Os **Elementos do Slide (Os Blocos de Construção do Conteúdo) são diferentes tipos de objetos, tais como você descreveu.**

**Vou repassar por cada um e detalhar algumas alterações que serão necessárias, pois para cada tipo de elemento de slide identificado vamos necessitar recriar o mesmo resultado do lado HTML. Acredito que vamos necessitar usar Flex Box**

**Preciso que cada tipo de elemento de slide tenha um modelo a ser seguido, permitindo assim futuramente novas inserções slides.**

**Tipo TextElement - Um trecho de texto – sugiro ter no css estilos pré definidos títulos, subtítulos e parágrafos.**

**ListElement: Uma lista com marcadores ou numerada – precisamos prever em nosso código (html, css e/ou js ) esta situação de "unorderedList" (lista não ordenada) e "numberList" (lista numerada).**

**VideoElement: Um vídeo. – aqui necessitamos de uma alteração. Os vídeos não estarão na pasta assets, mas sim em uma url de um vídeo do Youtube,**

**Com isso teremos que alterar o campo “video” do arquivo data.json, inserindo a url do vídeo no youtube.**

**Outro ponto de atenção é que nossa solução precisará de um player de vídeo.**

**SpacerElement: Usado para layout, para criar um espaço vazio. Este tipo de elemento possui um campo “width”. Precisamos criar uma forma de criar este recurso em nossa nova solução.**

**GridElement: Uma grade ou tabela – este tipo de elemento acredito que mereça uma análise extra, pois ele possui um array de arrays, onde cada array interno representa uma linha e seus itens são o conteúdo das células. Precisamos criar em nossa solução algo que pegue estas informações e recrie a tabela do lado html.**

**GroupElement: Um contêiner para agrupar outros elementos para fins de layout – aqui também precisaremos de análise extra. Ele possui o campo “mode” que pode ser "horizontalGroup" ou "verticalGroup" para empilhar elementos lado a lado ou um sobre o outro e o campo “horizontalAlign / verticalAlign” que controlam o alinhamento do conteúdo dentro do grupo. Uma sugestão é que o container use do lado HTML uma <div> e as configurações de alinhamento usar FlexBox.**

**InfoBoxElement: Um tipo especial de GroupElement que aparece como uma caixa de destaque estilizada – pensei aqui em usar a criação de modal do lado HTML para replicar este recurso. Para os elementos dentro da modal usaremos os recursos já discutidos acima.**

**Precisamos planejar em detalhes como renderizar cada elemento planejado usando como fonte destas configurações o arquivo data.json**

Sobre o **InfoBoxElement e sua afirmação tenho algumas observações.**

**Você afirmou “Usar um modal aqui seria inadequado, pois o JSON sugere que essas caixas são parte do fluxo do conteúdo do slide, não uma sobreposição. Criaremos um componente visualmente distinto.”**

**Porém este recurso deverá funcionar da seguinte forma:**

**O InfoBoxElement é um botão amarelo (veja a imagem) que possui as propriedades: título, posição x e y, layout (vertical ou horizontal), alinhamento vertical (topo, meio ou embaixo) e alinhamento horizontal (esquerda, centro ou direita) e por fim “Preencher 100%” que é um controle do tipo verdadeiro ou falso.**

**A única propriedade que refrete diretamente no botão amarelo do InfoBoxElement é a posição x e y, que define a posição do botão na página, que conforme comprova a imagem, pode fica sobre os demais elementos.**

**As demais propriedades se referem a caixa que é aberta ao clicar neste botão. Por este motivo sugeri usar um modal, mas para a caixa e não para o botão.**

**Replaneje este elemento e se precisar revisite o data.json para entender esta estrutura de forma completa**

**Considerando que você não passou por todas as propriedades dos Elementos do Slide previstos no data.json, farei isso manualmente para recapitular e evitar erros, pois quero replicar de forma mais fiel possível a estrutura do data.json.**

A estrutura de um curso pelo que entendi é formada por:

Nivel 1 - produto

"type": "SlideGroup",

"items": []

"id": 0,

"title": "Nosso Trânsito Multimídia",

"productId": "000001"

Nivel 2 – item do menu

"type": "SlideGroup",

"items": []

"id": 5,

"title": "Legislação de Trânsito"

Nivel 3 – sub item do menu

"type": "SlideGroup",

"items": []

"id": 15,

"title": " Formação do Condutor"

IMPORTANTE – um menu (SlideGroup), pode ter um submenu (SlideGroup), que pode ter outro submenu (SlideGroup) , até chegar no slide propriamente dito.

