



InputScope

banc de test MonoGame pour manettes et contrôleurs MIDI

CFPT Ecole informatique
stephane.garchery@edu.ge.ch

Table des matières

1 Contexte	1
2 Objectifs	1
3 Livrables	1
4 Contraintes techniques (communes à tous les périphériques)	2
5 Cahier des charges fonctionnels de base	2
5.1 Variante Joystick/Manette	2
5.2 Variante Contrôleur MIDI	3
6 Plan de réalisation suggéré	3
7 Manuel utilisateur (plan attendu, 4–8 pages)	3
8 Fiche des capacités (1–2 pages)	3
9 Tests unitaires (obligatoires)	4
9.1 Portée minimale	4
9.2 Exigences	4
10 Qualité et bonnes pratiques	4
11 Ressources	4
12 Exemple fiche des capacités du contrôleur	5

1 Contexte

Chaque binôme reçoit **un ou plusieurs périphériques physiques différents** (manette/joystick ou contrôleur MIDI).

L'objectif est de livrer une **application de démonstration simple** MonoGame qui affiche en temps réel l'état de **tous** les interrupteurs/axes/contrôleurs du périphérique, accompagnée :

- d'un **manuel utilisateur**,
- d'un **inventaire des capacités**,
- d'un **export PDF du code source**,
- et d'une **suite de tests unitaires** vérifiant le comportement de la logique métier.

2 Objectifs

- ☐ Interroger un périphérique d'entrée et traiter ses événements dans une boucle de jeu MonoGame (GamePad/Naudio).
- ☐ Concevoir une UI de diagnostic lisible (valeurs en temps réel, historique, calibration).
- ☐ Structurer un projet propre et testable (couches, dépendances. . .).
- ☐ Rédiger une documentation utilisateur exploitable.
- ☐ Mettre en place et exécuter des **tests unitaires** (logique d'acquisition, normalisation, mappage).
- ☐ Automatiser un **export PDF du code**.

3 Livrables

- 1 Application de démonstration** MonoGame exécutable (Windows).

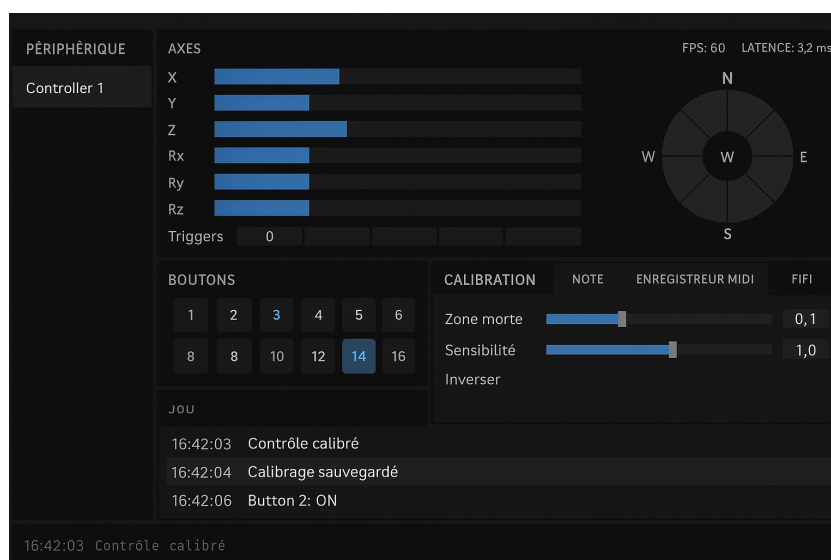


FIGURE 1 – Exemple d'application de test

- 2 Code source complet (dépôt) + export PDF des sources (tous les fichiers).
- 3 Manuel pour développeur (PDF, 4–8 pages).
- 4 Fiche des capacités du périphérique (PDF, 1–2 pages).
- 5 Suite de tests unitaires exécutable + instructions d'exécution.
- 6 Journal de tests (PDF court ou section README) : cas testés, résultats, limites.
- 7 README à la racine : présentation, prérequis, build, lancement, options, *commandes de test*...

