3D Studio - Two-Person Collaboration Guide

Team Division

This document outlines how the project work can be divided between two developers for collaborative development.

Developer A: 3D & Animation Specialist

Primary Responsibilities

1. 3D Editor Module

Files:

- `client/src/pages/Editor.tsx` - Main 3D editor layout
- `client/src/components/Viewport.tsx` - Three.js 3D viewport
- `client/src/components/Toolbar.tsx` - Tool selection and controls
- `client/src/components/HierarchyPanel.tsx` - Scene object tree
- `client/src/components/PropertiesPanel.tsx` - Object/material properties
- `client/src/components/Timeline.tsx` - Animation timeline
- `client/src/components/SceneManager.tsx` - Scene save/load
- `client/src/lib/store.ts` - 3D editor state management
- `client/src/lib/history.ts` - Undo/redo system
- `client/src/lib/export.ts` - GLTF export

Features to Implement/Maintain:

- 3D viewport with orbit controls
- Primitive creation (Cube, Sphere, Cylinder, etc.)
- Transform tools (Move, Rotate, Scale)
- Material system with PBR presets
- Lighting system (Point, Directional, Spot, Ambient)

- Keyframe animation
- Camera controls and animation
- Scene hierarchy and parenting
- GLTF/GLB export
- Undo/redo functionality

2. Character Animation Module

Files:

- `client/src/pages/CharacterAnimEditor.tsx` - Character animation layout
- `client/src/components/PoseEditor.tsx` - Bone hierarchy and pose controls
- `client/src/components/NlaEditor.tsx` - NLA timeline with strips
- `client/src/lib/characterAnimStore.ts` - Character animation state

Features to Implement/Maintain:

- Skeleton creation and humanoid preset
- Bone selection and rotation
- Pose library (save/load poses)
- Inverse Kinematics (IK) with CCD solver
- Action editor for animation clips
- NLA editor with blend modes
- Audio sync markers

3. Shared 3D Utilities

Files:

- `client/src/lib/performance.ts` - Performance utilities (debounce, throttle, pooling)

Developer B: Video & Compositing Specialist

Primary Responsibilities

1. Video Editor Module

Files:

- `client/src/pages/VideoEditor.tsx` - Video editor layout

- `client/src/components/VideoTimeline.tsx` - 32-track timeline
- `client/src/components/VideoPreview.tsx` - Video preview player
- `client/src/components/AudioMixer.tsx` - Audio mixing console
- `client/src/components/VideoScopes.tsx` - Waveform, vectorscope, histogram
- `client/src/components/MediaLibrary.tsx` - Media asset management
- `client/src/components/ClipProperties.tsx` - Clip editing panel
- `client/src/lib/videoStore.ts` - Video editor state management

Features to Implement/Maintain:

- 32-track timeline with multiple track types
- Video/audio/image clip handling
- Drag-and-drop media import
- Clip trimming and splitting
- Effect system (16 effect types)
- Effect presets
- Transitions between clips
- Audio mixer with volume, pan, mute, solo
- Video scopes for color analysis
- Keyframe animation for effects

2. Compositor Module

Files:

- `client/src/pages/CompositorEditor.tsx` - Compositor layout
- `client/src/components/NodeEditor.tsx` - Node graph editor
- `client/src/lib/compositorStore.ts` - Compositor state management

Features to Implement/Maintain:

- Node graph canvas with pan/zoom
- Node creation and deletion
- Bezier curve connections
- Keying nodes (Chroma, Luminance, Difference)
- Matte cleanup nodes (Dilate/Erode, Blur, Despill)
- Color correction nodes
- Alpha blending and mix nodes
- Viewer node with backdrop preview

- Node parameter editing

3. AI Integration

Files:

- `client/src/components/AIPanel.tsx` - AI assistant panel
- `server/ai/sceneAI.ts` - AI service for scene generation
- `server/routes.ts` - API routes (AI endpoints)

Features to Implement/Maintain:

- Text-to-scene generation
- Material suggestions
- Animation suggestions
- Scene enhancement
- Chat assistant
- Texture prompt generation

Shared Responsibilities (Both Developers)

Core Infrastructure

Files:

- `client/src/App.tsx` - Main application and routing
- `client/src/index.css` - Global styles and theme
- `shared/schema.ts` - TypeScript types and schemas
- `server/storage.ts` - Storage interface
- `server/index.ts` - Server entry point

