Pentago: Revolutionary Design with Big Data & Edge Computing

Owen Mastropietro
mastropi@sonoma.edu

Sonoma State University, CS470 Advanced Software Development

Template

This template is/was merely used in the planning stage of development and is bound to change.

Board

A component that represents the entire game (board), including the rotation options, in the browser.

Data Structures

None

State

const [hasWinner, setHasWinner] = **useState**(() => false);
const [turn, setTurn] = **useState**(() => ‘white’);

Child Components:

Single instance of *Top Banner*
Single instance of *Quadrants*
Single instance of *Rotators*

Props

None

Callback Function

None

Top Banner

A component within the game *Board* to display the “Pentago” title and conditionally render a victory notification (i.e. “black wins”).

Data Structures

None

State

None

Child Components

None

Props

{ turn, hasWinner }

Callback Function

None

Quadrants

A component within the Board that represents the four (3x3) quadrants within a Pentago game board.

Data Structures

A three dimensional array to represent the play area and a two dimensional array to represent the equivalent board for convenient validation of goal states/winners.

State

const [quadrants, setQuadrants] = useState(() => emptyQuadrants());
const [board, setBoard] = useState(() => null);
~~const [hasWinner, setHasWinner] = useState(() => false);~~
~~const [turn, setTurn] = useState(() => 'white');~~

Child Components

Single instance of *Quadrant* for each of the FOUR quadrants in a Pentago game board.

Props

{ hasWinner, turn }

Callback Function

Let the callback function, *onClickCallback*, be used when the components that represent elements in the four quadrants - cells - are clicked.
This callback function can be expected to, at least, update *Quadrants* and *Board*.
It can be expected to, at most, update *Quadrants*, *Board*, *hasWinner*, and *turn*.

Let the callback function, *onRotateCallback*, be used when the components that represent the rotators are clicked.
This function can be expected to, at least, update *Quadrants*, *Board*, and *turn*.
It can be expected to, at most, update *Quadrants*, *Board*, *turn*, and *hasWinner*.

Quadrant

A component within the Quadrants that represents ONE out of the FOUR (3x3) quadrants that make up a Pentago game board.

Data Structures

None

State

None

Child Components

Nine instances of *Cell* for a given *Quadrant* of a Pentago game *Board*.

Props

{ quadrant, onClickCallback }

Callback Function

This component has access to the *onClickCallback* function provided by - and described in - *Quadrants*.

Cell

A component within a Quadrant that represents a single, clickable, element of the game board as it is displayed in the browser.

Data Structures

None

State

None

Child Components

None

Props

{ cellColor }

Callback Function

None

Rotators

A component within the *Board* that represents the various options for rotating each of the Quadrant's.

Data Structures

A two dimensional array of objects to represent the TWO options, CLOCKWISE and COUNTER_CLOCKWISE, for each of the FOUR quadrants.

State

None

Child Components

None

Note: I'm not sure if I should include a Rotator component to mirror the Quadrants :: Quadrant structure or if it's fine to just work with Rotators.

Props

{ quadrant, onRotateCallback }

Callback Function

This component has access to the *onRotateCallback* function provided by - and described in - *Quadrants*.