

プログラミング問題PCIモジュール ソースコード説明書

BPS株式会社
2022年03月25日



目次

1. 適用	3
2. 概要	3
3. ファイル一覧	3
4. ソースコード説明	4
4.1 プログラムファイル	4
4.1.1 CodeBlockProgrammingInteraction.js	4
4.1.1.1 initialize: function(id, dom, config)	4
4.1.2 Panel.js	4
4.1.3 Rectangle.js	5
4.1.4 ToCode.js	5
4.1.5 Correct.js	5
4.1.6 Answer.js	5
4.1.7 Question.js	5
4.2. 設定ファイル	6
4.2.1 config.js	6
4.3 テンプレートファイル	6
4.3.1 choicePanel.tpl	6
4.3.2 interaction.tpl	6
4.3.3 markup.tpl	6
4.3.4 propertiesForm.tpl	7
4.3.5 rectangle.tpl	7
4.3.6 spacerectangle.tpl	7
4.4 スタイルシート	7
4.4.1 base.css	7
4.5 プロパティファイル	7
4.5.1 pciCreator.js	7
4.5.2 pciCreator.json	7

1. 適用

本書は、教科「情報」の CBT 試験を TAO で実施するための PCI に関する調査研究業務 (プログラミング問題) において開発したプログラミング問題のソースコード説明書に適用します。

2. 概要

PCIとは独自の形式の問題を作成するために必要なインタラクション(問題アイテムをオーサリングするための構成要素)のことです。TAO上ではQTI(標準で用意されている一般的なインタラクション)を使用することで典型的な問題アイテムを作成することができますが、本件での要件である「プログラミング問題(短冊式)」のような問題アイテムを作成するためのインタラクションが標準では用意されていないため、「プログラミング問題(短冊式)」の問題アイテムの作成を可能にするためにPCIの開発が必要となります。

- TAOアイテムのオーサリング画面において、プログラミング問題(短冊式)が作成できます
- CustomInteractions欄に「プログラミング問題(短冊式)」を作成するためのアイコンを追加し、当該アイコンを白紙の問題作成ページにドラッグ・アンド・ドロップすることでプログラミング問題(短冊式)を作成できます
- 短冊を並べて構成したプログラムコードの断片に関して、プログラム構文に応じて自動インデントーションを行います
- 短冊を並べて構成したプログラムに対する入力をまとめて付与して実行できます
- 短冊中の空欄に任意の値や式、条件文等を入力できます
- 短冊の表示ラベルには擬似言語など任意の文字列を設定することができ、短冊ごとに1行あるいは複数行のJavaScriptコードを対応させることができます
- 問題作成時及び出題時にもプログラムの実行結果を確認できます
- TAOの標準に備わっているInteractionsとを組み合わせることで問題を作成できます

3. ファイル一覧

本プログラムには以下のファイルが含まれます。なお、カスタムインタラクションの動作を保証するため、ディレクトリ構造、ファイル名は下記に示す内容の通りとします。

```
/
├── creator
│   ├── Panel.js
│   ├── Rectangle.js
│   ├── ToCode.js
│   ├── img
│   ├── model
│   ├── renderer
│   ├── tpl
│   └── widget
└── Widget.js
```

```

├── choice
├── states
│   ├── Answer.js
│   ├── Choice.js
│   ├── Correct.js
│   ├── Question.js
│   └── states.js
├── config.js
└── utils.js
lib
pciCreator.js
pciCreator.json
└── runtime
    ├── CodeBlockProgrammingInteraction.js
    └── css
        ├── base.css
        └── img

```

4. ソースコード説明

本項では、PCIの動作に必要なとなる主要なプログラムファイルを記載します。

4.1 プログラムファイル

4.1.1 CodeBlockProgrammingInteraction.js

場所: /runtime/CodeBlockProgrammingInteraction.js

内容: Interactionがロードされた時に呼ばれるファイルで、問題作成者・問題解答者共に動作します。また/creator/widget/statesのファイルからアクセス出来るため、ある程度関数を集めることができます。getResponseの戻り値が問題解答時の値となります。

※ このファイルのファイル名は、pciCreator.jsonの「typeIdentifier」およびpciCreator.jsの「_typeIdentifier」の値と一致させる必要があります。

4.1.1.1 initialize: function(id, dom, config)

