Министерство образования Самарской области

АССОЦИАЦИЯ ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ "ИНСТИТУТ НАПРАВО.РУ"

УТВЕРЖДЕНО
Приказ директора
от 07.08.2024 № 012/2/с

Воробьев К.А.

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

УП.01.01 Учебная практика по модулю ПМ.01

Специальность 09.02.10 Разработка компьютерных игр, дополненной и виртуальной реальности
Квалификация: разработчик компьютерных игр, дополненной и виртуальной реальности
Форма обучения: очная
На базе основного общего образования
Срок получения образования: 3 года 10 месяцев

Тольятти, 2024г.

Содержание

1. Паспорт фонда оценочных средств 3

1.1. Назначение и область применения 3

1.2. Сведения об образовательной программе 3

1.3. Объём подготовки и формы аттестации 3

1.4. Нормативные и учебно-методические основания 3

2. Проверяемые результаты и средства оценивания 3

3. Организация текущего контроля 4

4. Практические задания 5

5. Промежуточная аттестация и шкала оценивания 11

6. Вопросы для защиты результатов практики 11

Приложение 1. Индивидуальное задание 15

Приложение 2. Дневник практики 15

Приложение 3. Отчёт и протокол проверки 15

Приложение 4. Аттестационный лист 16

1. Паспорт фонда оценочных средств

1.1. Назначение и область применения

Фонд оценочных средств предназначен для текущего контроля успеваемости и промежуточной аттестации по практике «УП.01.01 Учебная практика по профессиональному модулю ПМ.01 «Разработка программных модулей»».

1.2. Сведения об образовательной программе

Основная образовательная программа подготовки специалистов среднего звена по специальности 09.02.10 «Разработка компьютерных игр, дополненной и виртуальной реальности» реализуется ИНСТИТУТ НАПРАВО.РУ. Присваиваемая квалификация — разработчик компьютерных игр, дополненной и виртуальной реальности.

1.3. Объём подготовки и формы аттестации

Общий объём практики — 36 академических часов.

3-й курс, 5-й семестр: 10.11.2026–16.11.2026; объём — 36 академических часов.

Форма промежуточной аттестации — дифференцированный зачёт в 5-м семестре. Защита проводится в пределах часов практики.

1.4. Нормативные и учебно-методические основания

— Федеральный государственный образовательный стандарт среднего профессионального образования по специальности 09.02.10 «Разработка компьютерных игр, дополненной и виртуальной реальности», утверждённый приказом Министерства просвещения Российской Федерации от 25 июня 2024 г. № 441.

— Учебный план по программе подготовки специалистов среднего звена 09.02.10 «Разработка компьютерных игр, дополненной и виртуальной реальности».

— Рабочая программа практики «УП.01.01 Учебная практика по профессиональному модулю ПМ.01 «Разработка программных модулей»» по специальности 09.02.10 «Разработка компьютерных игр, дополненной и виртуальной реальности».

— Положение о практической подготовке обучающихся, утверждённое приказом Министерства науки и высшего образования Российской Федерации и Министерства просвещения Российской Федерации от 5 августа 2020 г. № 885/390.

2. Проверяемые результаты и средства оценивания

Компетенция	Практическое подтверждение	Оценочное средство
ПК 1.1. Формировать алгоритмы разработки программных модулей в соответствии с техническим заданием	Блок-схема или псевдокод; трассировка для пустого, заполненного и частично заполненного инвентаря.	Задание 1; демонстрация и защита
ПК 1.2. Разрабатывать программные модули в соответствии с техническим заданием	Исходный код, инструкция запуска, автоматические тесты.	Задание 2; демонстрация и защита
ПК 1.3. Оформлять программный код в соответствии с установленными требованиями	Оформленный репозиторий, описание интерфейса и пример использования.	Задание 3; демонстрация и защита
ПК 1.4. Использовать систему контроля версий программного кода для коллективной разработки программного кода	История изменений, две ветки, описание разрешённого конфликта.	Задание 4; демонстрация и защита
ПК 1.5. Выполнять отладку программных модулей с использованием специализированных	Протокол воспроизведения, исправление, регрессионные тесты.	Задание 5; демонстрация и

