Министерство образования Самарской области

АССОЦИАЦИЯ ПРОФЕССИОНАЛЬНАЯ ОБРАЗОВАТЕЛЬНАЯ ОРГАНИЗАЦИЯ "ИНСТИТУТ НАПРАВО.РУ"

УТВЕРЖДЕНО
Приказ директора
от 07.08.2024 № 012/2/с

Воробьев К.А.

УЧЕБНО-МЕТОДИЧЕСКИЕ УКАЗАНИЯ

ПМ.06 Разработка компьютерных игр и мультимедийных приложений

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. Назначение и организация практикума 3

2. Состав работ и представление результатов 3

3. Практические работы 4

Работа 1. Игровая петля и критерий завершения 4

Работа 2. Движение и столкновения 4

Работа 3. Простое поведение NPC 5

Работа 4. Баланс и учебные данные прохождений 6

Работа 5. События и сетевые ограничения 6

Работа 6. Сборка, поддержка и выпуск 7

Развёрнутая лабораторная работа. Вертикальный срез игры с ключами и таймером 7

4. Комплексное учебное задание 9

5. Термины и понятия 9

6. Исходные данные и учебные примеры 9

7. Использованные источники 10

1. Назначение и организация практикума

Полный цикл учебного игрового продукта: правила, сцена, механики, баланс, поддержка и проверка результата.

Указания предназначены для обучающихся по специальности 09.02.10 «Разработка компьютерных игр, дополненной и виртуальной реальности», квалификация — разработчик компьютерных игр, дополненной и виртуальной реальности.

Компонент	Учебная нагрузка
Индекс и наименование	ПМ.06 Разработка компьютерных игр и мультимедийных приложений
Общий объём	742 академических часов
Практические занятия	254 академических часов
Промежуточная аттестация	Экзамен — семестр 6, 8; Дифференцированный зачёт — семестр 8

Перед работой изучаются её исходные данные и проверяемые результаты. Затем выполняются последовательные действия, сопоставляется учебный пример и решаются самостоятельные варианты. Ответ включает ход работы и подтверждение результата. Использование готового примера без выполнения своего варианта не подтверждает освоение действия.

Подготовленные материалы: Проект и исполняемая сборка; Паспорт механик и баланс; Тесты, журнал версий и инструкция.

2. Состав работ и представление результатов

Работа	Тема и результат
1	Игровая петля и критерий завершения. Состояние не остаётся одновременно Won и Lost.
2	Движение и столкновения. A: за 2 с путь 5.
3	Простое поведение NPC. Нет обращения к отсутствующей цели.
4	Баланс и учебные данные прохождений. Сохраняются сырые результаты.
5	События и сетевые ограничения. Порядок не меняет сумму независимых начислений.
6	Сборка, поддержка и выпуск. Поставка не зависит от рабочего пути автора.

Профессиональные результаты модуля

Код	Профессиональная компетенция
ПК 6.1	Использовать популярные платформы для сборки, настройки и развертывания контента.
ПК 6.2	Разрабатывать решения на основании игрового движка.
ПК 6.3	Разрабатывать механику игрового процесса.
ПК 6.4	Программировать игровую графику и специальные эффекты.
ПК 6.5	Разрабатывать системы игрового баланса.
ПК 6.6	Администрировать процесс разработки игровых продуктов.

В отчёте указываются тема, вариант, исходные данные, ход выполнения, проверка результата и вывод. Таблицы и графики снабжаются названиями и единицами измерения, где они применимы. Результаты оцениваются по критериям фонда оценочных средств. Приложите игровой проект и протокол проверки игровых сценариев.

3. Практические работы

Работа 1. Игровая петля и критерий завершения

Краткие теоретические сведения

Игровая петля связывает действие, обратную связь и следующую цель. Правила победы, поражения и перезапуска должны исключать неопределённое состояние.

Исходные данные

Игрок за 60 секунд собирает 5 ключей и открывает выход; после истечения времени — поражение.

Порядок выполнения

1. Описать основной цикл.

2. Задать состояния Menu, Playing, Won, Lost.

3. Определить одновременные события последнего кадра.

4. Подготовить проверяемые критерии завершения.

Разбор учебного примера

Приоритет спорного события задаётся заранее: например, выход до окончания таймера означает победу, при истечении времени новые действия не принимаются. Реализация должна следовать выбранному правилу.

