1.2.3. Расширение игровых механик: комбинирование модулей ПРИМС, внутренние события и победная точка

Блок: Блок 1.2. Программирование · **Полугодие:** 1 · **Класс:** 7 · **Продолжительность:** 40 мин

Тема: Комбинирование модулей: (движение, анимация, индикация, таймер), внутренние события и настройка победной точки · **Инструменты:** ПРИМС (Cyberiada Graph Editor), Мастерская игр «Урсула»

📋 Паспорт урока

🎯 Цель	Научить учащихся комбинировать в одном состоянии команды нескольких модулей (движение, анимация, индикация), освоить работу с внутренними событиями состояний для создания более сложных и выразительных игровых механик, а также создать диаграмму победной точки (флага) для завершения игрового уровня.
📚 Тип урока	Комбинированный урок с практической работой.
👥 Формы работы	Фронтальная беседа, демонстрация на проекторе, индивидуальная практическая работа.
🛠️ Оборудование	Сохраненные проекты локаций с прошлых уроков, редактор ПРИМС, «Урсула».

🎯 Планируемые результаты

Предметные

- Учащиеся умеют запускать в одном состоянии параллельные действия с участием ≥3 модулей

(запуск приводов, анимация, включение цветного маяка).

- Учащиеся понимают принцип работы внутреннего события (реакция без смены состояния) и настраивают остановку движения и запуск таймер внутри состояния.
- Учащиеся умеют создавать диаграмму победной точки с использованием МодуляПобедыИПоражения.

Метапредметные

Коммуникативные	• Умение описывать комплексное поведение персонажа понятными техническими шагами.
Регулятивные	• Пошаговая сборка и отладка сложной механики. • Сборка двух независимых диаграмм для одного проекта. • Декомпозиция задачи на модульные команды
Познавательные	• Понимание системности: игровой мир состоит из независимых объектов с разной логикой (моб патрулирует, флаг ждет клика для победы). • Анализ причинно-следственных цепочек между модулями.

📚 Ход урока

Этап 1. Введение и актуализация (3 мин)

🖥️ Слайд 1: Титульный

Иллюстрация: робот со светящимся маяком на маршруте и победный флаг.

Напоминание: «NPC уже ходит, помнит и выдает реплики. Но его поведение очень простое. Сегодня добавим анимацию, реакцию на мир — и еще настроим финальную точку игры».

📣 **Действия учителя:** Приветствие. Напоминание: на прошлых уроках NPC получил состояния

и переменные. Сегодня — добавляем «выразительность» через несколько модулей сразу.

Этап 2. Каталог модулей: повторение и расширение (5 мин)

🖥️ **Слайд 2: Основные модули «Урсулы»**

Содержание слайда: Сетка из 10 иконок модулей с краткими подписями.

«Каждый модуль — это отдельная „способность“ объекта. МодулеОбнаружения — глаза, МодульДвижения — ноги, МодульАнимаций — тело, Таймер и Счётчики — память, МодульВзаимодействия — руки. Сложное поведение = комбинация способностей».

(Полный список модулей можно найти здесь.)

Этап 3. Комбинирование модулей в одном состоянии (8 мин)

🖥️ **Слайд 3: Одно состояние — сразу много действий**

«Внутри одного блока состояния на событие Вход можно добавить сколько угодно команд от разных модулей. Они запускаются одновременно:

- МодульДвижения запускает моторы/ноги и перемещает модель;
- МодульАнимаций включает циклическую анимацию (если поставить число повторов = 0, то анимация будет бесконечной в этом состоянии — станет циклической);
- МодульИндикации зажигает шар-маяк над головой модели.

✨ **Пример: состояние «Патрулирование»**

Робот едет по маршруту, светит индикатором и работает манипулятором одновременно. Нам не нужно делать под каждое действие отдельное состояние, мы комбинируем их внутри одного состояния».

> 💡 **Логика «событие → реакция»**
>
> Хороший паттерн: на «вход в состояние» повесить стартовые действия (запустить анимацию, звук, движение), а на «выход из состояния» — завершающие (остановить движение, остановить атаку (применение вооружения)).

> 🖥️ **Слайд 4: Внутреннее событие — действие без смены состояния**

📣 **Действия учителя:**

«Обычно событие выталкивает персонажа по стрелке в другое состояние. Но что, если роботу, который движется по квадрату, нужно просто остановиться на 3 секунды в конце каждого отрезка своего маршрута?