Компетенция	Практическое подтверждение	Оценочное средство
программных средств		защита
ПК 1.6. Осуществлять рефакторинг и оптимизацию программного кода	Сравнение версий, результаты измерений, тесты эквивалентности.	Задание 6; демонстрация и защита
ПК 1.7. Разрабатывать процедуры интеграции программных модулей и платформы/фреймворка	Спецификация интерфейса, схема взаимодействия, план интеграционных тестов.	Задание 7; демонстрация и защита
ПК 1.8. Выполнять интеграцию программных модулей и платформы/фреймворка	Рабочая сцена, конфигурация, протокол интеграционных проверок.	Задание 8; демонстрация и защита
ПК 1.9. Осуществлять инсталляцию, настройку и обслуживание программного обеспечения	Пакет установки, перечень зависимостей, журнал проверки и инструкция отката.	Задание 9; демонстрация и защита

Общая компетенция	Подтверждение
ОК 01. Выбирать способы решения задач профессиональной деятельности применительно к различным контекстам	Наблюдение за работой; дневник; документация; обсуждение решений
ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности	Наблюдение за работой; дневник; документация; обсуждение решений
ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях	Наблюдение за работой; дневник; документация; обсуждение решений
ОК 04. Эффективно взаимодействовать и работать в коллективе и команде	Наблюдение за работой; дневник; документация; обсуждение решений
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста	Наблюдение за работой; дневник; документация; обсуждение решений
ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения	Наблюдение за работой; дневник; документация; обсуждение решений
ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях	Наблюдение за работой; дневник; документация; обсуждение решений
ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности	Наблюдение за работой; дневник; документация; обсуждение решений
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках	Наблюдение за работой; дневник; документация; обсуждение решений

3. Организация текущего контроля

Руководитель проверяет готовность рабочего места, понимание задания, соблюдение безопасности, ход работы, промежуточные версии и ведение дневника. Для каждой работы фиксируются дата, выполненное действие, практический результат и выявленные замечания. Ошибки устраняются с повторной проверкой.

Оценивание опирается на фактически выполненные действия. Конфиденциальные данные заменяются разрешёнными учебными данными; сведения об авторстве и лицензиях

сохраняются. Для обучающихся с инвалидностью и ограниченными возможностями здоровья предоставляются необходимые специальные условия и доступные материалы с сохранением проверяемых результатов.

4. Практические задания

4.1. Исходные условия и материалы практических заданий

Работы выполняются в рамках связанного проекта. Ниже приведены требования и исходные данные, необходимые для создания учебного прототипа и его проверки. Обучающийся сохраняет исходный проект, промежуточные результаты и протоколы. При использовании проекта профильной организации исходные данные и ограничения фиксируются в индивидуальном задании; проверяемые действия сохраняются.

ПМ.01. Программные модули игрового прототипа

Создайте минимальный прототип с инвентарём вместимостью пять единиц. Предметы: ключ (1), аптечка (2), монета (1). Число в скобках — занимаемый объём. При добавлении проверяются свободное место и корректность количества. Удаление отсутствующего предмета возвращает отказ без изменения состояния.

Контрольный сценарий: пустой инвентарь → добавить ключ и аптечку (занято 3) → добавить три монеты (отказ, состояние не изменилось) → добавить две монеты (занято 5) → удалить ключ (занято 4). Добавление нуля, отрицательного количества и неизвестного предмета должно завершаться контролируемым отказом.

Заложенный дефект для отладки: обработчик получения предмета подписывается на событие при каждом открытии окна, но не отписывается при закрытии. Объясните повторное добавление после трёх открытий окна; реализуйте симметричную подписку и отписку и регрессионную проверку.

Сохранение включает идентификатор предмета и количество. Загрузка некорректного файла не должна уничтожать последнюю корректную копию. Представьте репозиторий, интерфейс модуля, тесты, историю объединения двух веток и инструкцию установки.

На учебной практике действия выполняются с предусмотренными консультациями и демонстрацией каждого этапа. На производственной практике обучающийся самостоятельно планирует работу, согласует изменения, обосновывает решения и подтверждает результат на рабочем месте. Заимствованные ресурсы сопровождаются сведениями об авторстве и лицензии.

