестовых условий и чек-листов. Последовательность тестовых сценариев должна быть направлена на максимальное покрытие всех аспектов механик. Исследовательское тестирование и свободное тестирование являются наиболее эффективными для проверки взаимодействия различных механик друг с другом. Особенно это проявляется в масштабных многопользовательских играх, где количество способов взаимодействия между механиками и различными компонентами игры становится слишком большим, чтобы его можно было отразить в тестовых условиях.

В рамках нефункционального тестирования проверяется влияние механик на общую производительность игры. Тестируется использование ресурсов (например, ОЗУ, ЦПУ и т.д.) и фактический результат (например, возникновение зависаний в игровом клиенте, значительное снижение частоты кадров и т.д.). Совместимость различных компонентов и программного обеспечения подлежит проверке, особенно с антивирусными программами.

Если для игры требуется наличие сервера, то производительность сервера также тестируется. Тестирование производительности планируется и выполняется на этапе производства, когда реализуются запланированные механики, которые могут повлиять на производительность. Подробности планирования и выполнения тестирования производительности описаны в [ISTQB_FL_PT].

Значительное внимание уделяется мета-механикам при проведении бета-тестирования с большим количеством игроков. Оценивается удобство и понятность игры для игроков. Что касается тестирования неигрового ПО, основной целью данного тестирования является не поиск явных функциональных дефектов, которые в идеале должны быть выявлены на предыдущих этапах, а получение обратной связи от конечных игроков. Однако конечные игроки также могут участвовать и оставлять отзывы в более ранних альфа-тестах, например, в больших многопользовательских играх.

После того, как игра выпущена на рынок, задача тестировщиков состоит в том, чтобы обработать и проверить наличие дефектов в механиках, о которых сообщают игроки, а также протестировать любые другие механики, которые добавляются в игру.

2.2.2 Важность тестирования игровых механик

Игровые механики являются основой игры. Они формируют основные навыки игрока, которые необходимы для поддержания его интереса. Поскольку игровые механики являются сутью любой игры, проверка их корректной работы является первоочередной задачей для тестировщика, а дефекты, обнаруженные в ходе такого тестирования, как правило, наиболее критичные.

Дефект в ключевых механиках, связанный с неработающим прыжком персонажа, может привести к невозможности преодолеть определенное препятствие и, как следствие, завершить игру.

Но даже если механики редко встречаются в игре или связаны с действиями, ненужными для прохождения игры (например, умение персонажа ездить верхом), их неправильная работа негативно скажется на восприятии игры в целом.

2.2.3 Важность проверки игровых механик

Важным этапом создания игры является разработка документации игры, в которой, помимо прочего, описываются игровые механики.

Проверка спецификации игровых механик на ранней стадии разработки помогает в дальнейшем избежать многих проблем. Эти проблемы могут быть связаны как с самим процессом разработки, так и с реакцией игроков на реализованную игровую механику в конечном продукте.

При проверке тестировщик должен обратить внимание на следующие характеристики качества:

- Полнота описания,
- Соответствие реальности,

- Структура и удобство навигации в документе.

Проблемы, которые может предотвратить проверка, в процессе разработки игры:

- Неправильное понимание разработчиком сути механик и неверная оценка требуемых усилий, ввиду недостаточно четкого, подробного и однозначного описания механик,
- Несовместимость новых механик с уже существующими,
- Невозможность реализовать механики в конкретном проекте.

Проблемы, связанные с восприятием игроками механик, которые может предотвратить проверка:

- Общая неуместность механик в конкретной игре,
- Неинтересные механики,
- Несбалансированные механики,
- Слишком редкое использование механик в игре.

2.2.4 Тестирование игрового состояния после возобновления сеанса и при неактивности пользователя

Игровое состояние

Под игровым состоянием понимается значения всех параметров и переменных, описывающих все объекты в игре, в конкретный момент времени.

Чем сложнее игра и чем больше возможностей для действий она предоставляет игроку, тем больше параметров имеет каждый объект.

Все параметры можно условно разделить на видимые и скрытые.

Информация о значениях видимых параметров явно доступна игроку. Она может включать:

- Уровень внутриигрового опыта пользователя за выполненное задание,
- Характеристики выбранного оружия,
- Количество разрешенных предметов в инвентаре,
- Стоимость товара во внутриигровом магазине.

Скрытые параметры могут использоваться разработчиками в качестве воздействия на ход игры, без привлечения внимания игрока. Например, помимо известных игроку показателей атаки и защиты, у персонажа может быть параметр точности, который скрыт от игрока и определяет вероятность попадания выстрела в цель. Зачастую игрок не подозревает о существовании таких параметров и на что именно они влияют.

Скрытые параметры могут отвечать за разные аспекты игры:

- Шанс выпадения, то есть шанс получить в награду случайный предмет,
- Скорострельность или скорость передвижения техники по разным типам поверхностей,
- Влияние выбора игрока на развитие сюжета игры.

Набор скрытых и явных параметров однозначно определяет игрового персонажа, которым управляет игрок, и другие объекты.

Цели тестирования игрового состояния

Важным моментом при тестировании параметров игры является проверка того, действительно ли игра использует фактические значения параметров.

В случае явных параметров тестирование выполняется путем визуального сравнения отображаемого значения с фактически используемым. Например, если игровой персонаж с параметром атаки, равным единице, нанес противнику десять единиц урона, то после того, как игрок

увеличил параметр атаки до двух, количество нанесенного этому же врагу урона также должно увеличиться и стать, скажем, двадцать единиц.

Такой тест особенно важен, если информация о новом количестве урона доступна игроку, в виде, например, игровой подсказки: “Каждая единица параметра атаки добавляет 10 единиц урона”. При этом игрок также сможет сравнить значение, указанное в подсказке, с реальным уроном.

Обычный игрок как правило ориентируется только на адекватность и постоянство изменения урона, не учитывая конкретных значений параметров. Такой игрок может не считать дефектом нанесение неправильного количества очков урона, но, если урон не изменится или уменьшится, это обязательно будет замечено.

Профессиональный тестировщик руководствуется формулами и принципами расчета изменения параметров, указанными в документации игры. Если эту информацию нельзя получить из документации, тестировщик, как и игрок, оценивает ее приблизительно, исходя из здравого смысла.

Для проверки скрытых параметров тестировщику иногда может потребоваться доступ к базе данных игры. С его помощью тестировщик может сравнить фактические значения параметров с ожидаемыми.

Кроме того, если предполагается, что пользователь может изменять явные параметры своего персонажа во время игры, то проверяется сама возможность изменения параметров и корректность их отображения.

Тестирование игрового состояния тесно связано с сохранением и загрузкой игры.

Сохранение и загрузка

Многие игры невозможно пройти за одну сессию, в общем и целом, они могут включать и бесконечный игровой процесс. Поэтому в играх часто используются функции сохранения и загрузки игрового состояния, чтобы игроку не приходилось каждый раз начинать с самого начала.

Другими причинами использования сохранений являются побуждение пользователя исследовать игровой мир, находить различные способы решения проблем, предлагаемых игрой, и даже управлять сложностью игры. Если у игрока есть возможность сохраниться перед трудным препятствием, цена его ошибки будет ниже, чем если бы неверное действие полностью свело на нет его игровой прогресс.

Сохранение — это информация о состоянии игры, которая хранится на жестком диске или в облаке. Сохранение может быть представлено в виде файла, содержащего информацию о текущем состоянии игры. Размер такого файла в несколько раз меньше самой игры и обычно содержит следующую информацию:

- Список объектов, для которых необходимо хранить информацию о состоянии,
- Список изменяемых параметров для каждого объекта,
- Уникальный идентификатор для каждого параметра, по которому игра может определить, что с ней происходит в данный момент.

В качестве объекта может выступать практически любая переменная, состоянием которой управляет игровая система и которую нужно запоминать. Например, уровень здоровья игрового персонажа, количество изученных способностей и координаты персонажа на игровой карте.

Сохранение типов и областей тестирования

Метод сохранения, формат файла сохранения и список сохраняемых параметров могут отличаться от игры к игре. За всю историю видеоигр было придумано большое количество подходов к сохранению прогресса. Наиболее часто используемые описаны ниже.

Использование контрольных точек для сохранения

Сохранение происходит автоматически в указанные разработчиком моменты. При достижении каждой следующей контрольной точки информация о текущем прогрессе игрока перезаписывается.

Игровое состояние, сохраненное в контрольных точках, сохраняется только для текущего игрового сеанса и удаляется по окончании игры.

Иногда игра явно не показывает, что пользователь достиг контрольной точки, но прогресс сохраняется. Для проверки правильности сохранения и загрузки, тестировщику необходима документация игры, в которой указан список и точное расположение всех существующих контрольных точек. При использовании других способов сохранения эта информация в документации может отсутствовать или излагаться менее подробно.

Использование стационарных сохранений

В определенных местах игры разработчики размещают специальные точки, в которых игрок при желании может вручную сохранить свой текущий прогресс. Эти точки обычно находятся между несколькими игровыми объектами, что позволяет избежать создание больших файлов сохранения.

Областью проверки в этом методе является возможность сохранения в каждой точке, а также возможность повторного сохранения и загрузки каждого сохраненного состояния.

Автосохранение

Это комбинация контрольных точек и стационарных сохранений. Прогресс игрока автоматически сохраняется на протяжении всего игрового процесса в определенные моменты. Например, во многих играх есть функция автосохранения при выходе. При запуске игры пользователь может начать новую игру или загрузить ранее сохраненную.

Задача тестировщика в этом случае - проверить, произошло ли сохранение и есть ли возможность загрузки достигнутого состояния игры во всех точках.

Ручное или свободное сохранение

Пользователь сохраняет игру в любой момент с помощью специального элемента в игровом меню. Быстрое сохранение может быть реализовано, когда сохранение и загрузка происходит нажатием одной клавиши или выполнением одного клика.

Как и при автосохранении, при ручном сохранении создается специальный файл, который содержит информацию об игровом состоянии. Тестирование в данном случае включает проверку правильности загрузки информации из файла, правильности имени файла, расположение файла в структуре операционной системы, корректности работы в новой версии игры и даже возможности использования файла сохранения, например, на другом компьютере.

Для игр, подходящих под следующее описание, достаточно протестировать правильную загрузку необходимого игрового уровня:

- Игра предполагает единственно правильный путь и способ прохождения (например, только один правильный выход из лабиринта),
- Игровой персонаж, контролируемый игроком, не меняется от уровня к уровню (например, у него не увеличивается сила удара, дальность прыжка, скорость передвижения по уровню и другие характеристики, которые могли бы увеличить скорость прохождения уровня).

В играх, подходящих под следующее описание, сохранения содержат больше уникальной информации об игровой сессии, уровне, персонаже и т.д.:

- Игровой персонаж может иметь множество параметров (например, скорость передвижения, дальность и высота прыжка, сила удара, инвентарь, внешний вид трехмерной модели и другие характеристики)
- Описанные выше параметры могут меняться в зависимости от внутриигровых действий и решений игрока (например, игрок находит артефакт в лабиринте и скорость передвижения увеличивается).

Для таких игр тестировщикам необходимо проверить правильность загрузки необходимого уровня игры, уникальной информации, исключительной информации для каждого отдельного игрока и

каждого обособленного игрового персонажа. При тестировании сохранения тестировщику необходимо проверить следующее:

- Игрок находится на том игровом уровне, на котором он закончил предыдущую игровую сессию,
- У игрового персонажа есть артефакт,
- Параметр скорости игрового персонажа действительно увеличен из-за артефакта.

3. Тестирование графики - 165 минут (К3)

Ключевые термины тестирования

Тестирование во время игры

Ключевые термины, специфичные для игр

Анимация, визуальный эффект (VFX), игровой уровень, коллизия, маппинг, область попадания, освещение сцены, привязка модели персонажа к скелету (скиннинг), создание скелета модели (ригинг), текстура, трехмерная модель, уровень детализации (LoD)

Цели обучения Главы 3

3.1 Принципы и концепции игровой графики

GaMe-3.1.1 (K2) Объяснить особенности графического содержания игрового продукта

GaMe-3.1.2 (K2) Классифицировать типы дефектов графического содержания

3.2 Подходы к тестированию графики в игровых продуктах

GaMe-3.2.1 (K2) Обобщить основные подходы к художественному тестированию

GaMe-3.2.2 (K2) Обобщить основные подходы к техническому тестированию

GaMe-3.2.3 (K2) Обобщить основные подходы к тестированию игрового процесса

3.3 Выполнение тестирования графики

GaMe-3.3.1 (K3) Применить фундаментальные подходы тестирования графики

GaMe-3.3.2 (K2) Объяснить важность проверки графики на историческую достоверность

3.4 Инструментальная поддержка тестирования графики

GaMe-3.4.1 (K2) Обобщить использование инструментов тестирования графики

3.1 Принципы и концепции игровой графики

3.1.1 Особенности графического содержания игрового продукта

Любой игровой продукт, будь то небольшая мобильная игра или многоплатформенная высокобюджетная игра, имеет большое количество различных графических элементов, которые в совокупности позволяют пользователю визуализировать процесс игры.

В процессе создания графических объектов для игр различные специалисты выполняют те или иные функции:

Р о л ь	О т в е т с т в е н н о с т ь
Художник	Создает графическое содержание для игры
Проектировщик трехмерных моделей	Создает трехмерные модели игровых объектов
Художник по текстурам	Создает текстуры для игровых объектов
Специалист по анимации	Работает над анимацией игровых объектов
Технический художник	Проводит технические испытания
Тестировщик	Тестирует графические объекты в движении видеоигры

Тестировщику игрового продукта, независимо от бюджета игрового продукта и количества графического содержания, необходимо тестировать различные его области, которые частично, полностью или по отдельности присутствуют в каждой игре.

Рассмотрим основные направления игрового продукта для тестирования.

Уровни (игровые карты)

Уровни — это отдельная область виртуального мира игры, которая обычно представляет собой определенную локацию, например, здание или город. Этот термин пришел из ранних настольных ролевых игр (RPG), где он относился к уровням подземелья (то есть к среде, в которой проходила большая часть игры). Игроки начинали в глубинах подземелья (уровень 1) и должны были выбраться на поверхность, проходя все уровни, которые становились все более сложными, пока они не достигали последнего (например, уровня 100), тем самым побеждая в игре. Сейчас уровни в том или ином виде присутствуют почти в каждой видеоигре.

Модели

Модель – это любой объект компьютерной графики. По способам создания изображений модели графики можно разделить на следующие категории:

- Двухмерная графика,
- Трехмерная графика,
- Изображения, созданные компьютером (CGI).

Двухмерная графика

Двухмерная компьютерная графика классифицируется по типу представления графической информации и вытекающим из него алгоритмам обработки изображений. Обычно двумерную компьютерную графику делят на векторную и растровую, хотя существует и фрактальный тип представления изображения.

Трехмерная графика

Трехмерная графика работает с объектами в трехмерном пространстве. Трехмерная компьютерная графика широко используется в кино и компьютерных играх.

Трехмерная графика может быть:

- Полигональной,
- Воксельной.

Полигональная графика представляет собой набор вершин, ребер и граней, которые определяют форму многогранного объекта в трехмерной компьютерной графике и объемном моделировании. Грани обычно представляют собой треугольники, четырехугольники или другие простые выпуклые многоугольники, поскольку это упрощает визуализацию.

Воксель — это виртуальный элемент, соответствующий набору из шести прямоугольных многогранников. Все в виртуальном мире (то есть пиксели, многогранники и воксели) должно проецироваться на пиксели физического экрана. Воксельная графика похожа на растровую графику. Объект состоит из набора объемных фигур, чаще всего кубов.

Изображения, созданные компьютером (CGI)

CGI — это трехмерные изображения, полученные компьютером на основе расчета и используемые в изобразительном искусстве, полиграфии, кинематографических спецэффектах, на телевидении и в симуляторах. Движущиеся изображения создаются компьютерной анимацией, которая является более узкой областью графики CGI.

Любое изображение на мониторе в силу своей плоскостности становится растровым, так как монитор представляет собой матрицу, состоящую из столбцов и строк. Трехмерная графика существует только в нашем воображении, так как то, что человек видит на мониторе, является проекцией трехмерной фигуры, а зритель сам трактует это изображение как пространственное. Таким образом, визуализация графики может быть только растровой и векторной, а метод

рендеринга — только растровым (т. е. набором пикселей). Способ определения изображения зависит от количества этих пикселей.

Текстуры

Текстура — это растровое изображение, применяемое к поверхности полигональной модели для придания ей цвета или иллюзии рельефа. В широком смысле текстуры можно рассматривать как узор на поверхности скульптурного изображения. Использование текстур позволяет воспроизводить небольшие поверхностные объекты, создание которых с помощью многогранников было бы чрезмерно трудоемким, например, шрамы на коже, складки на одежде, мелкие камни и другие предметы на поверхности стен и почвы.

Основным аспектом тестирования текстур является проверка наличия текстур на видимых объектах, правильности их отображения и однородности. Если в игре все текстуры в высоком разрешении, то не должно быть текстур низкого качества.

Коллизии

Коллизии — это технология, отвечающая за прохождение сталкивающихся объектов друг через друга.

Компьютерные игры, особенно консольные, должны распределять многие свои задачи между ограниченными аппаратными ресурсами и очень ограниченным игровым временем. Несмотря на эти ограничения и использование относительно примитивных и неточных алгоритмов обнаружения коллизий, разработчики игр смогли создать визуально правдоподобные и относительно реалистичные способы отображения взаимодействия игровых объектов друг с другом.

В большинстве компьютерных игр основными объектами, с которыми необходимо избегать столкновений и проникновений, являются ландшафт и окружающая среда уровня. Это статичные, не интерактивные и неразрушаемые структуры, такие как горы, деревья, здания и заборы. В этом случае персонаж представляется в виде одной точки и используется метод бинарного разбиения пространства (т. е. метод рекурсивного разбиения евклидова пространства на выпуклые множества и гиперплоскости). В результате объекты представляются в виде структуры данных, используемой для эффективного выполнения операций с трехмерной компьютерной графикой, включая сортировку визуальных объектов по удаленности от наблюдателя и обнаружение коллизий. Это обеспечивает жизнеспособный, простой и эффективный способ проверить, находится ли точка, представляющая персонажа, в окружающей среде (элементах игровой поверхности (террейне)) или нет. Коллизии между персонажами и другими динамическими объектами рассматриваются и обрабатываются отдельно. Дополнительную информацию о коллизиях можно найти в Разделе 3.1.2.

Анимации

Перед анимацией модели в ней создается скелет. Этот процесс также называется риггингом. Модели живых существ и все объекты, которые предполагается анимировать, могут иметь виртуальные кости. Кости модели зависят друг от друга, так что, например, при смещении руки кости ладони также будут двигаться. Вся последующая анимация зависит от того, насколько хорошо выполнены риггинг и привязка модели персонажа к скелету (скиннинг).

Анимация — это техника создания иллюзии движущихся изображений, т. е. движения и/или изменения формы объектов. С помощью трансформации последовательность неподвижных изображений (кадров) сменяется друг другом с высокой частотой (от 12 кадров в секунду для рисованной анимации до 30 кадров в секунду для компьютерной анимации).

Эффекты

Визуальные эффекты можно разделить на два основных типа:

- Эффекты игрового процесса (эффекты взаимодействия),
- Природные эффекты (эффекты окружения).

Среди прочих жанров эффекты игрового процесса более важны в симуляторах единоборств и ролевых играх. Например, когда игровой персонаж толкает игрового персонажа другого игрока, для

каждого из них должна воспроизводиться правильная анимация взаимодействия. Есть и другие игровые жанры, такие как шутеры (в особенности реалистичные), где природные эффекты так же важны, как и сам игровой процесс. Примерами природных эффектов являются водопады, туман и дождь.

Освещение сцены

Освещение сцены необходимо игроку для обзора сцены. Свет влияет на эмоции. Связь между изображением и эмоциональной реакцией дает еще один мощный инструмент, помогающий работать с персонажем, повествованием, звуком и игровой механикой. При этом взаимодействие света с поверхностью позволяет ему влиять на яркость, цвет, контрастность, тени и другие эффекты.

Благодаря особенностям строения человеческий глаз распознает черно-белые объекты в трех измерениях в пределах всего 110°, а полноцветные в еще меньшем диапазоне. Поскольку центральное зрение (в пределах упомянутого выше диапазона) — это первое, что использует человек, оно должно получить критические элементы, которые игрок обязательно должен видеть, как задумано дизайнером. Периферийное зрение обеспечивает контекст и усиливает центральное зрение. В видеоигре, если элементы, попадающие в периферийное зрение, не обеспечивают необходимого контекста или противоречат элементам, попадающим в центральное зрение, то связь между дизайнером и игроком разрывается.

Ниже приведены примеры правильного использования освещения сцены:

- Свет, попадающий в центральное поле зрения, может направлять игрока.
- Освещение может менять кадры. Например, использование фонарика делает его основным источником света для игрока. Измененная перспектива фиксирует взгляд игрока на освещенную область и отсекает все остальное из-за сильного контраста.
- Освещение может также создавать атмосферу, например чувство страха. Игрок держится в постоянном напряжении, когда, например, где-то в темноте прячется страшный враг.
- Направление света может либо облегчить поиск элементов на игровом уровне, либо усложнить его.
- Недостаток или переизбыток освещения может подтолкнуть игрока к использованию специальных предметов или действий.

