## **Data Synchronization**

```
const State = {
    room: value
    info: value
    adjRooms: value
```

```
Initialization: connect();
```

- connect(Buttons)
- connect(HouseInfoPanel)

Wrapper::Buttons Buttons

Global State variable is passed down to the Wrapper parent class of both Buttons and HouseInfoPanel and local state is respectively set

Wrapper::HouseInfoPanel

```
Class Wrapper {
  state = {
    room: (value)
    info: (value)
    adjRooms: (value)
  }
}
```

```
Class Wrapper {
  state = {
    room: (value)
    info: (value)
    adjRooms: (value)
  }
}
```

```
* When Buttons and HIP
mount, the
componentDidMount()
call sends the value of
listener which is a call to
setState into the listeners
set, which is waiting on
standy by for a button
click a.ka.
changeRoomButton()
                  Wrapper
```

class Buttons {

Buttons as props

## Data Synchronization (cont)

```
const listeners = new Set()
componentDidMount() {
      listener = () => this.setState({etc...})
```

\* Wrapper parent passes its state to

class HouseInfoPanel { \* Wrapper parent passes its state to HIP as props

Wrapper

## Data Synchronization (cont)

changeRoom(roomSelection)

- 1. Global State is changed to the selection
- 2. Environment background is updated
- 3. updateComponents is triggered

Wrapper of both Buttons and HIP gets its local state changed by the cb() function in updateComponents. Afterwards, the new local state is passed to the children from Wrapper as props and then the cycle waits until changeRoom is triggered again.