1.2.1. Программирование автономного поведения неигровых персонажей для реализации квестового задания в редакторе ПРИМС

Блок: Блок 1.2. Программирование · **Полугодие:** 1 · **Класс:** 5-7 · **Продолжительность:** 40 мин

Тема: ПРИМС: состояния, переходы и базовое поведение NPC · **Инструменты:** ПРИМС (Cyberiada Graph Editor), Мастерская игр «Урсула»

📋 Паспорт урока

🎯 Цель	Познакомить учащихся с парадигмой программирования поведения через иерархические машины состояний. Научить создавать простую диаграмму поведения NPC (патруль / ожидание / реакция на игрока) в редакторе ПРИМС и подключать её к объекту в «Урсуле».
📚 Тип урока	Комбинированный урок с практической работой в редакторе
👥 Формы работы	Фронтальная беседа, демонстрация, индивидуальная практическая работа
🛠️ Оборудование	Проекты локаций в «Урсуле» (с урока 1.1.3), редактор ПРИМС (Cyberiada Graph Editor https://games.kruzhok.org/cyberiada-graphml/), проектная документация команд (концепция, квест, библии персонажей)

🎯 Планируемые результаты

Предметные

- Учащиеся понимают логическую связку «поведение персонажа = машина состояний»
- Учащиеся знают элементы ПРИМС: состояние, событие, действие
- Учащиеся умеют создавать диаграмму из 2–3 состояний с переходами и сохранять её в формате GraphML
- Учащиеся умеют подключать диаграмму к объекту в «Урсуле» и тестировать поведение

Метапредметные

Коммуникативные	• Умение задавать вопросы по интерфейсу нового инструмента.
Регулятивные	• Работа по пошаговой инструкции с новым ПО. • Самопроверка: соответствует ли поведение задуманному.
Познавательные	• Декомпозиция поведения NPC на состояния. • Анализ причинно-следственных связей между событиями и переходами.

📚 Ход урока

Этап 1. Введение (3 мин)

🖥️ **Слайд 1: Титульный**

📣 **Действия учителя:** Приветствие. Напоминание: на уроках геймдизайна команды придумали NPC, описали его в библии персонажа и поставили на локацию в «Урсуле». Но сейчас NPC просто стоит. Сегодня — научим его действовать.

Этап 2. Что такое машина состояний (7 мин)

> 🖥️ **Слайд 2: Поведение = машина состояний**

«Любое поведение NPC можно описать так: персонаж находится в каком-то состоянии (патрулирует, спит, бежит). Что-то происходит (игрок подошёл, истекло время) — и персонаж переходит в другое состояние. Это и есть машина состояний: набор состояний и переходов между ними».

Элемент	Что это	Пример
Состояние	Что персонаж делает прямо сейчас	«Патруль» — ходит по точкам; «Ожидание» — стоит
Событие	Что произошло в мире, что персонаж может заметить	Игрок вошёл в зону видимости; таймер истёк
Переход	Стрелка из одного состояния в другое; срабатывает по событию	Из «Патруль» в «Преследование» по событию «Заметил игрока»
Действие	Что персонаж делает внутри состояния (или при переходе)	В состоянии «Патруль» — двигаться к точке A; при переходе — зажечь индикатор

> ℹ️ **Иерархия состояний**
>
> Состояния могут быть вложенными: «Боевой режим» содержит внутри «Преследование» и «Атака», а «Спокойный режим» — «Патруль» и «Ожидание». Это иерархическая машина состояний (HSM) — на этом и построен ПРИМС. На первом уроке обходимся без иерархии.

Этап 3. Знакомство с редактором ПРИМС (Cyberiada Graph Editor) (7 мин)

> 🖥️ **Слайд 3: Интерфейс ПРИМС**
>
> Если есть возможность,- покажите интерфейс прямо в редакторе, так ученикам будет проще освоиться.

«ПРИМС — это визуальный язык: вы составляете диаграмму состояний и переходов между ними в специальном редакторе. Диаграмма сохраняется в файл формата GraphML (.graphml), который потом подключается к объекту в „Урсуле“».