Освещение сцены — одна из самых важных областей тестирования, потому что освещение создает эстетику игры и влияет на игровой процесс. Таким образом, многие методы освещения сцены, используемые в изобразительном искусстве, кино и архитектуре, используются в компьютерных играх, чтобы дополнить эстетику виртуального пространства и улучшить впечатления игрока. Однако игры сильно отличаются от кино или театра; их окружение динамично и непредсказуемо. Помимо статического освещения сцены используются динамические источники света. Они добавляют интерактивности и правильных эмоций.

Историческая достоверность

Любой элемент, например графический элемент, анимация или эффект, должен соответствовать общему стилю игры. Текстовое или звуковое описание объекта, модели или локации также должно соответствовать описанию исторического прототипа.

Богатое и сложное графическое содержание — одно из основных отличий компьютерных игр от любого другого типа программного обеспечения. Это существенно меняет подход к тестированию. Для тестирования графики в играх тестировщик должен иметь дополнительные знания по физике и оптике, иметь представление о технике цветопередачи, знание истории. Тестирование графики является одним из наиболее ответственных мероприятий по тестированию, так как именно здесь сосредоточена основная масса дефектов, напрямую влияющих на восприятие игры пользователем и его игровой опыт.

3.1.2 Типы дефектов графического содержания

Распространенной категорией дефектов при тестировании графики являются визуальные дефекты. Дефекты графики включают разрывы изображения на экране, отсутствие текстур и неожиданное кадрирование определенных областей изображения.

При создании графики и анимации для мобильных игр используются те же движки, что и для ПК. Разница в том, что одни и те же движки адаптированы под конкретные платформы, используемое в них аппаратное обеспечение и их технические возможности [Gregory18]. Поэтому дефекты, возникающие в таких играх, аналогичны.

Отсутствие текстур

Среди дефектов, наиболее часто встречающихся при тестировании текстур:

- Отсутствие текстур графических объектов,
- Потеря текстур во время игры,
- Временные текстуры и заглушки.

Как показывает опыт, первые две группы дефектов чаще всего связаны с недостаточной производительностью графических процессоров или устаревшими драйверами видеокарт. Для сравнения, временные текстуры могут появиться из-за дефекта разработки.

Неверный уровень детализации (LoD)

В современной трехмерной игре на сцене одновременно может отображаться множество объектов, находящихся на разном расстоянии от камеры (с точки зрения игрока). Чтобы снизить нагрузку на систему, разработчики используют технологию уровня детализации (LoD).

Когда разработчики создают новый игровой объект (например, дерево, здание или автомобиль), они добавляют в игру несколько вариантов моделей объектов. К ним относятся модели с уменьшенным количеством многогранников и упрощенной геометрией (низкополигональные модели) и более детализированные модели с большим количеством многогранников (высокополигональные модели). В зависимости от расстояния до камеры для отображения одного и того же объекта в игре используются модели с разным количеством многогранников. В непосредственной близости от камеры, когда требуется добиться максимального качества изображения, используются высокополигональные модели. По мере удаления камеры они сменяются менее детализированными моделями с меньшим количеством многогранников. На достаточно большом расстоянии модель отображается только в виде силуэта или вообще не визуализируется. Это позволяет уменьшить количество обрабатываемых многогранников и повысить производительность игры.

Например, для леса на горизонте можно использовать текстуры низкого разрешения с отображением только его цвета, без рельефа и отражений. Если игрок подойдет ближе, деревья предстанут во всех подробностях.

Технология LoD также используется для анимации, скелетов и искусственного интеллекта ботов, когда автоматизированная программа играет от имени игрока-человека. Например, в шутерах с большим количеством врагов в кадре у ботов будет разный уровень детализации. Боты на заднем плане будут изображены в упрощенном виде с низкой детализацией текстур, в то время как ближайшие противники будут тщательно прорисованы со сложной анимацией и кажутся достаточно умными, чтобы сражаться с игроком.

Разработчики также меняют уровень детализации в зависимости от текущей частоты кадров, скорости движения объекта на экране и общего количества одновременно видимых объектов.

Дефекты, возникающие при использовании технологии уровня детализации, в основном связаны с низкой производительностью системы. Разработчики стараются, чтобы увеличение детализации объекта или подмена модели происходили плавно и незаметно для игрока, хотя это не всегда так. В играх, работающих на слабом аппаратном обеспечении, низкополигональные модели могут не заменяться высокополигональными моделями достаточно быстро. В результате игрок,

приближающийся к объекту, сначала увидит «уродливую» модель, которая затем сменится на более качественную.

Сетка коллизии

Коллизии — это то, как объект в игре увеличивается в размерах и реагирует на столкновения с другими объектами. Сетка коллизии или коллайдер создается для того, чтобы готовая объектная модель могла взаимодействовать с окружающей средой и другими объектами. Это невидимая упрощенная форма объекта, которая привязана к нему для расчета коллизий с другими объектами. Коллайдер можно назвать физической моделью объекта. Он остается невидимым для игрока, но любые столкновения с коллайдерами обрабатываются движком игры [Buttfield19].

Форма коллайдера должна примерно соответствовать сетке модели, но часто достаточно грубого приближения. Для большинства статических объектов в окружающей среде, таких как здания, скалы и разбитые автомобили, разработчики используют низкополигональные коллайдеры. Это незаметно в игровом процессе, но повышает эффективность обработки всех видимых объектов.

При этом размер коллайдера должен соответствовать видимой модели, особенно если игрок может напрямую взаимодействовать с этим объектом.

Если коллайдер объекта намного меньше его визуальной модели, то персонаж сможет «провалиться в текстуру» или пройти сквозь нее. Строго говоря, фраза «провалиться в текстуру» (например, когда персонаж может пройти сквозь другой объект) некорректна, но прекрасно описывает то, что видит игрок, когда его персонаж частично или полностью погружен в другой объект.

Такие дефекты наиболее критичны в многопользовательских играх, так как могут дать игроку незаконное игровое преимущество. Например, въехавший в камень танк будет практически невидим для других игроков. В то же время такой камень не защитит от вражеского огня. Вот почему тестировщики должны уделять особое внимание таким объектам на картах в режиме игрок против игрока (PvP).

Тем не менее, некоторые объекты в игре вполне могут быть декоративными и вообще не иметь коллизий. Например, танк легко проедет через куст, а игрока не нужно заставлять объезжать или перепрыгивать через каждый мелкий камешек или консервную банку.

Здесь применяются игровые условности. Таким же образом можно отправить персонажа с длинным оружием в узкий низкий коридор, чтобы отбиваться от врагов, атакующих с разных сторон. В отличие от реальной жизни, персонаж сможет свободно поворачиваться в любом нужном направлении, и даже не заметит, что при повороте копые проходит сквозь стены. Кроме того, длинные волосы персонажа могут проходить сквозь массивные наплечники брони. Однако наличие низкого забора, через который нельзя перелезть, будет раздражать игрока и уменьшать его удовольствие от игрового процесса.

Если размер коллайдера объекта больше видимой модели, может возникнуть ситуация, когда персонаж упирается в невидимую стену или стоит в воздухе на крохотной опоре. Такие дефекты часто используются для незапланированного скоростного прохождения игр. Для этого игроки специально ищут места в локациях, где разработчики поленились или забыли изменить размер коллайдеров окружающих объектов. Иногда это позволяет игроку обойти значительную часть локации и сократить время прохождения.

Область поражения

Если предполагается, что персонаж или игровой объект может получить урон, например, от вражеской пули/снаряда, то в ходе боя необходимо вычислить положение каждой точки на поверхности этого объекта и определить, было поражение или нет.

Для объектов сложной формы это очень ресурсозатратно и не всегда оправдано.

Для упрощения таких расчетов используется понятие области поражения. Например, в двумерных платформах или симуляторах единоборств такая область может быть в виде одного или нескольких прямоугольников. Их положение относительно друг друга легко вычислить.

Для трехмерного объекта простейшей формой области поражения является сфера. В этом случае для расчетов достаточно знать ее центр и радиус. Таким образом, в любой момент времени легко определить, находится ли пуля или другой объект внутри этой сферы. Тем не менее, область поражения с параллельным каналом оказалась более подходящей формой, поскольку она содержит меньше пустого пространства по сравнению со сферой, и для расчета достаточно знать центральную точку объекта и три измерения: длину, ширину и высоту.

Для большей достоверности объект можно описать несколькими областями поражения, отдельно для каждой части модели, например, рук, ног, головы, хвоста и даже оружия.

Однако части модели, которые могут быть повреждены, и те, которые могут нанести ущерб, не всегда совпадают. Поэтому в некоторых играх область поражения и уязвимые области выделяются отдельно для одного и того же объекта, так что определяются области, наносящие урон (например, оружие, кулаки и т.д.), и области, по которым наносится урон (например, голова, конечности). Чтобы определить, должен ли персонаж получать урон, проверяется, перекрываются ли область поражения оружия и область уязвимости тела персонажа.

Основной задачей тестировщика при работе с областями поражения будет проверка соответствия видимой области воздействия реальному результату атаки.

Даже в больших играх бывают моменты, когда игрок четко видит, что его оружие коснулось врага, но не нанесло урона. Наоборот, если область поражения слишком велика, меч врага может навредить персонажу, даже если визуально он проходит мимо.

Прочие дефекты

При работе над крупным проектом каждый этап производства модели проходит множество согласований и подтверждений. Его тестируют художники, проектировщики трехмерных моделей и художники по текстурам. Даже в этом случае тестировщики могут обнаружить визуальные дефекты, появившиеся в процессе производства.

Несоответствие историческому прототипу. Игроки могут иметь нереалистичные доспехи с гигантскими наплечниками и неправдоподобный внешний вид летательного аппарата в фэнтезийных или научно-фантастических играх. Впрочем, если игра претендует на реалистичность, то наверняка найдутся дотошные игроки, которые укажут, что винты у тех или иных моделей самолетов располагались совершенно иначе, а пушек на тех или иных линкорах было гораздо больше.

Именно поэтому тестировщику стоит по возможности ознакомиться хотя бы с парой фотографий тестируемого объекта, чтобы представить, как он выглядел в реальности.

Предметы, висящие в воздухе. Одновременно над локацией может работать большое количество специалистов. Некоторые добавляют или удаляют объекты; другие изменяют геометрию элементов игровой поверхности (карты) или исправляют тайлы (текстуры, накладываемые на местность). Из-за этого появляются дефекты, когда ранее существовавший объект утопает в местности или наоборот зависает в воздухе.

Видимый стык между текстурами. Такие дефекты чаще всего встречаются на элементах игровой поверхности больших карт и локаций при смешивании нескольких текстур. Например, на переходах между поверхностью травы и песка или камня может быть виден «шов». Подробнее см. в главе 3.1.

Разрушение объектов. Как правило, модель имеет несколько состояний, такие как исходное, поврежденное и разрушенное, чтобы добавить реалистичности различным ситуациям. Если в игре предполагается реалистичное разрушение объектов, то для них создается специальная модель разрушений, возможно, даже со своей моделью коллизии. В определенный момент оригинальная модель заменяется моделью разрушенного объекта.

Здесь важно следить за тем, чтобы эффект разрушения соответствовал самой модели. Разрушенный грузовик должен отличаться от исходного, а части разбитого кирпичного здания не должны быть деревянными.

Дефекты освещения. К дефектам освещения относятся:

- Дефекты общего освещения,
- Дефекты точечных источников света,
- Дефекты прожекторных источников света,
- Дефекты плоскостных источников,
- Дефекты направленных источников света,
- Дефекты излучающих источников света,
- Дефекты источников рассеянного света.

Современные движки видеоигр включают в себя очень продвинутые возможности для создания различных типов источников света в полном соответствии с законами оптики. Однако все еще есть игры, в которых освещение не идеальное, например, из-за неправильных настроек освещения или отсутствия опыта у разработчиков. Рассмотрение визуальной композиции (положение источника света, его углы, цвета, поле зрения и движение) оказывает большое влияние на то, как игроки воспринимают игровое окружение, и создает правильную атмосферу для эмоционального вовлечения игрока. Дизайнеры должны позаботиться об этом, создавая визуальную целостность. [Lee16], [Tavakkoli18], [Romero19]

Дефекты анимации. Эти дефекты являются результатом привязки скелета к модели, называемой скиннингом. Чем больше костей у модели, тем более реалистичную анимацию можно создать. Можно анимировать разные объекты, например, волосы главного героя.

Кости модели зависят друг от друга. Следовательно, при смещении руки двигаются и кости ладони. Вся последующая анимация зависит от того, насколько реалистично все было сделано и настроено при риггинге и скиннинге.

Если дизайнер допустил ошибки, это может привести к удлинению рук/ног персонажей в различных анимациях, а также к отрыву частей моделей от остальных элементов. Часто эти дефекты появляются при столкновении с другими моделями или в различных анимациях модели.

Сбой визуальных эффектов (VFX) . В играх часто используются различные эффекты, связанные с действиями персонажей, событиями и природными явлениями, например, взрывы, искры, дым, исчезновение и появление.

При работе с анимацией визуальные эффекты обычно прикрепляются к костям, вспомогательным костям (т. е. служебным объектам, которые используются для создания и анимации моделей) и объектам сцены. Создание визуальных эффектов — это повторяющийся процесс, в котором взаимодействуют команды аниматоров и художники по визуальным эффектам. Последних могут попросить внести изменения или дополнения в анимацию, например:

- добавить больше ключевых кадров для более плавного следа от меча,
- повернуть вспомогательную кость, чтобы использовать ее вращение как направление для брызг крови,
- изменить переключение камеры, чтобы улучшить вид сетки частиц.

Прежде всего важно, чтобы такие эффекты были синхронизированы с событиями, которые их производят. Например, огонь и дым из дульного среза ружья должны появляться в момент выстрела и вылета пули из ствола. В противном случае реалистичность происходящего будет нарушена, что является дефектом.

Еще одна проблема, вызывающая дефекты визуальных эффектов, — несоблюдение технических условий, связанных с поддержанием частоты кадров и оптимизацией аппаратных ресурсов, используемых для корректного отображения эффекта. Каким бы масштабным ни был эффект, вряд ли игрок обрадуется, увидев на экране дергающиеся клубы пыли или застрявшие во времени взрывы. В результате спецэффекты в играх обременены жесткими техническими ограничениями, целью которых является поддержание определенного стабильного количества отображаемых кадров.

При тестировании визуальных эффектов тестировщик должен убедиться, что аппаратные возможности игрового устройства используются оптимально и что эффекты синхронизированы с событиями, которые их запускают.

На этапе производства модели желательно найти как можно больше дефектов. Например, увеличение количества многогранников модели или дефекты текстурирования значительно увеличивают нагрузку на оборудование, используемое при расчетах модели. Это неизбежно приводит к длительным задержкам в игре из-за перегрузки процессора видеоадаптера.

Чтобы проверить это, разработчик должен получить заранее определенные и фиксированные требования для таких аспектов, как уровень детализации (LoD) и модели коллизий.

Тестировщики, как правило, находят проявления этих дефектов, в том числе вызванные ими отказы. Если у объекта есть, например, модель коллизии со слишком большим количеством многогранников, дефект не будет обнаружен до тех пор, пока частота кадров при взаимодействии с этим объектом будет высокой.

3.2 Подходы к тестированию графики в игровых продуктах

3.2.1 Художественное тестирование

В процессе создания графический объект проходит множество различных этапов и проверок. Одни люди создают дизайн, а другие встраивают его в приложение. В итоге результат может быть не совсем таким, каким его задумал автор проекта. Тестировщик может проверить соответствие проекта стандартам и заметить дефекты, но есть дефекты, которые легче заметит художник. Например, недостаточная прозрачность фона или когда приложение ведет себя не так, как задумано при масштабировании.

Однако, художник не может всего предусмотреть и заметит дефект только в выполняющемся приложении, когда уже может случиться отказ. Поэтому хорошей практикой является отдавать приложение на проверку проектировщику после реализации, чтобы помочь выявить критичные дефекты. В этом процессе тестировщик подготавливает необходимую среду, создает тестовые данные, выполняет тесты и описывает дефекты после тестов, а затем может отправить их на проверку художнику.

На проектах с большим объемом графического содержания часто есть специалисты, которые занимаются тестированием графики. Это в первую очередь художники, досконально разбирающиеся в процессе создания моделей. Они подтверждают каждый из следующих этапов создания модели:

- Создание геометрии,
- Создание текстур,
- Создание модели коллизий.

Цель этой работы — улучшить качество моделей перед экспортом в движок видеоигры. Например, проверяют такие аспекты, как полигональность (т. е. количество многогранников) в геометрии, правильность их расположения, правильность развертки модели, наличие вытянутых многогранников, швов в модели.

Художественное тестирование — это большая сложная задача, которая может выполняться на различных этапах производства, начиная с создания простейшей геометрии и текстур и заканчивая экспортом модели в движок и размещением ее на карте. Для проведения художественного тестирования тестировщику часто не нужны специальные инструменты и редакторы графического содержания. Это можно сделать прямо во время игры. Однако наличие инструментов и редакторов графического содержания позволяет находить дефекты гораздо быстрее и намного эффективнее.

Художественное тестирование проводят:

Роль	Ответственности в художественном тестировании
Художники	При рецензировании объектов. При рецензировании и оценке объектов

Роль	Ответственности в художественном тестировании
Проектировщики трехмерных моделей	При рецензировании объектов.
Супервайзеры	При рецензировании и оценке объектов
Тестировщики	После финального экспорта моделей в движок
Игроки	При участии в игровом тестировании

3.2.2 Техническое тестирование

Техническое тестирование включает комплекс задач, связанных с техническими параметрами графики, таких как:

- Соблюдение ограничений на количество многогранников модели, о которых говорилось ранее,
- Форматы текстур,
- Дистанции переключения уровня детализации (LoD) , для которых достаточно сравнить имеющиеся ресурсы со спецификацией.

Само техническое тестирование выполняется тестировщиками, а также техническими художниками. Он включает процедуры, описанные ниже.

Поиск неиспользуемых и временных файлов в ресурсах графического содержания

При работе с художниками в репозиторий могут быть случайно добавлены ресурсы, которые в дальнейшем не будут использоваться. Например, разработчик может загрузить в репозиторий все ресурсы, включая неиспользуемые. В этом случае увеличивается объем места на жестком диске, занимаемый игровым клиентом, что снижает производительность и в итоге увеличивает общий объем доставляемых игроку обновлений (например, обновление отдельных файлов видеоигры). Для предотвращения этого выполняется поиск неиспользуемых ресурсов.

Поиск наличия всех требуемых записей в ресурсах клиента

В ресурсах игрового клиента графическое содержание не существует независимо от других ресурсов. Ссылки на текстуры пишутся в моделях, ссылки на модели пишутся на картах. Даже самые мелкие дефекты в таких записях могут привести к неожиданным последствиям.

Тестирование форматов текстур

К каждому типу текстуры технические художники предъявляют свои требования. Это связано с необходимостью соблюдения баланса между производительностью графического содержания и визуальной составляющей. Формат текстуры подразумевает тип сжатия и ее размер.

Тестирование клиентских протоколов при тестировании графического содержания

Иногда работа клиентских подсистем не очень заметна. Постэффекты — это эффекты, которые накладываются поверх изображения, например линза при взгляде на солнце. Для каждого постэффекта создается отдельная текстура в соответствии с наложением на экране. Глядя, например, на солнце через линзу, человек увидит линзу и солнечные лучи вне зависимости от того, назначена текстура для постэффекта или нет. В этом случае игрок просто не увидит дефекта, несмотря на то что он есть. Чтобы найти такие дефекты, необходимо изучить протоколы клиента, потому что в них обязательно будет указано, что рендер пытался использовать текстуру, которой нет в ресурсах.

Проверка соблюдения лимитов на количество текстур, моделей

Каждый тип графического содержания имеет свои ограничения, которые обычно являются общими для определенной группы. Например, для неразрушаемых многоэтажных зданий разрешено использовать до 15 тыс. многогранников, а для одноэтажных только 10 тыс. В связи с тем, что правки графического содержания продолжаются и количество загрузок в репозиторий велико, после каждого нового экспорта модели необходимо проводить проверку на соответствие лимитам и требованиям, чтобы предотвратить ситуации, связанные с нехваткой ресурсов аппаратного обеспечения, необходимых для расчета моделей видеопроцессора.

Несмотря на простоту тестов, необходимо понимать важность технического тестирования. Графика может иметь огромное влияние на общую производительность системы. Поэтому техническое тестирование включает в себя тестирование производительности, чтобы производительность и размер игры оставались на желаемом уровне. При увеличении количества текстур и моделей увеличивается и размер сборки. Такое несоответствие форматов текстур ухудшает показатель частоты кадров, что приводит к увеличению потребляемой клиентом памяти. Техническое тестирование гарантирует, что художественное графическое содержание будет доступно игроку в максимально полном объеме.

3.2.3 Тестирование во время игры (игровое тестирование)

Тестирование во время игры (игровое тестирование) — это подход к тестированию, направленный на те факторы, которые влияют на игровой процесс. Это может быть как оценка соответствия модели коллизии визуальной модели, так и проверка соответствия настроек игровых объектов требованиям игрового режима.