Если делать для этого отдельное состояние "Осмотр" на каждом углу, схема раздуется до 8 блоков и покроется паутиной стрелок.

Вместо этого мы используем **внутреннее событие**:

1. На Входе робот едет 10 метров с синим маяком.
2. Когда колеса доехали до точки, срабатывает событие ПеремещениеВыполнено. Но мы **не протягиваем стрелку наружу**! Мы настраиваем это событие **прямо внутри этого же состояния**.
3. На месте робот глушит моторы (Стоп), зажигает зеленый маяк и запускает Таймер на 3 секунды.
4. А вот стрелка перехода к следующей точке теперь ждет окончания таймера: ТаймерВыполнен.

Схема остается компактной: всего 4 точки вместо 8!».

Этап 5. Демонстрация: настройка финальной точки игры (3 мин)

> 🖥️ **Слайд 5: Демо: Финал уровня — МодульПобедыИПоражения**

📣 **Действия учителя:**

«Любой квест должен иметь финал. Чтобы игра поняла, что игрок справился с задачей, используется **МодульПобедыИПоражения**.

Его главная команда — Победа(). Когда автомат доходит до этой команды, игра выводит окно "Вы победили!".

Для победного флага нам нужна предельно простая схема:

- Состояние Ожидание: флаг стоит, ждет игрока и выводит реплику, чтобы показать игроку точку финала: МодульВыводаТекста.ПоказатьТекст(win).
- Переход по клику: игрок подошел ближе 3 метров и кликнул по флагу (МодульВзаимодействия.ВзаимодействиеИгрока).
- Состояние Победа: вызывается команда МодульПобедыИПоражения.Победа()».

Этап 6. Практика: сложная механика своего NPC (16 мин)

> 🖥️ **Слайд 6: Задание: добавьте ≥3 модулей в поведение NPC и запрограммируйте победный флаг**

Каждый ученик дорабатывает свою диаграмму, добавляя звук, анимацию, индикацию, вывод текста и/или таймер. Цель — поведение, в котором задействованы минимум 3 модуля.

Идеи для доработки (выберите 1-2)

- NPC меняет анимацию при переходе между состояниями (МодульАнимаций в каждом состоянии).
- NPC преследует игрока ограниченное время, потом возвращается (Таймер + МодульДвижения).
- NPC подсвечивает свои действия шаром-индикатором (МодульИндикации).
- NPC выводит разные реплики (МодульВыводаТекста).

Пошаговая инструкция

1. Откройте свою диаграмму с урока 1.2.2.
2. Изучите инструкцию по сборке диаграммы для робота, который патрулирует по маршруту и в конце

каждого отрезка останавливается на 3 секунды. Дополните свою диаграмму по аналогии

3. Выберите состояние, в котором хотите добавить анимацию, индикацию или текст. Например:
    - Добавьте команду: МодульАнимаций → событие «вход в состояние» → действие «запустить анимацию» → параметр.
    - Добавьте вторую команду: МодульВыводаТекста → действие «ВывестиТекст» → параметр (имя реплики)
    - Добавьте Таймер, чтобы ограничить длительность.
4. Сохраните, подключите, протестируйте: убедитесь, что анимация и вывод текста синхронны с движением.
5. Воспроизведите также диаграмму победного флага, которую демонстрировал учитель. Соберите ее, сохраните, подключите и протестируйте финал уровня.

Этап 6. Подведение итогов (5 мин)

Учитель просит показать поведение. Обсуждение: какие механики получились самыми живыми.

📣 **Действия учителя:**

«Сегодня мы подготовили два элемента будущей игры: автономного NPC с живым поведением и финальную точку победы. На следующем уроке мы объединим все наши наработки в единый играбельный проект».

> ℹ️ **Домашнее задание**
>
> Готовиться к сборке проекта (следующий урок). Проверить, что все диаграммы сохранены, файлы GraphML не потеряны, локация в «Урсуле» открывается. Подумать, что ещё нужно доработать до финальной сборки.

📄 Раздаточный материал

Карточка «Основные возможности модулей»

МодульДвижения:

- **Команды:** ЗадатьКоординатуВниз/Вправо/Вверх/Влево(метры) — задать направление и расстояние пути.
- **Команда:** ДвигатьсяПоКоординатам() — включить моторы и начать движение.
- **Команда:** ДвигатьсяКОбъекту()/ДвигатьсяОтОбъекта() — начать непрерывное движение за текущей целью (преследование) или от текущей цели (убегание).
- **Команда:** ВернутьсяНаИсходную() — вернуть моба на стартовую точку, где он начинал игру.
- **Команда:** Стоп() — заглушить моторы / остановиться на месте.
- **Событие:** ПеремещениеВыполнено — колеса или ноги дошли до заданной точки.