4.2. Задания и показатели выполнения

Задание 1. Алгоритм обработки инвентаря

Проверяемый результат: умение формировать алгоритмы разработки программных модулей в соответствии с техническим заданием.

Задание. Спроектируйте добавление предмета в инвентарь с ограничением вместимости и объединением одинаковых предметов. Определите входы, выходы и исключительные ситуации.

Представляемые материалы. Блок-схема или псевдокод; трассировка для пустого, заполненного и частично заполненного инвентаря.

Показатели правильного выполнения. Предмет не теряется и не дублируется; отказ при переполнении предсказуем; все ветви алгоритма проверены.

Порядок выполнения: проанализировать условие; подготовить проект и исходные данные;

выполнить действия; проверить результат по контрольным сценариям; сохранить исходники и доказательства проверок; пояснить принятые решения.

Показатель	Максимальный балл
Соответствие результата условию и полнота реализации	4
Корректность проверки и воспроизводимость результата	3
Качество исходных материалов и документации	2
Обоснование решений при защите	1
Всего	10

Ответ для проверяющего. Предмет не теряется и не дублируется; отказ при переполнении предсказуем; все ветви алгоритма проверены. Оценка подтверждается исходным проектом, демонстрацией выполнения сценариев и протоколом проверок. Скриншот без работоспособного результата не подтверждает выполнение задания.

Задание 2. Программный модуль инвентаря

Проверяемый результат: умение разрабатывать программные модули в соответствии с техническим заданием.

Задание. Реализуйте спроектированный алгоритм в отдельном модуле. Отделите модель данных от отображения. Добавьте проверки входных данных.

Представляемые материалы. Исходный код, инструкция запуска, автоматические тесты.

Показатели правильного выполнения. Корректные операции добавления и удаления; неверные данные не приводят к аварийному завершению; результат соответствует техническому заданию.

Порядок выполнения: проанализировать условие; подготовить проект и исходные данные; выполнить действия; проверить результат по контрольным сценариям; сохранить исходники и доказательства проверок; пояснить принятые решения.

Показатель	Максимальный балл
Соответствие результата условию и полнота реализации	4
Корректность проверки и воспроизводимость результата	3
Качество исходных материалов и документации	2
Обоснование решений при защите	1
Всего	10

Ответ для проверяющего. Корректные операции добавления и удаления; неверные данные не приводят к аварийному завершению; результат соответствует техническому заданию. Оценка подтверждается исходным проектом, демонстрацией выполнения сценариев и протоколом проверок. Скриншот без работоспособного результата не подтверждает выполнение задания.

Задание 3. Оформление и документирование кода

Проверяемый результат: умение оформлять программный код в соответствии с установленными требованиями.

Задание. Примените единый стиль именования и форматирования. Документируйте публичный интерфейс и нетривиальные ограничения.

Представляемые материалы. Оформленный репозиторий, описание интерфейса и пример использования.

Показатели правильного выполнения. Имена отражают назначение; нет неиспользуемого кода; документация позволяет вызвать модуль без чтения его реализации.

Порядок выполнения: проанализировать условие; подготовить проект и исходные данные; выполнить действия; проверить результат по контрольным сценариям; сохранить исходники и доказательства проверок; пояснить принятые решения.

Показатель	**Максимальный балл**
Соответствие результата условию и полнота реализации	4
Корректность проверки и воспроизводимость результата	3
Качество исходных материалов и документации	2
Обоснование решений при защите	1
Всего	10

Ответ для проверяющего. Имена отражают назначение; нет неиспользуемого кода; документация позволяет вызвать модуль без чтения его реализации. Оценка подтверждается исходным проектом, демонстрацией выполнения сценариев и протоколом проверок. Скриншот без работоспособного результата не подтверждает выполнение задания.

Задание 4. Совместная разработка

Проверяемый результат: умение использовать систему контроля версий программного кода для коллективной разработки программного кода.

Задание. Создайте ветку для новой функции, внесите изменения и объедините их через запрос на включение. Воспроизведите и разрешите конфликт.

Представляемые материалы. История изменений, две ветки, описание разрешённого конфликта.

Показатели правильного выполнения. Изменения прослеживаются; итоговая ветка собирается; конфликт разрешён без утраты требуемой функциональности.