Игровое тестирование объектов включает:

- Тестирование соответствия очков здоровья (ОЗ). Это значение в ролевых и компьютерных играх, определяющее максимальное количество повреждений, которые нужно нанести объекту для его уничтожения. Несмотря на то, что ОЗ присваиваются объектам автоматически с учетом материала, размера и типа объекта, дефекты все же могут возникать. Поэтому ОЗ проверяются вручную на соответствие характеристикам объекта.
- Тестирование соответствия модели коллизий визуальной модели. Как упоминалось ранее, модель коллизий значительно упрощена по сравнению с визуальной моделью. Это может привести к дефектам, когда игрок визуально скрыт за моделью, но на самом деле может быть получить урон от противника или быть убит.

Тестирование графики — сложный процесс, и его виды пересекаются друг с другом. Описанный выше подход к тестированию обеспечивает тщательное тестирование графического содержимого игры на различных этапах для выявления графических дефектов.

3.3 Выполнение тестирования графики

3.3.1 Выполнение тестов графики на разных этапах производства объекта

Создание серого ящика

Это упрощенная модель графического объекта, которая на самом деле является макетом, используемым для тестирования во время игры. Обнаруженные на данном этапе дефекты в первую очередь связаны с удобством работы художников с данной моделью.

Создание видимой геометрии

Следующий этап моделирования включает отображение и создание трехмерной проекции физического объекта в окружающей среде. При этом учитывается его геометрия и расположение в пространстве. Создается видимая геометрия и развертка модели и проверяется соответствие координат на поверхности трехмерного объекта (X, Y, Z) координатам на текстуре (U, V). Поскольку проверки часто проводятся с помощью редакторов трехмерной графики, дефекты на этом этапе также могут быть обнаружены художниками и проектировщиками трехмерных моделей. Как правило, к таким дефектам относятся:

- Дублированная геометрия. Дублирование многогранников может значительно снизить производительность.
- Излишняя детализация. Например, модель, находящаяся вне поля зрения игрока, имеет фаски, а значит, на обработку таких «украшательств», которые игрок просто не заметит, будут выделяться дополнительные аппаратные ресурсы.
- Недостаточная детализация. Этот тип дефекта является полной противоположностью дефекта излишней детализации. Если модель хорошо видна игроку, она должна быть достаточно детализирована, чтобы не выглядеть угловатой.
- Части геометрии, которые не визуализируются. Например, отсутствие необходимых деталей в модели может сделать ее нереалистичной.

Текстурирование модели

На этом этапе текстуры принимаются художником или арт-директором. В этом случае корректное наложение текстур приложением, отсутствие швов и натяжек (т.е. неверная топология многогранников) могут не обнаружиться. Тест касается только общей цветовой палитры и уровня загрязнения (например, наличие грязи, ненужных цветных вставок, случайных пикселей разных цветов на модели). Типичными дефектами, обнаруживаемыми на этом этапе, являются случаи, когда автоматический маппинг и последующее текстурирование выполняются таким образом, что на части модели, имеющие общий край, наносятся текстуры разных цветов, в результате чего появляются заметные швы.

Рецензирование LoD и модели коллизии объекта

На этом этапе к оценке качества моделей привлекаются как художники, так и проектировщики уровней. Проектировщики уровней оценивают модели коллизий с точки зрения их влияния на игровой процесс. Например, нужно ли использовать детальную модель в конкретном месте или ее можно упростить. Типичными дефектами в этом случае являются модели, которые либо слишком детализированы (например, содержат больше допустимого количества многогранников в модели коллизии), либо недостаточно детализированы (например, содержат недостаточно детализированную модель коллизии).

Экспорт модели в движок игры

На этом этапе учитывается правильность настроек расстояний, на которых следует переключать LoD, и количество очков здоровья объекта.

Тестирование проводится тестировщиками непосредственно в движке видеоигры как в редакторах графического содержания, так и непосредственно в игровом приложении. При этом большая часть художественных тестов сводится к визуальной оценке объекта, а ряд тестов дублирует тесты художников и проектировщиков трехмерных моделей. Тестирование направлено на:

- Наглядность переключения LoD объектов с учетом местоположения и ситуации, в которой будет использоваться объект,
- Зазоры и швы в геометрии объекта, заметные игроку,
- Видимость надписей на различных графических объектах,
- Эффект разрушения модели (например, с учетом цвета и размера),
- Визуальные дефекты анимации объектов.

В художественном тестировании могут принимать непосредственное участие игроки, которые визуально оценивают графическое содержание и могут найти другие дефекты графики.

Тестирование графики часто проводится вместе с тестированием игровых карт. В этом случае есть еще несколько этапов.

Размещение объектов на карте

Это один из самых важных этапов тестирования. Размещением объектов занимаются проектировщики уровней. При этом типичным возникающим дефектом являются объекты, которые висят или утоплены под карту (террейн). Это часто происходит, когда с картой параллельно

работает много специалистов, которые могут менять геометрию местности, выставлять объекты и редактировать саму карту.

Создание и размещение эффектов

На этом этапе художники по эффектам работают над созданием новых спецэффектов. Наблюдение за ним в различных ситуациях позволяет им обнаружить художественные и технические дефекты рендеринга. Примером такого тестирования является визуальная проверка эффекта в движке игры. Художники по эффектам просматривают каждый эффект в игровом клиенте, чтобы определить неправильные параметры эффекта (например, недостаточное время затухания эффекта).

Тестирование графики часто проводится с помощью игровых тестов. Это относится к любому тестированию, при котором карта тестируется в игровых ситуациях. Цели этого тестирования обширны и применимы к любой стадии производства (например, оценка игрового процесса на карте). Ближе к концу производства такое тестирование может включать оценку игрового процесса и художественную оценку карты с точки зрения игроков. Помимо сбора статистики, необходимой проектировщикам игры для настройки баланса на карте, такие тесты выявляют различные художественные и технические проблемы, начиная от висящих объектов и заканчивая дефектами отрисовки графики (например, зависаниями, дефектами и сбоями). При чисто художественном тестировании игровое тестирование выявляет проблемы, заметные с точки зрения игрока.

Тестирование карты

Это завершающий этап тестирования, когда карта полностью готова и над ней больше не ведутся работы (за редким исключением). Тестировщики проводят полную проверку карты на основе чек-листа проверки карты, от поиска висящих объектов до тестирования анимации. Тестировщики делают упор на такие тесты, которые позволяют им обнаруживать гораздо больше дефектов по сравнению с художниками. Во время художественных тестов художники визуально проверяют карты, тогда как тестировщики используют специализированный набор инструментов для выявления наиболее критичных проблем в графических объектах.

3.3.2 Тестирование графики на историческую точность

Отдельное направление тестирования графического содержания — проверка на историческую точность и историческую достоверность.

Историческая достоверность влияет на качество игры, в которой конкретный эпизод истории описывается с помощью характерных образов. В них не упоминаются конкретные события, личности и т.д. Историческая достоверность включает и историческую точность.

Историческая точность влияет на качество игры, в которой описывается конкретный период истории. Все ключевые события описываются в их динамике, включая всех соответствующих личностей и их действия с образами, характерными для периода. Они принципиально исключают образы, чуждые эпохе.

Как правило, в играх невозможно сохранить историческую достоверность, так как упор делается на затягивающий игровой процесс. Трудно вовлечь игроков в игру, исход которой predetermined с самого начала.

Однако историческая точность — это область игр, требующая особого внимания, особенно когда сюжет игры привязан к определенной исторической эпохе.

Типичные аспекты, которые следует учитывать при проверке графического содержания на историческую точность:

- Сходство персонажей с историческими прототипами,
- Точность воспроизведения архитектурных объектов,
- Точность воспроизведения оружия, снаряжения, транспортных средств, одежды той или иной эпохи и их характеристик,
- Точность воссоздания повседневной жизни,
- Точность исторических событий и дат.

При проведении проверки исторической точности тестировщик должен обладать широкими знаниями и кругозором. Очень часто разработчики привлекают исторических консультантов к тестированию своих продуктов, чтобы создать максимально реалистичную игру.

3.4 Инструментальная поддержка тестирования графики

Тестировщики графики используют широкий спектр инструментов, начиная от внутриигровых редакторов графического содержания и заканчивая автоматизированными инструментами и тестовыми сценариями. Как правило, игры создаются с использованием движков видеоигр, содержащих несколько встроенных инструментов, в том числе для тестирования графики: редактор объектов, редактор мира и редактор освещения.

Все инструменты предоставляют возможность настроить существующее графическое содержание для взаимодействия игрока с ним. Редакторы моделей позволяют привязывать к моделям такие эффекты, как различные триггеры, звуки и другие события, происходящие при взаимодействии с движком. Редактор мира позволяет создавать карту, размещать на ней объекты и настраивать игровую механику карты. Именно в этих редакторах тестировщики проводят большую часть тестов. При этом они используют те же инструменты, что и разработчики. Эти редакторы также предоставляют обширный набор инструментов, который позволяет настраивать графическое содержание и тестировать его.

Специализированные инструменты обычно разрабатываются производителями видеокарт. Эти инструменты позволяют захватывать кадры в игре и детально анализировать любое приложение, используя различные наборы программных интерфейсов приложений (API), используемые для создания двух- и трехмерной графики. Они также позволяют оценить следующее:

- Как готовится игровой кадр (например, геометрия, текстуры, вызовы отрисовки),
- Когда возникают проблемы с производительностью,
- Геометрию бега,
- Отладку для проектировщиков трехмерных моделей, художников графики и аниматоров.

Более подробная информация о том, как используется каждый из этих инструментов, описана в конкретной спецификации инструмента.

В средства тестирования графики также входят автоматизированные скрипты, обеспечивающие автоматическое прохождение персонажем заданного маршрута в игре. Они особенно часто используются при тестировании производительности и совместимости.

4. Тестирование звука - 190 минут (К3)

Ключевые слова, специфичные для игр

Бинауральный эффект, громкость, задержка звука (Фоли), звуковое окружение, звуковые зоны, звуковые прерывания, звуковые разрывы, звуковые эффекты, искажение, окклюзия, реверберация

Цели обучения главы 4

4.1 Особенности звукового содержания игрового продукта

GaMe-4.1.1 (K1) Вспомнить особенности звукового наполнения игрового продукта

4.2 Типы дефектов звукового содержания

GaMe-4.2.1 (K1) Вспомнить типы дефектов звукового содержания

GaMe-4.2.2 (K2) Классифицировать дефекты звукового содержания

4.3 Подходы к тестированию звукового содержания в игровых продуктах

GaMe-4.3.1 (K2) Обобщить основные подходы к тестированию звукового содержания

GaMe-4.3.2 (K2) Обобщить основные подходы к тестированию сочетания музыки и звуков

GaMe-4.3.3 (K2) Обобщить основные подходы к тестированию музыкальной композиции

4.4 Тестирование звука

GaMe-4.4.1 (K2) Объяснить уровни тестирования аудио-музыкального содержания

GaMe-4.4.2 (K1) Вспомнить особенности интеграции звуков в клиенте

GaMe-4.4.3 (K1) Вспомнить зоны ответственности звукового тестирования

GaMe-4.4.4 (K3) Применить подходы к тестированию звука

4.5 Инструментальная поддержка тестирования звука

GaMe-4.5.1 (K2) Обобщить использование инструментов тестирования звука

4.1 Особенности звукового содержания игрового продукта

Звук - важная часть любой современной игры. Звуки и музыка создают нужное настроение, предупреждают об опасности, передают эмоции персонажей, дополняют и создают целостную картину игрового мира.

Человек, столкнувшийся с фильмом и игрой, в которой вообще нет звука, скорее всего, испытывает легкий диссонанс. Это происходит, потому что наш мозг пытается активировать наш слух, но не может, а вместо этого сигнализирует об ошибке восприятия того, что происходит перед нашими глазами.

Некачественные, нереалистичные, неуместные или плохо функционирующие звуки могут раздражать игрока и нарушать погружение в игровой процесс.

Процесс создания звукового сопровождения в игре достаточно длительный и трудоемкий, так как необходимо учитывать сложность организации содержания игры:

- Звуки окружающей среды,
- Голоса персонажей,
- Фоновое звуковое сопровождение,
- Звуковые эффекты,
- Различные звуковые объекты,
- Голосовые файлы.

Это усложняет задачу тестирования.

4.1.1 Типы звуков

Звуковой дизайн может варьироваться от игры к игре и создаваться для разных целей. Рассмотрим, какие виды звуков принято различать, какие функции они выполняют и для каких целей создаются.

Фоновая музыка

Главная музыкальная тема игры, ее визитная карточка, может появиться даже в самой простой игре. Мелодии из старых 8-битных игр уже давно стали классикой.

Основные функции музыки:

- Подчеркивать “эпичность” момента,
- Усиливать динамику события,
- Усиливать атмосферу,
- Настраивать игрока и объяснять эмоциональное состояние персонажа,
- Подсознательно формировать у игрока определенную эмоциональную реакцию за счет повторяющейся мелодии.

Звуковые эффекты

Если игровой мир наполнен объектами, с которыми герой может взаимодействовать, то каждый из них должен иметь реалистичные звуки.

Открытие дверей, удары, стрельба и перезарядка оружия, использование аптечки, взрыв канистр с горючим, звук бьющегося стекла — все это необходимо озвучивать и воспроизводить в нужный момент.

Даже в небольшие игры разработчики добавляют звуковые эффекты взаимодействия с элементами интерфейса (кнопки, переключатели, пункты меню). Это дает игроку обратную связь и понимание того, что действия игрока воспринимаются и анализируются.

Звуки персонажей (Фоли)

Звуки, издаваемые персонажами в игре, включают шорох одежды, дыхание, возгласы, шаги, стоны и т.д. Термин назван в честь Джека Фоли, одного из пионеров звуковых эффектов. В некоторых играх, например в шутерах, такие звуки могут означать, что персонаж ранен. Это будет даже понятнее для игрока, чем всплывающие цифры урона.

Речь

Хоть и речь воспроизводится персонажами, но все же лучше выделить ее отдельно. В современных играх каждый неигровой персонаж (NPC) имеет уникальный голос, записанный профессиональным актером. Разработчики вкладывают много сил и средств в озвучку, потому что качественно записанные реплики позволяют передать эмоции персонажей и создать эмоциональную связь между ними и игроком.

Звуковое окружение

Звуковое окружение относится к звукам, характерным для определенной местности. Музыкальная тема и звуковое окружение подчеркивают атмосферу и создают необходимое настроение игроку. Звуковое окружение не предполагает никаких объектов, от которых исходит звук. Это звук окружающей среды, который просто сопровождает определенное место, ситуацию или этап игрового цикла.

Звуковое окружение обычно воспроизводится фоном и обычно не зависит от действий персонажа. Примерами таких звуков являются:

- Шелест листьев и пение птиц в лесу,
- Шум волн на море,
- Невнятный разговор посетителей салона,
- Полицейские сирены, шум проезжающих машин и т.д.

Все это помогает игроку ориентироваться в пространстве и лучше воспринимать окружающий мир игры.

4.1.2 Звуковые эффекты и технологии

Звуковые дорожки часто обрабатываются в звуковых редакторах для создания аутентичного эффекта с учетом возможностей современной звуковой аппаратуры и техники.

Для того, чтобы погрузить пользователя в виртуальный мир игры, использование только реалистичных звуков может быть недостаточно. Для повышения реалистичности происходящего разработчики часто увеличивают громкость некоторых звуков. Этот прием часто можно увидеть, например, в фильмах со сценами драк, где удары и даже движение бойцов звучат гораздо громче, чем в реальной жизни.

Для достижения такого эффекта используются различные приемы и технологии.

Окклюзия

Одним из наиболее распространенных приемов является эффект окклюзии, когда звук проходит через преграду (например, стену здания). Для большей реалистичности звукорежиссеры в таких ситуациях не просто убавляют громкость, а добавляют спецэффект.

Основной звук и отраженные звуки по-разному реализуются, когда между источником звука и его слушателем находится глухая стена, когда предметы находятся в разных комнатах, но в прямой видимости (дверной проем), а также когда предметы находятся в одной комнате, но не в зоне прямой видимости (столб между ними).

Реверберация

Эффект реверберации используется для передачи объема и глубины пространства (от глагола реверберировать, отражать).

В этом случае звук от источника в закрытом помещении отражается от стен и вызывает многочисленные эхо, которые постепенно затухают. Этот эффект хорошо работает, если персонаж находится в пещере, маленькой комнате или другом помещении с хорошо отражающими звук поверхностями.

С помощью продвинутых звуковых движков можно легко создать эхо и добавить громкости звукам выстрелов, взрывам, голосам, шагам и т.д.

Бинауральный эффект

Бинауральный эффект (от латинского *binī* - два, пара и *auris* - ухо) основан на том, что человек слышит двумя ушами одновременно, и когда он поворачивает голову в сторону, звук достигает одного уха раньше другого. Благодаря этому человек может определить, в каком направлении находится источник звука и даже расстояние до него.

Для записи звука с этим эффектом используются специальные микрофоны, а для его восприятия рекомендуется использовать наушники. Это необходимо для того, чтобы звуки могли поступать в каждое ухо с определенного направления. Звуковое окружение позволяет игроку глубже погрузиться в игру. Благодаря бинауральному слуху игрок знает, с какой стороны исходит звук (например, где враг появится через мгновение).

4.1.3 Звуковые зоны

В некоторых проектах для распределения звукового окружения по разным локациям используются так называемые звуковые зоны.

Например, когда персонаж в ролевой игре входит в деревню и появляются определенные характерные звуки, или в стратегической игре (RTS) пользователь приближает камеру к своей базе и слышит звуки собираемых ресурсов.

Звуковые зоны должны иметь переходы. Вблизи перехода звук одной зоны должен немного затухать, а звук другой зоны появляться.

Примером эффекта звукового зонирования в реальной жизни служит открывание и закрывание окон в комнате. При открытом окне отчетливо слышны звуки улицы, а при закрытом их либо вообще не слышно, либо они крайне приглушены.

4.2 Типы дефектов звукового содержания

Существуют три категории дефектов звукового содержания:

- Отсутствие звукового эффекта,
- Воспроизведение неправильного звука,
- Некорректное воспроизведение правильного звука.

Отсутствие звукового эффекта

Звук не воспроизводится, хотя должен воспроизводиться для определенного действия персонажа или в определенный момент.

Воспроизведение неправильного звука

Неправильно настроен звук предмета или окружения, например, пушечный выстрел звучит как выстрел из пистолета, моторная лодка звучит как самолет или персонажи (как внутриигровые, так и неигровые) произносят чужие реплики.

Звуковой эффект нарушает историческую достоверность, например, игровые персонажи используют современную лексику в то время, как действие игры происходит в средние века, цитаты из фильмов и музыки, танк Т-34 звучит как танк Т-72. Это особенно важно в тех случаях, когда реализм и аутентичность заявлены как ключевые ценности тестируемой игры.

Некорректное воспроизведение правильного звука

Пропадание звука

При возникновении такого дефекта могут исчезнуть некоторые воспроизводимые звуки, ровно так же, как при разговоре по телефону с плохой связью. Из-за этого игрок может, например, неправильно понять смысл фразы в диалоге. Если это единственный звук, который воспроизводится в данный момент, то тестировщики легко найдут проблему. А вот если одновременно воспроизводятся несколько звуков, а один из них пропадает, то обнаружить такой дефект гораздо сложнее.

Звуковые прерывания

Звуковые прерывания музыки или звуковых эффектов могут быть связаны не только с повреждением самого звукового файла. Прерывание часто возникает из-за проблем с производительностью. В данном случае они возникают при дефекте частоты кадров.

Искажение

Из-за проблем с производительностью некоторые фразы могут звучать искаженно и неразборчиво.

Задержка воспроизведения

В играх бывает моменты, когда звуковое оформление отстает от отображаемой анимации, предмета или игровой ситуации.

Примеры дефектов звукового содержания и их возможные причины

1. Отсутствие звука объекта/среды.

Дефект: при движении по металлической поверхности нет звука шагов, персонаж движется бесшумно.

Причина: разработчик забыл настроить или подключить звук на этой поверхности.

2. Звук объекта или окружения слишком громкий/тихий.

Дефект: горит некий объект на локации, но шум от огня слишком громкий, или наоборот, очень тихий, что выглядит нереалистично.

Причина: неправильно установлена громкость эффекта.

3. Звук объекта или окружения установлен неправильно.

Дефект: во время вождения моторной лодки по воде игра воспроизводит звук гоночного автомобиля.

Причина: в коде прописан некорректный звуковой файл для объекта.

4. Неправильное позиционирование звука в зависимости от его источника.

Дефект: в игре есть кузнец, который непрерывно ударяет молотом по мечу, но звук удара воспроизводится в другом углу комнаты.

Причина: в звуковом редакторе звуковой эффект установлен в неправильное положение.

5. Неправильно настроена звуковая зона.

Дефект: звук выстрела слышен только в радиусе пары метров. Если игрок уходит дальше, то звука вообще не слышно, что не реалистично.

Причина: неправильно задан радиус звука.

6. Звуковое искажение.

Дефект: игроки слышат, как звук начинает трещать или шипеть.

Причина: используется поврежденный аудиофайл.

7. Непрерывное повторение звука.

