2.1.1. Знакомство с языком Ink. Сюжетные возможности

Блок: Блок 2.1. Геймдизайн и Нарратив · **Полугодие:** 2 · **Класс:** 7 · **Продолжительность:** 40 мин

Тема: Язык Ink: узлы, выборы, переменные и переходы · **Инструменты:** Inky (официальный редактор Ink)

📋 Паспорт урока

🎯 Цель	Познакомить учащихся с синтаксисом классического языка интерактивной литературы Ink и средой разработки Inky. Научить проектировать ветвящийся диалог с переменными, условными репликами и финалами истории (-> END).
📚 Тип урока	Урок открытия новых знаний с практической работой в редакторе Inky.
👥 Формы работы	Фронтальная беседа, разбор примеров кода, индивидуальный практикум за компьютерами.
🛠️ Оборудование	Компьютеры с установленным редактором Inky, бланки шпаргалок по синтаксису Ink, документация команд (концепция, квест, библии персонажей) из первого полугодия.

🎯 Планируемые результаты

Предметные

- Учащиеся понимают, что такое Ink и для каких игр он подходит
- Учащиеся знают базовую структуру файла сценария Ink (узлы, переходы, выборы, концовки).
- Учащиеся знают, как объявлять переменные (VAR) и ветвить текст по условию { условие: ...- else:... }.
- Учащиеся создали и протестировали законченный интерактивный квест в Inky.

Метапредметные

Коммуникативные	• Умение формулировать диалоговые развилки, логически соответствующие характеру персонажа.
Регулятивные	• Работа по пошаговой инструкции. • Самоконтроль при наборе кода: отслеживание обязательных символов разметки, закрытия скобок и отступов.
Познавательные	• Декомпозиция ветвящегося сюжета на дискретные узлы графа. • Понимание логики булевых переменных (true/false) в контексте повествования.

📚 Ход урока

Этап 1. Введение (3 мин)

🖥️ Слайд 1: Титульный

📣 **Действия учителя:** Приветствие. Напоминание: в первом полугодии сделали игру в «Урсуле». Во втором — добавим настоящий нелинейный нарратив. Сегодня — знакомимся с Ink.

Этап 2. Что такое Ink (5 мин)

🖥️ Слайд 2: Ink — язык интерактивной литературы

Содержание слайда: Примеры игр на Ink: 80 Days, Sorcery!, Heaven's Vault, Overboard!. Скриншот ветвящегося диалога.

📣 **Действия учителя:**

«Ink — это язык для написания интерактивных историй с ветвящимся сюжетом. Создан студией Inkle Studios (игры 80 Days, Sorcery!, Heaven's Vault). Это не программирование в обычном смысле — вы пишете обычный текст, а специальными символами размечаете выборы и переходы».

«В первом полугодии мы научились оживлять объекты, задавать мобам маршруты и настраивать механику квеста на сцене. Но пока наши персонажи общались с игроком только короткими табличками над головой. Сегодня мы открываем большое направление нарративного дизайна — научимся писать настоящие ветвящиеся сюжеты с диалогами, выбором ответов и памятью. Для этого мы познакомимся с профессиональным языком игровой литературы — **Ink**. На нем пишут визуальные новеллы или интерактивные истории».

Полезные факты:

- Файлы Ink имеют расширение .ink
- Редактор Inky — официальный, бесплатный (Windows/Mac/Linux)
- Ink можно интегрировать в игру: «Мастерская игр “Урсула”» поддерживает воспроизведение ink-историй, есть даже специальный редактор для визуальных новелл, в котором можно быстро собирать игру из своего Ink-сценария. А дальше мы познакомимся с тем, как Ink встроен в 3D-игры.

💡 **Play-as-you-write**

- В Inky слева — текстовый редактор, где вы пишете реплики обычными словами, добавляя простые знаки разметки: плюсы, стрелочки и скобки.
- Справа — окно интерактивного плеера. Как только вы сохраняете файл (Ctrl + S), справа тут же появляется интерактивная книга: вы можете кликать по строчкам выбора и проверять, как развивается история».

Это делает написание интерактивной истории быстрым и увлекательным.