画面がオーサリングされた時に必ず呼ばれる関数で、短冊形コードのネストや、ドラッグアンドドロップの処理もこの中で実行しています。

また、DOMエレメントを「dom」で継承しているため、jQueryのイベントなどもこの関数の中に記載することができます。

4.1.2 Panel.js

場所: /creator/Panel.js

内容: 問題タブ時の短冊クリックされた際にパネル(右側のプロパティ)を作る処理や、パネルの入力値が変わった際に短冊に反映させる処理を行います

4.1.3 Rectangle.js

場所: /creator/Rectangle.js

内容: 短冊を生成したり、短冊内のテキストボックス・セレクトボックスを表示する処理を行います

4.1.4 ToCode.js

場所: /creator/ToCode.js

内容: 作成された短冊を実際のjavascriptコードに変換する処理を行います

4.1.5 Correct.js

場所: /creator/widget/states/Correct.js

内容: 解答タブを選択した時に呼ばれるファイルです。Interactionの「答え」の部分は、

```
interaction.onPci('responsechange', function (response) {  
    correctItems = JSON.parse(response).result  
    responseDeclaration.setCorrect([response]);  
});
```

この部分にあたり、onPciというTAO標準関数を利用し
responseDeclaration.setCorrect([response]);によって答えが保存されます。

4.1.6 Answer.js

場所: /creator/widget/states/Answer.js

内容: 解答タブを選択した時に呼ばれるファイルです。また、解答タブから離れた時の処理もこのファイルに記載します。

- 解答タブの短冊の状態の復元
- 解答タブで動かされた際の短冊の保存
- その他、onイベントなど動的なイベントが発生した際の処理
- 発生したイベントの解除処理

4.1.7 Question.js

場所: /creator/widget/states/Question.js

内容: 問題タブを選択した時に呼ばれるファイルです。また、問題タブから離れた時の処理もこのファイルに記載します。

- 問題タブの短冊の状態の復元
- 問題タブで動かされた際の短冊の保存

- 短冊をクリックした際に短冊のプロパティを作成する
- その他、onイベントなど動的なイベントが発生した際の処理
- 発生したイベントの解除処理

4.2. 設定ファイル

4.2.1 config.js

場所: /creator/widget/config.js

内容: 設定ファイルとなります。

- inputIdMax: 設定可能な最大短冊数
- inputShowValueMax: 短冊の「表示コード」に記入可能な最大文字数(半角文字を1とする)
- inputExecValueMax: 短冊の「実行コード」に記入可能な最大文字数(半角文字を1とする)
- inputarrowValueMax: 短冊の「入力許可ワード」に記入可能な最大文字数(半角文字を1とする)
- maxLineNum: 短冊の「表示コード」に記入可能な最大行数
- maxWhileNum: Whileの最大繰り返し回数
- inputTextFieldWidth: テキストボックスの横幅の初期値(単位: px)

4.3 テンプレートファイル

テンプレートファイルは、Handlebars.jsテンプレートに基づいたマスタッシュ構文が利用可能となっております。

例)

```
//変数のprint
{{<変数名>}}
{{'表示したい文字'}}
//IF
{{#if}}
{{else}}
{{/if}}
```

4.3.1 choicePanel.tpl

場所: /creator/tpl/choicePanel.tpl

内容: 短冊を選択したときに表示される右側プロパティのテンプレートファイルです。

4.3.2 interaction.tpl

場所: /creator/tpl/interaction.tpl

内容: Interactionの外側のテンプレートファイルです。

4.3.3 markup.tpl

場所 : /creator/tpl/markup.tpl

内容 : Interactionの内側のテンプレートファイルです。

4.3.4 propertiesForm.tpl

場所 : /creator/tpl/propertiesForm.tpl

内容 : 基本プロパティのテンプレートファイルです。

4.3.5 rectangle.tpl

場所 : /creator/tpl/rectangle.tpl

内容 : 短冊のテンプレートファイルです。

4.3.6 spacerectangle.tpl

場所 : /creator/tpl/spacerectangle.tpl

内容 : 空白パーツのテンプレートファイルです。

4.4 スタイルシート

4.4.1 base.css

場所 : /runtime/css/base.css

内容 : 本Interactionで呼び出されるスタイルシートです。

4.5 プロパティファイル

4.5.1 pciCreator.js

場所 : /pciCreator.js

内容 : カスタムPCIの標準関数を返すAMDモジュール群です。必ず読み込みが必要です。

4.5.2 pciCreator.json

場所 : /pciCreator.json

内容 : カスタムPCIのマニフェストファイルです。全ての項目を記入する必要があります。