МодульОбнаружения:

- **Команда:** ПоискИгрокаВРадиусе(метры) — включить круговой радар зрения моба.
- **Команда:** ВзаимодействиеИгрокаСОбъектом(имя, метры) — следить за кликом игрока по другому предмету в радиусе.
- **Команда:** ОстановкаПоиска() — выключить радар / прекратить сканирование.
- **События:** ИгрокОбнаружен и ИгрокНеОбнаружен — герой вошел в зону видимости или покинул ее.

МодульВзаимодействия:

- **Команды:** УстановитьПроходимость(True / False) — сделать объект проходимым (открыть створки) или твердой стеной (запереть).
- **Команда:** УстановитьДистанциюВзаимодействия(метры) — расстояние от героя до объекта, с которого сработает клик мыши (по умолчанию 3 метра).
- **Команда:** УничтожитьОбъект() — навсегда стереть предмет со сцены (подбор чипа, сбор ресурса, поломка преграды).
- **Событие:** ВзаимодействиеИгрока — клик левой кнопкой мыши по самому объекту на допустимой дистанции.

МодульАтаки:

- **Команда:** АтаковатьЦель() — включить непрерывный боевой режим (наносить урон цели по кулдауну).
- **Команда:** Стоп() — выключить боевой режим / опустить оружие (рекомендуется вызывать на

событии Выход).

- **События:** ЦельВошлаВЗону и ЦельВышлаИзЗоны — враг подошел на расстояние удара или отбежал назад.

Счётчик1 (и Счётчик2):

- **Команда:** ПрибавитьЗначение(N) — увеличить число в локальном блокноте объекта на N.
- **Команда:** ОтнятьЗначение(N) — уменьшить число в блокноте на N.
- **Команда:** ОбнулитьЗначение() — сбросить число обратно в 0.
- **Переменная:** ТекущееЗначение — сохраненное число для проверок в ограждающих условиях (<, >, ==, >=).

МодульВыводаТекста:

- **Команда:** ПоказатьТекст(ID_реплики) — зажечь парящую 3D-плашку с текстом рядом с моделью объекта.
- **Команда:** СкрытьТекст() — убрать текстовую плашку из игрового мира.
- **События:** ТекстПоказан и ТекстСкрыт — появление и исчезновение надписи.

МодульАнимаций:

- **Команда:** ВоспроизвестиАнимацию(имя, повторы) — запуск анимации модели (параметр 0 — крутить бесконечно, 1 — сыграть один раз).
- **Команда:** ОстановитьАнимацию() — зафиксировать позу модели.

МодульИндикации:

- **Команды:** ВключитьИндикатор() и ВыключитьИндикатор() — зажечь или погасить парящий цветной шар над объектом.
- **Команда:** УстановитьЦветИндикатора(Синий / Зеленый / Красный) — перекрасить световой маяк.

Таймер:

- **Команда:** ТаймерЗапуск(секунды) — запустить обратный отсчет секундомера.
- **Команда:** ТаймерСтоп() — остановить таймер / сбросить.
- **Событие:** ТаймерВыполнен — время вышло (триггер перехода по времени).

МодульПобедыИПоражения:

- **Команда:** Победа() — остановить игру и показать победный экран «Вы победили!».
- **Команда:** Поражение() — остановить игру и показать экран проигрыша.

Карточка «Инструкция по сборке диаграммы для патрульного робота»

Собираем умный патруль (robot_patrol_timer.graphml):

