1.2.4. Сборка проекта, проверка работы автономных агентов, презентация игровой концепции

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

Тема: Финальная сборка и презентация проекта · **Инструменты:** ПРИМС (Cyberiada Graph Editor), Мастерская игр «Урсула»

📋 Паспорт урока

🎯 Цель	Объединить все созданные компоненты (локацию, персонажей, диаграммы поведения) в единый игровой проект, провести итоговое тестирование, устранить ошибки и подготовить проект к презентации.
📚 Тип урока	Урок-практикум с элементами презентации.
👥 Формы работы	Индивидуальная работа над проектом, взаимное тестирование, итоговая презентация.
🛠️ Оборудование	Все файлы проектов команд: локации в «Урсуле», GraphML-диаграммы, проектная документация.

🎯 Планируемые результаты

Предметные

- Учащиеся умеют собирать единый проект из локации, объектов и диаграмм поведения.
- Учащиеся проводят плейтест своего и чужого проекта по чек-листу.
- Учащиеся выявляют и устраняют ошибки в логике поведения.
- Учащиеся представляют проект, оценивая степень реализации концепции.

Метапредметные

Коммуникативные	• Конструктивная обратная связь при взаимном плейтесте. • Презентация результатов своей работы.
Регулятивные	• Планирование доработок в ограниченное время. • Приоритизация задач перед дедлайном.
Познавательные	• Анализ соответствия результата задуманной концепции. • Системный взгляд на проект как единое целое.

📚 Ход урока

Этап 1. Введение (2 мин)

> 🖥️ **Слайд 1: Титульный**

> 📣 **Действия учителя:** Приветствие. План урока: сборка → взаимный плейтест → доработка → итоговая презентация.

Этап 2. Чек-лист готовности проекта (3 мин)

> 🖥️ **Слайд 2: Что должно быть в проекте**
>
> Чек-лист из 7 пунктов: локация, игрок, ≥1 NPC с поведением, ≥1 квест, ≥1 переменная, ≥3 модуля задействованы в ПРИМС-диаграмме, игра запускается.

«Прежде чем собирать — проверьте, что все компоненты на месте. Это ваш контрольный список перед дедлайном».

- Локация создана в «Урсуле» и открывается.
- Добавлен игрок (через «Загрузка героя»)

- Добавлен ≥1 NPC с подключённой диаграммой поведения (GraphML)
- Реализован ≥1 квест (цель → препятствие → награда)
- Использована ≥1 переменная (Счётчик/Таймер) для ветвления
- В поведении задействовано ≥3 модулей
- Игра собирается и запускается в браузере без ошибок

Этап 3. Сборка и самоотладка (12 мин)

> 🖥️ **Слайд 3: Сборка + самоотладка**

Каждый ученик собирает свой проект и проходит его самостоятельно, отмечая ошибки. Время — 12 минут.

Алгоритм самоотладки

1. Соберите игру (кнопка «Перейти к игре» в «Урсуле»).
2. Пройдите свой уровень от старта до финала. Запишите, что не работает.
3. Если есть ошибка в консоли отладки — найдите соответствующий переход в диаграмме.
4. Исправьте диаграмму, сохраните GraphML, пересоберите.
5. Пройдите снова. Повторяйте, пока главный сценарий не будет работать.

> 💡 **Приоритет — главный сценарий**
>
> Если не работает второстепенная ветка — оставьте её на доработку. Главное — чтобы основной квест проходился от старта до финала. Лучше маленькая работающая игра, чем большая неработающая.

Этап 4. Взаимный плейтест (10 мин)

> 🖥️ **Слайд 4: Взаимный плейтест**

Ученики разбиваются на пары и меняются проектами. Каждый играет в игру соседа и заполняет бланк отзыва по чек-листу. Время — 10 минут (по 5 минут на каждую игру).

> ℹ️ **Что проверяет плейтестер**
>
> Понятно ли, куда идти? Понятно ли, что делать? Заставляет ли NPC задуматься/действовать? Какая механика зацепила? Что сломалось?