Nivel 4 – slide

"type": "Slide",

"bookReference": "4",

"id": 61,

"audio": "",

"elements": [ ],

"title": "Pré-requisitos Para Habilitação",

"subtitle": null

Nivel 5 – elemento do slide

É aqui, neste último nível que teremos os diversos tipos de elementos, sendo:

**TIPO TextElement**

{

"type": "TextElement",

"styleName": "title1",

"text": "Pré-requisitos Para "

}

**TIPO GroupElement**

{

"type": "GroupElement",

"elements": [],

"verticalAlign": "middle",

"mode": "horizontalGroup",

"horizontalAlign": "left",

"fillHeight": false

}

**TIPO ListElement**

{

"type": "ListElement",

"styleName": "unorderedList",

"text": "Ser penalmente imputável (que possa ser processado na forma da Lei);\nSaber ler e escrever;\nPossuir documento de identidade;\nPossuir Cadastro de Pessoa Física - CPF.\n",

"startIndex": 1

}

IMPORTANTE: Uma lista com marcadores ou numerada – precisamos prever em nosso código (html, css e/ou js ) esta situação de "unorderedList" (lista não ordenada) e "numberList" (lista numerada).

**TIPO InfoBoxElement**

{

"type": "InfoBoxElement",

"elements": [

{

"type": "TextElement",

"styleName": "paragraph",

"text": "Você não poderá se habilitar ou se reabilitar se estiver respondendo por um delito de trânsito ou com sua habilitação suspensa ou cassada."

}

],

"verticalAlign": "top",

"y": 476,

"mode": "verticalGroup",

"horizontalAlign": "left",

"title": "Info",

"fillHeight": false,

"x": 932

},

**TIPO ImageElement:**

{

"legend": "",

"source": "exames para habilitação.swf",

"width": 0,

"type": "ImageElement",

"height": 0,

"title": "",

"searchText": "avaliação psicológica\nexame de aptidão física e mental\nexame direção veicular\nexame teórico-técnico"

}

**TIPO GridElement**

{

"type": "GridElement",

"alternateRowColor": true,

"columnNumber": 3,

"featureColumn": 1,

"columnSizes": [

2.5,

2.5,

2.5

],

"content": [

["", "TEÓRICA", "PRÁTICA"],

["ACC", " 20 h.", " 5 h."],

["CNH Inicial A", " 45 h.", " 20 h."],

["Adição Cat A", " ---", " 15 h."],

[ "CNH Inicial B", " 45 h.", " 20 h."],

["Adição cat B", " ---", " 15 h."]

],

"isFirstRowHeader": true

}

**TIPO VideoElement**

{

"type": "VideoElement",

"previewImage": "direcao-defensiva-condicoes-do-condutor-via-usuario.jpg",

"title": "Vídeo",

"video": "8.a.TRATAMENTO\_DE\_USUARIOS\_DA\_VIA.flv"

}

IMPORTANTE: a propriedade “vídeo” do tipo **VideoElement** para nossa versão HTML será uma URL de vídeo do Youtube.

**TIPO AppLauncherElement**

{

"path": "flecha-ao-alvo.zip",

"type": "AppLauncherElement"

}

Este tipo de elemento são jogos educacionais. Para nossa versão HTML a propriedade “path” também poderá ser a URL de um jogo em WebGL.

Leve todas estas informações em consideração para construir a solução de renderização renderer.js