UI Components (Shared Library)

Directory: `client/src/components/ui/`

- All Shadcn UI components are shared
- Both developers can use but should coordinate changes

Documentation

- `TECHNICAL_REPORT.md` - Both contribute to their module sections

- `PRESENTATION.md` - Both contribute to their slides
- `replit.md` - Both keep updated

Work Division Summary

Area	Developer A	Developer B
3D Editor	Primary	Support
Character Animation	Primary	Support
Video Editor	Support	Primary
Compositor	Support	Primary
AI Integration	Support	Primary
Shared UI Components	Both	Both
Backend/Storage	Both	Both
Testing	Their modules	Their modules
Documentation	Their modules	Their modules

Feature Breakdown by Sprint

Sprint 1-2: Core Foundation (Both)

- Developer A: 3D viewport, primitive creation, transforms
- Developer B: Video timeline structure, preview player

Sprint 3: Materials & Effects

- Developer A: Material system, PBR presets
- Developer B: Effect system, effect pipeline

Sprint 4: Animation & Clips

- Developer A: Keyframe animation, timeline
- Developer B: Clip handling, trimming, transitions

Sprint 5: Advanced Features

- Developer A: Lighting, camera animation, export
- Developer B: Audio mixer, video scopes, media library

Sprint 6: Character Animation & Compositor

- Developer A: Skeleton system, pose editor, IK
- Developer B: Node editor, keying nodes, color correction

Sprint 7: AI & Polish

- Developer A: Integration testing, performance
- Developer B: AI integration, chat assistant

Sprint 8: Testing & Documentation

- Developer A: 3D/Character module tests, documentation
- Developer B: Video/Compositor tests, documentation

Communication Guidelines

Daily Sync Points

1. **Morning standup:** Share what you're working on today
2. **End of day:** Commit and push all changes

Code Review

- All pull requests reviewed by the other developer
- Focus on your specialty area for thorough reviews

Conflict Resolution

- Shared files (App.tsx, schema.ts): Coordinate before editing
- UI components: Announce changes in advance
- Backend routes: Use separate endpoint prefixes if needed

Git Workflow

```
main
  └── develop
       ├── feature/3d-editor (Developer A)
       ├── feature/character-anim (Developer A)
       ├── feature/video-editor (Developer B)
       ├── feature/compositor (Developer B)
       └── feature/ai-integration (Developer B)
```

File Ownership Quick Reference

Developer A Owns:

```
client/src/pages/Editor.tsx
client/src/pages/CharacterAnimEditor.tsx
client/src/components/Viewport.tsx
client/src/components/Toolbar.tsx
client/src/components/HierarchyPanel.tsx
client/src/components/PropertiesPanel.tsx
client/src/components/Timeline.tsx
client/src/components/SceneManager.tsx
client/src/components/PoseEditor.tsx
client/src/components/NlaEditor.tsx
client/src/lib/store.ts
client/src/lib/characterAnimStore.ts
client/src/lib/history.ts
client/src/lib/export.ts
```

Developer B Owns:

```
client/src/pages/VideoEditor.tsx
client/src/pages/CompositorEditor.tsx
client/src/components/VideoTimeline.tsx
client/src/components/VideoPreview.tsx
client/src/components/AudioMixer.tsx
client/src/components/VideoScopes.tsx
client/src/components/MediaLibrary.tsx
client/src/components/ClipProperties.tsx
client/src/components/NodeEditor.tsx
client/src/components/AIPanel.tsx
client/src/lib/videoStore.ts
client/src/lib/compositorStore.ts
server/ai/sceneAI.ts
```

Shared (Coordinate Changes):

```
client/src/App.tsx
client/src/index.css
shared/schema.ts
server/routes.ts
server/storage.ts
client/src/lib/performance.ts
client/src/lib/queryClient.ts
```

Estimated Effort Distribution

Module	Lines of Code	Developer A	Developer B
3D Editor	~2000	100%	-
Character Animation	~1500	100%	-
Video Editor	~2000	-	100%
Compositor	~1200	-	100%
AI Integration	~800	20%	80%
Shared Infrastructure	~1000	50%	50%
Total	**~8500**	**~45%**	**~55%**

Guide created: December 2024