Базовые элементы диаграммы ПРИМС

Элемент	Как создать	Что настраивать
Состояние (прямоугольник со скошенными или скругленными краями)	Кнопка «Добавить состояние» на холсте	Имя (например, «Патруль»); внутри — команды: модуль + событие + действие + параметр
Внешнее событие (располагается в желтой рамочке на стрелке-переходе)	Перетащить от «+» на краю состояния к другому состоянию	Событие-триггер (что запускает переход); можно добавить условие
Внутреннее событие (располагается в желтой рамочке внутри состояний)	Кнопка «Новое событие» на холсте	Событие-триггер (что запускает новые действия внутри этого же состояния)
Вход	Предзаданный специальный значок «entry», стрелка из него в первое состояние	Выбранное состояние будет начальным — с него начинается выполнение программы

Действия внутри состояния

Действия многосоставные и собираются в редакторе по формуле «модуль + команда + параметр». Например: модуль «МодульОбнаружения», действие «ПоискИгрокаВРадиусе», параметр «5»

> ℹ️ **Модули поведения в «Урсуле»**
>
> У объекта в «Урсуле» доступны модули: МодульОбнаружения (видит игрока/объекты), МодульДвижения (перемещение по карте), МодульАнимаций (запускает анимацию), Таймер, МодульИндикации, Счётчик1 и Счётчик2, МодульВзаимодействия. Каждый модуль предоставляет события (🚩), действия (👉) и переменные (1️⃣).

Этап 4. Демонстрация: простой NPC-патрульный (5 мин)

> 🖥️ **Слайд 4: Демо: NPC с двумя состояниями**

Учитель в реальном времени собирает простую диаграмму и комментирует:

1. Создаём состояние «Патруль», добавляем действия — 1) МодульДвижения → команда «ЗадатьСлучайныеКоординаты (20); 2) МодульДвижения → команда «ДвигатьсяПоКоординатам» ».
2. Создаём состояние «Ожидание», добавляем действие Таймер → команда «ТаймерЗапуск (3)» .
3. Переход от состояния «Патруль» к состоянию Ожидание» по событию «ПеремещениеВыполнено» (от «МодульДвижения»).
4. Переход от состояния «Ожидание» к состоянию «Патруль» по событию «ТаймерВыполнен» (от «Таймер»).
5. Назначаем начальным состоянием «Патруль».
6. Сохраняем в GraphML.

> 🔑 **Ключевой вывод:** Получился NPC, который ходит по случайным точкам и каждые 3 секунды останавливается отдохнуть. Минимальное поведение — минимальная диаграмма.

Этап 5. Практика: поведение своего NPC (13 мин)

🖥️ **Слайд 5: Задание: оживите NPC из вашего квеста**

Каждый ученик (или пара) создаёт диаграмму поведения для NPC из своего проекта.

Используйте библию персонажа и карту квеста.

Пошаговая инструкция

1. Откройте Cyberiada Graph Editor — редактор ПРИМС в «Урсуле» (в карточке моделей «Интерактивный объект» или «Моб» есть ссылка на переход в редактор).
2. Создайте новое состояние и назовите его «Патруль» (или, например, «Ожидание» — зависит от вашего NPC).
3. Внутри состояния: выберите событие (Общие), триггер Вход. Добавьте команду МодуляДвижения: действие («ЗадатьСлучайныеКоординаты»), параметр. Затем команду из этого же модуля: действие («ДвигатьсяПоКоординатам»).
4. Внутри этого же состояния добавьте команду из МодуляОбнаружения: действие «ПоискИгрокаВРадиусе» и установите параметр (например, 5).
5. Создайте второе состояние «Реакция на игрока» (например, «Идёт к игроку»).
6. Внутри второго состояния добавьте команду через МодульОбнаружения (событие «ИгрокОбнаружен») и МодульДвижения (действие «ДвигатьсяКОбъекту»).
7. Создайте переход из «Патруль» в «Реакция на игрока» по событию «ИгрокОбнаружен» от МодуляОбнаружения.
8. Создайте обратный переход «Реакция → Патруль» по событию «ИгрокНеОбнаружен».
9. Сделайте «Патруль» начальным состоянием.
10. Сохраните диаграмму в файл формата GraphML.
11. В «Урсуле» откройте проект, выберите NPC, в окне редактирования модели подключите сохранённый GraphML-файл.
12. Соберите игру и протестируйте: подходит ли NPC к игроку, когда тот рядом?

