**philbert — Yesterday at 8:34 PM**

three: {

text: 'Are you free to help with some more stuff? \nWe can use all the help we can get.',

options: [

{ text: 'Yes', event: 'mine', goto: 'four', origin: 'four' },

{ text: 'No' },

],

},

const [mineActive, setMineActive] = useState(false)

<Dialog

schema={dialog}

onView={setView}

onEvent={event => {

if (event === 'mine') {

setMineActive(true)

}

}}

>

<model

onClick={mineActive ? doClick() : null}

>

{visible && (

<>

<model src/>

<model src/>

</>

)}

      <Erika position={[0, 0, 0]} armorPosition={[-82, 25, 83]} swordPosition={[140,11,-588]}/>

      <Dialog

        position={position}

        schema={schema}

        onRequire={name => {

          if (name === 'armor') return hasArmor

        }}

        onView={setView}

        onEvent={(event, setView) => {

          if (event === 'complete') {

            setGivenArmor(true)

          }

          // If you wanted to you could run async stuff like checking a wallet

          // here in response to an event, and then call setView(String) to continue

          // the conversation.

        }}

      >

      </Dialog>

      {!hasArmor && (

        <model

          src="armor.glb"

          position={armorPosition}

          onClick={() => setHasArmor(true)}

        />

      )}

Graphical user interface

Description automatically generated

Graphical user interface, application

Description automatically generatedDiagram

Description automatically generated with medium confidence

Graphical user interface, application

Description automatically generatedGraphical user interface

Description automatically generated