Порядок выполнения: проанализировать условие; подготовить проект и исходные данные; выполнить действия; проверить результат по контрольным сценариям; сохранить исходники и доказательства проверок; пояснить принятые решения.

Показатель	**Максимальный балл**
Соответствие результата условию и полнота реализации	4
Корректность проверки и воспроизводимость результата	3
Качество исходных материалов и документации	2
Обоснование решений при защите	1
Всего	10

Ответ для проверяющего. Изменения прослеживаются; итоговая ветка собирается;

конфликт разрешён без утраты требуемой функциональности. Оценка подтверждается исходным проектом, демонстрацией выполнения сценариев и протоколом проверок. Скриншот без работоспособного результата не подтверждает выполнение задания.

Задание 5. Отладка дефектного модуля

Проверяемый результат: умение выполнять отладку программных модулей с использованием специализированных программных средств.

Задание. Найдите причины повторного начисления очков и обращения к отсутствующему объекту. Используйте точки останова и наблюдение переменных.

Представляемые материалы. Протокол воспроизведения, исправление, регрессионные тесты.

Показатели правильного выполнения. Указана первопричина каждого дефекта; исправление устраняет её; ранее работавшие сценарии сохраняются.

Порядок выполнения: проанализировать условие; подготовить проект и исходные данные; выполнить действия; проверить результат по контрольным сценариям; сохранить исходники и доказательства проверок; пояснить принятые решения.

Показатель	**Максимальный балл**
Соответствие результата условию и полнота реализации	4
Корректность проверки и воспроизводимость результата	3
Качество исходных материалов и документации	2
Обоснование решений при защите	1
Всего	10

Ответ для проверяющего. Указана первопричина каждого дефекта; исправление устраняет её; ранее работавшие сценарии сохраняются. Оценка подтверждается исходным проектом, демонстрацией выполнения сценариев и протоколом проверок. Скриншот без работоспособного результата не подтверждает выполнение задания.

Задание 6. Рефакторинг и оптимизация

Проверяемый результат: умение осуществлять рефакторинг и оптимизацию программного кода.

Задание. Устраните дублирование в обработчиках игровых событий. Измерьте время выполнения до и после изменения на одинаковом наборе данных.

Представляемые материалы. Сравнение версий, результаты измерений, тесты эквивалентности.

Показатели правильного выполнения. Поведение сохранено; сравнение воспроизводимо; улучшение подтверждено измерениями, а не субъективной оценкой.

Порядок выполнения: проанализировать условие; подготовить проект и исходные данные; выполнить действия; проверить результат по контрольным сценариям; сохранить исходники и доказательства проверок; пояснить принятые решения.

Показатель	**Максимальный балл**

Показатель	Максимальный балл
Соответствие результата условию и полнота реализации	4
Корректность проверки и воспроизводимость результата	3
Качество исходных материалов и документации	2
Обоснование решений при защите	1
Всего	10

Ответ для проверяющего. Поведение сохранено; сравнение воспроизводимо; улучшение подтверждено измерениями, а не субъективной оценкой. Оценка подтверждается исходным проектом, демонстрацией выполнения сценариев и протоколом проверок. Скриншот без работоспособного результата не подтверждает выполнение задания.

Задание 7. Проектирование интеграции

Проверяемый результат: умение разрабатывать процедуры интеграции программных модулей и платформы/фреймворка.

Задание. Опишите контракт обмена данными между модулем инвентаря и игровым движком: события, структуры данных, жизненный цикл и ошибки.

Представляемые материалы. Спецификация интерфейса, схема взаимодействия, план интеграционных тестов.

Показатели правильного выполнения. Для каждого события определены отправитель и получатель; учтены повторные вызовы, отсутствие данных и освобождение ресурсов.

Порядок выполнения: проанализировать условие; подготовить проект и исходные данные; выполнить действия; проверить результат по контрольным сценариям; сохранить исходники и доказательства проверок; пояснить принятые решения.

Показатель	Максимальный балл
Соответствие результата условию и полнота реализации	4
Корректность проверки и воспроизводимость результата	3
Качество исходных материалов и документации	2
Обоснование решений при защите	1
Всего	10

Ответ для проверяющего. Для каждого события определены отправитель и получатель; учтены повторные вызовы, отсутствие данных и освобождение ресурсов. Оценка подтверждается исходным проектом, демонстрацией выполнения сценариев и протоколом проверок. Скриншот без работоспособного результата не подтверждает выполнение задания.