💡 **Совет**

Начните с 2 состояний — это проще отладить. Если работает — добавьте третье (например, «Возврат на стартовую точку»).

Этап 6. Подведение итогов (5 мин)

🖥️ Слайд 6: Что должно быть готово

Учитель просит показать работающее поведение. Короткое обсуждение: что получилось, где NPC ведёт себя странно.

ℹ️ Домашнее задание

Подумать: какое поведение вашего NPC зависит от того, что игрок уже сделал в игре (например, NPC по-разному реагирует, если игрок уже выполнил часть квеста). Это — повод для переменных, о которых узнаем на следующем уроке.

📄 Раздаточный материал

Шпаргалка по элементам ПРИМС

Элемент	Значок	Зачем нужен
Состояние	Прямоугольник со скругленными/скошенными краями	Что NPC делает сейчас (протяженный процесс)
Событие	пиктограмма в желтой рамке на стрелке-переходе или внутри состояний	Осуществляет смену состояний или действий внутри одного состояния

Элемент	Значок	Зачем нужен
Вход	Закрашенный кружок со стрелкой к начальному состоянию	Точка старта выполнения
Действие	Пиктограммы внутри состояний	Связка: модуль + команда + параметр (при необходимости)

Модули «Урсулы» (кратко)

Модуль	Что даёт
МодульОбнаружения	**События:** «ИгрокОбнаружен», «ОбъектОбнаружен», «ОбъектовНеОбнаружено»,«ИгрокНеОбнаружен» **Действия:** «ПоискИгрокаВРадиусе», «УстановитьРадиусОбнаружниея», «ОстановкаПоиска»
МодульДвижения	**События:** «Перемещение выполнено» **Действия:** «ДвигатьсяКОбъекту», «ДвигатьсяОтОбъекта», «ДвигатьсяПоКоординатам», «ВернутьсяНаИсходную» «ЗадатьСлучайныеКоординаты», «СбросКоординат», «ЗадатьКоординатуВверх/Вправо/Влево/Вниз»
МодульАнимаций	**События:** «АнимацияЗавершилась» **Действия:** «ВоспроизвестиАнимацию», «ОстановитьАнимацию»
МодульИндикации	**События:** «ИндикаторВключен», «ИндикаторВыключен», «СцетИндикатораИзменился» **Действия:** «УстановитьЦветИндикатора» (параметры: красный, синий, зеленый), «ВключитьИндикатор», «ВыключитьИндикатор»

Модуль	Что даёт
Таймер	**События:** «ТаймерВыполнен», «Тик», «Тик1Секунда». **Действия:** «ТаймерЗапуск», «ТаймерСтоп», «ТаймерСброс»
Счётчик1 / Счётчик2	**Событие:** «ЗначениеИзменилось». **Действия:** «ПрибавитьЗначение», «ОтнятьЗначение», «ОбнулитьЗначение»
МодульВзаимодействия	**События:** «ОбъектУничтожен», «ПроходимостьИзменена», «ВзаимодействиеИгрока» **Действия:** «УстановитьПроходимость», «ИзменитьПроходимость», «УничтожитьОбъект»

Чек-лист «Поведение NPC готово»

- ☐ Диаграмма содержит ≥2 состояний
- ☐ Есть ≥2 перехода с событиями-триггерами
- ☐ В каждом состоянии есть хотя бы одна команда
- ☐ Диаграмма сохранена в GraphML
- ☐ GraphML подключён к объекту в «Урсуле»
- ☐ Поведение протестировано: NPC реагирует на игрока

🎓 Методические рекомендации учителю

Урок знакомит с языком ПРИМС - программированием расширенных иерархических машин состояний — это может быть непривычно детям, знакомым только со Scratch. Концептуально оба редактора похожи, но в Scratch исполнение последовательное сверху вниз (императивно), а в ПРИМС она может жить в текущем состоянии и реагировать на заданные события (событийно). На этом уроке обходимся без переменных и иерархии — это темы следующих уроков.