Этап 3. Базовый синтаксис: узлы, выборы, переходы (7 мин)

🖥️ **Слайд 3: Таблица базовой разметки**

Элемент разметки	Как пишется в коде	Что это делает в игре
Узел (knot)	=== имя_узла === (обязательно латиницей)	=== captain_talk === (Отдельная сюжетная сцена или глава диалога)
Старт истории	-> имя_узла (например -> start)	Первая строчка файла: указывает, с какого узла начать чтение
Строка выбора	+ [Кратко на кнопке] (Полная фраза (или мысли) героя)	Интерактивная строка, на которую кликает игрок. Создает вариант выбора и разделяет надпись на кнопке и речь (мысли) персонажа
Переход (divert)	-> имя_узла	Мгновенный переброс игрока к следующей сцене
Финал истории	-> END	Точка полного завершения сюжета игры
Комментарий	//	// Написанное после двойной косой черты не выводится в игре, но помогает разработчику

📣 **Действия учителя:**

«По сути, Ink выглядит как просто текст, который можно читать, будто это рассказ:

```
Ваш корабль совершил аварийную посадку. Оборудование повреждено и дальше лететь нельзя, но тебе и капитану Орбите удалось добраться до шлюзовой камеры станции. Со станции можно будет попросить о помощи. Вот только вам еще далеко до спасения...

Капитан Орбита склонился над аварийным пультом управления связи. На экранах тревожно мигают красные индикаторы сбоя питания станции.
```

Теперь добавим выбор — и переход дальше.

```
+ [Обратиться к капитану] Я: Капитан, мой скафандр в порядке, я могу быть полезным на поверхности. Какова обстановка?
    -> brief
+ [Присмотреться, что делает капитан] Кажется, капитану сейчас нельзя мешать...
    -> brief
```

И добавим реплику NPC:

```
=== brief ===
Капитан Орбита: Рад, что ты на ногах, пилот. Обстановка скверная: энергосеть жилого купола отключилась при аварийной посадке. Я держу реактор вручную, отойти от пульта не могу ни на секунду.
```

Узел (knot) — это сцена или глава. Переход (divert) `->` — переход к другому узлу. Переход происходит мгновенно, без участия игрока.

Каждая ветка диалога обязательно должна заканчиваться стрелкой перехода: либо -> следующий_узел, либо завершением истории — -> END».

Прямая речь NPC оформляется как Имя NPC: Реплика NPC.

ℹ️ Suppression [..] — спрятать текст выбора из вывода

Обратите внимание на то, как устроен выбор с квадратными скобками:

+ [Обратиться к капитану] Капитан, скафандр в порядке. Какова обстановка на станции? -> brief

Это специальный механизм языка, который называется **подавлением (suppression)**:

- Текст **внутри квадратных скобок [...]** отображается только на самой кнопке выбора. Игрок видит краткую суть действия: «*Обратиться к капитану*».
- Текст **после квадратных скобок** печатается в истории диалога как развернутая прямая речь космонавта или его мысли, когда игрок нажимает на эту кнопку.

Кнопка остается короткой и удобной, а диалог выглядит связным и литературным!

Этап 4. Работа с переменными (7 мин)

🖥️ Слайд 4: Переменные и развилки в тексте

Объявляем переменные: VAR, изменяем переменные через тильду ~ , проверяем переменные и выбираем, как отвечать — через условный блок { ... : ... - else: ... }

📣 Действия учителя:

«Сюжет становится игрой только тогда, когда мир помнит решения игрока. В Ink есть переменные:

1. **Создание переменной:** пишется («объявляется») в самом верху файла со словом VAR:

   VAR has_chip = false *(в начале игры запасного чипа питания у героя нет)*.
2. **Изменение переменной:** любая логическая операция в сценарии пишется со знака тильды ~:

   ~ has_chip = true *(герой нашел резервный модуль в контейнере)*.

```
=== rocks ===

Ты стоишь среди массивных темных скал северной гряды. Где-то здесь должен лежать спасательный контейнер.

+ [Искать контейнер среди камней] Ты осматриваешь трещины в скалах и находишь контейнер с чипом! Теперь можно и вернуться к капитану.

    ~ has_chip = true

    -> captain
```