Этап 5. Доработка по отзывам (10 мин)

> 🖥️ **Слайд 5: Доработка по отзывам**

Ученики получают бланки отзывов от соседей и дорабатывают проекты по самым критичным замечаниям. Не пытайтесь исправить всё — выберите 1–2 самые важные правки.

> ⚠️ **Частые правки**
>
> «Не понятно, куда идти» → добавьте ориентир. «NPC не реагирует» → проверьте радиус обнаружения. «Квест не завершается» → проверьте переход в финальное состояние и условие.

Этап 6. Итоговая презентация (5 мин)

> 🖥️ **Слайд 6: Презентация проектов**

Каждая команда быстро представляет проект: концепция (1 фраза), что реализовано (2–3 пункта), что не успели (1 пункт). Класс и учитель задают один вопрос каждой команде.

> ℹ️ **Домашнее задание**
>
> Зафиксировать итоги блока «Программирование»: что получилось, что не успели, что хочется улучшить. Подготовиться к блоку 1.3 «3D-моделирование» — подумать, какой объект для своей игры хотелось бы сделать в Blender.

📄 Раздаточный материал

Бланк взаимного плейтеста

Заполняется плейтестером во время прохождения чужого проекта.

Проект: _____ Плейтестер: _____

Вопрос	**Ответ (да/нет + комментарий)**
Понятно, куда идти от старта?	
Понятно, что нужно делать?	
NPC реагирует на игрока?	
Квест можно завершить?	
Есть ветвление (разные сценарии)?	
Звук/анимация работают?	

Что зацепило (1 пункт)

_____

Что сломалось (1–2 пункта)

_____

Итоговый чек-лист проекта

- ☐ Локация в «Урсуле» открывается
- ☐ Добавлен игрок
- ☐ ≥1 NPC с диаграммой поведения
- ☐ ≥1 квест реализован и проходим
- ☐ ≥1 переменная используется
- ☐ ≥3 модуля в поведении NPC
- ☐ Игра собирается без ошибок

- ☐ Главный сценарий проходится от старта до финала

🎓 Методические рекомендации учителю

Урок завершает блок «Программирование». Главная цель — чтобы каждый ученик ушёл с играбельным прототипом. Не пытайтесь довести всё до идеала; лучше маленький работающий проект, чем амбициозный неработающий.

⚠️ Возможные трудности и решения

Ситуация	Решение
Проект не собирается — ошибка в «Урсуле»	Чаще всего проблема в экране «Формирование сцены» — например, случайно добавлены глобальные переменные, которые не задействованы, или битый сценарий игры (в поле сценарий игры). Удалите те компоненты, которые ненужны и попробуйте заново нажать на кнопку «Перейти к игре» в окне редактора сцены.
Ученик обнаружил, что весь квест не работает	Не паникуйте. Определите минимальный сценарий: что должно работать обязательно? Сосредоточьтесь на нём. Остальное — в список «не успели».
Взаимный плейтест выявил непонятные места	Это нормально — автор всегда слеп к своим ошибкам. Возьмите 1–2 самых частых замечания и исправьте. Не пытайтесь угодить всем.
Ученик расстроен, что «получилось некрасиво»	Напомните: это программный прототип, а не финальный продукт. Визуал — следующий блок (1.3). Главное сейчас — работает ли механика.

✅ Критерии оценки

✅ Успешно	❌ Требует доработки
• Игра запускается и проходится от старта до финала • Поведение NPC соответствует задуманному • Использовано ≥3 модулей и ≥1 переменная • Презентация честная: что работает — показано, что не успели — названо	• Игра не запускается • Квест невозможно завершить • NPC не реагирует на игрока • Команда скрывает неработающие части

💡 Дополнительные советы

- Дайте ученикам право на «не успели» — это честнее, чем скрывать проблемы.
- Если проект явно не успевает собраться — сократите масштаб: оставьте 1 NPC и 1 квест.
- Сохраните документацию по финальным версиям проектов в общую папку — они понадобятся во втором полугодии.

🏠 Домашнее задание

Зафиксировать итоги блока: что получилось, что не успели. Подготовиться к блоку 1.3 — подумать, какой объект из своей игры сделать в Blender.