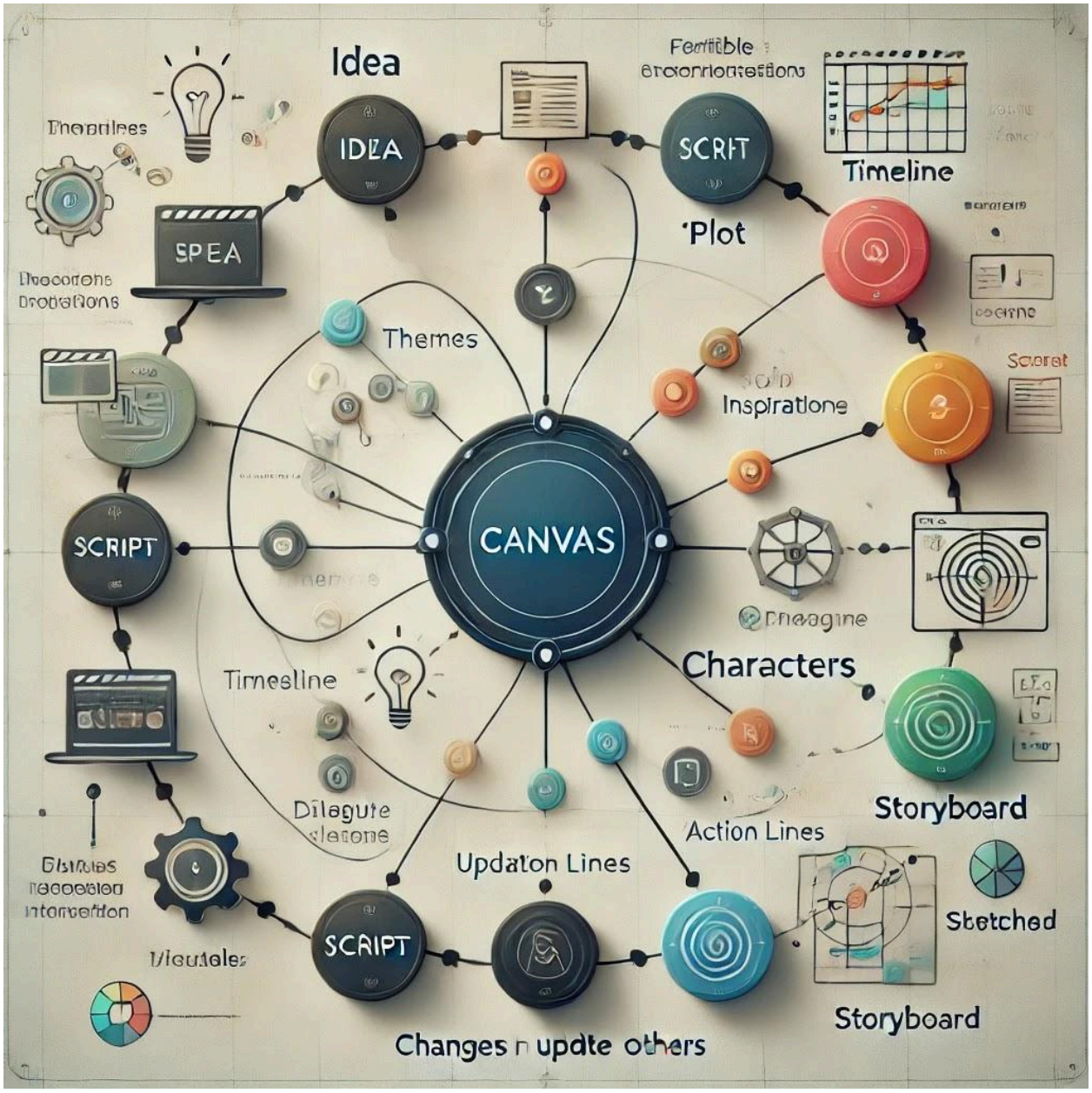
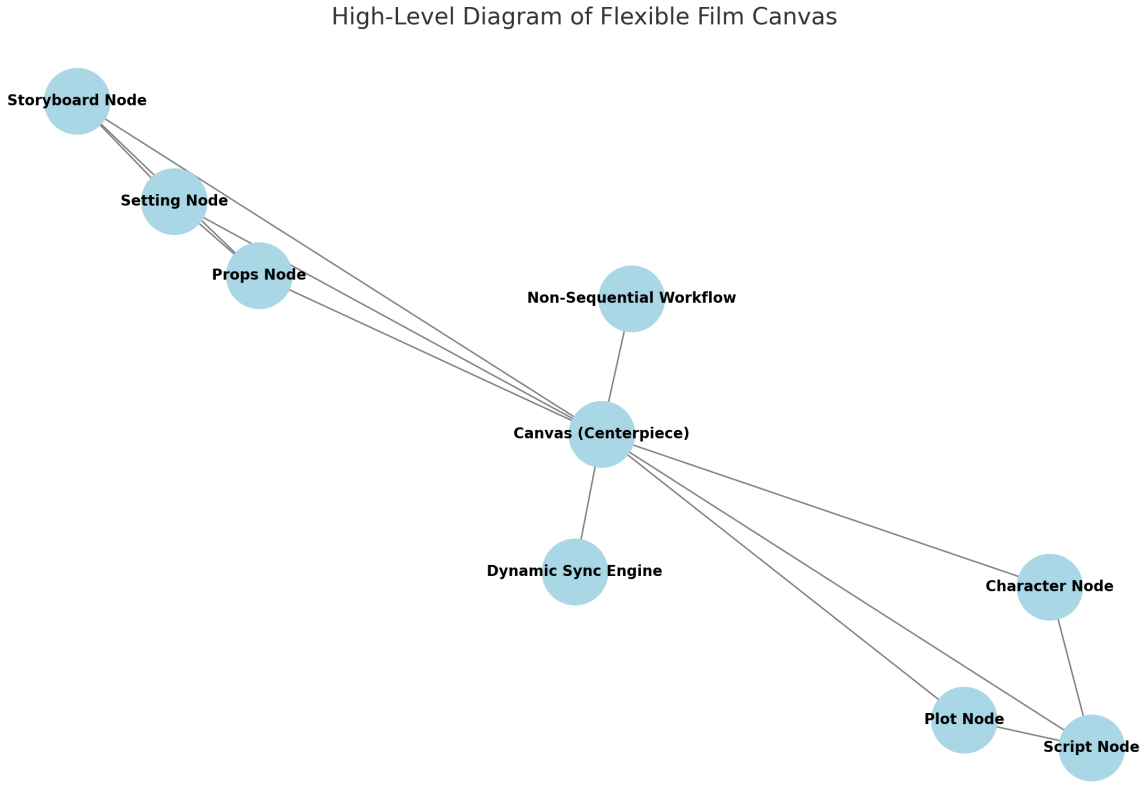
overview