Проверка в диалоге: условная развилка позволяет NPC произносить разные реплики в зависимости от ситуации: если чип есть — Капитан радуется и запускает системы базы. Если чипа нет — отправляет на поиски. Сценарий помнит, где мы были и что нашли!».

```
=== captain ===
```

```
{ has_chip:

    Капитан Орбита: Отличная работа, пилот! Вставляю чип в считыватель... Есть контакт! Системы купола перезапущены, шлюз открыт. Мы спасены!

        + [Зайти в безопасный купол] Наконец-то мы в безопасности и можем вызвать спасателей! -> win

- else:

    Капитан Орбита: Ты вернулся с пустыми руками? Без чипа мы не сможем открыть шлюз купола. Возвращайся к скалам и найди контейнер!

        + [Пойти к скалам за чипом] Понял, иду искать. -> rocks

}

//И не забываем закрывать такие скобки!
```

Этап 5. Сюжетная карта квеста (4 мин)

> 🖥️ **Слайд 5: Схема квеста в игре**
>
> *Содержание слайда:* Простая схема квеста: **старт** (описание ситуации) → **доклад Капитану** у аварийного пульта → **развилка**: безопасный поиск контейнера у скал или опасная попытка замкнуть искрящий кабель → **финалы** (win или bad_end).

📣 **Действия учителя:**

«Давайте свяжем наш сценарий с миром игры. Вспомните NPC-наставника, которого мы использовали в ранних уроках — Капитана Орбиту. Капитан работает с аварийным реактором базы после жесткой посадки челнока. Отойти от пульта он не может ни на секунду, поэтому ему нужен игрок.

Шлюз жилого купола заблокирован автоматикой защиты. Чтобы его открыть, нужен резервный чип питания. Капитан отправляет нашего космонавта на поиски контейнера к скалам и предупреждает: рядом искрит силовой кабель с опасным напряжением!
У игрока есть выбор:

1. Выполнить распоряжение командира: отправиться к скалам, найти контейнер, забрать чип (~ has_chip = true), вернуться и успешно завершить игру (win-> END).
2. Проявить неоправданный риск: попробовать замкнуть искрящий высоковольтный кабель напрямую в обход инструкций Капитана, получить удар током и проиграть (bad_end -> END)».

Этап 6. Практика: диалог для своего NPC (12 мин)

> 🖥️ **Слайд 6: Задание: ветвящийся диалог с NPC из вашего проекта**
>
> *Содержание слайда:* Чек-лист: ① открыть Inky ② Объявить переменную ③ создать ≥3 узла и ≥ по 3 реплики игрока и NPC ④ создать 1 сюжетную развилку -> END ⑥ запустить в Inky и пройти обе ветки.

Каждый ученик пишет ветвящийся диалог с NPC из своего проекта (по предоставленному шаблону), тестируя каждую развилку.

Пошаговая инструкция

1. Откройте Inky (если не установлен — скачайте с github.com/inkle/inky/releases).
2. Создайте новый файл: File → New. Сразу сохраните, например, как dialog.ink.
3. Объявите переменную.
4. Далее в первой строке напишите: -> start
5. Создайте узел === start === с приветствием NPC и 2–3 выборами (используйте suppression [..]).
6. В одном из узлов переменная должна изменяться с false на true.
7. Каждый выбор должен вести -> в новый узел или к -> END.
8. Создайте ≥3 узлов: start, как минимум 2 промежуточных, и финальный.
9. Финальный узел завершайте -> END.
10. Сохраните. Запустите диалог в плеере справа — пройдите все ветки.
11. Исправьте ошибки, если они есть (Inky подсветит warning о «loose ends», если нет перехода к следующему узлу или имя узла написано неверно).

> 💡 **Связь с проектом**
>
> Диалог должен быть частью вашего квеста. Например: NPC даёт задание, игрок выбирает, как

реагировать, диалог приводит к одной из двух реплик NPC.

Этап 7. Подведение итогов (2 мин)

Учитель просит показать работающий диалог. Обсуждение: какие диалоги получились самыми живыми.

📣 **Действия учителя:**

«Вы только что написали законченную интерактивную новеллу. Сейчас она живет как текстовый квест внутри Inky: игрок выбирает действия строчками на экране. Но в следующем уроке мы совершим важный шаг: перенесем этот текстовый сценарий прямо в наш трехмерный мир Урсулы и соединим его с настоящими 3D-моделями!».

