

Space Management System

基于 React 和 Three.js 构建的现代化空间设计管理系统，提供 2D/3D 可视化编辑功能。

核心功能

- **双模式画布系统**
 - 2D 平面编辑器：基于 Canvas 2D 的精确绘图 (src/components/Canvas/Canvas2D.tsx)
 - 3D 实时预览：Three.js 驱动的空间渲染 (src/components/Canvas/Canvas3D.tsx)
 - 视图同步引擎：通过 Redux store 实现状态双向绑定 (src/store/spaceSlice.ts)
- **智能墙体系统**
 - 自动端点捕捉 (src/utils/geometryUtils.ts#findSnapPoints)
 - 墙体厚度实时计算 (src/types/wall.ts)
 - 相交墙体自动断开处理 (src/utils/wallConnectionManager.ts)
- **状态管理系统**
 - 基于 Redux Toolkit 的原子化状态切片：
 - 认证状态 (src/store/authSlice.ts)
 - 空间数据 (src/store/spaceSlice.ts)
 - 文件操作 (src/store/fileSlice.ts)
 - 命令历史 (src/store/commandSlice.ts)
 - 持久化存储：localStorage 集成 (src/services/fileApi.ts)
- **安全认证体系**
 - JWT 令牌自动刷新机制 (src/services/api.ts)
 - 路由守卫系统 (src/App.tsx#PrivateRoute)
 - 加密敏感操作 (src/utils/commandUtils.ts)

技术架构

```
space-management/  
├── src/  
│   ├── commands/           # 命令模式实现  
│   │   └── elementCommands.ts # 图元操作命令集  
│   ├── components/        # 组件层  
│   │   ├── Auth/          # 认证模块  
│   │   │   ├── Login.tsx  # 登录界面  
│   │   │   └── Register.tsx # 注册界面  
│   │   ├── Canvas/        # 画布核心  
│   │   │   ├── Canvas2D.tsx # 2D 渲染引擎  
│   │   │   ├── Canvas3D.tsx # 3D 渲染引擎  
│   │   │   └── WallDrawing.tsx # 墙体绘制逻辑  
│   └── Elements/          # 图元组件
```

```

├── BaseElement.tsx # 图元基类
├── WallElement.tsx # 墙体组件
├── SeatElement.tsx # 座位组件
├── services/      # 服务层
│   ├── api.ts     # 网络请求封装
│   └── fileApi.ts  # 文件操作接口
├── store/         # 状态管理
│   ├── authSlice.ts # 认证状态管理
│   ├── commandSlice.ts # 命令历史管理
│   └── spaceSlice.ts # 空间数据管理
├── types/         # 类型定义
│   ├── command.ts  # 命令系统类型
│   └── space.ts     # 空间数据类型
└── utils/         # 工具库
    ├── geometryUtils.ts # 几何计算工具
    └── commandUtils.ts # 命令系统核心

```

开发配置

```

// store/index.ts (Redux 配置)
export const store = configureStore({
  reducer: {
    auth: authReducer,
    space: persistReducer(spacePersistConfig, spaceReducer),
    command: commandReducer,
    file: fileReducer
  },
  middleware: (getDefaultMiddleware) =>
    getDefaultMiddleware({
      serializableCheck: {
        ignoredActions: [FLUSH, REHYDRATE, PAUSE, PERSIST, PURGE, REGISTER],
      },
    }).concat(commandMiddleware),
});

// main.tsx (渲染入口)
ReactDOM.createRoot(document.getElementById('root')!).render(
  <React.StrictMode>
    <Provider store={store}>
      <PersistGate loading={null} persistor={persistor}>
        <ConfigProvider locale={zhCN}>
          <App />
        </ConfigProvider>
      </PersistGate>
    </Provider>
  </React.StrictMode>
);

```

关键实现细节

1. 状态持久化机制

```
// src/store/spaceSlice.ts
const spacePersistConfig = {
  key: 'space',
  storage: localStorage,
  whitelist: ['elements', 'viewMode'],
};

export const spaceReducer = persistReducer(
  spacePersistConfig,
  createSlice({ /* ... */ }).reducer
);
```

2. 命令系统架构

```
// src/commands/elementCommands.ts
export const createElementCommand = (element: BaseElement): Command => ({
  execute: (dispatch) => {
    dispatch(addElement(element));
    return { inverse: () => deleteElementCommand(element.id) };
  },
  type: 'ELEMENT_ADD'
});
```

3. 3D 渲染优化

```
// src/components/Canvas/Canvas3D.tsx
useEffect(() => {
  const animate = () => {
    controls.update();
    renderer.render(scene, camera);
    frameId = requestAnimationFrame(animate);
  };
  animate();
  return () => cancelAnimationFrame(frameId);
}, []);
```

开发工作流

```
# 安装依赖 (确保使用 pnpm)
pnpm install

# 开发模式运行
```

```
pnpm dev -- --port 3001

# 生产构建
pnpm build && pnpm preview

# 代码质量检查
pnpm lint

# 类型检查
pnpm typecheck
```

贡献指南

- 功能分支命名: **feat**/[功能简写] 或 **fix**/[问题编号]
- 提交信息格式:

[类型]: [简明描述] (关联问题 #123)

- 详细说明变更内容
- 技术实现细节
- BREAKING CHANGE (如有)

- PR 需包含: 测试用例 + 文档更新 + TypeScript 类型定义

许可协议

Apache-2.0 License | Copyright 2024 Spatial Systems Team