Дефект: звук заикливается и воспроизводится непрерывно.

Причина: указатель, отвечающий за остановку звукового цикла, установлен неправильно.

8. Искажение звука в виде “заикания” воспроизведения:

Дефект: искажение звукового оформления движения происходит во время движения одного из игровых персонажей.

Причина: проблема может возникнуть из-за неполадок со звуковой сборкой или из-за общей нестабильности игрового клиента.

9. Задержка воспроизведения звука.

Дефект: во время заставки персонаж начинает говорить, но его реплика воспроизводится чуть позже, чем когда его губы начинают двигаться.

Причина: анимация движения губ персонажа не синхронизирована с началом воспроизведения соответствующего звукового файла.

4.3 Подходы к тестированию звукового содержания в игровых продуктах

Звуковое оформление игры может быть представлено в виде эффектов, голосов персонажей или музыки. Тестирование аудио содержания игр может выполняться формальным или неформальным способом. Формальные подходы к тестированию звука включают проверку наличия, правильности, громкости и целостности всех аудиофайлов. Неформальное звуковое тестирование — это своего рода оценка реалистичности или сохранения атмосферы игры. Например, тяжелое музыкальное

сопровождение локации в играх жанра хоррор для акцентирования обстановки и периодического поддержания игрока в состоянии напряжения.

Тестировщик получает тестовые сценарии из ранее рассмотренной спецификации конкретного звукового объекта. Из этой спецификации тестировщик должен однозначно понимать, в каких локациях, игровых сценах и ситуациях должен воспроизводиться звук. В некоторых случаях для этого может потребоваться консультация аудио проектировщика, чтобы получить подробности о созданном звуковом содержании. В других случаях тестировщик может использовать редактор игровых ресурсов для проверки синхронизации звуковых файлов с действиями игрока или окружением в игре.

Тестирование звукового оформления игры тестировщиком может состоять из нескольких этапов.

4.3.1 Тестирование звукового содержания

Тестировщик прослушивает звуки и тестирует звуковые параметры созданного содержания с помощью звукового редактора или редактора ресурсов игры. Все звуковые файлы должны быть проверены.

Это один из самых масштабных этапов. Следующие звуковые характеристики проверяются на соответствие требованиям:

- Соответствие звука объекта обстановке. Например, если персонаж носит железную броню, то скрежет металла должен доминировать над шелестом ткани при его движении. Едва слышный взрыв динамитной шашки или упавшей скалы, явный топот по тротуару при движении по песку может нарушить погружение в виртуальный мир.
- Голоса актеров. Голоса актеров в полностью озвученных фразах должны соответствовать персонажам, которые их произносят, а также настройкам игры. Обычно голос должен соответствовать полу и возрасту персонажа. Это делается для поддержания общей атмосферы, например, у персонажа-южанина южный акцент (так, например, в США), а не современный ирландский.
- Не раздражающий. Образцы нескольких звуков вместе иногда могут стать раздражающими для человеческого уха.
- Уровень громкости. Громкость каждого звукового файла должна быть одинаковой при использовании нескольких файлов в одной игре.
- Звуковые эффекты. Эффекты, применяемые как к звукам, так и к озвучке, также должны использоваться правильно. Если локация или сцена наполнена различными деталями, то детали подчеркиваются озвучкой. В большом офисе человек слышит гудение системных блоков, жужжание принтеров, шум кондиционеров, нажатие клавиш клавиатуры, разговоры работающих.

Тщательное дублирование всех объектов способствует созданию полной картины. Например, оказавшись в деревне, игрок не обязательно услышит мычание коров, но скорее всего будет ожидать характерные звуки лесопилки, если она появится перед ним. Небольшой ручеек в лесу вполне может не иметь звукового оформления, но, если герой стоит рядом с водопадом, отсутствие шума воды можно объяснить только дефектом (или отключенным звуком в настройках).

4.3.2 Тестирование сочетания музыки и звуков игры

Размер проекта и профессиональные навыки команды разработчиков варьируются от проекта к проекту, что делает невозможным четкое разделение выполняемых функций между ресурсами команды. Могут быть ситуации, когда актер по озвучке не требуется, работа Фоли художника (специалиста по созданию шума) доверена аудио дизайнеру или даже один специалист сам создает или покупает готовые образцы звука, впоследствии встраивает их в клиента и тестирует полученный звук.

Роль аудио проектировщика заключается в создании звуков и записи аудиофайлов. Эти звуковые файлы должны быть протестированы для различных случаев. Звуковая система игрового автомата разительно отличается от системы мобильного телефона, и звук нужно тестировать. Звук в

консольных играх должен корректно воспроизводиться как на небольших встроенных динамиках, так и на системах домашнего кинотеатра.

Даже качественный и, казалось бы, уместный звук может вызвать у игрока замешательство и раздражение, если финальное сведение сделано неправильно. Например, звук действия длится намного дольше, чем само действие, или его громкость неуместна.

Тестировщик тестирует готовую звуковую картину и правильность реализации новых звуковых элементов в клиенте, используя редактор ресурсов, встроенный в движок игры, опираясь на собственный опыт или предоставленную документацию игры.

Так как во время игрового процесса игрок слышит большое количество звуков, исходящих от разных предметов, и музыку, играющую фоном, важно, чтобы все звуковые составляющие гармонизировались друг с другом. Тестировщик должен оценить звуковую составляющую проекта и выявить возможные отклонения, нереальные моменты, касающиеся громкости, а также качества и естественности звука, исходя из здравого смысла. Также важно оценить точность расположения источника звука по отношению к самим объектам. Например, в игре, когда персонаж открывает дверь, игрок слышит звук открываемой двери не от двери, а по ошибке из-за спины персонажа. Тестировщику также важно уметь правильно оценивать своевременность звуков по отношению к тому, что происходит на экране.

4.3.3 Тестирование музыкальной композиции

Композиция или сведение — это этап создания финальной записи из индивидуально записанных треков. Это следующий после аудиозаписи шаг, состоящий в выборе и редактировании (иногда восстановлении) исходно записанных треков, объединении их в единый проект и наложении эффектов. Редактирование часто является самостоятельным этапом работы.

Микширование игр, в которых используется электронная музыка - следующий шаг после создания аудио содержания. Стадия звукозаписи электронного проекта чаще всего отсутствует. Грань между созданием и микшированием электронной музыки стирается; проект приходит к микшированию уже частично смикшированным, т.е. многие виртуальные синтезаторы уже имеют шаблонную обработку разных звуков.

В результате многоканальный проект выводится на монофоническую, стереофоническую или многоканальную фонограмму, которая обычно получает окончательную форму в процессе, называемом мастерингом.

После завершения мастеринга начинается тестирование обработанной музыки. Тестировщик должен проверить музыкальный компонент на наличие дефектов, звуковых разрывов и помех и сообщить обо всех обнаруженных сбоях.

4.4 Тестирование звука

4.4.1 Уровни тестирования аудио-музыкального содержания в жизненном цикле разработки игрового программного обеспечения

Часто звуковое тестирование начинается на поздних стадиях разработки. Сначала разработчик создает звуковое содержание для игры (модели объектов/персонажей, карты, предметы и т.д.), а звуковая составляющая прорабатывается на более позднем этапе. На этапе разработки объект может не иметь звука, но сам объект уже имеет готовую модель с настроенной анимацией, интегрированную в клиент и готовую к тестированию.

Образцы звука должны быть предварительно протестированы специалистами, создавшими звук (то есть аудио проектировщиками, Фоли художником, специалистами по звуковым эффектам). Они могут принять участие в тестировании отдельных звуков, а затем отправить их в разработку для дальнейшей интеграции в клиент.

4.4.2 Интеграция звуков в игру

После интеграции звукового оформления в игру тестировщик оценивает следующие характеристики звуков:

- Способность включать и выключать звуки и/или музыку.
- Возможность изменить громкость.
- Соблюдение единого формата имен аудиофайлов и правильное их распределение по каталогам в проекте.
- Расположение источников звука, зоны распространения звука на игровой карте.
- Производительность системы с интегрированными звуками.
- Совместимость звука на разных аудиосистемах: наушники, колонки, мониторные колонки.

4.4.3 Зоны ответственности вовлеченных лиц

В процессе создания музыки и звукового дизайна в проекте участвуют несколько разнотипных специалистов, выполняющих определенные функции:

Роль	Обязанности
Композитор	Участие в создании звуков и музыки для игры.
Звукоинженер	Занимается эксплуатацией звукозаписывающей, звукоусиливающей и радиовещательной аппаратуры.
Звукооператор	Специалист, обладающий максимальной полнотой звуковой экспертизы и определяющей итоговую звуковую картину каждого объекта, игровой сцены и всей видеоигры в целом.
Аудио проектировщик	Создает содержание из оригинальных аудиофайлов с записанным звуком и настраивают его в звуковом редакторе.
Фоли художник	Делает существующие в реальности звуки более яркими и насыщенными, а также отвечает за создание звуковых эффектов, которых в реальности не существует. Комбинируя звуки обычных предметов, Фоли художник записывает сэмплы и создает из них эффекты, например, гудения лазерного меча, рычания зомби или хлопка открывающегося портала.
Голос актера	Озвучивает своим голосом, чтобы вдохнуть жизнь в персонажа, у которого будут звуковые реплики в игре.
Разработчик	Интегрирует созданное и настроенное аудио содержание в код игрового клиента.
Проектировщик уровней	Распределяет источники звука по уровням, а затем привязывает звуковые файлы к источникам звука.
Повествовательный проектировщик	Специалист, отвечающий за сюжет и повествование в разработке компьютерных игр, работает над звуком и музыкой, проигрываемой в моменты, связанные с развитием сюжетной линии (ролики, заставки, диалоги, значимые моменты в игре и т.д.).
Проектировщик игр	Участвует в разработке документации (например, документации игры). В этом документе простым языком описаны правила и особенности игры. Таким образом, еще до разработки движка, проектировщик игр разрабатывает целостное видение игры. В процессе разработки проектировщик также частично участвует в роли “консультанта”, утверждая соответствие предложений/изменений основной идее проекта. Также он лично участвует в тестировании игры.

Роль	Обязанности
Тестировщик	Тестирует готовую звуковую картину и правильность внесения новых звуковых изменений в игровом клиенте.

4.4.4 Процедуры и подходы к тестированию звуковых объектов

При тестировании звуковых объектов тестировщик должен реализовать процедуры и действия, необходимые для получения полной информации о том, насколько правильно настроен звук в той или иной версии игры. Список действий напрямую зависит от того, какие объекты необходимо протестировать. Например, для финальной проверки звукового оформления добавляемого оружия в клиенте тестировщик выполняет ряд действий: добавление оружия в случае, если оно изначально отсутствует в клиенте, загрузка нового звукового содержания на тестовую карту и в целом список действий, необходимых для максимальной проверки всех звуков, прилагаемых к оружию (выстрел, перезарядка, повторная перезарядка, переключение режима стрельбы и т.д.).

Важно понимать, что перечень процедур и подходов для каждого объекта разный. Например, если тестируется объект окружения, а не объект, с которым взаимодействует персонаж, то тестировщик тестирует включение самого звукового оформления - например, журчание ручья или реки при приближении к ним персонажа. Тестировщик должен понимать, где найти информацию обо всех доступных звуках для конкретного объекта. Он также должен воспользоваться встроенным редактором ресурсов и звуковым редактором, чтобы проверить целостность звукового содержания и синхронизировать их с действиями персонажа и их окружением.

4.5 Инструментальная поддержка тестирования звука

Звуковой редактор движка игры

Тестировщик не всегда имеет доступ к звуковому редактору движка игры. Поэтому при тестировании используется стандартный набор программ для проверки звукового содержания. Такой редактор используется тестировщиками, работающими в команде разработчиков (если таковая имеется) или специалистами, работающими непосредственно в аудио редакторе (например, композиторами, аудио проектировщиками и т.д.). Аудио проектировщики должны участвовать в тестировании. Они тестируют активы, включая уровни балансировки, настройку эквалайзеров и микширование/выравнивание всех активов как вместе, так и по отдельности. Как только активы будут добавлены в рабочую версию игры, аудио проектировщики смогут легко их получить и при необходимости их редактировать.

Чтобы облегчить обслуживание звукового содержимого, все звуковые образцы должны быть на разных звуковых дорожках, расположены в папках, иметь имена, временные метки и версии. Также эффективно конвертировать звуковые дорожки в сжатый цифровой формат, чтобы не нагружать аппаратные ресурсы звукового адаптера.

Редактор карт/локаций

Иногда тестировщик получает доступ к разработке программного обеспечения, чаще всего к редактору карты/локации. Это позволяет тестировщику расширить свои возможности при тестировании аудио содержания и всего содержания, которое есть в игровом клиенте. Разработчики игр добавляют в редакторы карт функциональность, поддерживающую симуляцию игрового процесса непосредственно на карте уровня. Таким образом, тестировщик может проходить необходимые тесты, не переключаясь с редактора карт на сам игровой клиент, что существенно экономит время.

Большинство звуковых эффектов и источников звука, доступных на карте, расположены и настроены с помощью редактора уровней. В редакторе карт тестировщик может видеть расположение источников звука на клиентских локациях, привязанных к разным объектам на локациях или персонажам. Соответственно, когда специалист тестирует источник звука через редактор, он может тестировать воспроизводимый звук и дальность распространения звука, отображаемые либо в виде сферы, либо в виде зон, где тестировщик или конечный пользователь

начинает слышать воспроизводимый звук/музыку, исходящий от привязанного источника звука, при пересечении их границ.

Если распространение звука реализовано в виде “зон”, тестировщик проверяет воспроизведение звука от источника при пересечении границы “зоны”. Также важно протестировать увеличение громкости при приближении к источнику звука, чтобы сделать связанный звук более реалистичным. Тест на дальность распространения звука в виде сферы отличается только формой зоны распространения. Редактор также позволяет протестировать привязку, расположение, и, соответственно, конфигурацию источника звука каждого клиента.

Обладая этой информацией, тестировщик может оценить реалистичность звукового наполнения, дальность распространения звука, правильность звука (на локации, объекте, у персонажа) и указать разработчику на дефекты в случае их обнаружения. Например, создать описание дефекта о неправильно установленной дистанции у того или иного источника звука. Или громкость того или иного звука (относящегося к конкретному источнику звука) слишком большая на фоне “соседних источников”. При описании дефектов тестировщик, работая через редактор, может помочь разработчикам, облегчая поиск проблемных мест, указав на ту или иную проблему.

5. Тестирование игровых уровней - 65 минут (K2)

Ключевые термины тестирования

Тестирование во время игры

Ключевые термины, специфичные для игр

Недоступное место, повествование, прототип уровня, редактор уровней, структурная геометрия, сценарное событие, уровень видеоигры, цветовое кодирование

Цели обучения для Главы 5

5.1 Принципы и концепции проектирования уровней видеоигр

GaMe-5.1.1 (K1) Запомнить компоненты уровней видеоигр

GaMe-5.1.2 (K2) Классифицировать дефекты, характерные для уровней видеоигр

5.2 Этапы и выполнение тестирования уровней видеоигр

GaMe-5.2.1 (K2) Обобщить тесты, проводимые на разных этапах создания уровней видеоигр

GaMe-5.2.2 (K2) Сравнить зоны ответственности специалистов, принимающих участие в тестировании уровней видеоигр

5.3 Инструментальная поддержка тестирования уровней видеоигр

GaMe-5.3.1 (K2) Обобщить использование инструментов для тестирования уровней видеоигр

5.1 Принципы и концепции проектирования уровней видеоигр

5.1.1 Термин «уровень» и его специфика в зависимости от жанра игрового проекта

Уровень видеоигры (или игровой уровень) — это отдельная область виртуального мира игры, в которой игрок должен выполнить определенную задачу: найти сокровище, победить всех противников или просто добраться до выхода. Общая задача, необходимая для прохождения уровня, неразрывно связана с локацией и часто состоит из множества мелких задач и головоломок, решение которых необходимо для прохождения уровня, или побочных головоломок, которые не являются необходимыми для прохождения уровня. Во многих движках видеоигр уровни разделены экраном загрузки.

Проектирование игровых уровней — этап разработки игры, связанный с созданием карт, локаций, заданий, миссий и прочего окружения для уровней игры. Проектирование игровых уровней включает в себя внешний вид игровых объектов и игровой механики, препятствия на пути игрока, сюжетную линию игры и другие элементы, которые в совокупности создают предполагаемый игровой опыт.

Для создания игровых уровней обычно используется специальное программное обеспечение — редактор уровней.

Уровни игры состоят из множества компонентов, объединенных в единое целое. Среди основных частей можно выделить следующие.

Структурная геометрия

Структурная геометрия, или базовая геометрия игровых уровней, является одним из наиболее важных и формообразующих элементов любой игровой поверхности. Именно структурная геометрия задает рельеф местности и террейн, по которому могут передвигаться игровые персонажи.

Структурная геометрия также ограничивает пространство игрового уровня, доступное для перемещения персонажей. Например, на границах игрового уровня могут быть непроходимые горы или другие особенности местности.

Если игровой уровень имеет четко обозначенные точки входа и выхода, то структурная геометрия может подсказать, куда игроку следует идти, чтобы добраться до конца уровня.

В зависимости от локации, где происходит игра, в качестве объектов, образующих игровой уровень, могут выступать различные элементы. Например, в городе: это здания, улицы, мосты и подземные переходы; за его пределами – элементы природного ландшафта: поля, холмы, скалы, овраги и пещеры.

Игровая среда

Объекты игровой среды используются проектировщиком уровней для создания более правдоподобного образа необходимого игрового пространства.

Например, деревенская улица состоит из домов, хозяйственных построек и дорог, поэтому проектировщик наполняет ее объектами разного размера. Различают крупные объекты, такие как деревья, заборы, телеги, объекты среднего размера – колодцы, бочки, скамейки, а также мелкие объекты, такие как растения в саду, камни и т.п.

Освещение

Использование различных типов источников света позволяет создать нужную атмосферу на игровом уровне. Например, направляет игрока в нужном направлении, привлекает его внимание или скрывает от противников.

Звуковое сопровождение

Фоновая звуковая дорожка или звуковые эффекты, воспроизводимые для различных действий или событий. Подробнее читайте в Главе 4. Тестирование звука.

Игровой функционал

Игровой функционал включает систему настроек для организации игрового процесса на определенном уровне видеоигры. Различают следующие виды настроек:

- Настройки, связанные с прохождением игрового уровня, включают:
 - точки появления персонажей и их противников;
 - условия победы и поражения;
 - точки автоматического сохранения игры;
 - систему сценарных событий, запускающих определенные игровые события или сценарии. Например, каждые две минуты над полем боя пролетает самолет, или при входе персонажа в здание проигрывается сюжетное видео, или при пересечении границы охраняемой территории поднимается тревога.
- Настройки объектов, с которыми игрок может взаимодействовать, например, двери, автомобили, разрушаемые объекты, ловушки и т.д.
- Настройки физической оболочки игрового уровня, предотвращающие выход персонажа за пределы уровня или застревание в его геометрии.

Как правило, настройки игрового функционала скрыты от игроков и видны только в редакторе уровней.

5.1.2 Понимание типов дефектов в проектировании уровней видеоигры

Существует ряд дефектов, наиболее часто встречающихся при создании уровней видеоигр. Отследить такие дефекты бывает сложно, поэтому они могут какое-то время оставаться незамеченными.

Некоторые специальные инструменты создания карт имеют встроенные функции обнаружения дефектов на картах. Проектировщики уровней часто используют эти инструменты на последних этапах создания карты или уровня игры. Но в большинстве случаев лучший способ найти дефекты на карте — протестировать ее опытными игроками, способными обнаружить проблему и сообщить о ней.

Значительная доля дефектов, встречающихся на уровнях видеоигр, связана с внешним видом и расположением игровых объектов, их освещением, моделью коллизий и т.д. Подробное описание таких дефектов дано в Главе 3. Тестирование графики.

Однако следующие дефекты напрямую относятся к проектированию уровней видеоигры.

Геометрия

Дефекты геометрии — это ситуации, когда игровой персонаж полностью или частично застревает в некоторых элементах уровня/карты. Как правило, такие дефекты встречаются на краях карты, уступах, склонах. Игрок направляет своего персонажа к объекту или подходит близко к объекту и застревает в нем. Часто с помощью прыжков, наклонов и других движений персонажу удается выйти из такой ситуации, но в некоторых случаях это не помогает. В этих случаях игрок должен перезапустить уровень/карту, вернуться к последней точке сохранения или использовать специальную комбинацию кнопок/клавиш, чтобы перейти в ближайшее место без дефектов коллизий.

Например, игровой персонаж может застрять в геометрии карты, не имея возможности выбраться, или попасть в области, в которые запрещен доступ. Особенно важно выявлять такие дефекты в многопользовательских играх, где дефекты игрового уровня могут поставить одного игрока или одну команду в более выгодное положение.

Еще одна распространенная ситуация — смерть при застревании, а именно, персонаж визуально проваливается под поверхность уровня/карты и падает вниз. В конечном итоге персонаж либо умирает и игровой уровень приходится начинать заново, либо бесконечно падает в пустоту, и игру приходится начинать заново.