ℹ️ **Домашнее задание**

Расширить диалог до 5–6 узлов. Подумать: какие выборы должны зависеть от того, что игрок уже сделал в игре (это — тема следующего урока: переменные и условия).

📄 Раздаточный материал

Шпаргалка по базовому синтаксису Ink

Конструкция	Синтаксис	Пример использования	Назначение
Узел сцены	=== имя ===	=== brief ===	Задает границы диалоговой сцены
Точка старта	-> имя_узла	-> start	Указывает первую сцену при старте игры
Строка выбора с	+ [Кратко]	+ [Спросить о базе] Что со	Разделяет текст кнопки и речь героя

репликой	Полная фраза	станцией?	
Переход	-> имя_узла	-> search_rocks	Направляет сюжет к следующему узлу
Финал	-> END	-> END	Завершает сценарий полностью
Переменная	VAR имя = знач	VAR has_chip = false	Хранит состояние мира в памяти сценария
Изменение	~ имя = значение	~ has_chip = true	Меняет значение переменной по ходу сюжета
Ветвление	{ условие: ... - else: ... }	{ has_chip: ... - else: ... }	Показывает разные реплики в зависимости от переменной

Шаблон квеста

```
VAR has_chip = false

-> start

=== start ===

Ваш корабль совершил аварийную посадку. Оборудование повреждено и дальше лететь нельзя, но тебе и капитану Орбите удалось добраться до шлюзовой камеры станции. Со станции можно будет попросить о помощи. Вот только вам еще далеко до спасения...

Капитан Орбита склонился над аварийным пультом управления связи. На экранах тревожно мигают красные индикаторы сбоя питания станции.

+ [Обратиться к капитану] Капитан, мой скафандр в порядке, я могу быть полезным на поверхности. Какова обстановка?

    -> brief

+ [Тихо кашлянуть, вдруг его нельзя отвлекать] Кхм-кхм…

    -> brief

=== brief ===

Капитан Орбита: Рад, что ты на ногах, пилот. Обстановка скверная: энергосеть жилого купола отключилась при аварийной посадке. Я держу реактор вручную, отойти от пульта не могу ни на секунду.

+ [Спросить про жилой купол] Мы сможем открыть шлюз и укрыться внутри?
```

```
    -> lock

+ [Предложить помощь] Чем я могу помочь прямо сейчас?

    -> task

=== lock ===

Капитан Орбита: Автоматика шлюза заблокировала створки из-за потери питания. Нужен резервный чип доступа, чтобы перезапустить систему.

+ [Узнать местонахождение чипа] Где нам взять запасной чип?

    -> task

=== task ===

Капитан Орбита: Грузовой контейнер с запасными модулями упал при спуске в каменистую гряду к северу. Отыщи его и принеси чип сюда! И будь осторожен: аварийный кабель искрит, не вздумай подходить к оголенным проводам.

+ [Отправиться на поиски к скалам] Принято! Выдвигаюсь к северной гряде.

    -> rocks

+ [Предложить замкнуть кабель] Капитан, а если я просто замкну кабель перемычкой в обход защиты?

    -> bad_end

=== rocks ===

Ты стоишь среди массивных темных скал северной гряды. Где-то здесь должен лежать спасательный контейнер.

+ [Искать контейнер среди камней] Ты осматриваешь трещины в скалах и находишь контейнер с чипом! Теперь можно и вернуться к капитану.

    ~ has_chip = true

    -> captain

+ [Вернуться к капитану без чипа] Скалы слишком опасны. Лучше сначала посоветоваться.

    -> captain

=== captain ===

{ has_chip:

    Капитан Орбита: Отличная работа, пилот! Вставляю чип в считыватель... Есть контакт! Системы купола перезапущены, шлюз открыт. Мы спасены!

    + [Зайти в безопасный купол] Наконец-то мы в безопасности и можем вызвать спасателей!

        -> win

- else:

    Капитан Орбита: Ты вернулся с пустыми руками? Без чипа мы не сможем открыть шлюз купола. Возвращайся к скалам и найди контейнер!

    + [Пойти к скалам за чипом] Понял, иду искать.

        -> rocks

}

=== win ===
```

```
Створки шлюза с шипением открываются. Вы с Капитаном входите в жилой модуль станции.
-> END
=== bad_end ===
Капитан Орбита: Стой, безумец! Там напряжение в десять киловольт!..
Но ты уже касаешься оголенного провода. Мощнейший электрический разряд выводит скафандр из строя...
-> END
```

Шаблон можно адаптировать под учеников (уменьшить количество узлов, убрать вторую развилку), пояснить на схеме квеста, какие узлы являются обязательными.

