Web Frameworks

Laboratory 04

mgr Sara Jurczyk

React – state

Let's develop the project from our last class. We start creating a game library application.

Let's add a new file moreInfo.js to the src directory and prepare a component that displays data conditionally:

```
import React, {Component} from 'react'

class MoreInfo extends Component{
    render() {
        var infoMessage = 'Lorem ipsum ...'
        const isShown = true
        return <div>
            <h2>About: </h2>
            {isShown?<p>{infoMessage}</p>:'More info about...'}
        </div>
    }
}

export default MoreInfo;
```

After placing it on the website (remeber to add it to the App component), we can see that it is rendered only once while we create a component. To make the content appear and hide on the page, we need a component state and also some methods to change it. Class components by default can hold a state, but functional components by default are stateless. State is hold in a special value provided by React.Component by which we extend a class.

Let's modify the MoreInfo component so that we can actually hold a state and change it. Now the message is hidden by deafault, and we would like to show it after we prompt the change:

```
import React, {Component} from 'react'

class MoreInfo extends Component{
    state = {
        showInfo: false
    }
    changeShowInfoState = () => {
        this.setState({showInfo: true})
    }

    render() {
        var infoMessage = 'Lorem ipsum ...'
        return <div>
            <h2>About: </h2>
            {this.state.showInfo?<p>{infoMessage}</p>:'More info about...'}
        </div>
    }
}

export default MoreInfo;
```

We have to also add a button with an onClick method that allows to make the change:

```
import React, {Component} from 'react'

class MoreInfo extends Component{
    state = {
        showInfo: false
    }
    changeShowInfoState = () => {
        this.setState({showInfo: true})
    }

    render() {
        var infoMessage = 'Lorem ipsum ...'
        return <div>
            <h2>About: </h2>
            {this.state.showInfo?<p>{infoMessage}</p>:
                <button onClick={this.changeShowInfoState}>More info
about...</button>}
        </div>
    }
}

export default MoreInfo;
```

After we save changes, the message should be visible on the page only after we click on the button.

EXERCISE:

Let's modify the component above, so that one can constantly change the state and show or hide the message. The button should now be visible all the time, and has a label „Show/Hide info". On the page the is either the message or „More info about…" text.

Hint: the method can be modified like in the example below

```
changeShowInfoState = () => {
    this.setState({showInfo: !this.state.showInfo})
}
```

Let's add now the second component: a list of favourite game titles that is shown or hidden using a button. Example solution:

```
import React, {Component} from 'react'

class FavouriteGamesList extends Component{
    state = {
        isListShown: false
    }
    changeIsShownFavListState = () => {
        this.setState({isListShown: !this.state.isListShown})
    }

    render() {
        var games = ['CS:GO','World of Warcraft','Osu']
        const gameList = games.map(game => <li key={game}>{game}</li>)
        return <div>
            <h2>Favourite games list: </h2>
            {this.state.isListShown? <div>
                    <button onClick={this.changeIsShownFavListState}>
                        Hide games list</button>
                    <ul>{gameList}</ul></div>:
                <button onClick={this.changeIsShownFavListState}>
                    Show games list</button>}
        </div>
    }
}

export default FavouriteGamesList;
```

Exercises:

1. Create a button that after clicking on it, displays a link to some game website.
2. Create a component with header „Click button to subscribe" and a button that allows to change it to „Now you subscribe".
3. Add to the website a section in which you display „Log in" button or „Log out" button (as one visits the site, he is log out by deafault).