Rendering Tab Panels

Right now the tab bar only shows the tabs themselves, but no content. We really need to show a different component as the content for each tab, and swap which component is visible as the active tab changes.

One way to do this would be to actually change which component is being rendered for the content panel each time the `currentTab` prop changes. Another would be to always render all the tabs, but toggle the visibility instead of un-rendering them. We’re going to go with toggling visibility, just because I feel like it.

There’s some existing libraries out there like `react-toggle-display` which implement this functionality already, but it’s pretty simple to do. So, we’ll do it ourselves.

> **Commit b7ce9e5: Add a ToggleDisplay component**

common/components/ToggleDisplay.jsx

```
import React from "react";

const ToggleDisplay = ({show, children}) => {
    const displayStyle = {
        display : show ? undefined : "none"
    };

    return (
        <span style={displayStyle}>
            {children}
        </span>
    );
}

export default ToggleDisplay
export default ToggleDisplay;
```

After adding that to the project, we need to update `TabBar` to look for components in the tab definitions, and render those wrapped in a `<ToggleDisplay>`. We also need to add some initial dummy components to our existing tab definitions:

> **Commit 823e9c5: Add ability to swap visible tab component based on active tab**

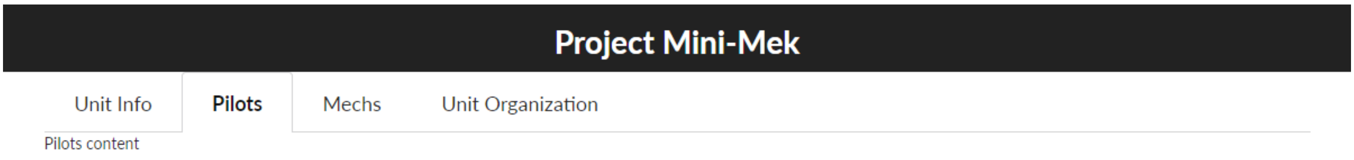
features/tabs/TabBar.jsx

```
import React from "react";
import {Menu} from "semantic-ui-react";
+
+import ToggleDisplay from "../../common/components/ToggleDisplay"

import Tab from "./Tab";

const TabBar = (props) => {
    const {tabs, currentTab, onTabClick, ...otherProps} = props;

    const tabItems = tabs.map(tabInfo => {
        const {name, label} = tabInfo;

        return (
            <Tab
                key={name}
                name={name}
                label={label}
                active={currentTab === name}
                onClick={onTabClick}
            />
        );
    });

+   const tabPanels = tabs.map(tabInfo => {
+       const {name, component : TabComponent} = tabInfo;
+
+       return (
+           <ToggleDisplay show={name === currentTab} key={name}>
+               <TabComponent />
+           </ToggleDisplay>
+           </ToggleDisplay>
+       )
+   })

    return (
        <div>
            <Menu tabular attached="top" {...otherProps}>
                {tabItems}
            </Menu>
+           {tabPanels}
        </div>
    )
}

export default TabBar;
```

App.jsx

```

+const UnitInfo = () => <div>Unit Info content</div>;

+const Pilots = () => <div>Pilots content</div>;

+const Mechs = () => <div>Mechs content</div>;

+const UnitOrganization = () => <div>Unit Organization content</div>;

 class App extends Component {
     render() {
         const tabs = [
+           {name : "unitInfo", label : "Unit Info", component : UnitInf
o,},
+           {name : "pilots", label : "Pilots", component : Pilots,},
+           {name : "mechs", label : "Mechs", component : Mechs,},
+           {name : "unitOrganization", label : "Unit Organization", compo
nent : UnitOrganization}
         ];

         // ....
     }
}
```

Now, as we click between tabs, we can see the visible panel change:



Now that we’ve got the tabs working, it’s about time to start filling out the content for those tabs, and that means new components and new files. But, we’re going to take a detour first. Creating new files means deciding where to put where to put them, and that’s a topic in and of itself.