Задание 8. Интеграция с движком

Проверяемый результат: умение выполнять интеграцию программных модулей и платформы/фреймворка.

Задание. Подключите модуль к сцене. Реализуйте получение предмета, отображение остатка и сохранение состояния между запусками.

Представляемые материалы. Рабочая сцена, конфигурация, протокол интеграционных проверок.

Показатели правильного выполнения. Повторная загрузка восстанавливает состояние;

события обрабатываются однократно; модуль не зависит от конкретного элемента интерфейса.

Порядок выполнения: проанализировать условие; подготовить проект и исходные данные; выполнить действия; проверить результат по контрольным сценариям; сохранить исходники и доказательства проверок; пояснить принятые решения.

Показатель	Максимальный балл
Соответствие результата условию и полнота реализации	4
Корректность проверки и воспроизводимость результата	3
Качество исходных материалов и документации	2
Обоснование решений при защите	1
Всего	10

Ответ для проверяющего. Повторная загрузка восстанавливает состояние; события обрабатываются однократно; модуль не зависит от конкретного элемента интерфейса. Оценка подтверждается исходным проектом, демонстрацией выполнения сценариев и протоколом проверок. Скриншот без работоспособного результата не подтверждает выполнение задания.

Задание 9. Установка и сопровождение

Проверяемый результат: умение осуществлять инсталляцию, настройку и обслуживание программного обеспечения.

Задание. Подготовьте развёртывание проекта на чистом рабочем месте: зависимости, настройки и восстановление после неудачного обновления.

Представляемые материалы. Пакет установки, перечень зависимостей, журнал проверки и инструкция отката.

Показатели правильного выполнения. Установка воспроизводима по инструкции; версии зависимостей указаны; откат возвращает работоспособное состояние.

Порядок выполнения: проанализировать условие; подготовить проект и исходные данные; выполнить действия; проверить результат по контрольным сценариям; сохранить исходники и доказательства проверок; пояснить принятые решения.

Показатель	Максимальный балл
Соответствие результата условию и полнота реализации	4
Корректность проверки и воспроизводимость результата	3
Качество исходных материалов и документации	2
Обоснование решений при защите	1
Всего	10

Ответ для проверяющего. Установка воспроизводима по инструкции; версии зависимостей указаны; откат возвращает работоспособное состояние. Оценка подтверждается исходным проектом, демонстрацией выполнения сценариев и протоколом проверок. Скриншот без работоспособного результата не подтверждает выполнение задания.

5. Промежуточная аттестация и шкала оценивания

Для дифференцированного зачёта обучающийся представляет индивидуальное задание, дневник, отчёт, исходные материалы, работающий результат и протоколы проверок. Руководитель оформляет характеристику и аттестационный лист. При защите проверяются личный вклад, правильность решений, воспроизводимость результата и ответы на вопросы.

Каждое практическое задание оценивается максимум в 10 баллов по приведённым показателям. Итоговый процент равен сумме полученных баллов, делённой на максимально возможную сумму за обязательные задания соответствующего периода и умноженной на 100. Перечень обязательных заданий и их связь с компетенциями фиксируются в индивидуальном задании до начала работ.

Выполнение заданий, %	Оценка
90–100%	Отлично
75–менее 90%	Хорошо
60–менее 75%	Удовлетворительно
Менее 60%	Неудовлетворительно

Результат	Оценка

Для положительной оценки должны быть представлены обязательные работы и подтверждены основные действия по каждому проверяемому результату. Если обязательное действие не выполнено или результат не представлен, соответствующая компетенция не подтверждается независимо от общей суммы. Нарушение безопасности требует остановки работы и устранения опасности; невыполненные действия оцениваются по фактически достигнутому результату.

Результаты фиксируются в аттестационном листе и ведомости. Порядок повторного прохождения промежуточной аттестации определяется установленной процедурой ИНСТИТУТ НАПРАВО.РУ. При комплексной аттестации по каждому входящему компоненту отдельно фиксируются результаты; часы не суммируются повторно.

