Renz Ian P. Parreño IT ELECTIVE 3

BSIT32-E2 Sr. Alamo

**Create a sample code snippet that demonstrates passing a callback function from a parent component down to its child(ren) component.**

import React, { Component } from 'react';

import Zombie from '../Zombie';

const Humbug = {

name: 'Humbug',

level: 5,

hp: 20

}

class GameMgr extends Component {

constructor(props) {

super(props);

}

render() {

return(

<div>

<Zombie name={"Geek"} level={5} hp={21}></Zombie>

<Zombie {...Humbug}></Zombie>

</div>

)

}

}

export default GameMgr;

**Explain props-drilling.**

Each zombie battle includes two opposing zombies chosen by their respective trainers. The zombie with the normal status and the faster pace is the first to attack (of course there are other factors to consider, but we're sticking to the basics for the sake of simplicity). Whenever the hit points (HP) of a zombie drop to 0, the zombie faints and the opposing zombie is declared the winner. On the other hand, in order for a zombie to attack, its trainer must pick one of the moves that the zombie learned earlier.

Back to React and its components. Passing values from a parent component to a child component is simple; we only have to pass the values as props of the child element. To illustrate this concept, take a look at the following code. This code is equivalent to the zombie selection process performed by the trainers. In this battle, we are sending a Humbug and a Geek to fight.

It can clearly see that the information regarding Geek is passed to the first instance of the Zombie component that is in the render function of the GameMgr component. The props are initiated separately and in succession. The information regarding Humbug, however, is passed by means of a JavaScript object (defined as a constant after import) through the spread operator (ES6). This way, every key-value pair of this object is addressed as a prop of the child component.