4 Contraintes techniques (communes à tous les périphériques)

- Langage : **C#** (.NET 8 recommandé).
- Framework de rendu : **MonoGame** (obligatoire).
- Plateforme cible : **Windows**.
- L'application doit fonctionner *hors ligne* une fois les dépendances installées.
- UI libre (scènes MonoGame + overlays texte / panneaux (widgets) simples).
- **Fréquence d'acquisition configurable** (p.ex. 60 Hz par défaut).
- **Journalisation** minimale : console & fichier.
- **Calibrage** :
 - Joystick : deadzones par axe, inversion éventuelle, normalisation $[-1, +1]$.
 - MIDI : mappage CC/Note/Aftertouch, affichage valeurs 0–xxx et vélocités.
- **Tests unitaires obligatoires** : couverture de la logique pure (normalisation, filtrage, mappage, sérialisation, calculs), **pas de rendu graphique dans les tests**.
- **Export code en PDF** avec structure logique : script latex reproductible documenté.

5 Cahier des charges fonctionnels de base

5.1 Variante Joystick/Manette

- ☐ Détection et sélection du périphérique actif.
- ☐ Affichage temps réel :
 - **Boutons** : ON/OFF + horodatage récent.

- **Axes** (X, Y, Z, Rx, Ry, Rz, triggers) : brut, normalisé, bargraph.
- **POV/Hat** : angle/direction (N, NE, E, ...).
- ☐ **Calibrage** par axe (deadzone, sensibilité, inversion).
- ☐ **Profils de mappage** (JSON ou XML) : nom de contrôle → action symbolique.
- ☐ Option *feedback* visuel/sonore : flash/bip à l'activation.
- ☐ Activation / Tests des vibrations

D'autres options peuvent être disponibles en fonction du périphérique fourni.

5.2 Variante Contrôleur MIDI

- Détection du port MIDI IN/OUT, sélection du périphérique.
- Affichage temps réel :
 - **Notes** : canal, n°, vitesse, on/off, durée.
 - **CC** : n° CC, valeur 0–127 (curseur/bargraph).
 - **Pitch Bend/Aftertouch** : valeur normalisée.
- **Enregistreur** court (10–30 s) avec horodatage et rejouabilité (si OUT dispo).
- **Mappage** CC → nom lisible (profil JSON ou XML).
- **Latence** affichée (moyenne et variance approximatives).
- **Sortie** : jouer les notes

6 Plan de réalisation suggéré

- 1 **Inventaire / test périphérique** : lister contrôles détectables / évaluer commandes.
- 2 **Squelette MonoGame & acquisition minimale** : boucle de jeu + capture événements → log/console.
- 3 **UI diagnostic** : bargraphs, LED virtuelles, timeline d'événements.
- 4 **Calibrage & mappage** : deadzones, normalisation/CC mapping, profils JSON/XML.
- 5 **Enregistreur/bonus** : replay MIDI ou profils multiples joystick.
- 6 **Tests, polish & docs** : manuel, fiche capacités, export PDF du code, tests.

7 Manuel utilisateur (plan attendu, 4–8 pages)

- ☐ Présentation du périphérique et de l'application.
- ☐ Installation (prérequis, .NET, MonoGame, drivers).
- ☐ Branchement et sélection du périphérique.
- ☐ Écran principal : légendes, couleurs, échelles, interprétation.
- ☐ Calibrage (pas à pas, sauvegarde/chargement profils).
- ☐ Scénarios rapides (Joystick et MIDI).
- ☐ Dépannage (non-détection, valeurs instables, latence).
- ☐ Références (versions, licences tierces).

8 Fiche des capacités (1–2 pages)

- ☐ Identité : marque/modèle/ID.
- ☐ Liste / schéma exhaustive des contrôles :
 - Joystick : axes (plage brute, deadzone par défaut), boutons (n°), POV.
 - MIDI : canaux, Note On/Off, CC, Pitch Bend, Aftertouch, CC courants.

- ❑ Limitations/particularités observées (jitter, latence, pas de MIDI OUT, etc.).

Un exemple est fourni à la section 12 page suivante



9 Tests unitaires (obligatoires)

9.1 Portée minimale

- **Normalisation axes** (joystick) : brut $\rightarrow [-1, +1]$ avec deadzone.
- **Filtrage deadzone** : valeurs proches de 0 \rightarrow 0.
- **Inversion/sensibilité** par axe : paramètres pris en compte.
- **Mappage** (joystick/MIDI) : contrôles \leftrightarrow actions symboliques.
- **Sérialisation JSON** de profils (sauvegarde/chargement fidèles).
- **Calcul latence** simple (moyenne, écart-type sur timestamps).
- **Timeline** d'événements : ordre/horodatage cohérents.

9.2 Exigences

- **Aucune dépendance graphique MonoGame dans les tests.** Tester la logique pure.
- Cible : **80% de couverture** sur le *Core logique* recommandé (indicatif).
- Commandes d'exécution documentées dans le README (ex. : `dotnet test`).