6. Вопросы для защиты результатов практики

Вопросы используются совместно с демонстрацией выполненных работ и анализом отчётных материалов. Ответ связывается с собственной реализацией, исходными данными и проверками. Приведённые пояснения являются ориентирами для проверяющего.

1. Что такое алгоритм A*? Алгоритм A* — это алгоритм поиска пути, который находит кратчайший путь от одной точки до другой, учитывая стоимость перемещения между узлами.

2. Как работают алгоритмы генерации уровней в играх? Алгоритмы генерации уровней создают уникальные игровые карты на основе заданных параметров и случайных элементов.

3. Что такое проверка столкновений? Проверка столкновений — это процесс определения, пересекаются ли два или более объекта в игровом мире.

4. Как алгоритм A* используется в играх? Алгоритм A* в играх используется для управления движением NPC (неигровых персонажей) и позволяет им находить оптимальные маршруты, избегая препятствий.

5. Какими методами реализуется проверка столкновений? Приведите хотя бы один пример. Проверка столкновений может реализовываться с помощью различных методов, таких

как Bounding Box или Bounding Sphere, которые помогают быстро определить, нужно ли проводить детальную проверку.

6. Как алгоритмы искусственного интеллекта (AI) улучшают игровой процесс? Алгоритмы AI позволяют NPC принимать решения на основе действий игрока и состояния игры. Это делает игровой процесс более динамичным и реалистичным, так как NPC могут адаптироваться к изменениям в окружении и реагировать на действия игрока.

7. Что такое процедурная генерация контента? Процедурная генерация контента — это метод создания игровых элементов (уровней, объектов, текстур) с использованием алгоритмов и случайных чисел.

8. Каково основное преимущество процедурной генерации контента? Процедурная генерация контента позволяет разработчикам создавать уникальный контент без необходимости ручного проектирования каждого элемента.

9. Как работает алгоритм Дейкстры? Алгоритм Дейкстры используется для нахождения кратчайшего пути в графе с неотрицательными весами. Он последовательно выбирает узел с минимальной стоимостью и обновляет расстояния до соседних узлов, пока не будет найден кратчайший путь до целевой точки.

10. Что такое система управления состоянием (State Machine)? Система управления состоянием — это структура, которая позволяет объектам (персонажам) изменять свое поведение в зависимости от текущего состояния.

11. Что такое программный модуль в контексте разработки игр? Программный модуль — это самостоятельная часть кода, отвечающая за определенную функциональность в игре, такую как управление персонажем или обработка столкновений.

12. Вопрос: Что такое юнит-тестирование? Юнит-тестирование — это метод тестирования, при котором отдельные модули кода проверяются на корректность выполнения своих функций.

13. Какова цель прототипирования в разработке игр? Прототипирование помогает визуализировать и тестировать идеи и механики игры до начала полноценной разработки.

14. Каковы основные принципы объектно-ориентированного программирования (ООП)? Инкапсуляция, наследование, полиморфизм.

15. Каковы преимущества использования системы контроля версий в разработке игр? Система контроля версий позволяет отслеживать изменения в коде, управлять совместной работой команды и восстанавливать предыдущие версии проекта.

16. Что такое Agile-методология? Agile-методология — это подход к разработке, основанный на итеративном процессе, который позволяет гибко реагировать на изменения и улучшать продукт на каждом этапе.

17. Что проверяет интеграционное тестирование? Интеграционное тестирование проверяет взаимодействие между различными модулями или компонентами системы для выявления проблем на стыках.

18. Каковы основные метрики для оценки качества программного обеспечения? Надежность, удобство использования, производительность.

19. Что такое геймдизайн-документ (GDD)? GDD — это документ, который описывает концепцию игры, механики, персонажей и другие ключевые аспекты разработки.

20. Что такое комментарии в C? Комментарии — это текст в коде, который игнорируется компилятором и используется для пояснения кода.

21. Как обозначаются однострочные комментарии в C? Однострочные комментарии начинаются с символов //.

22. Как правильно объявить переменную типа int? Объявление переменной типа int выглядит так: int number;.

23. Что такое функция в языке C? Функция — это блок кода, который выполняет определенную задачу и может быть вызван из других частей программы.

24. Какой символ используется для многострочного комментария? Многострочные комментарии начинаются с /* и заканчиваются */.

