

Interfaz

Crear

- Grabar h-state (5 angles, handness, anchor) / seleccionar h-state
- asociar comando (conjunto de instrucciones) / crear instruccion
- Poder probar activacion de h-state

Eliminar

- Eliminar Gesto (h-state y comando asociado)

--

En el main loop, crear modo personalizado: El main loop se trata del ciclo de vida del main process Nos comunicamos por el a travez de la api expuesta a los renderers

```
ipcMain.handle("saveMappings", async (event, newMappings) => {
  try {
    await saveMappings(newMappings);
    return { success: true };
  } catch (error) {
    return { success: false, error: error.message };
  }
}
```

});

saveMappings: (newMappings) => ipcRenderer.invoke("saveMappings", newMappings),

Crear sistema de guardado y retribucion de gestos personalizados Tenemos un sistema de guardado y lectura de json que podria guardar los h-state:

Si se van a guardar los h-states como json, las instrucciones como texto, nombreGesto, {hstate, comando [...instrucciones]}, metadata{allow debouncing, slackIndex, }

```
const saveMappings = (mappings) => { try { ensureDirectoryExists();
const jsonObject = Object.fromEntries(mappings);
fs.writeFileSync(mappingsPath, JSON.stringify(jsonObject, null, 2),
"utf8"); console.log("Stored successfully in:", mappingsPath); } catch
(error) { console.log("Error saving mappings", error); } };
```

Forma de evaluar Gesto: if ResemblanceIndex <= slackIndex then return true; ResemblanceIndex

Forma de evaluar Ejecucion: idea.- No realizar accion hasta que se deje de hugar

data: hugging(t,f)

NO tenemos una forma de guardar una funcion personalizada por nombre?: Existe mapeo:

Descripcion del sistema: Usaremos json para obtener el h-state Usaremos