Недоступные места

В видеоиграх, особенно на картах в многопользовательских играх, могут быть преднамеренные локации, в которых игрок занимает лучшую позицию, чем другие игроки. Например, у пулеметчика на холме будет большой сектор обстрела. Хотя игрок, первым занявший такую позицию, получает некоторое преимущество перед противниками, это замысел разработчика, элемент проектирования уровней, а не дефект.

Но бывают ситуации, когда такое место создается случайно из-за ошибки проектировщика уровней игры. По замыслу разработчика, эта область должна быть недоступна для игроков. Однако определенный игровой персонаж с помощью своих уникальных особенностей (например, более высоких прыжков, чем остальные) может там оказаться. Таким образом, игрок может получить незаконное игровое преимущество над своими соперниками. Например, другие игроки не смогут его обнаружить, нанести урон и т.д.

Сложный игровой процесс

Проблемы, связанные с отсутствием указателей целей или не очевидностью необходимых действий для прохождения игрового уровня. Например, игрок победил всех противников на доступной территории, но не может понять, что еще нужно сделать, чтобы пройти дальше.

Баланс игрового уровня

Очень много дефектов возникает при настройке баланса игрового уровня. Например, слишком сложные на текущем этапе игры враждебные mobs, которые полностью исключают возможность прохождения этого игрового уровня. В других играх, где несколько команд пытаются добраться до нужной точки карты раньше соперников, дефектом баланса игрового уровня могут быть неравные стартовые условия для команд.

Неуместные ограничения и условности

Эти проблемы не влияют на игровой процесс и не дают игрового преимущества, но ухудшают восприятие и погружение в игру.

Например, игрок видит, что путь персонажу преграждает забор высотой по пояс, но перелезть через него невозможно. По задумке проектировщика уровней, это место является недоступной для игрока

областью, и так оно и должно быть. Но с точки зрения игрока такая ситуация выглядит недоработкой, портящей погружение в игру. Точно так же игрок сочтет дефектом ситуацию, когда персонаж заперт за решеткой и не может выбраться, хотя и видит, что расстояние между прутьями решетки достаточно велико.

Повествование

Дефекты игрового уровня, связанные с нарушением общего стиля повествования, сюжета игры. Влияние таких дефектов на сам игровой процесс минимально, однако они негативно сказываются на погружении пользователя и общем восприятии игры. Например, если персонаж окажется в городе, который по сюжету подвергся ядерной бомбардировке, появление в нем неповрежденных зданий вызовет диссонанс.

Изменение цветового кодирования объектов

Объекты, с которыми сталкивается пользователь во время игры, имеют разные свойства. Некоторые из них являются декорациями, а с другими игрок может взаимодействовать. Цветовое кодирование объектов с различными свойствами — важная часть проектирования уровней видеоигр. Поэтому желательно, чтобы единожды принятое цветовое кодирование не менялось на протяжении всей игры.

Например, игрок обнаружил на игровом уровне красную бочку, которая взорвалась от выстрела. Если игрок снова столкнется с такой бочкой, он будет ожидать от нее такого же поведения. Поэтому все бочки, взрывающиеся от выстрела, всегда должны быть одного цвета (обычно красного).

Ящики, которые игрок может разбить, двери, которые игрок может открыть, камни и уступы, за которые игрок может ухватиться и перелезть, и т.д. должны отличаться по форме и/или цвету от других объектов.

В противном случае игрок будет вынужден тратить больше времени на то, чтобы понять, что делать дальше.

5.2 Этапы и выполнение тестирования уровней видеоигр

5.2.1 Основные этапы проектирования и тестирования уровней видеоигр

Уровень видеоигры — это сущность, объединяющая в себе игровую механику, визуальные объекты, звуковое сопровождение, искусственный интеллект и другие составляющие. Подходы к тестированию для каждого из этих компонентов подробно обсуждаются в соответствующих главах этой программы обучения.

Тестирование уровней видеоигры включает тестирование всех этих компонентов вместе и похоже на системное тестирование, когда все компоненты рассматриваются и тестируются как единое целое.

Тестирование игровых уровней начинается с самых ранних этапов их создания. Объекты тестирования, характер тестов и порядок их проведения, а также зоны ответственности различных специалистов зависят от текущего этапа разработки игрового уровня.

Рассмотрим основные этапы проектирования и тестирования игровых уровней на примере трехмерной игры.

Прототипирование уровней видеоигры (Черновик/серый ящик проектировщика)

На этом этапе на основе ранее разработанного эскиза из специальных объектов создается трехмерная модель будущего игрового уровня. Используются только элементы, непосредственно влияющие на игровой процесс: серые кубы, сферы, цилиндры и плоскости, которые схематически формируют макет игрового уровня.

Такие монохромные, не детализированные геометрические фигуры позволяют быстро менять структуру игрового уровня, например, удалять или добавлять огромные куски локации, а при

необходимости быстро стирать все начисто и начинать заново. Иногда на этом этапе не все игровые механики готовы к использованию, поэтому проектировщику приходится реализовывать их постепенно, адаптируя игровой уровень к постоянно меняющимся условиям разработки.

Макет создается с учетом всех пропорций и масштабов. Для этого проектировщик уровней вместе с проектировщиком игры формирует набор метрик — размеров и параметров, которыми обладает игровой персонаж. Так, проектировщик игры предоставляет информацию о расстоянии и высоте прыжка персонажа, о том, в каких положениях может находиться персонаж (например, стоя в полный рост, на корточках, лежа) и как при этом изменяются размеры фигуры, максимальный угол наклона поверхности, по которому персонаж может двигаться без скольжения и т.п. Эти метрики напрямую влияют на размеры объектов и элементов среды игрового уровня.

После того, как проектировщик уровней определился с размером игрового уровня и установил правильный масштаб для объектов, настраивается прочий функционал для поддержки игрового процесса на этом уровне. Сюда входят точки появления персонажей, сценарные события и т.д.

Разработка игрового уровня на этом этапе, как правило, проходит в несколько итераций, в ходе которых тестирование помогает выявить и устранить проблемные элементы.

Тестирование прототипа уровня видеоигры

В связи с тем, что требования не финализированы на этапе прототипирования, наиболее эффективными считаются исследовательское тестирование и другие методы тестирования на основе опыта.

На созданном макете проводятся первые тесты во время игры для проверки игрового процесса на этой карте. Участниками тестирования, проводящими первые тесты во время игры, могут быть как тестировщики из команды разработчиков игрового ПО, так и тестировщики из сторонней организации. Получая отзывы от участников тестирования, разработчик может сделать вывод об удачных и неудачных решениях и при необходимости изменить, переместить или убрать объекты на макете.

Также путем проведения тестирования во время игры на прототипе проверяются игровые механики и то, как они сочетаются с расположением и размером объектов на макете.

Например, чтобы укрытия, расположенные на игровом уровне, были достаточно большими, чтобы персонаж мог спрятаться за ними от врагов, чтобы ширина расщелины в поверхности позволяла персонажу перепрыгнуть через нее, или чтобы персонаж мог взобраться на все необходимые уступы и не удариться головой о слишком низкие потолки.

Отдельно проверяется, что макет уровня создает нужную сложность игры и правильное ощущение от игрового процесса в целом. Например, если предполагается, что на заданном интервале игры должна быть перестрелка со многими противниками, то игровой уровень должен иметь достаточное количество укрытий. Если предполагается сражение с одним сильным живучим врагом (боссом), то пространство игрового уровня ограничивается и очищается от ненужных объектов.

Геометрическое прототипирование (Художественный черновик/белый ящик)

Временные трехмерные модели, созданные художником по окружению, заменяют объекты, из которых проектировщик уровней сделал прототип, и добавляются дополнительные объекты. Основная задача — проработать визуальную составляющую игрового уровня, не меняя сам игровой процесс.

Однако такая замена может вызвать дефекты на игровом уровне, затрудняя игровой процесс. Например, временная модель дерева перекрывает выход из игрового уровня. В этом случае игровой уровень снова переходит в руки проектировщика уровней. Теперь требуется еще раз протестировать, правильно ли работает механика, везде ли игрок может пройти и видит ли все, что ему нужно.

Тестирование геометрического прототипа

Тестирование на этом этапе проводится, чтобы убедиться, что после художественной прорисовки на игровом уровне не осталось дефектов, мешающих игровому процессу.

Примерами таких дефектов могут быть временная трехмерная модель камня, блокирующего проход игрового уровня, или крона дерева, закрывающая обзор игроку.

Для выявления таких проблем снова используются тесты во время игры, где проверяются игровые механики на предмет соответствия размеров и расположения моделей объектов требуемым метрикам и корректности игрового процесса.

Параллельно с этим может проводиться тестирование внешнего вида объектов, освещенности, игровой среды и т.д. (подробнее см. в Главе 3. Тестирование графики.)

Создание финальной версии

На этом этапе проектировщик уровня интегрирует программное обеспечение всей команды разработки в единое целое. Из всего созданного содержания он собирает единое игровое пространство. На этом этапе на уровень видеоигры добавляются декоративные детали, объекты становятся похожи на настоящие не только по внешнему виду, но и по назначению и расположению. Например, на столе появляются бумаги, канцтовары, компьютерные мониторы, которые игрок может разбить и сбросить со стола.

Тестирование финальной версии

На этом этапе проводится тестирование производительности и совместимости. Тестируется работоспособность игрового уровня на разных устройствах, измеряется количество кадров в секунду, проверяется качество отображения визуальных моделей объектов на разном расстоянии, освещения и теней.

В рамках тестирования во время игры тестировщики проверяют наличие моделей коллизий у всех игровых объектов, соответствие физических и визуальных моделей объектов и т.д.

Тестирование моделей коллизий может выполняться следующим образом: тестировщик подводит игрового персонажа к игровому объекту, например, к большому камню, и пытается «пройти сквозь камень» с разных сторон. При правильно настроенной модели коллизий камней тело персонажа визуально будет только касаться камня, но не пробивать его или полностью входить в него.

5.2.2 Области ответственности вовлеченных лиц

Над созданием уровня или карты видеоигры работают разные специалисты, и зачастую эти задачи выполняются параллельно, что приводит к появлению различных дефектов.

Рассмотрим задачи, которые входят в обязанности этих специалистов с точки зрения получения качественного продукта:

Роль	Ответственность
Проектировщик уровня видеоигры	Отвечает за геометрию и форму уровня видеоигры, расположение объектов и сценарных событий, размеры укрытий и т.д. Все это должно поддерживать основной игровой процесс и способствовать вовлечению пользователей в игру.
Художник	С художественной точки зрения все объекты на карте должны иметь правильные текстуры, освещение и другие визуальные эффекты. Это касается как внешнего вида (правильное размещение темных и светлых участков), так и реалистичности изображения. Например, хижина в лесу из бетонных блоков выглядела бы странно и неуместно.
Тестировщик	Тестировщики, как правило, работают с законченными версиями уровней и карт видеоигры, когда все объекты уже расставлены и настроены. Основная задача тестировщика — протестировать, может ли игрок играть на данной карте, не сталкиваясь с техническими и художественными дефектами. Тестируется взаимодействие со всеми объектами на карте, а именно: <ul style="list-style-type: none">• Объекты имеют физическую модель (модель коллизий),

Роль	Ответственность
	<ul style="list-style-type: none">• Отсутствие невидимых препятствий,• Невозможность выйти за пределы карты.

5.3 Инструментальная поддержка тестирования уровней видеоигры

Для облегчения тестирования игровых уровней или карт используются различные инструменты, включая трехмерные редакторы, редакторы уровней и движки видеоигр.

При этом тестировщики используют те же инструменты, что и разработчики, создавшие игровой уровень.

Движок видеоигры обычно имеет функционал, позволяющий настраивать карту и объекты на ней для различных тестов и поиска дефектов. Движок собственной разработки может быть изменен по запросу команды тестирования с целью повышения эффективности самих тестов.

Например, в результате изменения геометрии поверхности игрового уровня некоторые предметы могут быть подвешены в воздухе или утоплены в поверхность. Такие дефекты можно обнаружить при визуальном осмотре всех объектов или, если позволяет редактор карты, с помощью специального алгоритма, который может упростить проверку.

Кроме того, отключив в редакторе отображение текстур различных объектов, тестировщик может найти на карте непроходимые места или наоборот непредусмотренный проектировщиком карты путь прохождения. Это особенно важно для многопользовательских игр, поскольку в некоторых случаях это может дать некоторым игрокам нежелательное преимущество над другими.

6. Тестирование игровых контроллеров - 95 минут (K2)

Ключевые слова

Соблюдение требований, тестирование эргономичности, функциональное тестирование

Ключевые слова, специфичные для игр

Акселерометр, гироскоп, двуручный игровой контроллер, игровой контроллер, игровой руль, сенсорный экран, трекбол

Цели обучения для Главы 6

6.1 Принципы и концепции игровых контроллеров

GaMe-6.1.1 (K2) Классифицировать типовые и специализированные устройства ввода

GaMe-6.1.2 (K2) Привести примеры различных устройств ввода с точки зрения их применения

GaMe-6.1.3 (K1) Запомнить различные типы игровых контроллеров

GaMe-6.1.4 (K2) Классифицировать дефекты игрового продукта, связанные со спецификой игровых контроллеров, и возможные причины их возникновения

6.2 Подходы к тестированию контроллеров в игровых продуктах

GaMe-6.2.1 (K2) Привести примеры тестовых условий, которые необходимо учитывать при тестировании игровых контроллеров

GaMe-6.2.2 (K2) Классифицировать задачи для UX-специалистов, тестировщиков и проектировщиков игр во время тестирования игр

6.3 Инструментальная поддержка тестирования игровых контроллеров

GaMe-6.3.1(K2) Обобщить использование инструментов для тестирования поведения игровых контроллеров

6.1 Принципы и концепции игровых контроллеров

6.1.1 Типы игровых контроллеров

Игровой контроллер — это устройство ввода, которое используется в консольных и компьютерных играх. Контроллер обычно подключается к игровой приставке или персональному компьютеру. С помощью игрового контроллера игрок управляет движением и действиями игровых объектов. При этом тип элементов зависит от самой игры, но чаще всего это один из персонажей игры.

Типовые устройства ввода

Игровые устройства, такие как телефоны, ПК, консоли и игровые автоматы, гарантированно позволяют использовать одно из следующих устройств.

Устройство ввода	Описание
Двуручный игровой контроллер	Основное устройство ввода в игровых консолях. Также называется «геймпад» (gamepad).
Клавиатура и мышь	Поскольку эти устройства стали обычными устройствами ввода для персональных компьютеров, они также широко используются для компьютерных игр. Некоторые игровые приставки также позволяют подключать мышь и клавиатуру и управлять с помощью них в играх.
Игровой руль	Контроллер, который имеет ручку для вращения и одну или несколько кнопок действий.

Устройство ввода	Описание
Трекбол	Выглядит как шар, наполовину выступающий из основания; шар вращается, когда по нему водят ладонью.
Сенсорный экран	Используется в телефонах, КПК, портативных консолях и современных игровых автоматах. В некоторых играх есть элементы игрового процесса, специально предназначенные для сенсорного экрана.

Специализированные устройства — это устройства, ориентированные на определенные виды игр.

Специализированное устройство ввода	Описание
Джойстик	Изначально это было универсальное игровое устройство. Клавиатура/мышь или двуручный игровой контроллер оказались удобнее в динамичных играх, а джойстик стал специализированным устройством для игр в жанре авиасимуляторов. Однако по привычке двуручные игровые контроллеры часто называют «джойстиком».
Гоночный двуручный игровой контроллер	Используется для упрощения игры в гоночные симуляторы на консоли. По сути, это обычный двуручный игровой контроллер, в который встроена дополнительная рулевая ось (а иногда и аналоговые кнопки газа и тормоза). Такое устройство намного дешевле полноценного игрового руля.
Штурвал	Используется для гражданских авиасимуляторов (военные авиасимуляторы используют джойстик).
Педали	Используется для гоночных и авиасимуляторов, имеющих принципиально разные конструкции.
Компьютерный дроссель	Используется для авиасимуляторов. Также известен как рычаг управления двигателем.
Рычаг переключения передач	Используется для симуляторов вождения.
Легкий пистолет	Используется для стрельбы по объектам на экране.
Графический планшет	Используется для управления курсором вместо мыши.
Контроллеры ритма	Используется в музыкальных играх для имитации музыкальных инструментов, таких как гитары, барабаны или пульт ди-джея.
Танцевальная платформа	Танцпол (как на игровых автоматах). Это платформа с несколькими кнопками, на которые можно наступать ногами. Игровой процесс таких игр заключается в том, чтобы наступать на нужную последовательность кнопок, повторяя танец, демонстрируемый на экране.
Игровая клавиатура	Это специализированная клавиатура с кнопками, расположенными в соответствии со спецификой конкретной игры, и дополнительными кнопками для создания макросов.
Контроллер удочки	Используется для симуляторов рыбалки.
Микрофон	В качестве дополнительного устройства ввода используется микрофон или гарнитура, с помощью которых в видеоигре подаются команды, чтобы персонажи и игроки могли общаться.

Специализированное устройство ввода	Описание
Панель управления поездом	Тренажер пульта управления тяговым железнодорожным/трамвайным подвижным составом.

Технологии захвата движения

- С начала 2000-х годов системы отслеживания движений головы использовались для игр в авиасимуляторы и для людей с ограниченной подвижностью.
- Устройства дистанционного управления, отслеживающие свое положение в пространстве с помощью инфракрасных датчиков и акселерометров.
- Камеры, отслеживающие движения игрового контроллера в трехмерном пространстве и распознающие изображения.
- Устройства, которые позволяют использовать словесные команды, положения тела и отображаемые объекты или изображения.

6.1.2 Дефекты, связанные со спецификой игровых контроллеров

Причины дефектов, связанных с игровыми контроллерами, могут быть разными. Появление дефектов может быть вызвано самим программным обеспечением, браком компонентов контроллера и даже несоблюдением разработчиком инструкции по использованию контроллера от производителя:

- Устаревший драйвер игрового контроллера,
- Несовместимость модели игрового контроллера с игровым приложением,
- Дефект отдельного устройства или всей партии,
- Несоответствие игрового процесса инструкции.

Наиболее частый дефект — недостаточность или полное отсутствие всплывающей подсказки при переключении игровых контроллеров во время игры. Привязки клавиш могут различаться от игры к игре. Они также могут быть переназначены игроком по своему усмотрению.

Устаревшая версия драйверов или их отсутствие может привести к тому, что игровой контроллер не будет работать должным образом. Если программные дефекты можно устранить обновлением, то технические неисправности и недостатки игровых контроллеров можно исправить только выпуском их новой версии.

Однако неточность в считывании движения гоночного руля или любого другого игрового контроллера может быть вызвана аппаратным дефектом или дефектом программных расчетов. Для игр, где критична точность считывания управляющих сигналов игрового контроллера, документация видеоигры должна содержать требуемые значения в градусах наклона.

Кроме того, при выпуске видеоигры на популярной платформе владелец платформы может указать требования к внутриигровым изображениям своего игрового контроллера. Внутриигровые изображения — это изображения игрового контроллера, используемые в игре. Например, это может быть схематичное изображение двуручного игрового контроллера в меню настроек привязки клавиш или в подсказках по игровому процессу. Изображения могут быть не только внутри игры; игровые контроллеры также могут отображаться на упаковке программного обеспечения или на цифровой обложке. Это требование обычно распространяется на известных издателей и одновременно производителей консолей и контроллеров. Корпорации в своей документации по тестированию для разработчиков игр могут указывать требования к изображению игровых контроллеров в приложении, включая как контуры, так и товарные знаки.

Также для видеоигр, где используются акселерометр и гироскоп игрового контроллера, владелец платформы предъявляет требования к безопасности. Простейшим примером является необходимость делать махи с игровыми контроллерами для реализации игрового процесса. Датчики движения внутри него позволяют использовать его в качестве элемента управления в трехмерном пространстве. В таких случаях перед запуском игры необходимо указать, что пользователю

необходимо надеть удерживающие ремни, которые крепятся к игровым контроллерам. В противном случае, вследствие резких движений игровые контроллеры могут выскользнуть из рук пользователя и повредить окружающее оборудование или, что еще хуже, стать опасными для здоровья.

6.2 Подходы к тестированию контроллеров в игровых продуктах

Тестирование с использованием игровых контроллеров обычно начинается в тот момент, когда игровой функционал готов к использованию какого-либо стандартного игрового контроллера: для ПК это клавиатура/мышь, для консолей — двуручный игровой контроллер.

Этот тип тестирования может включать:

- подключение/отключение игрового контроллера к ПК/консоли,
- использование игрового контроллера при низком уровне заряда батареи контроллера,
- тестирование поддержки игр для некоторых производителей игровых контроллеров,
- тестирование поддержки определенных программных интерфейсов приложения (API), которые позволяют приложению получать данные от игрового контроллера,
- использование одного и/или нескольких игровых контроллеров,
- нетривиальное использование игрового контроллера (негативное тестирование),
- тестирование вибрации (наличие и степень ее интенсивности).

