Keyboard Overlay (Java 21) – Maven Project

Description: A JavaFX-based keyboard remapper + WASD-to-mouse mover with virtual DPI and HUD. Target: Windows, Java 21, IntelliJ IDEA.

Run (Development)

1. Open project in IntelliJ → Import as Maven project.
2. Run with **admin privileges** (recommended for `SendInput` reliability).
3. From terminal:

   ```
   mvn clean javafx:run
   ```

Build a Runnable JAR

```
mvn clean package
```

- The assembled JAR with dependencies will be in: `target/keyboard-overlay-1.0-SNAPSHOT-jar-with-dependencies.jar`

Notes

- Use conservative DPI values when testing.
- If JNA `SendInput` structures produce compile issues:
- Update dependency version or report error to adjust the wrapper.

Team Members

- Minh Thành
- Nguyễn Hoàng Vinh
- Quang Hải
- Sơn Tùng
- Tuấn Minh

Global Shortcuts

All non-numeric controls have editable global-shortcut fields in `Settings → Global Shortcuts`.

Action	Shortcut Field Name
Toggle macros on/off	`toggleEnabled`

Action	Shortcut Field Name
Toggle lock overlay show/hide	`toggleOverlay`
Open settings overlay	`openSettings`
Minimize/hide overlay	`minimizeOverlay`
Toggle the app	`toggleApp`
DPI increment / decrement	`dpiInc`, `dpiDec`
IPS increment / decrement	`ipsInc`, `ipsDec`

- Shortcut format: `CTRL+ALT+K`, `CTRL+SHIFT+X`, `F9`, `CTRL+ALT+UP`, etc.
- Press **Save Shortcuts** to persist changes in `%APPDATA%/KeyRemap/settings.json`.
- `MacroManager` immediately reloads shortcuts.

Assigned Inputs (Settings Section)

- Shows current key bindings: `Key → Action`
- Buttons:
- **Refresh Assigned Inputs** → reload current bindings
- **Edit Selected Binding** → edit quickly without leaving overlay
- Safe read/edit area; no accidental overwrite.

Behavior / How It Works

1. Edit text fields → click **Save Shortcuts** → updates JSON → reloads `MacroManager`.
2. Pressing a configured global shortcut triggers mapped action (with debounce).
3. `toggleApp` → alias for enable/disable toggle.
4. Assigned Inputs list populated from `manager.getBindings()`.

Numeric Controls

- DPI, IPS, Sprint multiplier remain numeric spinners.
- Increment/decrement shortcuts: DPI `+100/-100`, IPS `+0.5/-0.5`.
- Step sizes can be configurable.

Files Changed (High Level)

File	Changes
`MainApp.java`	Settings overlay extended → extra shortcut for `toggleApp` + Assigned Inputs ListView with refresh/edit buttons
`MacroManager.java`	Shortcut parser extended to read `toggleApp`; `performShortcutAction` handles `toggleApp`

File	Changes
`Settings.java`	Added `toggleAppShortcut` default
`ShortcutHandler.java`	Added `toggleApp()` method; `MainApp` implements it

Next Suggestions / Improvements

1. Add record button for each global-shortcut field.
2. Make DPI / IPS increment step sizes configurable from the UI.
3. Add on-screen toast confirmation when global shortcut runs.
4. Assigned Inputs enhancements:
5. Show native key icons
6. Enable drag-to-edit for faster workflow