Задания для самостоятельного выполнения

А: требуется 3 ключа за 30 секунд.

Б: перезапуск после проигрыша.

Проверка результата

— Состояние не остаётся одновременно Won и Lost.

— Перезапуск сбрасывает все нужные данные.

Вопросы для самоконтроля

1. Что делает цель проверяемой?

2. Почему нужен порядок обработки одновременных событий?

Работа 2. Движение и столкновения

Краткие теоретические сведения

Изменение позиции связывают со временем, если требуется независимость от частоты кадров. Столкновения и визуальное перемещение проверяют отдельно.

Исходные данные

Скорость 4 единицы/с, прямое движение 2 с.

Порядок выполнения

1. Записать модель перемещения.

2. Проверить результаты при разных шагах времени.

3. Добавить препятствие и правило остановки.

4. Отделить изменение состояния от отображения.

Разбор учебного примера

Без препятствий путь равен 8 единицам и при 30, и при 60 шагах в секунду. Умножение на число кадров без учёта времени нарушает требование.

Задания для самостоятельного выполнения

А: скорость 2,5 единицы/с.

Б: движение прекращено через 1,2 с.

Проверка результата

— А: за 2 с путь 5.

— Б: при исходной скорости путь 4,8 до учёта препятствий.

Вопросы для самоконтроля

1. Когда временной шаг влияет на точность?

2. Почему проверяют столкновение на большой скорости?

Работа 3. Простое поведение NPC

Краткие теоретические сведения

Конечный автомат разделяет состояние, условие перехода и выполняемое действие. Гистерезис или согласованные пороги уменьшают дрожание на границе.

Исходные данные

NPC патрулирует; замечает игрока ближе 5 единиц, прекращает преследование дальше 7.

Порядок выполнения

1. Построить автомат Patrol→Chase→Return.

2. Задать пороги и порядок проверки.

3. Проиграть расстояния 8,4,6,8.

4. Проверить возврат после исчезновения игрока.

Разбор учебного примера

Последовательность состояний: Patrol, Chase, Chase, Return. При расстоянии 6 преследование сохраняется, так как порог выхода 7 отличается от входа 5.

Задания для самостоятельного выполнения

А: одинаковые пороги 5 и 5.

Б: цель удалена во время Chase.

Проверка результата

— Нет обращения к отсутствующей цели.

— Переходы соответствуют таблице.

Вопросы для самоконтроля

1. Зачем разные пороги входа и выхода?

2. Что должно происходить при отсутствии цели?

Работа 4. Баланс и учебные данные прохождений

Краткие теоретические сведения

Баланс исследуют по заданной цели и нескольким показателям. Изменение всех параметров одновременно не позволяет понять причину результата.

Исходные данные

Пять времён прохождения: 40,45,50,55,60 с; целевой ориентир — около 50 с.

Порядок выполнения

1. Вычислить среднее и диапазон.

2. Выбрать один изменяемый параметр.

3. Подготовить план повторного прохождения.

4. Записать ограничения малой учебной выборки.

Разбор учебного примера

Среднее 50 с, диапазон 20 с. Это не доказывает одинаковую сложность для всех пользователей; дополнительно нужны ошибки и доля завершений.

Задания для самостоятельного выполнения

А: уменьшить время уровня на 10 с.

Б: изменить число обязательных предметов.

Проверка результата

— Сохраняются сырые результаты.

— Вывод связан с заранее выбранной метрикой.

Вопросы для самоконтроля

1. Почему одной средней недостаточно?

2. Как отличить эффект обучения от эффекта изменения?

Работа 5. События и сетевые ограничения

Краткие теоретические сведения

Обмен сообщениями требует идентификатора события, порядка и поведения при повторе. Учебная имитация позволяет проверить правила до настройки реальной сети.

Исходные данные

Сообщения начисления: event1,+10; event2,+5; повтор event1,+10.

Порядок выполнения

1. Определить уникальный идентификатор.

2. Реализовать защиту от повторной обработки.

3. Проиграть изменение порядка.

4. Проверить неизвестное и повреждённое сообщение.

Разбор учебного примера

При повторе event1 итог остаётся 15, а не 25. Отсутствие дубликата в одном запуске не заменяет специальный отрицательный тест.

Задания для самостоятельного выполнения

А: сообщения пришли event2,event1.

Б: поле количества отсутствует.

Проверка результата

— Порядок не меняет сумму независимых начислений.