🎓 Методические рекомендации учителю

Урок знакомит учащихся с языком Ink в его автономном виде. Главная педагогическая задача — дать детям уверенность в работе со средой Inky и развить навык алгоритмического мышления через связный текст, прежде чем переходить к интеграции с 3D-движком.

Главная сложность для учеников — привыкнуть к символам +, ->, ===, { … }. В Inky это сделать легче, потому что есть мгновенный предпросмотр: написал — увидел. Используйте это.

💬 Скрипт-помощник для работы с группами

Ситуация	Фраза учителя
Ученик не понимает, что такое узел	*«Узел — это сцена. У неё есть имя. Ты можешь из любой точки истории сделать -> имя_узла — и игрок туда попадёт. Это как глава в книге, только с переходами».*
Ученик путает + и ->	*«+ — это кнопка, которую игрок нажимает. -> — это невидимый прыжок, который происходит сам, чтобы повествование могло идти дальше».*

Ситуация	Фраза учителя
Ученик не понимает, зачем [..]	*«Без скобок текст выбора повторится в выводе: кнопка «Обратиться к капитану» и строка «Капитан, скафандр в порядке». Со скобками [Обратиться…] — кнопка есть, а в выводе только продолжение. Это чище и живее».*
Ученик забыл поставить переход в конце ветки	*«Посмотри на окно Inky: снизу появилось предупреждение "loose end". Это значит, что сюжет зашел в тупик. Каждая строчка выбора обязана заканчиваться стрелкой: либо -> имя_узла, либо финалом -> END».*
Ученик путает изменение переменной и проверку	*Запомни правило: когда мы меняем значение переменной, пишется тильда и одно равно: ~ has_chip = true. Когда проверяем условие в фигурных скобках, пишется только имя: { has_chip: ... }».*

⚠️ Возможные трудности и решения

Ситуация	Решение
Inky не установлен	Скачайте с github.com/inkle/inky/releases/latest. Распакуйте, запустите. Альтернатива — онлайн-редактор на inklestudios.com/ink/ (без сохранения в файл).
Inky ругается на «loose end»	Ветка не заканчивается -> END или -> узел. Inky подсветит строку. Добавьте переход в конце каждой ветки.
Кириллица отображается кракозябрами	Файл сохранён не в UTF-8. В Inky: File → Save As → убедитесь, что кодировка UTF-8. В крайнем случае — создайте новый файл и скопируйте текст.

✅ Критерии оценки

✅ Успешно	❌ Требует доработки
• Скрипт компилируется в Inky без предупреждений и ошибок (*нет loose ends*) • Есть ≥3 узла с осмысленными именами • Есть ≥3 выбора • В выборах используются квадратные скобки [...], а после них идет осмысленная прямая речь персонажа • Переменная создана через VAR, изменяется через ~ и проверяется в блоке { …: … -else } • Диалог связан с NPC из проекта	• Файл не запускается в плеере Inky или сразу высвечивается End of story • Есть loose ends • Только 1 выбор или 1 узел • Выборы оформлены без скобок либо после скобок нет текста речи • Переменная объявлена, но не изменяется или не влияет на реплики. • Диалог не связан с проектом

💡 Дополнительные советы

- Показывайте Inky на проекторе — дети быстрее поймут, как «код → предпросмотр».
- Дайте готовый шаблон (из раздаточного) — пусть начинают с него и модифицируют.
- Не требуйте литературного качества — важна структура, а не стиль.

🏠 Домашнее задание

Завершить работу над диалогом для своей игры. Расширить его до 5–6 узлов и 2 выборов. Подумать, какую еще переменную нужно задействовать в своей игре.