Функциональное тестирование

Если приложение требует использования игрового контроллера в качестве устройства ввода, то все функции программного обеспечения должны корректно взаимодействовать с подключенным устройством. Пользователь должен иметь возможность управлять как элементами пользовательского интерфейса, так и непосредственно игровым процессом, не меняя контроллеры. Назначенные элементы управления: кнопки, рукоятки, повороты игрового руля, голосовые команды или движения в пространстве должны соответствовать действиям игрового персонажа. Тестировщику также необходимо убедиться, что при отключении игрового контроллера от устройства приложение приостанавливается, если это возможно. Если реализация игрового процесса осуществляется в реальном времени и не может быть остановлена, то отключение игрового контроллера не должно приводить к отключению игрока от сервера и должно сопровождаться информационным сообщением.

Тестирование безопасности

Игровой контроллер может содержать технологическую уязвимость и открывать доступ к взлому консоли. Тестирование безопасности должно быть выполнено для снижения рисков безопасности: получение несанкционированного доступа к режиму разработчика, предотвращение блокировки устройства по сети с использованием режима полета и т.д. [ISTQB_AL_SEC]. Тестирование безопасности игровых контроллеров включает подходы к обнаружению аппаратных дефектов, влияющих на функциональность программного обеспечения.

Тестирование эргономичности

Кнопки на игровых контроллерах самых популярных брендов имеют устоявшееся назначение. Кнопки X, A и , X на двуручных игровых контроллерах очень часто используются в приложениях как кнопки «принять», «принять выбор» или «взаимодействовать» с интерактивным элементом, а кнопки B и O как «отмена», «отказ», или «возврат».

Комбинации кнопок настраиваются и контролируются программным обеспечением. При тестировании комбинаций кнопок тестировщику необходимо обращать внимание на то, удобны ли они анатомически для разных групп игроков (в зависимости от жанра, возраста, инвалидности и т.д.): есть ли у игрока возможность нажать несколько кнопок одновременно или последовательно.

Тестирование игровых контроллеров на соблюдение требований

Роли UI/UX специалиста и проектировщика игр обязательны при разработке игр. Их задача — разработать интерфейс, который будет радовать глаз, быть функционально удобным и

соответствовать общепринятым условностям или стандартам жанра. Целью тестирования будет как проверка работоспособности взаимодействия отправляемых с игрового контроллера входных данных и выполняемых в интерфейсе действий, так и проверка общего пользовательского опыта (UX) в целом.

Тестировщику необходимо убедиться, что разработчик использует традиционные раскладки для популярных игровых контроллеров, например, как было сказано ранее, кнопки X, A и □, X используются как кнопки «принятия», а кнопки B и O — как кнопки «отмена». На клавиатуре кнопки W, A, S и D используются в качестве кнопок направления, Ctrl или C — для приседания или ползания, а пробел — для прыжка. Традиционной функцией левой кнопки мыши можно считать выстрел или атаку для шутеров от первого лица или выбор для стратегий. Правая кнопка мыши обычно зарезервирована для прицеливания в шутерах от первого лица или перемещения войск в стратегиях.

Тестировщикам также необходимо убедиться, что использование любого игрового контроллера не дает игроку значительного преимущества перед другими. Поскольку двуручные игровые контроллеры значительно уступают паре клавиатура/мышь по скорости и точности наведения на цель, в них часто добавляют автонаведение на враждебную цель и последующее преследование. Прицел просто «прилипает» к противнику. В этом случае в обязанности тестировщика входит слежение за тем, как далеко прицел начинает следовать за противником и захватывает ли он его голову, которая часто является уязвимой зоной, попаданием в которую наносится дополнительный урон.

6.3 Инструментальная поддержка тестирования игровых контроллеров

Несколько программных инструментов могут использоваться для поддержки тестирования игровых контроллеров путем видеозаписи или схематического представления контроллеров.

Инструменты захвата/воспроизведения позволяют записывать видео с экрана ПК. Они используются при подготовке отчетов о дефектах, чтобы показать характер дефекта и шаги для воспроизведения.

Иные типы инструментов используются для отображения схематического представления двуручного игрового контроллера и команд на видео. Эти инструменты упрощают понимание, обнаружение и анализ дефектов в действиях контроллера.

Существуют сервисы, которые поддерживают тестирование контроллеров ввода и степени отклонения рукояток двуручного игрового контроллера, а также характеристик вибрации игрового контроллера (см. [URL3]). В таких сервисах есть все необходимые инструменты для тестирования двуручного игрового контроллера, помогающие оценить работоспособность контроллера.

Также доступны специальные инструменты для отображения нажатий клавиш на клавиатуре, которые могут быть полезны во время выполнения тестов.

7. Тестирование локализации - 155 минут (К3)

Ключевые термины тестирования

Интернационализация, локализация, соблюдение требований

Ключевые термины, специфичные для тестирования игр

Историческая точность, культурная адаптация, локаль

Цели обучения для Главы 7

7.1 Принципы и концепции тестирования локализации

GaMe-7.1.1 (К1) Узнать шаги тестирования локализации

GaMe-7.1.2 (К1) Запомнить основные цели интернационализации и локализации

GaMe-7.1.3 (К2) Сравнить возможности интернационализации и локализации

7.2 Типы дефектов локализации и их причины

GaMe-7.2.1 (К2) Классифицировать дефекты локализации и их причины

7.3 Подходы к тестированию локализации и их выполнение

GaMe-7.3.1 (К1) Узнать полное и частичное тестирование локализации

GaMe-7.3.2 (К3) Классифицировать типы тестирования локализации

GaMe-7.3.3 (К2) Обобщить задачи тестирования для писателя, редактора, переводчика и тестировщика локализации

7.4 Инструментальная поддержка тестирования локализации

GaMe-7.4.1 (К2) Обобщить использование инструментов для тестирования локализации игр

7.1 Принципы и концепции тестирования локализации

7.1.1 Локализация и интернационализация

Процесс локализации любого программного продукта, будь то видеоплеер, игровой продукт или операционная система, имеет схожие этапы, подходы, методики и отличается уникальными внутренними процедурами, принятыми в компании-разработчике программного обеспечения [Chandler11].

Разработка игры — это идея, которая реализуется через написание кода, дополняется необходимым содержанием и предоставляется игрокам. Важным ранним этапом разработки является определение рынка, для которого будет выпущена игра:

- Этому этапу не всегда уделяется достаточное внимание, что может негативно сказаться на продажах игры в будущем.
- Наличие локальной версии игры обязательно для ее успешного продвижения на национальном рынке.
- Большинство игр разрабатываются с учетом необходимости адаптации продукта под целевой рынок.
- Адаптация игры под целевой рынок может многократно увеличить ее продажи.

Так как выход игры на другие рынки может произойти после окончания разработки локальной версии игры (например, у компании появились деньги на запуск игры на другом рынке или появились партнеры, готовые продвигать продукт в определенных странах), то при разработке игры команда должна учитывать, что игра может быть дополнительно адаптирована под некоторые другие рынки. В противном случае последующие адаптации могут потребовать больших финансовых и ресурсных затрат.

Интернационализация

Чтобы избежать рисков и трудностей адаптации игры для определенного региона, используется процесс, называемый интернационализацией. Интернационализация — это адаптация продукта для потенциального использования практически в любой стране, а локализация — это изменения для использования в конкретном регионе.

Интернационализация программного обеспечения представляет набор действий, выполняемых на начальных этапах разработки, для облегчения последующего перевода и локализации программного обеспечения.

Интернационализация в большинстве случаев осуществляется без привлечения лингвистов-переводчиков и является обязанностью разработчика программного обеспечения.

Интернационализация включает:

- Создание и разработку программного обеспечения таким образом, чтобы не было препятствий для локализации и интернационализации. Возможно использование юникода или предоставление доступа к кодировке символов (при необходимости).
- Создание возможности для использования элементов, которые нельзя использовать до процесса локализации. Например, добавление интегрированной среды для двунаправленного текста в язык разметки DTD (Document Type Definition). Другой пример - добавление в CSS (каскадную таблицу стилей) основы для вертикального текста или нелатинских типографских символов.
- Возможность поддержки региональных, языковых или культурных ссылок. Обычно это включает введение предопределенных локализованных данных или перевод, созданный ранее и сохраненный в специализированном программном обеспечении. Например, форматы даты и времени, местные календари, числовые форматы и системы счисления (римские и арабские цифры), выбор и представление списков, а также использование личных имен и обращений.
- Извлечение локализованных элементов из кода или содержания, чтобы локализованные версии можно было загрузить позже или выбрать в соответствии с предпочтениями пользователя. Как правило, это реализуется в виде загружаемых языковых пакетов.

Этот список не обязательно включает локализацию содержания, программы или продукта. Это методы и подходы к разработке программного обеспечения, которые позволяют легко перейти к локализации в будущем.

Локализация

Локализация — это процесс адаптации программного обеспечения к культуре страны. Частным случаем локализации является перевод пользовательского интерфейса, документации и соответствующих программных файлов [Retsker81]. В локализации участвуют специалисты-переводчики (в ряде случаев носители языка) с большим объемом дополнительных знаний.

Локализация игры включает:

- Перевод текстов игровых диалогов и субтитров, всплывающих подсказок, описаний и сообщений, имен персонажей, названий предметов,
- Перевод заставок, элементов интерфейса и меню,
- Перерисовку текстур и графики,
- Подбор актеров, дубляж и запись звуковых файлов,
- Интеграцию локализованных материалов в игру,
- Перевод и адаптация веб-сайта и печатных материалов проекта,
- Перевод рекламных материалов (новостей, пресс-релизов и маркетинговых материалов),
- Поддержку игры после выхода локализованной версии (обновления, новости и патчи).

Таким образом, тестирование локализации — это проверка того, насколько игровой продукт адаптирован для конкретной целевой аудитории в соответствии с ее культурными, языковыми, религиозными, политическими и другими характеристиками и ориентирами. Обычно здесь учитываются культурно-лингвистические аспекты, в частности перевод пользовательского интерфейса, документации и файлов на другой язык, а также форматы валют, чисел, времени, номеров телефонов и т.д.

Тестирование локализации включает проверку содержимого игрового приложения на соответствие языковым и культурным требованиям, а также специфике конкретной страны или региона. Этот тип тестирования помогает найти дефекты локализации или ошибки перевода в локализованной версии до того, как конечный продукт достигнет пользователя. Цель тестирования локализации — найти и исправить дефекты в разных локализованных версиях продукта для разных рынков и региональных настроек (локалей).

Важно отметить, что локализация — это не просто перевод на несколько языков, а тестирование локализации и лингвистическое тестирование — это не одно и то же. Лингвистическое тестирование в основном состоит из тестирования на предмет обнаружения орфографических, грамматических и стилистических дефектов.

7.1.2 Разница между локализацией игрового продукта и прикладного ПО

Различие между тестированием локализации игровых продуктов и прикладного программного обеспечения основано на понимании того, что эти два типа продукта существенно отличаются друг от друга по различным аспектам, учитываемым при локализации.

Основные отличия:

- Адаптация графического содержания под целевую аудиторию,
- Локализация аудио содержания,
- Локализация и адаптация текстового содержания,
- Соответствие жанровым и литературным особенностям.

Как правило, большой объем графического содержания может потребовать адаптации под целевую аудиторию. К этому типу графики относятся знаки, персонажи, игровые объекты, карты уровней, символы и атрибутика, игровые предметы, рекламные заставки и т.д.

Аудио содержание также может потребовать тщательной локализации. Часто звуковое содержание игрового продукта должно быть продублировано актерами на языке целевой аудитории, а само содержание должно быть адаптировано.

Смешанные стили текстового содержания требуют надлежащего перевода и адаптации. В игре могут использоваться следующие стили:

- Научный стиль (описание механизмов, инструкции),
- Публицистический стиль (газеты, журналы, статьи),
- Художественный стиль (личные дневники, книги),
- Деловой стиль (досье, различные документы),
- Разговорно-бытовой стиль (диалоги персонажей).

Например, в игре в жанре приключений (квесте) будут научный, публицистический и разговорно-бытовой стили, так как игроку необходимо отремонтировать или собрать какое-то устройство по инструкции, найти нужную информацию в газетах и опросить людей.

Существует требование обязательной адаптации содержания игрового продукта с учетом таких факторов, как: историческая достоверность, религиозные, культурные, политические и мировоззренческие особенности целевой аудитории и др. В ряде случаев нарушение данного требования может привести к неоднозначному восприятию продукта целевой аудиторией и повлечь запрет его использования на территории той или иной страны или региона.

При локализации игрового продукта необходимо соблюдать жанровые и литературные особенности. Например, в многопользовательских ролевых играх (RPG) при переводе необходимо учитывать названия рас игроков, названия предметов и т.д., принятые в жанре. Иногда требуется адаптация содержания с использованием стилистически близких и жанровых понятий.

Для осуществления качественного перевода игрового продукта необходимы дополнительные знания и информация. Например, различные ссылки на другие игровые продукты, медиа-работы или реалии-шоу, которые могут содержаться в игре.

Ниже приведены некоторые подходы к локализации игровых продуктов, лежащие в основе тестирования локализации.

7.1.3 Этапы тестирования локализации

Тестирование локализации включает проверку правильности переведенного содержания, различных элементов интерфейса, дефектов и системных сообщений, а также тестирование разделов «Часто задаваемые вопросы» и «Помощь».

Тестирование перевода

Целью тестирования перевода является проверка многоязычного интерфейса игры на наличие дефектов перевода, корректность формата почтовых адресов, имен и фамилий, валют, форматов даты и времени и т.д.

Иногда при разработке необходимо адаптировать внешний вид числа к национальным стандартам (например, единицам измерения). Тестировщик всегда должен помнить, какая система единиц измерения принята в той или иной стране, чтобы четко информировать пользователя о скорости, длине, весе, температуре и т.д.

Сообщение «Вы движетесь со скоростью 62 мили в час» может быть трудно понять игроку из страны, использующей метрические единицы измерения. В этом случае недостаточно просто изменить числовое значение скорости, необходимо еще и адаптировать единицу измерения.

Вышесказанное также относится к валюте, используемой в играх для совершения покупок. Самый простой пример локализации цены — автоматический перевод в валюту региона, где был активирован продукт. В дополнение к конвертации по обменному курсу также должен присутствовать соответствующий символ валюты. Однако при разработке приложения тестировщик также должен учитывать тот факт, что службы распространения программного обеспечения могут устанавливать региональные цены.

При тестировании игр следует обращать внимание на графику. Она должна соответствовать реалиям страны, для которой издается игра. Например, дорожные знаки в разных странах могут выглядеть по-разному. Кроме того, могут быть добавлены картинки и сцены местных праздников. В мусульманских странах кардинально пересматривается графика — убираются все изображения людей и животных и добавляются арабески (сложные восточные средневековые орнаменты, состоящие из геометрических и растительных элементов).

Особой популярностью пользуется празднование китайского Нового года. Во многих играх интерфейс украшен национальной символикой: фонарями, драконами и фейерверками. Также игра может быть дополнена доступным ограниченным временем содержанием: скины персонажей, стилизованные под китайских мифических героев, анимация запуска петард и ракет, катание на свиньях и драконах.

В играх часто приходится адаптировать шутки, а иногда даже приходится адаптировать сюжет под менталитет страны, в которой будет издаваться видеоигра.

Локализация не ограничивается работой с текстом — она включает в себя также согласование культурных особенностей и морально-этических аспектов.

Традиционно выделяют несколько тем и направлений в тестировании локализации, на которые тестировщики должны обратить самое пристальное внимание:

- Религия, включая оккультизм и сатанизм;
- Секс, откровенно одетые персонажи и нецензурная лексика;
- Предвзвещения и стереотипы, связанные с культурой и самими людьми;
- Войны, военные конфликты, терроризм;
- Политика, в том числе специфический взгляд разных стран на историю.

Например, если в видеоигре присутствует отрывок из Корана при озвучивании видеоигры, большинство исламских стран, скорее всего, запретят эту видеоигру. В условиях жесткого контроля за политкорректностью в современном мире к ее соблюдению следует относиться серьезно.

Помимо трудностей, вызванных технической составляющей игры или недостаточной подготовкой к локализации игры, переводчик видеоигры может столкнуться и с рядом сложностей, связанных не с переводимой им игрой, а с необходимостью наличия дополнительных знаний для выполнения качественного перевода.

Под дополнительными знаниями также понимается предыстория других игровых продуктов, медиа-работы или реалити-шоу, которые могут содержаться в игре. Такие отсылки должны быть правильно переведены в соответствии с тем, как они были переведены ранее. Переводчик должен быть в курсе последних событий, происходящих в мире, а также различных популярных фильмов и игр.

Тестирование соблюдения требований

Система возрастных ограничений для игр и приложений учитывает особенности законодательства и культуры отдельных стран. Это позволяет разработчикам точнее определять ограничения содержания и распространять приложения среди той аудитории, для которой они предназначены.

При тестировании локализации следует учитывать законодательные требования, которые поддерживаются в разных регионах. Например, в России возраст совершеннолетия наступает в 18 лет, в США, в зависимости от штата, этот возраст варьируется от 18 до 21 года, а в Японии молодые люди считаются взрослыми с 20 лет. В большинстве игр необходимо соблюдать систему возрастных ограничений, действующую в разных странах.

Одна и та же игра может иметь разные возрастные рейтинги в разных странах.

ESRB (USA)	PEGI (EU)	RARS (Russia)	ACB (Australia)	USK (Germany)
				
				
				
				
				
				

Возрастные рейтинги игр в разных странах
<https://www.kaspersky.com/blog/gaming-age-ratings/11647/>

При тестировании локализации на соответствие законодательству и правовым требованиям необходимо учитывать множество обстоятельств, связанных иногда не столько с целевым языком, сколько с правовыми и культурными особенностями страны, на рынке которой предлагается игровой продукт.

Помимо этого, также стоит обратить внимание на поддержку в игре валюты и операций с ней для каждой страны. Например, если в какой-то стране операции с иностранной валютой запрещены всем, кроме банка этой страны, то попытка запустить игру на этом рынке обязательно приведет к серьезным проблемам у издателей игр.

Косвенно к валютным операциям относится покупка контейнеров с добычей — буквально ящиков с призами, которые, как правило, содержат различные образы персонажей, расходимые предметы, обновления или достижения.

В декабре 2016 года Министерство культуры Китайской Народной Республики объявило о вступлении в силу закона, обязывающего издателя онлайн-игр публиковать информацию о вероятности приобретения всех виртуальных предметов и услуг, начиная с мая 2017 года.

В Австралии на игры с контейнерами с добычей распространяются ограничения на азартные игры, если в них можно играть на деньги или что-либо ценное. Вопросы остаются даже в том случае, когда ценность предмета, существующего только в игре, можно определить только в связи с престижностью этого предмета.

Цвета и символы

Конкретные цвета и символы, которые могут иметь разное значение в зависимости от страны, также следует учитывать при тестировании локализации.

Например, красный цвет для жителей Китая является символом стойкости и веры, в Индии он символизирует чистоту. Европа, напротив, видит в этом цвете грех и жертву. Для жителей Южной Африки это цвет горя. В США и Японии красный цвет символизирует опасность и террористическую угрозу, а египтяне ассоциируют его с трауром. Поэтому при тестировании игрового проекта тестировщику необходимо обращать внимание на цветовые схемы, так как они могут быть важны.

Кроме того, при тестировании следует обратить внимание на поддержку региональных раскладок клавиатуры и горячих клавиш.

В случае, если приложение использует интеграции со сторонними ресурсами (например, облачным хранилищем или социальной сетью), необходимо учитывать их доступность для регионов.

Таким образом, процессы тестирования локализации и интернационализации будут отличаться:

Интернационализация	Локализация
Кодировки UTF	Перевод
Форматы дат	Правовые требования
Направление текста	Валютные операции
	Цвета и символы
	Раскладка клавиатуры и горячие клавиши
	Интеграция со сторонними ресурсами

7.2 Типы дефектов локализации и их причины

7.2.1 Возможные причины дефектов локализации игровых продуктов

Даже крупные компании не застрахованы от «культурных» дефектов в продуктах. При выпуске любого приложения на соответствующий рынок тестировщик должен ознакомиться с прецедентами и действующими регламентами. В некоторых случаях одна фраза или шутка могут привести к полному запрету игры в определенных странах. Необходимо устранить следующие причины возможных дефектов локализации (например, детали или содержание, запрещенный в определенном регионе):

- Изображения,
- Звуки и саундтреки,
- Реалистичные или исторические сцены,
- Фразы или цитаты.

7.2.2 Дефекты и риски локализации

Плохое или недостаточное знание целевого языка или самого продукта может значительно усложнить тестирование локализации, особенно если продукт является представителем жанра симуляторов: медицинских, технических или спортивных [URL4].

Также следует учитывать, что тестирование локализации может быть достаточно длительным процессом, так как требуется время для изучения особенностей разных регионов.

Основные дефекты локализации можно сгруппировать следующим образом.