— Повреждённое сообщение не изменяет состояние.

Вопросы для самоконтроля

1. Чем повтор доставки отличается от нового события?

2. Какие операции зависят от порядка?

Работа 6. Сборка, поддержка и выпуск

Краткие теоретические сведения

Выпуск — проверенная версия продукта с ресурсами и инструкцией, а не папка исходников. Проверяют запуск в чистом каталоге, ограничения и возможность повторить сборку.

Исходные данные

Нужно передать игру с меню, сохранением, звуком и одним уровнем.

Порядок выполнения

1. Составить перечень обязательных файлов.

2. Подготовить сборку с номером версии.

3. Выполнить проверки по инструкции другим участником.

4. Оформить известные ограничения и порядок сообщения об ошибке.

Разбор учебного примера

Отсутствующая библиотека или ресурс выявляются до передачи. Известное ограничение описывается конкретным сценарием, а не фразой «возможны ошибки».

Задания для самостоятельного выполнения

А: отсутствует звуковой файл.

Б: сборка запущена в каталоге без права записи.

Проверка результата

— Поставка не зависит от рабочего пути автора.

— Сохранение и отказ записи проверены отдельно.

Вопросы для самоконтроля

1. Почему исходный проект и сборка различаются?

2. Какие сведения нужны службе поддержки?

Развёрнутая лабораторная работа. Вертикальный срез игры с ключами и таймером

Средства выполнения

Игровой движок Unity 2022.3 либо реализация эквивалентных объектов в предусмотренном рабочей программой движке. Для независимой проверки правил дан

файл game_rules.py.

Задание

Собрать один завершённый уровень: игрок перемещается, собирает пять ключей, открывает выход за 60 секунд, получает однозначный результат и может перезапустить игру.

Последовательность выполнения

1. До создания сцены выполните python game_rules.py. Разберите проверку победы и защиты от повторного начисления. Модель правил не заменяет проверки сцены, но задаёт ожидаемое поведение.

2. Создайте сцену Main и включите её в перечень сцен сборки. Используйте плоскость пола, стены из примитивов, игрока, пять ключей и выход; не загружайте ассеты неизвестного происхождения.

3. Задайте единицы, стартовую позицию и скорость 4 единицы/с. Сначала проверьте свободное движение на двухсекундном интервале; после этого добавьте столкновения и убедитесь, что стены не проходят насквозь.

4. Для каждого ключа задайте уникальное состояние «собран». После успешного контакта объект скрывается или удаляется и не может повторно увеличить счётчик. Проверяйте быстрые повторные входы в зону контакта.

5. Реализуйте состояния Playing, Won и Lost. Таймер уменьшается по времени; при завершении новые игровые действия не принимаются. Заранее определите приоритет событий на границе окончания таймера.

6. Добавьте экран текущего количества ключей, оставшегося времени и результата. Он должен отражать состояние, а не независимо хранить собственный счётчик, который может разойтись с логикой.

7. Проверьте выход при четырёх и пяти ключах. В первом случае игра продолжается с понятной обратной связью, во втором — завершается победой при соблюдении времени.

8. Реализуйте перезапуск. После него число ключей равно нулю, таймер 60, все ключи снова доступны, результат предыдущего запуска не сохраняется в текущем состоянии.

9. Проведите 15 проверок: меню, движение, стены, каждый ключ, повторный контакт, выход до и после набора, истечение времени, завершение, перезапуск и запуск сборки из отдельного каталога.

10. Передайте вертикальный срез с исходниками, ресурсами, инструкцией запуска, журналом версии и известными ограничениями. После проверки другим участником исправьте выявленные несоответствия и повторите связанные сценарии.

Контрольные сценарии и ожидаемые результаты

Сценарий	Ожидаемый результат
Начало игры	0 ключей, 60 секунд, Playing
Один ключ дважды	Счётчик увеличен только один раз
Выход при 4 ключах	Нет победы
Выход при 5 ключах до окончания времени	Won
Время закончилось	Lost
Перезапуск	Исходные данные уровня восстановлены
Повтор event1	Очки за событие не начислены повторно

Усложнённый вариант

Добавьте NPC с порогами преследования 5 и 7 единиц либо второй уровень сложности. Изменение должно сопровождаться новыми проверками, сохранением первоначальной версии и объяснением влияния на баланс.