10 Qualité et bonnes pratiques

- ❑ Code clair, commenté, KISS/DRY ; pas de logique lourde dans l'UI MonoGame.
- ❑ Noms cohérents (FR ou EN, mais uniformes).
- ❑ Constantes nommées (fréquence, tailles, couleurs).
- ❑ Gestion d'erreurs user-friendly (messages clairs, pas de stacktrace brute).

11 Ressources

- <https://docs.monogame.net/api/Microsoft.Xna.Framework.Input.html>
- <https://docs.monogame.net/api/Microsoft.Xna.Framework.Input.GamePad.html>
- https://docs.monogame.net/articles/tutorials/building_2d_games/10_handling_input/index.html
- Tutoriels Code Master (vidéos) : <https://www.youtube.com/@CodeMaster101/playlists>
<https://youtube.com/playlist?list=PLS9G7A6kaaH0m76Wn4Jr0ba8veJru6Bjz&si=YrqS4V0hbusqL-nW>

12 Exemple fiche des capacités du contrôleur

1. Identité du périphérique

- **Marque** : _____
- **Modèle** : _____
- **Type** : (joystick / gamepad / clavier maître MIDI¹ / hybride / autre)
- **Identifiants système** :
 - USB² : VID³ = _____, PID⁴ = _____
 - Nom exposé au système (OS⁵) : _____
 - Ports MIDI visibles : _____
- **Firmware / version** : _____
- **Date du test** : _____
- **Testeur** : _____

2. Contrôles joystick

☐ Axes analogiques

Axe	ID / Nom	Plage brute	Deadzone	Remarques
X	_____	_____	_____	_____
Y	_____	_____	_____	_____
Z / Throttle	_____	_____	_____	_____
Rot. X/Y/Z	_____	_____	_____	_____
Autres	_____	_____	_____	_____

☐ Boutons

N°	Étiquette physique	Remarques
0	_____	_____
1	_____	_____
2	_____	_____
3	_____	_____
...	_____	_____

☐ POV⁶ / Hat switch

- Nombre de POV : _____
- Valeurs retournées : _____ (ex. : -1 = neutre, 0 = haut, 9000 = droite, etc.)
- Retour haptique / vibration : oui / non Type (simple / multi-moteur) : _____
- Comportement particulier : _____

3. Contrôles MIDI

☐ Canaux et ports

- Canal(aux) MIDI utilisé(s) par défaut : _____
- Possibilité de changer de canal : oui / non (si oui : comment ?) _____
- Ports MIDI IN / OUT visibles : _____

☐ Messages Note On / Note Off

Zone	Plage de notes	Remarques
Clavier principal	_____ (ex. : C2–C6)	_____
Pads	_____	_____
Autres	_____	_____

1. Musical Instrument Digital Interface
 2. Universal Serial Bus
 3. Vendor ID
 4. Product ID
 5. Operating System
 6. Point of View Hat, croix directionnelle

☐ **CC⁷ (contrôles continus)**

Contrôle	N° CC	Remarques
Mod wheel		
Expression		
Fader 1..n		
Rotary 1..n		
Autres		

☐ **Autres messages**

- Pitch Bend : oui / non, plage : _____
- Aftertouch : aucun / channel aftertouch / poly aftertouch
- Program Change, MMC⁸, etc. : _____

4. Comportement temporel et stabilité

☐ **Latence**

- Latence moyenne observée (joystick) : _____ms
- Latence moyenne observée (MIDI) : _____ms
- Méthode de mesure (logiciel, protocole) : _____

☐ **Jitter**

- Variabilité de la latence (écart typique) : _____ms
- Niveau ressenti : faible / moyen / fort
- Effet perçu joystick : _____
- Effet perçu MIDI (rythme / arpèges) : _____

☐ **Stabilité des valeurs**

- Jitter des axes au repos (bruit, petites variations) : oui / non, remarques : _____
- Pertes de messages / comportements erratiques : _____

5. Limitations et particularités observées

☐ **Limitations techniques**

- Pas de MIDI OUT physique : oui / non
- Résolution limitée de certains contrôles : _____
- Nombre max. de boutons / axes gérés par le driver : _____

☐ **Particularités**

- Fonctions spéciales (modes, layers, presets internes) : _____
- Problèmes connus (drivers, compatibilité OS) : _____
- Autres remarques utiles pour l'intégration logicielle : _____

7. Control Change

8. MIDI Machine Control