25. Что делает директива #include? Директива #include подключает внешние библиотеки или заголовочные файлы к программе.

26. Что возвращает функция main в конце выполнения программы? Функция main обычно возвращает 0, что указывает на успешное завершение программы.

27. Что означает тип возвращаемого значения void? Тип void указывает, что функция не возвращает никакого значения.

28. Как правильно сравнить два значения в C? Для сравнения используется оператор ==.

29. Что делает функция scanf()? Функция scanf() считывает данные с клавиатуры и сохраняет их в переменные.

30. Что такое система контроля версий (VCS)? Это инструмент, который отслеживает изменения в исходном коде и управляет ими.

31. Какое основное преимущество использования VCS? Позволяет командам разработчиков работать параллельно, предотвращая конфликты изменений.

32. Что такое репозиторий в контексте VCS? Репозиторий — это хранилище, где сохраняется весь код и история изменений.

33. Приведите пример системы контроля версий? Git.

34. Что такое коммит в Git? Коммит — это сохранение изменений в репозитории с добавлением описания этих изменений.

35. Что такое ветка (branch) в Git? Ветка — это независимая линия разработки, позволяющая работать над функциями или исправлениями без влияния на основную кодовую базу.

36. Как объединить (слить) изменения из одной ветки в другую в Git? С помощью команды git merge <имя_ветки>.

37. Что такое конфликт при слиянии (merge conflict)? Конфликт возникает, когда изменения в разных ветках затрагивают одни и те же строки кода, и Git не может автоматически определить, какие изменения оставить.

38. Что такое тег (tag) в Git? Тег — это метка на определенном коммите, обычно используемая для обозначения релизов.

39. Как посмотреть историю коммитов в Git? Используя команду git log.

40. Что такое "интеграционное тестирование"? Интеграционное тестирование — это процесс проверки взаимодействия между различными модулями или компонентами системы, чтобы убедиться, что они работают вместе корректно.

41. Что такое комплект средств разработки (SDK)? SDK (Software Development Kit) — это набор инструментов и библиотек, который предоставляет разработчикам необходимые

средства для создания приложений на определенной платформе.

42. Какой из методов интеграции позволяет выявить проблемы на ранних этапах разработки? Метод непрерывной интеграции (CI) позволяет автоматически собирать и тестировать код при каждом изменении, что помогает быстро выявлять и исправлять ошибки.

43. Что подразумевает термин "системное тестирование"? Системное тестирование — это процесс проверки всей системы как единого целого, чтобы убедиться, что все компоненты работают совместно в соответствии с требованиями.

44. Какова основная функция дебаггера? Дебаггер позволяет разработчикам отслеживать выполнение кода, находить и исправлять ошибки, а также анализировать состояние программы в реальном времени.

45. Какова роль пользовательского тестирования в процессе разработки игр? Пользовательское тестирование позволяет разработчикам получать обратную связь от реальных пользователей, что помогает выявить проблемы с интерфейсом и игровым процессом до релиза продукта.

46. Что такое параллельная разработка в контексте интеграции? Параллельная разработка подразумевает одновременное создание нескольких модулей или компонентов, что может ускорить процесс разработки, но требует тщательной координации.

47. Какую роль играют пользовательские сценарии в разработке игр? Пользовательские сценарии помогают разработчикам понять, как игроки будут взаимодействовать с игрой, что позволяет улучшить дизайн и функциональность приложения.

48. Что такое версионирование API? Версионирование API — это практика управления изменениями в API, позволяющая поддерживать совместимость с предыдущими версиями при добавлении новых функций.

49. Каковы преимущества использования облачных сервисов для разработки игр? Облачные сервисы обеспечивают гибкость, масштабируемость и доступность ресурсов, что позволяет разработчикам сосредоточиться на создании контента без необходимости управления физической инфраструктурой.

6.1. Дополнительные вопросы по выполненным работам

ПК 1.1. Какие решения приняты при выполнении работы «Алгоритм обработки инвентаря», чем подтверждено качество результата и как воспроизвести его проверку? Ориентир: Предмет не теряется и не дублируется; отказ при переполнении предсказуем; все ветви алгоритма проверены.