Технические факторы

- Искаженный текст / Мусорные символы. Искаженный текст появляется в результате декодирования текста с использованием неправильно выбранной кодировки символов. Результатом является систематическая замена символов совершенно не связанными с ними символами, часто из другой системы письма. Как правило, это приводит к нечитаемым текстам.
- Локализованная строка не укладывается в ограничения, заданные интерфейсом (обрезается, прокручивается). Термин, переведенный с одного языка на другой, может использовать разное количество символов.
- Смещение. После локализации приложения макеты элементов пользовательского интерфейса могут потребовать реконфигурации для сохранения исходного выравнивания.
- Нет построчного перевода на локализованный язык. Тестировщикам следует обращать внимание на тексты в диалоговых окнах, изображения или снимки экрана в документах или пользовательском интерфейсе. Все это содержание должно быть локализовано, чтобы соответствовать ожиданиям пользователей.
- Перекрывание. Это происходит, когда определенные элементы управления накладываются друг на друга.
- Отсутствующие тексты. При переводе или создании приложений иногда могут теряться тексты.
- Неправильный шрифт/размер. В разных странах используются разные шрифты и размеры по умолчанию. Например, в азиатских странах по умолчанию обычно используется размер шрифта 9, в то время как в Америке используется размер шрифта 8. Эта проблема обычно не влияет на какую-либо функциональность, но сильно влияет на пользовательский опыт, затрудняя чтение текста.
- Неверные горячие клавиши. В некоторых случаях горячая клавиша недоступна из-за отсутствия буквы на локализованном языке или клавиатуре.
- Переменные в текстовых шаблонах. В разных типах языков (аналитических, синтетических) используются разные формы склонений, падежей и множественного числа, они могут изменять или не изменять окончания и звуки, таким образом переменные могут влиять на формы слов. Одна из самых распространенных ошибок разработчиков — разделение на части предложений, содержащих переменные. Грамматический строй целевого языка является едва ли не главным краеугольным камнем локализации. Ведь при добавлении переменных разработчики часто не учитывают, что в ряде языков их значение может изменить конструкцию фразы.
- Рассинхронизация звуковой последовательности оригинала и перевода. При переводе фразы персонажей могут быть длиннее или, наоборот, короче, чем в оригинале, из-за разного количества слогов. Видеоклипы в играх часто предварительно записываются, что заставляет авторов локализации более тщательно подходить к переводу реплик: важно соблюдать баланс между точностью перевода и количеством слогов, при этом также учитывать синхронизацию. звука с движениями губ персонажей.

Дефекты перевода

- Неверный перевод или транслитерация имен собственных, дат, числовых значений, названий исторических событий, праздников, реалий, жаргонизмов, ненормативной лексики, разговорной лексики, аббревиатур, «запоминающихся» имен и прозвищ и т.д.
- Противоречивые термины. Один и тот же термин должен иметь согласованный перевод во всем приложении. Часто переводы несовместимы друг с другом.

Культурная адаптация содержимого

- Юмор. Отношение к предметам шутки, юмора и сатиры, учитывание допустимости юмора по отношению к отдельным предметам и т.д.
- Религия. Отношение целевой аудитории к религиозным вопросам, включая религиозные объекты, ритуалы и церемонии. Отношение целевой аудитории к маргинальным культурам (сатанизм, язычество).
- Историческая достоверность и восприятие событий. Интерпретация исторических событий, военных конфликтов, открытий в области науки, внешнего вида и характеристик конкретных существующих объектов, особенностей быта народов и т.д.
- Особенности национальной культуры и мировоззрения. Национальные стереотипы, национальная кухня, одежда, образ жизни народов, отношение к детям, отношение к сексуальным меньшинствам, насилие над животными и т.д.
- Правовые ограничения. С учетом возрастных рейтингов, законодательства в сфере защиты детей, религиозных убеждений, пропаганды насилия, наркотиков, сексуальных отношений, расы, терроризма, выступлений против власти и государственного строя, запретов на использование определенных символов и т.д.
- Чрезмерная локализация. Не все нужно локализовать, и некоторые элементы должны сохранять свой первоначальный вид без перевода, например: товарные знаки, логотипы, аббревиатуры, названия продуктов.

7.3 Подходы к локализации и их выполнение

7.3.1 Разница между полным и частичным тестированием локализации

Полное тестирование локализации

Подразумевает тщательную проверку локализации на наличие дефектов. Этот тип эффективен только при разработке новой локализации для клиента, при условии, что все строки переведены и не тестировались ранее. Он самый дорогой, так как включает в себя все виды тестов.

Частичное тестирование локализации

Используется при изменении текста в ранее протестированных локализациях. Проверка определяет строки, которые были изменены в рамках крупного обновления, затронувшего основной сюжет и его эталонную локализацию. Тестируется только модифицированный текст, который должен совпадать с ранее представленным в игре текстом. Такой подход к тестированию считается оптимальным с точки зрения достижения эффективности при минимальных затратах.

7.3.2 Процедуры и подходы в тестировании локализации в жизненном цикле разработки игрового продукта

Тестирование локализации проверяет перевод, вспомогательные файлы, корректность расположения и адаптации элементов интерфейса, а также правила написания текста.

Цель тестирования локализации — убедиться, что игра поддерживает многоязычный интерфейс и функционал, а также отсутствуют проблемы с локализацией (перевод на другой язык, форматы дат и чисел, почтовые адреса, порядок имени и фамилии, валюта и т.д.).

Процесс тестирования локализации включает в себя:

- Определение и изучение списка поддерживаемых языков,
- Проверка правильности перевода, включая элементы пользовательского интерфейса, системные сообщения и дефекты,
- Проверка перевода раздела «Помощь» и сопроводительной документации (при наличии).

Тестирование локализации включает в себя сравнение строк, переведенных командой локализации, со строками эталонной локализации с целью нахождения:

- Грамматических, пунктуационных, синтаксических ошибок,
- Нарушений региональных требований тестируемой локализации (формат времени, даты, меры, соответствие законодательству и т.д.),
- Отсутствия необходимых технических данных (переменных, разделов, используемых для форматирования текста и т.д.),
- Нарушений художественного стиля и локализации контекста отсылок,
- Дефектов отображения текста в игровых интерфейсах (слишком длинные строки, нарушение формата отображения шрифтов и т.п.).

Все вышеперечисленные тесты должны быть выполнены командой тестирования. Однако не всегда есть возможность протестировать грамматику, пунктуацию, синтаксис или художественный стиль и контекст эталонной локализации, поскольку найти тестировщиков со знанием каждого из языков, на которые переведена игра, достаточно сложно. Выполнение этих тестов лежит на стороне команды локализации. Тестировщики проверяют нарушения региональных требований тестируемой локализации, проверяют отсутствие необходимых технических данных, выявляют дефекты отображения текста в игровых интерфейсах. (Подробнее см. в разделе «7.3.3. Типы тестирования локализации»).

Предварительный этап тестирования локализации

Этот этап включает:

- Обеспечение тестировщиков всей необходимой документацией по продукту,
- Создание глоссария и памяти переводов, чтобы помочь тестировщикам правильно интерпретировать используемые термины,
- Предоставление предыдущей версии продукта, если он уже был локализован ранее, для ознакомительных целей,
- Выбор и настройка инструментов управления дефектами — документ или платформа, где будут фиксироваться все дефекты, обнаруженные при тестировании локализации.

Тестирование региональных и культурных характеристик

Это один из самых важных шагов в тестировании локализации. При этом используются снимки экрана или локализованная сборка игры. Необходимо протестировать следующее:

- Форматы даты и времени для выбранного региона,
- Форматы телефонных номеров и адресов,
- Цветовые схемы,
- Соответствие наименований продукции региональным стандартам,
- Формат валюты,
- Единицы измерений.

Лингвистическое тестирование

На этом этапе проверяются особенности языка. Тестировщик должен убедиться, что:

- Используется единая терминология,
- Нет грамматических дефектов,
- Отсутствуют орфографические дефекты,
- Соблюдаются правила пунктуации,
- Используется правильное направление текста (справа налево или слева направо),
- Установлены правильные названия торговых марок, городов, мест, должностей и т.д.

Тестирование пользовательского интерфейса (или внешнего вида)

Важно убедиться в следующем:

- Все подписи к картинкам локализованы,
- Макет локализованной версии такой же, как и в оригинале,
- Разрывы строк на страницах/экранах размещаются в соответствии с правилами целевого языка,
- Разговоры, всплывающие окна и уведомления отображаются правильно,
- Длина строк не превышает существующих ограничений, а текст отображается корректно (иногда текст перевода длиннее оригинала и не помещается на кнопках).

Тестирование функциональности

Необходимо проверить, правильно ли работает локализованное приложение, поэтому обращается внимание на:

- Функциональность локализованного продукта,
- Функции ввода информации,
- Поддержку специальных символов для разных локалей и языков,
- Поддержку сочетаний клавиш,
- Поддержку различных шрифтов,
- Поддержку различных разделителей форматов.

Тестирование видео и звукового ряда

Это тестирование включает проверку:

- Соответствия звукового ряда элементу игры, к которому он принадлежит,
- Соответствия звукового ряда полу персонажа,
- Чистоты звучания (то есть отсутствие помех, а также синхронизация гаммы оригинала и перевода и их громкости по отношению друг к другу),
- Достоверности звукового ряда, включая историческую достоверность,
- Стилистических и культурологических особенностей звукового ряда (акценты, особенности речи).

7.3.3 Типы тестирования локализации

«Коробочная» локализация

Если игра выпущена и продается на физическом носителе, то, что написано на упаковке, должно быть локализованным. Если игра продается не на физическом носителе, а на платформе, то переводится ее страница в магазине: описание и снимки экрана. «Коробочная» локализация на этом ограничивается.

Локализация интерфейса

В игре будут переведены описание и «коробка», интерфейс, страница справки, надписи на кнопках и т.д. Необычный тип локализации возникает в ситуации, когда надпись на кнопке «Играть» выполнена на одном языке, а сюжет полностью на другом языке.

Локализация текста

Обычно все тексты в игре должны быть переведены вплоть до субтитров. Это означает, что пользователь может слушать и пытаться понять, например, афроамериканский сленг в игре, но при этом видеть субтитры на другом языке.

Локализация с озвучкой

Переводятся речь и диалоги, которые озвучиваются актерами. Локализация с озвучкой, сделанная на хорошем уровне, не воспринимается как нечто чужеродное.

Графическая локализация

Любая игра содержит какой-то движок, дизайн, графические объекты, текстуры — все то, что не является текстом, например, надпись на заборе в игре. Графическая локализация подразумевает, что все надписи внутри должны быть переведены. Это могут быть газеты, вывески магазинов, какие-то записки.

Часто действия в игре происходят в определенных местах. Подход к локализации может быть разным. Если какой-то предмет (например, газетные заметки) помогает в сюжете, их необходимо перевести, иначе будет утеряна важная сюжетная составляющая. При этом надписи на стенах на языке того места, где происходит игра, переводить не нужно, если это просто аранжировка.

Глубокая локализация — культурная адаптация

Культурная адаптация – это когда игра полностью переделана. Остается только исходный код и механики. Разработчики могут переделать текстуры, сюжет, диалоги и модели персонажей и сделать совершенно другую игру на движке видеоигры. Делается это достаточно редко, но такой способ локализации все же встречается. Это делается в тех случаях, когда игра без такой адаптации не может использоваться аудиторией и продаваться на конкретном рынке.

7.3.4 Области ответственности вовлеченных специалистов

Роль	Области ответственности
Сценарист Проектировщик повествования	Разработка стратегии и плана локализации. Определение терминологии, перевод понятий, характерных для игрового жанра, и труднопереводимых терминов, связанных с проектом. Объяснение контекста редакторам и переводчикам.
Переводчик	Перевод текстов.
Редактор	Вычитка переведенного материала.

Роль	Области ответственности
	Тестирование текста на соответствие культурным особенностям и морально-этическим аспектами языка, на котором осуществляется локализация. Контроль единообразия стиля перевода.
Тестировщик локализации	Проверка правильности перевода (контекст и смысл реплик, связность текста, соответствие стилю игры). Проверка технического соответствия перевода (внешний вид, интерфейсы, функциональность шрифтов, разделителей, спецсимволов и т.д.). Контроль технической интеграции перевода (с внутренними и внешними приложениями).

7.4 Инструментальная поддержка тестирования локализации

Инструменты тестирования локализации можно использовать на разных этапах тестирования и для решения разных задач. Эти инструменты включают:

- Визуальные инструменты сравнения строк. Эти инструменты помогают сравнивать строки эталонной и целевой локализации.
- Инструменты автоматического сравнения строк. К инструментам этого типа относятся любые вспомогательные скрипты, программы, утилиты, в результате работы которых можно получить данные о наличии дефектов любого рода в тестируемой локализации.
- Инструменты для проверки файловой структуры локализации на наличие отсутствующих или избыточных файлов.
- Примеры автоматизированных инструментов для тестирования локализации включают:
 - инструменты, сравнивающие снимки экрана для двух локализаций (эталонной и тестируемой) и генерирующие отказы при наличии существенного несоответствия,
 - инструменты, определяющие наличие изменений в целевой локализации на основании наличия изменений в эталоне,
 - инструменты, которые сравнивают переменные и числовые значения в каждой паре ссылок на строки и целевую локализацию.

8. Ссылки

8.1 Стандарты

[ISO25000] ISO/IEC 25000:2014, Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE

[ISO25010] ISO/IEC 25010:2011, Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models

8.2 Документы ISTQB

[ISTQB_AL_SEC] ISTQB Advanced Level Security Testing Syllabus, Version 2016; Программа обучения Продвинутого уровня Тестировщик безопасности, Версия RSTQB 2022

[ISTQB_ALTA_SYL] ISTQB Advanced Level Test Analyst Syllabus, Version 3.1.2

[ISTQB_ALTTA_SYL] ISTQB Advanced Level Technical Test Analyst Syllabus, Version 4.0

[ISTQB_ALTM_SYL] ISTQB Advanced Level Test Manager Syllabus, Version 2012; Программа обучения Продвинутого уровня Руководитель тестирования, Версия RSTQB 2017

[ISTQB_CTFL_MAT] ISTQB Mobile Application Tester, Version 2019; Программа обучения Базового уровня Тестирование мобильных приложений, Версия RSTQB 2019

[ISTQB_EXAM_S&R] ISTQB Exam Structure and Rules, Game Testing, Version 1.0

[ISTQB_FL_AT] ISTQB Foundation Level Agile Tester Syllabus, Version 2014; Программа обучения Сертифицированный тестировщик в сфере гибких методологий, Версия RSTQB 2014

[ISTQB_FL_PT] ISTQB Foundation Level Performance Testing Syllabus, Version 2018; Программа обучения Базового уровня Тестирование Производительности, Версия RSTQB 2018

[ISTQB_FL_SYL] ISTQB Foundation Level (Core) Syllabus, Version 2018; Программа обучения Базового уровня, Версия RSTQB 2018

[ISTQB_GLOSSARY] Глоссарий терминов тестирования ISTQB, <https://glossary.istqb.org/>

[ISTQB_UT_SYL] ISTQB Foundation Level Usability Testing Syllabus, Version 2018

8.3 Книги

[Nystrom14] Nystrom, R. (2014). Game programming patterns. Genever Benning., ISBN: 978-0990582908. URL: <https://www.gameprogrammingpatterns.com/>

[Gregory18] Gregory, J. (2018). Game engine architecture. AK Peters/CRC Press., ISBN: 978-1466560017. URL: <https://www.gameenginebook.com/>

[Buttfield19] Buttfield-Addison, P., Manning, J., & Nugent, T. (2019). Unity game development cookbook: essentials for every game. O'Reilly Media., ISBN: 9781491999158

[Lee16] Lee, J. (2016). Learning unreal engine game development. Packt Publishing Ltd., ISBN: 9781784398156

[Tavakkoli18] Tavakkoli, A. (2018). Game Development and Simulation with Unreal Technology. CRC Press., ISBN-13: 978-1498706247

[Romero19] Romero, M., & Sewell, B. (2019). Blueprints Visual Scripting for Unreal Engine: The faster way to build games using UE4 Blueprints. Packt Publishing Ltd., ISBN: 9781789347067

[Chandler11] Chandler, H. (2011). The Game Localization Handbook 2nd Edition, Jones & Bartlett Learning, ISBN: 0763795933

[Retsker81] Retsker, Ya. I. (1981). Textbook for translation from English into Russian. M.: Prosveschenie. (In Russian).

8.4 Ссылки (Интернет)

Примечание: все ссылки актуальны на 28 апреля 2022.

- [URL1] <https://research.ncl.ac.uk/game/mastersdegree/workshops/technicalrequirementschecklists/Technical%20Requirements%20Checklist%20Workshop.pdf>
- [URL2] https://docs.microsoft.com/en-us/gaming/gdk/_content/gc/live/get-started/live-xbl-overview
- [URL3] <https://gamepad-tester.com/>
- [URL4] <https://igda-website.s3.us-east-2.amazonaws.com/wp-content/uploads/2021/04/09142137/Best-Practices-for-Game-Localization-v22.pdf>

9. Приложение А – Цели обучения/Когнитивные уровни знаний

Следующие цели обучения применимы к данной программе обучения. Каждая тема в программе обучения рассматривается в соответствии с поставленной целью обучения.

Формулировка цели обучения начинается с глагола действия, соответствующего ее когнитивному уровню знаний, как указано ниже.

Уровень 1: запомнить (K1)

Кандидат запомнит, узнает и вспомнит термин или понятие.

Глаголы действия: запомнить, узнать.

Примеры
Запомнить концепцию пирамиды тестирования.
Узнать типичные цели тестирования.

Уровень 2: понять (K2)

Кандидат может выбрать причины или объяснения утверждений, относящихся к теме, а также может обобщить, сравнить, классифицировать и дать примеры понятий тестирования.

Глаголы действия: классифицировать, сравнить, различить, объяснить, привести примеры, интерпретировать, обобщить.

Примеры	Примечания
Классифицировать инструменты тестирования в соответствии с их назначением и действиями по тестированию, которые они поддерживают.	
Сравнить разные уровни тестирования.	Может использоваться для поиска сходств, различий или того и другого.
Отличить тестирование и отладку.	Ищет различия между понятиями.
Различить риски проекта и продукта.	Позволяет отдельно классифицировать два (или более) понятия.
Объяснить влияние контекста на процесс тестирования.	
Привести примеры, почему тестирование необходимо.	
Сделать вывод о первопричине дефектов из заданного профиля отказов.	
Обобщить итоги процесса рецензирования рабочего продукта.	

Уровень 3: применить (K3)

Кандидат может выполнить процедуру, столкнувшись со знакомой задачей, или выбрать правильную процедуру и применить ее к заданному контексту.

Глаголы действия: применить, реализовать, подготовить, использовать.

Примеры	Примечания
Применить анализ граничных значений для получения тестовых сценариев из заданных требований.	Должно относиться к процедуре/технике/процессу и т.д.
Реализовать методы сбора метрик для поддержки технических и управленческих требований.	
Подготовить тесты на возможность установки мобильных приложений.	
Использовать трассируемость для мониторинга хода тестирования на предмет его полноты и согласованности с целями тестирования, стратегией тестирования и планом тестирования.	Может использоваться в целях обучения, которые предполагают, что кандидат может использовать технику или процедуру. Похоже на: «применить».

10. Приложение Б – Матрица трассируемости бизнес-результатов и целей обучения

В этом разделе приведена трассируемость бизнес-результатов и целей обучения программы Сертифицированный тестировщик игр ISTQB®.

