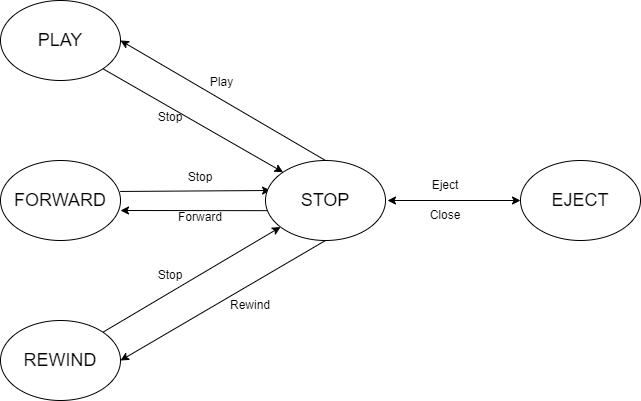
**IE403 – Human-Computer Interaction**

**Programming interaction**

**201901100 – Shubham Patel**

**State Transition Diagram and code for Walkman’s play, pause, rewind, forward buttons.**

* **State transition diagram**

****

* **Code**

import java.awt.\*;

import java.awt.event.\*;

const styles = {

container: {

margin: '10px',

width: '400px',

backgroundColor: 'grey',

padding: '20px',

borderRadius: '8px',

border: '2px solid darkgrey',

},

state: {

marginBottom: '20px',

textAlign: 'center',

fontSize: '24px',

fontWeight: 'bold',

color: 'yellow',

},

buttons: {

display: 'flex',

justifyContent: 'space-around',

},

button: {

padding: '15px 25px',

fontSize: '24px',

textAlign: 'center',

cursor: 'pointer',

outline: 'none',

color: '#fff',

backgroundColor: 'lightGrey',

border: 'none',

borderRadius: '8px',

boxShadow: '0 9px #999',

},

active: {

backgroundColor: 'darkgrey',

boxShadow: '0 5px #666',

transform: 'translateY(4px)',

}

}

class WalkmanControls extends React.Component {

get state() {

return this.props.model.active\_states[0].name;

}

emit(event) {

this.props.model.emit(event)

}

render() {

const Button = ({ children, event, active }) => {

const style = Object.assign({}, styles.button, this.state === active && styles.active)

return (

<button

style={style}

onClick={() => this.emit(event)}>

{children}

</button>

)

}

return (

<div style={styles.container}>

<div style={styles.state}>{this.state}</div>

<div style={styles.buttons}>

<Button event='STOP' active='Stopped'>STOP</Button>

<Button event='PLAY' active='Playing'>PLAY</Button>

<Button event='REWIND' active='Rewinding'>REWIND</Button>

<Button event='FAST\_FORWARD' active='FastForwarding'>FROWARD</Button>

</div>

</div>

)

}

}

function render(model){

return <WalkmanControls model={model} />

}

* **Prototype**

****