1. В Кибериаде GE создайте 4 состояния: Точка 1, Точка 2, Точка 3, Точка 4.
2. В Точке 1 настройте:
    - На Входе: МодульДвижения.ЗадатьКоординатуВверх(15), МодульДвижения.ДвигатьсяПоКоординатам(), МодульИндикации.УстановитьЦветИндикатора(Синий), МодульИндикации.ВключитьИндикатор(), МодульАнимаций.ВоспроизвестиАнимацию(work_1, 0).
3. Нажмите «+ Добавить событие» внутри этого же блока → выберите МодульДвижения.ПеремещениеВыполнено. Добавьте команды:
    - МодульДвижения.Стоп(),
    - МодульИндикации.УстановитьЦветИндикатора(Зеленый),
    - МодульВыводаТекст.ВывестиТекст(wait),
    - Таймер.ТаймерЗапуск(3).
4. Настройте точки 2, 3 и 4 по аналогии. На входе каждой следующей точки обязательно вызывайте ДвигатьсяПоКоординатам().
5. Соедините точки стрелками по кругу: каждая стрелка использует для перехода событие Таймер.ТаймерВыполнен.
6. Сохраните как robot_patrol_timer.graphml.
7. Настройте в Урсуле реплику модели по ID wait: «Произвожу осмотр».
8. Прикрепите диаграмму к модели, протестируйте в игре.

🎓 Методические рекомендации учителю

Урок расширяет набор инструментов. Главная идея: сложное поведение получается из комбинации возможностей разных модулей. Важно показать, что модули обмениваются данными через события и переменные, образуя систему.

Главная методическая задача этого урока — показать, что сложность поведения достигается не раздуванием графа, а слаженной работой независимых модулей внутри состояний.

💬 Скрипт-помощник для работы с группами

Ситуация	Фраза учителя
Ученик спрашивает, почему робот после остановки на точке 1 не едет к точке 2	*«Посмотри на команды в точке 2. На точке 1 ты вызвал команду Стоп(), моторы выключены. Чтобы робот поехал снова, на Входе в точку 2 обязательно должна стоять команда ДвигатьсяПоКоординатам()».*
Ученик запутался, где настраивать внутреннее событие, а где переход.	*«Если мы кликаем по самому блоку состояния и жмем «+ Добавить событие» — это внутреннее событие (плашка внутри блока). Если кликаем по стрелке между блоками и жмем гаечный ключ — это переход (плашка на стрелке)».*
Поведение получилось слишком сложным и непредсказуемым	*«Убери по одной команде и тестируй после каждого удаления. Когда станет понятно — это и есть твой рабочий набор».*
Не успевают сделать обе диаграммы за урок.	*«Приоритет — доработать поведение NPC (это сложная часть). Схему победного флага из двух блоков можно собрать за последние 3 минуты урока либо доделать дома».*

⚠️ Возможные трудности и решения

Ситуация	Решение
Анимация не запускается	Проверьте: (1) у модели NPC настроены анимации в карточке; (2) имя анимации в параметре совпадает с настроенным; (3) выставлен параметр повторов анимации — попробуйте поставить 5, чтобы точно не пропустить (анимации могут быть очень короткими), затем поменяйте на 0, что сделает анимацию циклической.
Слишком много команд в одном состоянии — трудно читать диаграмму	Разнесите команды по разным событиям: стартовые на «вход в состояние», завершающие на «выход из состояния».
Таймер не запускается повторно	После срабатывания «таймер истёк» нужно снова добавить действие «запустить таймер» — он не перезапускается автоматически.
Несмотря на установленный таймер, модель не делает паузы в 3 секунды	Проверьте стрелку перехода между Состояниями. Если там стоит ПеремещениеВыполнено — модель будет игнорироваь таймер. Замените событие на стрелке на Таймер.ТаймерВыполнен».

✅ Критерии оценки

✅ Успешно	❌ Требует доработки
• Использовано ≥3 модулей • Анимация и индикация добавлены и работают • Использована настройка через внутреннее событие. • Модули связаны через события, а не висят	• Используется только 1-2 модуля • Команды добавлены, но не запускаются (нет события) • Анимация или индикация не синхронизированы с движением • Созданы лишние промежуточные

✅ Успешно	❌ Требует доработки
изолированно • Поведение легко понять со стороны игрока • Собрана и сохранена отдельная диаграмма победного флага с вызовом команды Победа()	состояния вместо использования внутренних событий. • Диаграмма победного флага не создана или не содержит вызова Победа().

💡 Дополнительные советы

- Используйте консоль отладки — она показывает, какие события срабатывают. Это главный инструмент отладки.
- Если время поджимает — дайте готовый шаблон состояния с 3 командами, пусть ученики только меняют параметры.
- Свяжите с уроком 1.3: анимации — это контент, который создаётся в Blender. Можно упомянуть, что на следующих уроках они создадут свои ассеты.

🏠 Домашнее задание

Подготовить проект к сборке: проверить сохранение всех файлов, открыть локацию в «Урсуле». Записать, что нужно доработать.