ПК 1.2. Какие решения приняты при выполнении работы «Программный модуль инвентаря», чем подтверждено качество результата и как воспроизвести его проверку? Ориентир: Корректные операции добавления и удаления; неверные данные не приводят к аварийному завершению; результат соответствует техническому заданию.

ПК 1.3. Какие решения приняты при выполнении работы «Оформление и документирование кода», чем подтверждено качество результата и как воспроизвести его проверку? Ориентир: Имена отражают назначение; нет неиспользуемого кода; документация позволяет вызвать модуль без чтения его реализации.

ПК 1.4. Какие решения приняты при выполнении работы «Совместная разработка», чем подтверждено качество результата и как воспроизвести его проверку? Ориентир: Изменения прослеживаются; итоговая ветка собирается; конфликт разрешён без утраты

требуемой функциональности.

ПК 1.5. Какие решения приняты при выполнении работы «Отладка дефектного модуля», чем подтверждено качество результата и как воспроизвести его проверку? Ориентир: Указана первопричина каждого дефекта; исправление устраняет её; ранее работавшие сценарии сохраняются.

ПК 1.6. Какие решения приняты при выполнении работы «Рефакторинг и оптимизация», чем подтверждено качество результата и как воспроизвести его проверку? Ориентир: Поведение сохранено; сравнение воспроизводимо; улучшение подтверждено измерениями, а не субъективной оценкой.

ПК 1.7. Какие решения приняты при выполнении работы «Проектирование интеграции», чем подтверждено качество результата и как воспроизвести его проверку? Ориентир: Для каждого события определены отправитель и получатель; учтены повторные вызовы, отсутствие данных и освобождение ресурсов.

ПК 1.8. Какие решения приняты при выполнении работы «Интеграция с движком», чем подтверждено качество результата и как воспроизвести его проверку? Ориентир: Повторная загрузка восстанавливает состояние; события обрабатываются однократно; модуль не зависит от конкретного элемента интерфейса.

ПК 1.9. Какие решения приняты при выполнении работы «Установка и сопровождение», чем подтверждено качество результата и как воспроизвести его проверку? Ориентир: Установка воспроизводима по инструкции; версии зависимостей указаны; откат возвращает работоспособное состояние.

Приложение 1. Индивидуальное задание

ИНСТИТУТ НАПРАВО.РУ. Обучающийся: _____. Группа: _____. Вид и индекс практики: _____. Место проведения: _____. Период и объём: _____.

Тема проекта: _____. Исходные данные и целевая платформа: _____. Виды работ и проверяемые компетенции: _____. Требования к результату и контрольные сроки: _____.

Руководитель практики: _____. Руководитель от профильной организации (при проведении практики в профильной организации): _____. Дата выдачи и подписи: _____.

Приложение 2. Дневник практики

Дата	Выполненная работа	Полученный результат	Подпись

Приложение 3. Отчёт и протокол проверки

Структура отчёта: титульный лист; индивидуальное задание; содержание; характеристика рабочего места и применённых средств; описание выполненных работ и личного вклада; результаты испытаний и исправления дефектов; выводы; список использованных источников; приложения. Материалы отражают фактически выполненную работу.

Проверяемое	Действия / условия	Ожидаемый результат	Фактический результат

Приложение 4. Аттестационный лист

Обучающийся: _____. Практика и период: _____. Руководитель: _____.

Компетенция	Выполненная работа и подтверждение	Баллы / заключение
ПК 1.1	Алгоритм обработки инвентаря	
ПК 1.2	Программный модуль инвентаря	
ПК 1.3	Оформление и документирование кода	
ПК 1.4	Совместная разработка	
ПК 1.5	Отладка дефектного модуля	
ПК 1.6	Рефакторинг и оптимизация	
ПК 1.7	Проектирование интеграции	
ПК 1.8	Интеграция с движком	
ПК 1.9	Установка и сопровождение	

Общие компетенции: применение способов решения задачи; поиск и оценка информации; организация собственной деятельности; взаимодействие; устная и письменная коммуникация; ответственное и безопасное выполнение работ; использование документации. Наблюдения и примеры: _____.

Характеристика самостоятельности, качества работы и соблюдения безопасности: _____. Итоговый результат: _____. Дата и подписи руководителей: _____. Печать профильной организации — при наличии.