

react cop2

ono

2023 年 12 月 18 日

目次

1	はじめに	1
1.1	研究背景	1
1.2	研究課題	1
1.3	研究目的	1
1.4	本論文の構成	1
2	関連研究	1
2.1	COP の概要	1
2.2	React COP	1
2.3	EventCJ に複合層を導入	1
3	提案手法	1
3.1	代替する機能	1
(1)	アクティブなレイヤーの一つを取得できる	1
(2)	レイヤー名を指定してレイヤーを取得できる	1
(3)	レイヤー名を指定してレイヤーをアクティブにできる	1
(4)	レイヤー名を指定してレイヤーを非アクティブにできる	1
3.2	改善点	1
(1)	改善点一覧	1
(2)	layer の de/active 時に新しいレイヤーを定義できないようにする	2
(3)	layer params は in/active の両方の状態を持つ	2
(4)	layer params は layer の in/active 状態に依存をするようにしたい	2
(5)	layer params の値を入れるときに新しい layer を定義できないようにする	2
(6)	layer grop 的なものをついか	2
(7)	layer の活性化条件を定義できる	2
(8)	layer の活性化は排反	2
3.3	実装する内容	3
(1)	typescript での実装	3
(2)	テストの追加	3
3.4	評価方法	3
4	実装	3
4.1	layer の de/active 時に新しいレイヤーを定義できないようにする	3
4.2	ts の導入	3
4.3	テストの導入	3
4.4	各種、具体的な実装内容	3
5	評価	3
5.1	できるようになったこと	3
6	まとめ	3
7	参考文献	3

1 はじめに

1.1 研究背景

1.2 研究課題

1.3 研究目的

1.4 本論文の構成

2 関連研究

2.1 COP の概要

2.2 React COP

2.3 EventCJ に複合層を導入

3 提案手法

本研究では、関連研究にある複合層、多層の機能追加 react cop の改善点を洗い出し、それを解決するための機能を追加した。本章では、本研究で提案する手法について述べる。

react cop では、レイヤーの活性化情報を useState を用いて管理している。state 変数はクラスのインスタンスを用いている。しかし、state 変数は参照型のため、state 変数の値を変更しても再レンダリングが行われない。そのため、レイヤーの活性化情報を更新しても、再レンダリングが行われない。これを解決するために、state 変数を参照型のクラスのインスタンスから、イミュータブルなデータに変更する必要がある。この変更は react cop を大きく刷新することと同等であるため、react cop2 として新しく実装することとした。

react cop では、レイヤーのパラメーターと layer の活性化情報を別で管理している。そのため、レイヤーの活性化とレイヤーのパラメーターで 2 度レイヤー名を指定する必要がある。これは、レイヤーの管理が煩雑になる原因となっている。この問題を解決するために、レイヤーを一つのオブジェクトとして管理することとした。

以降では、react cop2 の実装について述べる。react cop で代替となる機能を提供するために、react cop2 では、以下の機能を提供する。react cop から改善された機能を以下に示す。

3.1 代替する機能

- (1) アクティブなレイヤーの一つを取得できる
- (2) レイヤー名を指定してレイヤーを取得できる
- (3) レイヤー名を指定してレイヤーをアクティブにできる
- (4) レイヤー名を指定してレイヤーを非アクティブにできる

3.2 改善点

(1) 改善点一覧

- layer の de/active 時に新しいレイヤーを定義できないようにする
- layer params は in/active の両方の状態を持つ
- layer params は layer の in/active 状態に依存をするようにしたい

- layer params の値を入れるときに新しい layer を定義できないようにする
- layer grop 的なをついか
- layer の活性化条件を定義できる
 - 複合層
 - 多層
- layer の活性化は排反
- typescript での実装
- テストの追加

(2) layer の de/active 時に新しいレイヤーを定義できないようにする

react cop では、layer の de/active 時に新しいレイヤーを定義できてしまう。layer の de/active 時に新しいレイヤーを定義できると意図しないレイヤーが簡単に定義できてしまう。またレイヤーの管理が煩雑になる。

react cop2 では、layer の de/active 時に新しいレイヤーを定義できないようにする。これによって、意図しないレイヤーが簡単に定義できなくなり、レイヤーの管理が煩雑にならない。

- (3) layer params は in/active の両方の状態を持つ
- (4) layer params は layer の in/active 状態に依存をするようにしたい
- (5) layer params の値を入れるときに新しい layer を定義できないようにする
- (6) layer grop 的なをついか
- (7) layer の活性化条件を定義できる
- (8) layer の活性化は排反

ソースコード 1: hoge

```

1      // このとき、Float と Integer のレイヤの活性化は排反でよい気がする
2      const [getHoge, setHoge] = useLayerParams('', ["Float", "Integer"]);
3
4      // いちいち切り替えがめんどろ
5      layerManager.deactivateLayer("Integer");
6      layerManager.activateLayer("Float");
7
8
9      // 排反ではないから 2 つ条件含むのどうなん?
10     <Layer condition={layerState.Float && !layerState.Integer}>
```

3.3 実装する内容

- (1) typescript での実装
- (2) テストの追加

3.4 評価方法

- 実装前と後で、できることの違いを比較する。

4 実装

4.1 layer の de/active 時に新しいレイヤーを定義できないようにする

コードは以下ようになる。レイヤーの名前が存在するかどうかを確認し、存在しない場合はエラーを出すようにしている。

4.2 ts の導入

4.3 テストの導入

4.4 各種、具体的な実装内容

5 評価

本章では、提案手法の評価を行う。

5.1 できるようになったこと

6 まとめ

7 参考文献