4. Комплексное учебное задание

Подготовьте завершённый учебный игровой прототип: основной цикл, движение, инвентарь, NPC, сохранение, звук и интерфейс. Предъявите минимум 15 проверок, журнал двух версий, анализ баланса, реестр ресурсов и воспроизводимую сборку с инструкцией.

Этапы и предъявляемые материалы

Этап	Результат
1	Проект и исполняемая сборка
2	Паспорт механик и баланс
3	Тесты, журнал версий и инструкция

Выполненная работа представляется с исходными данными, объяснением принятых решений и проверками. Изменения исходного условия согласуются до выполнения расчётов или реализации. При защите обучающийся воспроизводит один основной и один граничный либо отрицательный сценарий и объясняет полученный результат.

Контроль готовности

— Состояние не остаётся одновременно Won и Lost.

— A: за 2 с путь 5.

— Нет обращения к отсутствующей цели.

— Сохраняются сырые результаты.

— Порядок не меняет сумму независимых начислений.

— Поставка не зависит от рабочего пути автора.

5. Термины и понятия

Термин	Содержание
Игровая петля	Повторяемая связь действия, результата и следующей цели.
Состояние	Определённый режим поведения системы.
Баланс	Согласование параметров игры с заданными целями опыта.
Сборка	Подготовленная к запуску версия продукта и необходимых ресурсов.

6. Исходные данные и учебные примеры

Используйте учебный пример для проверки условий победы, поражения и перезапуска игры.

game_rules.py

```
class Game:
    def __init__(self):
        self.state, self.keys, self.remaining = "Playing", 0, 60.0
        self.score, self.events = 0, set()
```

```
    def tick(self, dt):
        if dt < 0:
            raise ValueError("negative delta time")
        if self.state == "Playing":
            self.remaining = max(0.0, self.remaining-dt)
            if self.remaining == 0:
                self.state = "Lost"

    def collect(self):
        if self.state == "Playing":
            self.keys += 1

    def exit(self):
        if self.state == "Playing" and self.keys >= 5:
            self.state = "Won"

    def award(self, event_id, points):
        if points < 0:
            raise ValueError("negative points")
        if event_id not in self.events:
            self.events.add(event_id)
            self.score += points

game = Game()
for _ in range(5): game.collect()
game.exit()
assert game.state == "Won"
game.tick(100)
assert game.state == "Won"
game.award("event1",10)
game.award("event2",5)
game.award("event1",10)
assert game.score == 15
print("Game rules checked")
```

Проверьте условия победы, поражения и перезапуска. Фактические результаты сопоставьте с ожидаемыми результатами по сценариям практической работы.

7. Использованные источники

1. Линовес, Д. Виртуальная реальность в Unity : практическое руководство / Д. Линовес ; пер. с англ. Р. Н. Рагимова. — 2-е изд.- Москва : ДМК Пресс, 2023. - 317 с. - ISBN 978-5-89818-578-7. - Текст : электронный. - URL: https://znanium.ru/catalog/product/.

2. Паласиос, Х. Unity 5.x. Программирование искусственного интеллекта в играх / Х. Паласиос ; пер. с англ. Р.Н. Рагимова. - Москва : ДМК Пресс, 2017. - 272 с. - ISBN 978-5-97060-436-6. - Текст : электронный. - URL: https://znanium.com/catalog/product/1028125.

3. Разработка 3D-игр в Unity : практическое руководство / Э. Дэвис, Т. Батист, Р. Крейг, Р. Станкел ; пер. с англ. П. М. Бомбаковой. – Москва : ДМК Пресс, 2023. - 299 с. – ISBN 978-5-93700-254-9. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2205049

4. Торн, А. Искусство создания сценариев в Unity. Практические советы и приемы создания игр профессионального уровня на C# в Unity : практическое руководство / А. Торн ; пер. с англ. Р. Н. Рагимова. - 2-е изд - Москва : ДМК Пресс, 2023. - 362 с. - ISBN 978-5-89818-343-1. - Текст : электронный. - URL: https://znanium.ru/catalog/product/2102630

5. Unity Technologies. Unity User Manual 2022.3. Creating and Using Scripts. URL: https://docs.unity3d.com/2022.3/Documentation/Manual/CreatingAndUsingScripts.html

6. Godot Engine. Documentation 4.2. Your first 2D game. URL: https://docs.godotengine.org/en/4.2/getting_started/first_2d_game/index.html