# Project Development Guide

**Project:** DriftOS — Controller-as-Mouse n Keyboard for Windows 10/11  
**Repository:** https://github.com/josepraveenkaruppiah/DriftOS   
**Status:** Draft (MVP phase)  
**Owner:** Jose Praveen Karuppiah  
**Last updated:** 06 Sep 2025

## 1) Vision & Scope

**Vision:** Make desktop navigation comfortable with a game controller, no keyboard/mouse required for casual use.

**MVP Scope:** - Reliable cursor movement via left stick (sensitivity, acceleration, deadzone). - Click/drag mapping with A/B (rebindable later). - Scroll via right stick or triggers. - Tray app with enable/disable, profile switch, quit. - Local settings persisted to JSON.

**Out of Scope (for MVP):** macros, per-app profiles, cloud sync, telemetry.

## 2) Users & Stories

**Personas** - **Couch User:** Wants to control the PC from the sofa. - **Streamer:** Needs quick, silent pointer control between scenes. - **Accessibility-Seeking User:** Finds controller easier than trackpad.

**User Stories (MVP)** - *As a couch user,* I can toggle controller-as-mouse from a tray icon so I can quickly disable it when gaming.  
- *As a streamer,* I can adjust cursor speed so small stick motions make fine movements on high‑DPI screens.  
- *As an accessibility user,* I can hold a button to drag windows without accidental releases.

**Acceptance Criteria Examples** - Cursor moves smoothly at 60–120 Hz polling with adjustable deadzone (0–0.3) and max speed (DPI scaled).  
- Left click = A/X; Right click = B/Y; hold for drag; all remappable in future release.  
- Tray icon shows active/inactive state and allows quit.

## 3) Architecture

**Overview** - **UI Layer:** WinUI 3 tray + settings window.  
- **Input Layer:** XInput polling (native) → normalised axes with deadzone & acceleration curves.  
- **Action Mapper:** Maps input events → OS actions (click, move, scroll, drag).  
- **System Adapter:** Win32 SendInput for synthetic mouse; registry‑safe (no drivers).  
- **Config:** JSON profiles (global for MVP; per‑app later).

**Key Decisions** - WinUI 3 > WPF for modern UX; fallback allowed. - XInput first (broad support). Raw Input/GameInput later. - Privacy‑first: no network calls in MVP; telemetry opt‑in post‑MVP.

**Data Flow (simplified)**

Controller → XInput Poller → Normaliser (deadzone/accel) → Mapper → SendInput → Windows Pointer

## 4) Tech Stack

* **Runtime:** .NET 8 (C# 12)
* **UI:** WinUI 3 (Windows App SDK)
* **Input:** XInput
* **System:** Win32 SendInput
* **Testing:** xUnit + FluentAssertions
* **CI:** GitHub Actions (Windows), build & test
* **Packaging:** MSIX (preferred) and portable zip (later)

## 5) Repository Layout

src/ # app projects (UI + input libs)  
 DriftOS.App/ # WinUI 3 tray + settings  
 DriftOS.Input/ # XInput poller + normalisation  
 DriftOS.Core/ # mapping, config, services  
  
tests/ # unit/integration  
 DriftOS.Tests/  
  
docs/ # diagrams, QA checklists  
.github/  
 workflows/ci.yml # build & test  
 ISSUE\_TEMPLATE/ # bug/feature forms  
 PULL\_REQUEST\_TEMPLATE.md

## 6) Development Workflow

**Branching**: Trunk‑based. main always releasable.  
**Branches**: feat/<topic>, fix/<bug>, chore/<task>  
**Commits**: Conventional Commits (feat:, fix:, docs:, test:, refactor:)  
**PRs**: Small, linked to Issues, include a **Test Plan** and screenshots/GIFs for UI.

**Definition of Done (MVP)** - Code builds locally; tests added/updated & passing.  
- No obvious regressions in manual QA checklist.  
- Docs updated (README/CHANGELOG as needed).  
- Reviewer approved; CI green.

## 7) Coding Standards

* Nullable enabled; treat warnings as errors where feasible.
* Keep methods small; prefer DI for testability.
* Avoid static state; isolate SendInput calls behind an interface.
* Use .editorconfig & analyzers; format on save.

## 8) Testing Strategy

**Unit Tests** - Deadzone & acceleration curve maths.  
- Mapping: button → click/drag; stick → velocity.

**Integration** - Fake input provider → verify mapper outputs expected SendInput calls.

**Manual QA (docs/qa-checklist.md)** - Cursor smoothness on 60/120/144 Hz displays.  
- DPI scaling checks (100–200%).  
- Drag reliability over 30s hold.  
- Tray toggle works during active drag (should cancel safely).

## 9) CI/CD

**CI (per PR & main):** Restore, Build (Release), Test on windows-latest.  
**Release (on tag):** Publish win-x64 artifacts; attach to GitHub Release.  
**Versioning:** SemVer (v0.1.0 for MVP).  
**Changelog:** Keep a Changelog format.

## 10) Packaging & Distribution

* **MSIX**: default; user‑scope install; signed when possible.
* **Portable zip**: Squirrel.Windows or plain zip as a fallback.
* Provide SHA256 hashes in Releases.

## 11) Security & Privacy

* No network access in MVP; all data local.
* Security contact: josepraveenk@gmail.com.
* If telemetry added later: explicit opt‑in, local anonymisation, documented data schema.

## 12) Risks & Mitigations

* **Anti‑cheat conflicts** (overlay/SendInput): document safe‑mode and exclusions.
* **Controller variance** (drift, deadzones): calibration wizard post‑MVP.
* **High‑DPI quirks**: test on multiple scales; clamp velocity.

## 13) Milestones & Roadmap

| Version | Target | Highlights |
| --- | --- | --- |
| v0.1.0 | MVP | Cursor move, click/drag, scroll, tray, JSON config |
| v0.2.0 | Rebinds | Key remapping UI, multiple profiles, per‑app sensitivity |
| v0.3.0 | UX polish | Accel editor, onboarding, startup with Windows |
| v1.0.0 | Stable | Signed installer, docs, accessibility passes |

## 14) Task Backlog (Seed)

* Input: implement XInput polling at 120 Hz with cancellation token.
* Maths: circular deadzone + exponential accel curve.
* Mapper: hold‑to‑drag with threshold; scroll via triggers.
* Tray: context menu (enable/disable, settings, quit).
* Settings: load/save JSON; basic UI sliders.

## 15) Contribution Guide (Summary)

* Discuss via Issue; claim with assignee.
* Fork/branch; keep PRs < ~300 lines where possible.
* Add tests for logic changes; update docs.
* Be constructive and respectful.

## 16) Decision Log (Rationales)

* **WinUI 3** chosen for modern Windows UX and future support.
* **XInput first** for broad compatibility; Raw Input/GameInput later.
* **No telemetry** in MVP to prioritise privacy and simplicity.

## 17) Appendix

**Build from source**

# Windows 10/11, .NET 8 SDK, VS 2022  
git clone https://github.com/<your-username>/driftos  
cd driftos  
 dotnet build  
# optional: dotnet test  
# optional: dotnet publish -c Release -r win-x64

**PR Template (short)**

## Summary  
## Screenshots / GIF (if UI)  
## Test Plan  
- [ ] Built locally  
- [ ] Unit tests updated/passing  
- [ ] Manual QA done  
## Related Issues  
Closes #...

**Issue Templates**: Bug report & feature request exist under .github/ISSUE\_TEMPLATE.

Copy this file to PROJECT\_DEVELOPMENT.md at the root of your repository. Update repo URL and any details that change (tech stack, owner, roadmap dates).