Наставнику рекомендуется заранее пройти курс «Программирование в Кибериаде»

(https://games.kruzhok.org/courses/10), чтобы ознакомиться с основными модулями ПРИМС, которые могут потребоваться на уроках.

💬 Скрипт-помощник для работы с группами

Ситуация	Фраза учителя
Ученик не понимает, что такое «событие»	*«Событие — это сигнал от модуля: что-то произошло. МодульДвижения кричит „достиг точки!“, МодульОбнаружения — „вижу игрока!“. Ты в переходе выбираешь, на какой крик реагировать».*
Ученик путает состояние и действие	*«Состояние — это процесс/режим работы, в котором NPC находится продолжительно, пока не произойдёт событие“. Действие — это разовая команда внутри состояния. Патруль — состояние; „двигаться по координатам“ — действие».*
Ученик не понимает, как сохранить GraphML	*«В Кибериаде → Сохранить как (значок дискеты) → выбери папку проекта → назови robot_patrol.graphml. В „Урсуле“ потом выбери этот файл и прикрепи в окне редактирования модели (в раздел «Файл алгоритма»)».*

⚠️ Возможные трудности и решения

Ситуация	Решение
Ученик не знает, где искать Cyberiada Graph Editor	Используйте переход в редактор ПРИМС из «Урсулы» (кнопка редактирования модели, ссылка содержится в разделе «Файл алгоритма»).
NPC не реагирует на игрока в	Проверьте: (1) подключён ли GraphML к нужному объекту; (2) есть ли переход по событию «игрок

Ситуация	Решение
собранной игре	обнаружен»; (3) настроен ли радиус обнаружения в МодулеОбнаружения.
Ученик создаёт слишком много состояний сразу	Напомните про правило минимума: начните с 2 состояний. Если работает — добавляйте по одному. Большая диаграмма трудно отлаживается.
NPC хаотично переключается между состояниями или реагирует непонятно	Используйте «МодульИндикации» для отладки: «раскрасьте» состояния разными светодиодами и следите за их переключением – это поможет локализовать проблему
Нормали/анимации NPC отображаются криво	Это задача 3D-блока (1.3). На этом уроке главное — поведение, а не визуал. Запишите проблему для доработки.

✅ Критерии оценки

✅ Успешно	❌ Требует доработки
• Диаграмма минимальна (2–3 состояния) и работает • Верно определено начальное состояние • Переходы реагируют на осмысленные события (обнаружение игрока, достижение точки, срабатывание таймера) • Поведение соответствует библии персонажа (например, NPC-охранник патрулирует и реагирует на игрока)	• Слишком много состояний, NPC не отлаживается • Переходы без событий или с зацикливанием • Диаграмма не подключена к объекту • Поведение не соответствует задуманному персонажу

💡 Дополнительные советы

- Покажите консоль отладки «Урсулы» —в разных браузерах консоль может вызываться по разному, информацию об этом можно найти в интернете. Из-за количества записей этот режим отладки может несколько дезориентировать ученика и требует внимания со стороны педагога. Консоль помогает понять, почему NPC ведёт себя не так, поскольку в ней фиксируются все действия объектов в игре и переходы между состояниями.
- Если у ученика не получается продумать диаграмму самому — попросите его нарисовать диаграмму на бумаге или доске а затем объяснить вслух. Часто объяснение выявляет ошибку.
- Сохраняйте GraphML в общую папку команды, чтобы не потерять.

🏠 Домашнее задание

Подумать, какое поведение вашего NPC должно зависеть от действий игрока (например, по-разному реагировать до и после выполнения части квеста). Записать 2–3 идеи.