Idea
IDEA
SCRIPT
Timeline
Plot
SPEA
Themes
CANVAS
SCRIPT
Inspirations
Characters
Storyboard
Timeline
Action Lines
Updaton Lines
SCRIPT
Sketched
Storyboard
Changes n update others


High-Level Diagram of Flexible Film Canvas
Storyboard Node
Setting Node
Props Node
Non-Sequential Workflow
Canvas (Centerpiece)
Dynamic Sync Engine
Character Node
Plot Node
Script Node

high level

Flexible Film Canvas: Functional Design and Workflow Requirements

High-Level Overview (Functional View):

The "Flexible Film Canvas" is a virtual tool designed to mimic real-life filmmaking workflows while offering unparalleled flexibility. Unlike traditional methods, this tool empowers users to work non-sequentially, allowing dynamic changes across all interconnected components.

Core Functional Components:

1. **Canvas (Central Hub):**
    - Acts as the primary interface where all elements of the workflow connect.
    - Provides a holistic view of the entire project.
    - Enables direct interaction with nodes such as idea generation, plot, script, characters, storyboards, scenes, and more.
2. **Idea Node:**
    - Capture rough ideas, themes, and inspirations.
    - Utilize AI tools to brainstorm concepts or generate high-level outlines.
3. **Plot Node:**
    - Create and visualize timelines, scenes, and key events.
    - Allow reordering of scenes dynamically, updating other components accordingly.
4. **Script Node:**
    - Write or generate dialogue, action lines, and scene descriptions.

- Synchronize dynamically with plots, scenes, and characters.