Бизнес-результаты: Сертифицированный тестировщик игр ISTQB®		GaMe-1	GaMe-2	GaMe-3	GaMe-4	GaMe-5	GaMe-6
GaMe-1	Описывать базовые концепции игр и тестирования игрового программного обеспечения.	17					
GaMe-2	Определять риски, цели и требования к игровому ПО для нужд и ожиданий заинтересованных сторон.		12				
GaMe-3	Концептуально проектировать, реализовывать и выполнять базовые тесты игрового программного обеспечения.			6			
GaMe-4	Знать подходы к тестированию игрового ПО и их назначение.				9		
GaMe-5	Знать инструменты, поддерживающие тестирование игр.					7	
GaMe-6	Определять, как мероприятия по тестированию согласуются с жизненным циклом разработки программного обеспечения, и снижение стоимости разработки и публикации видеоигр.						6

			GaMe-1	GaMe-2	GaMe-3	GaMe-4	GaMe-5	GaMe-6
1. Специфика тестирования игр								
GaMe-1.1.1	K1	Узнать цели и специфику тестирования игр	x					
GaMe-1.1.2	K2	Привести примеры продуктовых рисков в игровом ПО	x	x				
GaMe-1.1.3	K2	Привести примеры дефектов, специфичных для игрового ПО	x					
GaMe-1.1.4	K2	Резюмировать, как могут быть снижены риски тестирования игр		x				
GaMe-1.1.5	K2	Сравнить мероприятия по тестированию игр и самой игры	x					
GaMe-1.2.1	K1	Узнать роль и задачи команды разработки игрового ПО	x					
GaMe-1.3.1	K1	Запомнить мероприятия тестирования в жизненном цикле разработки игрового ПО					x	
2. Тестирование игровых механик								
GaMe-2.1.1	K2	Классифицировать виды игровых механик	x					
GaMe-2.1.2	K2	Различить тестирование игровой и неигровой механики			x			
GaMe-2.1.3	K2	Различить тестирование ключевой механики и мета-механики			x			
GaMe-2.1.4	K2	Различить тестирование клиентской, серверной и клиент-серверной механик			x			
GaMe-2.1.5	K2	Привести примеры дефектов игровой механики		x				

			GaMe-1	GaMe-2	GaMe-3	GaMe-4	GaMe-5	GaMe-6
GaMe-2.2.1	K2	Обобщить основные подходы и объекты тестирования на разных этапах создания игрового продукта				x	x	
GaMe-2.2.2	K2	Выделить важность тестирования игровых механик	x					
GaMe-2.2.3	K2	Выделить важность рецензирования документации, описывающей игровые механики	x					
GaMe-2.2.4	K3	Применить фундаментальные подходы тестирования игровых механик				x		
3. Тестирование графики								
GaMe-3.1.1	K2	Объяснить особенности графического содержания игрового продукта	x					
GaMe-3.1.2	K2	Классифицировать типы дефектов графического содержания		x				
GaMe-3.2.1	K2	Обобщить основные подходы к художественному тестированию				x		
GaMe-3.2.2	K2	Обобщить основные подходы к техническому тестированию				x		
GaMe-3.2.3	K2	Обобщить основные подходы к тестированию игрового процесса				x		
GaMe-3.3.1	K3	Применить фундаментальные подходы тестирования графики			x			
GaMe-3.3.2	K2	Объяснить важность проверки графики на историческую достоверность		x				
GaMe-3.4.1	K2	Обобщить использование инструментов тестирования графики					x	
4. Тестирование звука								
GaMe-4.1.1	K1	Запомнить особенности звукового наполнения игрового продукта	x					
GaMe-4.2.1	K1	Запомнить типы дефектов звукового содержания		x				
GaMe-4.2.2	K2	Классифицировать дефекты звукового содержания		x				
GaMe-4.3.1	K2	Обобщить основные подходы к тестированию звукового содержания				x		
GaMe-4.3.2	K2	Обобщить основные подходы к тестированию сочетания музыки и звуков				x		
GaMe-4.3.3	K2	Обобщить основные подходы к тестированию музыкальной композиции				x		
GaMe-4.4.1	K2	Объяснить уровни тестирования аудио-музыкального содержания	x					
GaMe-4.4.2	K1	Вспомнить особенности интеграции звуков в клиенте	x					
GaMe-4.4.3	K1	Вспомнить зоны ответственности звукового тестирования						x
GaMe-4.4.4	K3	Применить подходы к тестированию звука			x			
GaMe-4.5.1	K2	Обобщить использование инструментов тестирования звука					x	
5. Тестирование игровых уровней								
GaMe-5.1.1	K1	Запомнить компоненты уровней видеоигр	x					
GaMe-5.1.2	K2	Классифицировать дефекты, характерные для уровней видеоигр		x				
GaMe-5.2.1	K2	Обобщить тесты, проводимые на разных этапах создания уровней видеоигр				x		x
GaMe-5.2.2	K2	Сравнить зоны ответственности специалистов, принимающих участие в тестировании уровней видеоигр						x
GaMe-5.3.1	K2	Обобщить использование инструментов для тестирования уровней видеоигр					x	
6. Тестирование игровых контроллеров								
GaMe-6.1.1	K2	Классифицировать типовые и специализированные устройства ввода	x					
GaMe-6.1.2	K2	Привести примеры различных устройств ввода с точки зрения их применения		x				

			GaMe-1	GaMe-2	GaMe-3	GaMe-4	GaMe-5	GaMe-6
GaMe-6.1.3	K1	Запомнить различные типы игровых контроллеров	x					
GaMe-6.1.4	K2	Классифицировать дефекты игрового продукта, связанные со спецификой игровых контроллеров, и возможные причины их возникновения		x				
GaMe-6.2.1	K2	Привести примеры тестовых условий, которые необходимо учитывать при тестировании игровых контроллеров		x				
GaMe-6.2.2	K2	Классифицировать задачи для UX-специалистов, тестировщиков и проектировщиков игр во время тестирования игр						x
GaMe-6.3.1	K2	Обобщить использование инструментов для тестирования поведения игровых контроллеров					x	
7. Тестирование локализации								
GaMe-7.1.1	K1	Узнать шаги тестирования локализации	x					
GaMe-7.1.2	K1	Запомнить основные цели интернационализации и локализации			x			
GaMe-7.1.3	K2	Сравнить возможности интернационализации и локализации		x				
GaMe-7.2.1	K2	Классифицировать дефекты локализации и их причины		x				
GaMe-7.3.1	K1	Узнать полное и частичное тестирование локализации				x		
GaMe-7.3.2	K3	Классифицировать типы тестирования локализации						x
GaMe-7.3.3	K2	Обобщить задачи тестирования для писателя, редактора, переводчика и тестировщика локализации						x
GaMe-7.4.1	K2	Обобщить использование инструментов для тестирования локализации игр					x	

11. Приложение В – Описание изменений

Это первая версия программы обучения «Сертифицированный тестировщик игр ISTQB®». Разработка этого модуля началась в 2020 году как реакция на растущий рынок интерактивного развлекательного программного обеспечения, где тестирование является высокотехнологичной областью, но не существует стандарта для тестирования игр.

Команда разработчиков модуля «Сертифицированный тестировщик игр ISTQB®» оценивает количество специалистов в области разработки игр более чем в 400 000 человек.

Модуль «Сертифицированный тестировщик игр ISTQB®» дает возможность получить международный сертификат, чтобы быть признанным среди коллег, работодателей и клиентов.

12. Приложение Г – Термины, специфичные для тестирования игр, и прочие термины

Русскоязычный термин	Англоязычный термин	Определение
Акселерометр	Accelerometer	Датчик, который позволяет программному обеспечению определять и передавать информацию о том, как игровой контроллер расположен в пространстве.
Анимация	Animation	Техника создания иллюзии движущихся изображений с использованием последовательности неподвижных изображений, сменяющих друг друга с высокой частотой.
Белый ящик	White box	Этап разработки игрового уровня, на котором из временных моделей создается трехмерный макет для проверки геометрии уровня.
Бинауральный эффект	Binaural effect	Эффект, возникающий при восприятии звука на двух соседних частотах в каждое ухо одновременно. Он позволяет определить направление на источник звука, что делает звуковое восприятие объемным.
Босс	Boss	Управляемый компьютером противник, победить которого гораздо сложнее, чем обычного врага в игре. См. также мобильный объект (моб), неигровой персонаж (NPC)
Взлом	Hacking	Нечестный способ получить преимущество в игре.
Визуальный эффект (VFX)	Visual effect (VFX)	Создание или манипулирование любыми изображениями на экране, которые физически не существуют в реальной жизни.
Гироскоп	Gyroscope	Датчик, измеряющий ориентацию и угловую скорость тела в системе покоя.
Громкость	Volume	Степень громкости или интенсивности звука. Merriam-Webster
Движок игры	Video game engine	Программное обеспечение, в которое встроены все остальные компоненты игры.
Двухручный игровой контроллер	Gamepad	Тип игрового контроллера чаще всего для двух рук, используемого с игровыми приставками. См. также игровой контроллер.
Документация игры	Video game design document	Подробное описание разрабатываемой игры.
Задержка	Lag	Снижение производительности игр из-за медленного подключения к Интернету.
Звуковая зона	Sound zone	Технология, которая позволяет создавать внутриигровые области с реальными звуковыми свойствами.
Звуковое окружение	Ambient	Звук окружающей среды, который характерен для определенного места, ситуации или этапа игры.
Звуковой разрыв	Sound discontinuity	Неожиданное длительное отсутствие звука или резкое прекращение звука.
Звуковой эффект	Sound effect	Звук, отличный от речи или музыки, искусственно созданный для звучания объекта игры.
Звуковые прерывания	Skipping	Неисправный звуковой фон, который звучит так, будто части звуковой дорожки пропущены.

Русскоязычный термин	Англоязычный термин	Определение
Игра	Video game	Развлекательное программное обеспечение, в которой игрок управляет изображениями на экране.
Игровая приставка	Video game console	Электронное устройство, предназначенное для игр.
Игровое состояние	Video game state	Значение всех параметров и переменных, которые описывают все объекты в игре в определенный момент времени.
Игровой контроллер	Video game controller	Устройство, используемое для ввода данных в игру.
Игровой объект	Video game object	Любой объект, с которым игрок может взаимодействовать в игре.
Игровой процесс	Gameplay	Правила, описывающие взаимодействие между миром игры и игроком.
Игровой ресурс	Video game resource	Сущность или характеристика игры, важная для игрока и достижения его целей в игре.
Игровой руль	Racing wheel	Тип двуручного игрового контроллера, используемого для имитации рулевого колеса, педалей и рычага переключения передач. См. также двуручный игровой контроллер
Игровой уровень синоним: уровень игры	Video game level	Отдельная локация в виртуальном мире игры.
Игровые механики	Video game mechanics	Действия в рамках игры, которые определяются конкретными правилами.
Игрок против игрока (PvP)	Player versus player (PvP)	Режим игры, в котором игрок взаимодействует с одним или несколькими игроками.
Издатель	Publisher	Компания, которая финансирует разработку, распространяет и продает игры. Википедия
Изображения, созданные компьютером (CGI)	Computer-generated imagery (CGI)	Неподвижные и движущиеся изображения, сгенерированные при помощи трёхмерной компьютерной графики.
Искажение	Distortion	Изменение формы звуковой волны во время обработки. The Oxford Companion to the English Language.
Историческая точность	Historical accuracy	Соответствие игрового содержания реальной жизни.
Квест	Quest	Жанр игр, включающий задания, основанные либо на интерактивном сюжете, либо на повышении уровня персонажа. Википедия
Клиентские механики	Client mechanics	Механики, которые работают на стороне клиента игры. См. также игровые механики.
Ключевые механики	Core mechanics	Механики, определяющие действия, которые игрок повторяет на протяжении всей игры. См. также игровые механики.
Коллизия	Collision	Технология, отвечающая за взаимодействие объектов игры.
Контейнер с добычей	Loot box	Виртуальный предмет в играх, при использовании которого игрок получает случайные виртуальные предметы различной ценности и назначения.
Культурная адаптация	Cultural adaptation	Процесс адаптации условий окружения игры для создания характерных черт культуры, которые будут восприниматься аудиторией и продаваться на каком-либо рынке.

Русскоязычный термин	Англоязычный термин	Определение
Локаль	Locale	Набор параметров, определяющих язык, регион и другие специальные варианты предпочтений в пользовательском интерфейсе игры.
Магазин	Store	Сервис, обеспечивающий цифровое распространение игр и другого игрового содержания.
Маппинг	Mapping	Дисциплина разработки игр для создания уровней. Создание трехмерной проекции на физический объект окружающей среды с учётом его геометрии и местоположения в пространстве. См. также игровой уровень.
Место действия	Setting	Среда игры, в которой происходит действие.
Мета-механики	Meta mechanics	Механики, которые проходят вне основного игрового процесса, но которые влияют на него тем или иным образом. См. также игровые механики.
Минимальный элемент игрового поля (тайл)	Tile	Небольшой объект или фрагмент, из которого строится уровень или другое крупное изображение в игре.
Многоплатформенность	Multiplatform	Возможность использования программного обеспечения на разных платформах.
Мобильный объект (моб)	Mobile object (Mob)	Любой нестатичный объект в игре, который может перемещаться по определенной области или стоять на месте.
Недоступное место	Inaccessible place	Место в мире игры, доступ к которому не предоставляется игроку.
Неигровой персонаж (NPC)	Non-player character (NPC)	Персонаж, управляемый компьютером. См. также мобильный объект (моб).
Область попадания. Синоним: поражения	Hit box. Synonym: affected area	Упрощенный трехмерный объект модели для обнаружения и обработки столкновений объектов в игре.
Окклюзия	Occlusion	Процесс, имитирующий восприятие звуков в «закрытых» средах.
Освещение сцены	Scene lighting	Количество, размер, цвет и резкость света, которые окружают персонажа, должны соответствовать сцене игры.
Платформер	Platformer	Жанр игр, который включает в себя прыжки по платформам, подъем по лестнице и сбор предметов, необходимых для победы над врагами или прохождения уровня.
Повествование	Narrative	История, переданная с помощью игровых механик. См. также дизайн игр, игровые механики.
Полигон	Polygon	Набор вершин, ребер и граней, определяющих форму многогранного объекта в трехмерной компьютерной графике и объемном моделировании.
Помощь	Help	Документация, которая объясняет особенности программного обеспечения и помогает пользователю понять ее возможности.
Предварительный этап	Pre-production stage	Процесс планирования и документирования элементов игры.
Привязка модели персонажа к скелету (скиннинг)	Skinning	Процесс привязки трехмерного персонажа к скелету модели в игре. См. также создание скелета модели (риггинг).

Русскоязычный термин	Англоязычный термин	Определение
Проектирование игр	Video game design	Процесс создания формы и содержания игрового процесса разрабатываемой игры.
Проектирование уровней	Level design	Этап разработки игр, включающий создание уровней и миссий.
Проектировщик игр	Video game designer	Лицо, выполняющее разработку правил и содержания игрового процесса. См. также проектирование игр.
Прототип уровня	Level prototype	Ранний образец уровня игры, созданный в качестве доказательства концепции. См. также игровой уровень.
Растр	Raster	Тип графики, представляющий сетку пикселей, видимых на экране.
Реверберация	Reverb	Эффект эха, создаваемый электроникой. Merriam-Webster
Редактор уровней	Level editor	Программное обеспечение, которое используется для создания и редактирования локаций и окружения в игре.
Ролевая игра (RPG)	Role-playing game (RPG)	Жанр игр, в которых игрок берет на себя роли персонажей в вымышленном месте действия.
Рукоятка (стик)	Stick	Рукоятка на базе игрового контроллера, подвижность которой ограничена двумя степенями свободы. См. также игровой контроллер.
Свободное тестирование	Ad hoc testing	Неформальное тестирование, проводимое без анализа и проектирования тестов.
Сенсорный экран	Touchscreen	Контроллер для игр, использующий физическое прикосновении игрока к экрану. См. также игровой контроллер.
Серверные механики	Server mechanics	Игровая механика, работающая на сервере игры. См. также игровые механики.
Серый ящик	Grey box	Этап разработки игрового уровня, на котором создается "черновик" трехмерного макета из одноцветных неопределенных форм для тестирования игрового процесса.
Симулятор единоборств	Fighting	Жанр игр, имитирующих рукопашный бой небольшого числа (как правило двух) персонажей в ограниченном пространстве и времени.
Скоростное прохождение игры	Speedrunner	Попытка завершить игру как можно быстрее. Википедия
Соблюдение требований	Compliance	Соответствие рабочего продукта стандартам, соглашениям или правилам, содержащимся в законах и других предписаниях.
Создание скелета модели (риггинг)	Rigging	Процесс создания скелета модели в игре для последующей его анимации.
Стратегическая игра (RTS)	Real-time strategy (RTS)	Жанр игр, в которых игрок берет на себя управление подразделениями и сооружениями, находящимися под его контролем.
Структурная геометрия	Structural geometry	Технология, обеспечивающая рельеф местности и поверхность, по которой могут передвигаться персонажи игры.
Сценарное событие (триггер)	Trigger	Одно или несколько действий, инициирующих событие.

Русскоязычный термин	Англоязычный термин	Определение
Текстура	Texture	Растровое изображение, наложенное на поверхность многоугольной модели для придания ей цвета или имитации рельефа.
Тестирование во время игры синоним: игровое тестирование	Playtest	Свободное тестирование игры игроками для выявления отказов и сбора отзывов.
Тестирование эргономичности	Ergonomics testing	Тестирование для определения удобства использования компонента или системы
Трансформация	Morphing	Визуальный эффект, заключающийся в плавном превращении одного объекта в другой.
Трекбол	Trackball	Игровой контроллер, в котором свободно вращающийся шар используется в качестве устройства ввода для игры. См. также игровой контроллер.
Трёхмерная модель	3D model	Объёмное цифровое изображение реального или вымышленного объекта
Уровни детализации (LoD)	Level of detail (LoD)	Технология, с помощью которой создается менее ресурсоемкая и упрощенная копия объекта для его отображения игроку на дальнем расстоянии.
Фоновый звук	Background sound	Внутриигровой звук для улучшения атмосферы игры.
Функциональное тестирование	Functional testing	Вид тестирования, которое выполняется для оценки того, удовлетворяет ли компонент или система функциональным требованиям.
Цветовое кодирование	Color coding	Процесс маркировки объектов различными цветами в качестве средства идентификации.
Частота кадров	Frame rate	Количество измененных кадров в единицу времени.
Шум	Noise	Звук, который нежелателен или мешает воспринимать что-либо на слух
Шутер от первого лица	First person shooter	Поджанр игр, основанный на боевых действиях с применением оружия, где игрок видит происходящее глазами главного героя.
Элементы игровой поверхности (террейн)	Terrain	Набор элементов, относящихся к игровой поверхности, по которой перемещается персонаж в игре.
Этап концепта	Concept stage	Начальный этап планирования проекта разработки игры, который включает создание основных концепций и написании первоначальных проектных документов, описывающих будущую игру.
Этап постпроизводства	Post-production stage	Процесс поддержки и распространения игры, а также маркетинга после ее выпуска.
Этап производства	Production stage	Процесс разработки игры.

13. Приложение Д – Предметный указатель

- Акселерометр, 62
Анимация, 13, 31, 32, 33, 36, 41, 46, 47, 49, 68
Белый ящик, 57
Бинауральный эффект, 45
Босс, 57
Взлом, 63
Визуальный эффект (VFX), 23, 31, 36
Гироскоп, 62
Громкость, 45, 47, 48, 49, 50, 52, 75
Движок игры, 29, 34, 37, 38, 40, 41, 42, 49, 50, 51, 53, 59, 76
Двухручный игровой контроллер, 60, 61, 62, 63, 64
Документация игры, 16, 17, 23, 24, 26, 27, 49, 50, 62
Задержка, 20, 37, 46, 47
Звуковая зона, 47
Звуковое окружение, 44, 45
Звуковой разрыв, 49
Звуковой эффект, 46, 47, 49
Звуковые прерывания, 46
Игра, 12, 13, 14, 15, 24, 25, 27, 30, 32, 33, 34, 35, 38, 55, 62, 65, 67, 68, 69, 74, 75, 76
Игровая механика, 12, 13, 15, 17, 32, 42, 53, 56, 57, 58
Игровая приставка, 14, 60
Игровое состояние, 19, 22, 25, 26, 27
Игровой контроллер, 60, 62, 63, 64
Игровой объект, 33, 34, 58
Игровой процесс, 15, 17, 19, 20, 21, 23, 26, 27, 31, 32, 34, 39, 40, 41, 43, 49, 51, 54, 55, 56, 57, 58, 61, 62, 63
Игровой ресурс, 48
Игровой руль, 60
Игровой уровень, 21, 23, 27, 53, 54, 55, 56, 57, 59
Игрок против игрока (PvP), 21, 34
Издатель, 14, 62, 70
Изображения, созданные компьютером (CGI)., 30
Интернационализация, 66, 71
Искажение, 46, 47
Историческая точность, 41, 42
Квест, 67
Клиентская механика, 20, 21
Ключевая механика, 19, 20, 21, 23, 24
Коллизия, 31, 34, 35, 37, 39, 40, 55, 58
Контейнер с добычей, 70
Культурная адаптация, 73, 76
Локализация, 15, 65, 66, 67, 68, 69, 71, 72, 73, 74, 75, 76, 77
Локаль, 67, 75
Магазин, 22, 25, 75
Маппинг, 40
Мета-механика, 19, 20, 21, 24
Минимальный элемент игрового поля (тайл), 35
Многоплатформенность, 13, 29
Мобильный объект (моб), 22, 23, 55
Недоступное место, 55
Неигровой персонаж (NPC), 44, 46
Область поражения, 34, 35
Окклюзия, 45
Освещение сцены, 32
Платформер, 34
Повествование, 32, 50, 56
Полигон, 30, 31, 33, 34, 37
Помощь, 68, 74
Предварительный этап, 17
Привязка модели персонажа к скелету (скиннинг), 31, 36
Проектирование уровней, 54, 55, 56
Проектировщик игр, 16, 17, 19, 23, 41, 50, 57, 63
Прототип уровня, 57
Растр, 30, 31
Реверберация, 45
Редактор уровней, 53, 54, 59
Ролевая игра (RPG), 30, 31, 39, 45, 68
Рукоятка (стик), 63
Свободное тестирование, 23, 24, 27
Сенсорный экран, 61

Серверная механика, 20, 21, 22
Серый ящик, 56
Симулятор единоборств, 31, 34
Скоростное прохождение игры, 34
Соблюдение требований, 63, 69
Создание скелета модели (риггинг),
36
Стратегическая игра (RTS), 45
Структурная геометрия, 53
Сценарное событие (триггер), 42,
54, 57, 58
Текстура, 29, 31, 33, 34, 35, 37, 38,
39, 40, 42, 58, 59, 66, 76
Тестирование во время игры, 15,
39, 41, 57, 58
Тестирование эргономичности, 63
Трансформация, 31
Трекбол, 61

Трехмерная модель, 13, 27, 35, 39,
40, 56, 57, 58
Уровень детализации (LoD), 33, 37,
38
Фоновый звук, 43, 54
Функциональное тестирование, 14,
17, 22, 24, 63
Цветовое кодирование, 56
Частота кадров, 24, 31, 33, 36, 37,
39, 46
Шум, 44, 47, 48
Шутер от первого лица, 32, 33, 44,
64
Элементы игровой поверхности
(террейн), 31, 35, 40, 53
Этап концепта, 16
Этап постпроизводства, 18
Этап производства, 21, 24, 37