Инструкция по использованию движка StoryForge  
для создания визуальных новелл на Unity

1. Общая информация

Движок StoryForge предназначен для создания интерактивных визуальных новелл с разветвлённым сюжетом, использованием JSON и собственного текстового формата сценариев. Система реализована на Unity с использованием языка программирования C# и Zenject для внедрения зависимостей.

1. Установка и настройка

* Установите Unity 2021 или выше.
* Используйте Visual Studio для редактирования кода.
* Клонируйте репозиторий проекта через Git.
* Откройте проект в Unity Editor.

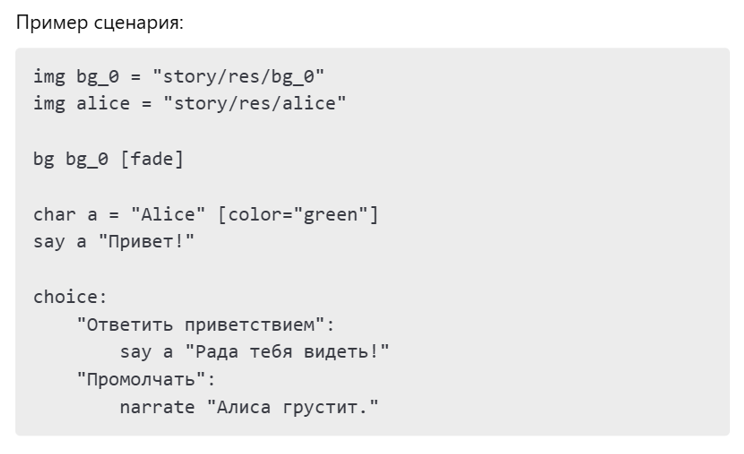
1. Структура проекта

Папки проекта:

* **Resources/story/chapters/ — главы и сцены новеллы.**
* **Resources/story/map/ — ресурсы карты сюжета.**
* **Resources/story/music/ — аудиофайлы.**
* **Resources/story/characters/ — спрайты персонажей.**

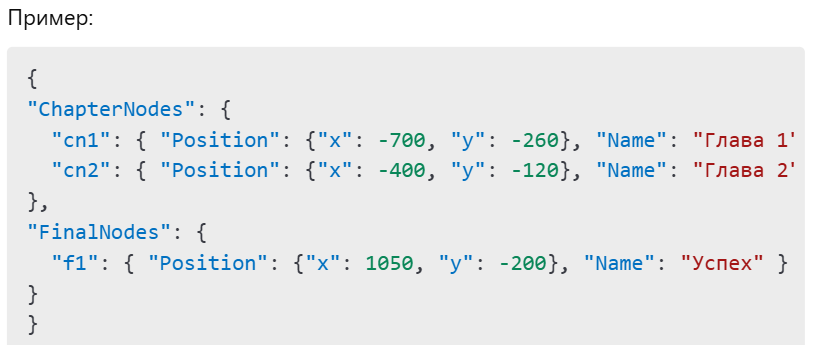
1. Создание сценария

Файлы сценария хранятся в формате .txt внутри папок глав:



1. Конфигурация через JSON

Файл story.json определяет структуру сюжета, связи и карту повествования:

Внутри JSON-файла вы задаёте:

* Имя истории
* Главы и их параметры (сцены, позиция на карте)
* Ачивки
* Концовки
* Карту сюжета

1. Работа с движком

* Запуск: Запустите Unity Editor, затем сцену с начальной точкой входа (SceneContext).
* Проверка сценариев: Сценарии автоматически подгружаются через ReaderService.
* Интерфейс и отображение управляются через контроллеры:
  + **BGController — фоны.**
  + **CharacterController — персонажи.**
  + **DialogueController — диалоги.**
  + **UIManager — элементы интерфейса (диалоги, кнопки и т.д.)**

1. Система сохранений

Сохранения реализованы через PlayerPrefs:

* Автоматическое сохранение после каждой главы.
* Ручное сохранение доступно в любой момент.

1. Карта сюжета

* Отображается прогресс игрока и доступные пути.
* Заблокированные ветки затемнены, разблокировка происходит через выборы игрока.

1. Диаграмма взаимодействий
2. Игрок запускает игру.
3. Система загружает сохранения или начинает новую игру.
4. Отображаются сцены и выборы.
5. Игрок делает выбор.
6. Состояние сюжета обновляется.
7. Автоматическое сохранение состояния.
8. Процедурная генерация и динамические выборы

* Парсер автоматически формирует древо сюжета.
* Ветки сюжета могут быть скрыты и отображаться только после разблокировки.
* Поддерживаются таймеры на выборы, по истечении которых выбор блокируется автоматически.

1. Работа с ошибками

* В случае ошибок парсера или отсутствия ресурсов выводятся сообщения в консоль Unity.
* Проверяйте логи в Unity Editor для отладки сценария.

1. Технические требования

Минимальные требования:

* ОС: Windows 10/11 (64-bit)
* Процессор: Intel Core i3
* ОЗУ: 4 ГБ
* GPU: DirectX 11

1. Поддержка и расширение

* Архитектура движка позволяет легко расширять функциональность.
* Для добавления новых функций используйте Zenject и паттерны DI.

Используйте данную инструкцию как основу для быстрого старта работы с движком StoryForge и разработки собственных визуальных новелл.

**Синтаксис Языка**

# Инструкция к синтаксису

# comment

# сказать что-то

narrate "что-то"

# загрузить спрайт

img bg\_0 = "story/res/bg\_0"

img bg\_1 = "story/res/bg\_1"

img alice = "story/res/alice"

img bob = "story/res/bob"

# в некоторые методы вот так: [arg1, arg2="val"] можно передавать доп аргументы, примеры сразу показаны

# порядок не важен

# если аргумент передан без "=", то это bool = True

# сменить фон (изначально стоит черный фон) | fade -> затемнить анимацией, изначально ее нет

bg bg\_0 [fade]

# создать персонажа | color -> цвет текста, изначально серый

char a = "Alice" [color="green"]

char B = "Bob"

# сказать за персонажа

say a "Hi, Bob"

say и "Hi, Alice"

bg bg\_1 [fade]

# объявить переменные (локально - сцена, глобально - на все главы, постоянно - даже вне игры, т.е. при повторной прохождении)

# переменные двух типов: флаги (bool) и числа (float)

# флаг объявляем true

local val\_0

# число

global val\_1 = 5.55

# только числа через "="!, после "=" идет выражение

persistent val\_2 = 55 + speed

# удаление

remove local val\_0

# условия, пишутся в "", можно добавлять несколько elif

if "a + b > 5":

narrate "Sum is greater than five"

elif "c":

narrate "Variable c is truthy"

narrate "..."

else:

narrate "Neither condition was met"

# меню-выбора | auto -> если истечет таймер, то выберется автоматически, time - время таймера, сколько доступен выбор, enif - условие открытия

choice:

"choice 1" [auto, time=3.3, enif="a+b=5"]

narrate "this was choice 1"

narrate "this was choice 1.."

choice:

"choice nested1"

narrate "this nested1"

"choice nested2"

narrate "this nested2"

"choice 2"

narrate "this was choice 2"

# перейти на сцену этой главы

move "c0s1"

# перейти на первую сцену другой главы, сохраняет

enter "chap1"

# финал, сохраняет

final "final\_0"