5. **Character Node:**
    - Develop profiles, traits, and backstories for characters.
    - Integrate with scripts (e.g., dialogue) and storyboards (e.g., visual representation).
6. **Storyboard Node:**
    - Sketch visuals or upload references for scenes.
    - Link scenes to characters, props, and settings dynamically.
7. **Scene Node:**
    - Define specific scenes (e.g., the protagonist's office) with settings, props, and actions.
    - Auto-synchronize with the "Settings (佈景)" and "Props (道具)" nodes.
8. **Settings (佈景) Node:**
    - Maintain details about locations and environments.
    - Allow ad hoc creation of settings when defining new scenes.
9. **Props (道具) Node:**
    - Catalog props associated with scenes or characters.
    - Enable dynamic creation and updates from any workflow step.

Key Features and Workflow Advantages:

1. Centralized, Flexible Canvas:

- The canvas is the heart of the system, connecting all components.
- Users can navigate between nodes, review relationships, and make adjustments in real time.

2. Non-Sequential Workflow:

- Users can begin work at any stage—from idea to script to storyboard—without adhering to a linear process.
- Adjustments in one component automatically update connected nodes.

3. Dynamic Synchronization:

- Example 1: Reordering scenes in the "Plot Node" updates the corresponding script, storyboard, and timeline.
- Example 2: Changing a character's trait in the "Character Node" updates dialogue, storyboards, and character arcs.
- **Example 3 (New):** Creating a scene (e.g., the protagonist's office) in the "Scene Node" automatically generates an object in the "Settings (佈景) Node." Users can then expand on this setting by adding visual details, layout, and props. Similarly, props defined within scenes update the "Props (道具) Node" dynamically.

4. Mimicking Real-Life Filmmaking with Enhanced Flexibility:

- Traditional filmmaking workflows require completing scripts and settings before shooting begins. This tool breaks that constraint, allowing users to define settings (佈景) and props (道具) on the fly, even as scripts or storyboards evolve.

- For example, a director (user) can create a new setting for a scene and add details progressively, rather than predefining everything. This mimics the organic, iterative nature of real filmmaking but allows far greater flexibility.

---

functional design

Modified Design Document:

1. User Goals:

- Empower users to explore, create, and iterate on film components dynamically.
- Provide a tool that adapts to the user's workflow rather than imposing a rigid sequence.

2. Functional Modules:

- **Canvas (Main Interface):** Displays interconnected nodes and provides navigation.

- **Nodes:**

    - Idea Node (Themes, Inspirations, High-Level Outlines)
    - Plot Node (Timeline, Key Events, Scene Order)
    - Script Node (Dialogue, Action Lines, Descriptions)
    - Character Node (Profiles, Traits, Arcs)
    - Storyboard Node (Visual Sketches, Scene Representation)
    - Scene Node (Detailed Scene Descriptions)
    - Settings (佈景) Node (Location Details)
    - Props (道具) Node (Catalog and Details of Props)

3. Interconnectivity Examples:

- A change in the plot timeline updates scripts and storyboards.
- Adding a scene auto-generates a setting object in the "Settings (佈景) Node" and links it to the appropriate storyboard and script components.

- Updating a character's backstory adjusts their dialogue and interactions across relevant scenes.

4. Non-Sequential Workflow Implementation:

- All nodes are accessible and editable at any time.
- Changes propagate dynamically to maintain consistency across the project.
- Users can define ad hoc elements (e.g., settings, props) without waiting for other steps to be complete.

5. User Interface Design:

- **Canvas View:**
    - Central hub displaying nodes with visual connections.
    - Clickable nodes to open detailed views.
- **Node Views:**
    - Focused interfaces for individual components (e.g., Plot Editor, Script Editor).
    - Options for creating, modifying, and visualizing relationships.

Next Steps:

1. Develop wireframes for the canvas and node interfaces.
2. Define technical requirements for dynamic synchronization across nodes.
3. Create sample workflows to validate non-sequential editing capabilities.
4. Gather user feedback on key use cases to refine the design.

more details

wireframe & tech

Wireframes, Technical Requirements, Sample Workflows, and Feedback Plan for the Flexible Film Canvas

1. Develop Wireframes

1.1 Canvas Overview Wireframe

- **Central Hub:** Displays nodes connected by labeled pathways indicating their relationships (e.g., Plot -> Script -> Storyboard).
- **Interactive Nodes:** Clickable elements for "Idea," "Plot," "Script," etc., each with hover descriptions.
- **Navigation Panel:** Sidebar or top menu allowing quick access to all nodes or global settings.
- **Dynamic Updates Indicator:** Visual cues (e.g., color changes or icons) to show which components are dynamically updated after a change.
- **Search/Filter Functionality:** A top bar for filtering scenes, characters, or other elements.

1.2 Individual Node Interfaces

1.2.1 Plot Node Wireframe:

- **Timeline View:** A horizontal timeline with draggable scene markers.
- **Scene Details Panel:** Opens on marker click, showing scene name, description, and relationships (e.g., linked script, storyboard).
- **Add/Remove Scene Buttons:** Quick tools to modify the plot structure.

1.2.2 Script Node Wireframe:

- **Text Editor:** Central area for script writing with sections for action lines, dialogues, and descriptions.
- **Sidebar:** Links to related nodes (e.g., characters, scenes).
- **Real-Time Updates:** Highlight areas auto-modified based on changes in linked nodes.

1.2.3 Storyboard Node Wireframe:

- **Scene Thumbnails:** Scrollable gallery of sketches or images.
- **Detail Panel:** Opens upon selecting a thumbnail, showing linked settings, props, and character visuals.

1.2.4 Settings and Props Nodes:

- **Catalog View:** Grid or list format for easy navigation of existing entries.
- **Detail Editor:** Panels to edit or add specific settings/props.
- **Link Management:** Tools to associate settings/props with scenes or characters.

---

2. Define Technical Requirements

2.1 Dynamic Synchronization

- **Database Design:**
    - Centralized schema with interconnected tables for nodes (e.g., Plot, Script, Storyboard).
    - Example: Scene table links to Plot, Script, Settings, and Props tables.
- **Event-Driven Updates:**

    - Implement real-time event propagation using frameworks like WebSocket or Firebase.
    - Example: A "Scene Updated" event triggers updates in all linked components.

2.2 Scalability and Performance Optimization

- **Caching:**
    - Use caching to reduce the load on the database for frequently accessed data.
    - Example: Cache storyboard thumbnails to speed up node transitions.
- **Incremental Updates:**
    - Only refresh modified elements instead of reloading the entire canvas.
- **Error Handling:**
    - Define fail-safes to maintain data integrity during synchronization.

2.3 Integration APIs

- Enable exporting/importing to tools like Final Draft, Celtx, or Trello.
- Allow third-party plugin development to extend functionality.

---

3. Create Sample Workflows

3.1 Workflow 1: Adding a New Scene

1. Open "Plot Node."
2. Drag a new marker into the timeline.

3. Add details (e.g., name, brief description).
4. Automatically:
    - A script template appears in the "Script Node."
    - A blank storyboard thumbnail is generated.
    - A new entry is created in "Settings Node" with placeholders.

3.2 Workflow 2: Modifying a Character Trait

1. Open "Character Node" and update a trait (e.g., "Brave" to "Cautious").
2. Automatically:
    - Linked dialogues in "Script Node" adjust for tone and language.
    - Visual representation updates in "Storyboard Node."
    - Impacted scenes are flagged for user review in "Plot Node."

3.3 Workflow 3: Updating a Setting

1. Open "Settings Node" and modify details (e.g., "Modern Apartment" to "Rustic Cabin").
2. Automatically:
    - Linked storyboard visuals reflect new style notes.
    - Props in "Props Node" update to match the setting.

---

4. Gather User Feedback

4.1 Key Use Cases for Feedback

1. Creating a project from scratch.
2. Revising an existing script or storyboard.
3. Synchronizing multiple changes across nodes.

4. Exporting/importing workflows to external tools.

4.2 Feedback Collection Methods

- **Surveys:** Structured questionnaires focusing on usability, intuitiveness, and feature gaps.
- **Prototype Testing:** Early-stage testing with a small group of users (e.g., screenwriters, directors).
- **Focus Groups:** Sessions discussing specific workflows and challenges faced.
- **Analytics:** Track in-app behavior to identify bottlenecks or underused features.

4.3 Feedback Implementation

- **Prioritize High-Impact Changes:** Address user pain points that significantly improve workflow efficiency.
- **Iterative Updates:** Release incremental updates, validating each change with further feedback.
- **Transparent Communication:** Share release notes and future plans with users to maintain engagement.

Summary and Next Steps

- The wireframes outline user interfaces and their interactive components.
- Technical requirements focus on scalability, real-time synchronization, and seamless integration.
- Sample workflows validate the system's non-sequential editing capabilities.
- User feedback will shape refinements and ensure alignment with real-world needs.

Next Steps:

- Develop low-fidelity prototypes based on wireframes.
- Implement the first set of features for a minimum viable product (MVP).
- Conduct initial user testing to iterate on functionality and design.