

motion-system

motion-system

“ Catalogue généré le 2026-05-11

En une phrase

Conçoit un motion system complet pour un projet : principes directeurs, tokens (durations/easings/distances), 6 patterns réutilisables (enter, hover, focus, modal, list, page) et règles reduced-motion intégrées.

Quand l'utiliser

- Sur un projet ambitieux où le motion fait partie de l'identité (portfolio, marketing premium, app produit)
- Quand `motion-tokens` ne suffit plus et qu'il faut formaliser les patterns
- Pour aligner une équipe sur "quelle animation pour quel usage" avant de commencer à coder
- Avant de générer la doc d'un design system pour avoir un chapitre motion solide
- Sur une refonte de site où le mouvement est un levier de différenciation

Comment l'invoquer

- **Slash command** : `/motion-system`
- **Auto-invocation** : ☐ Oui — pour produire une grammaire motion cohérente d'un projet

Description détaillée

Un motion system, c'est l'équivalent du design system mais pour le mouvement. Au lieu d'avoir 47 animations différentes choisies au feeling, on a un vocabulaire restreint et cohérent qui se réutilise partout — ce qui rend le site mémorable et facile à maintenir.

Ce skill produit le motion system en 4 temps. D'abord les **intentions** : pour quoi anime-t-on ? Hiérarchie (un élément attire l'œil), causalité (un clic ici déclenche un changement là), feedback (l'app me dit qu'elle a compris), émotion (un détail délicieux). Ensuite les **tokens** : 5-7 durations et 3-5 easings nommés sémantiquement, plus des distances standard. Puis 6 **patterns** prêts à l'emploi : enter (apparition à l'arrivée d'un élément), hover (réaction au survol), focus (signal clavier), modal (ouverture/fermeture), list (animations en série sur une liste), page (transition d'écran). Enfin les **règles reduced-motion**, déclinées pattern par pattern (pas juste un kill switch global).

À la sortie, tu as 6-10 lignes de principes directeurs, les tokens en JSON + CSS variables, les 6 patterns avec snippets fonctionnels, et une matrice normal vs reduce pour chacun. Conseil intégré : préférer `transform` et `opacity` (gratuits pour le GPU), éviter `width/height/top/left` qui forcent du layout. Pour Thymon : c'est le skill "boss" du motion. Une fois ce système posé, tous les autres skills motion (framer-patterns, gsap, view-transitions) viennent s'y appuyer naturellement.

Source

- **Plugin** : `perso (standalone)`
- **Nom interne** : `motion-system`
- **Fichier** : `/home/thymon/.claude/skills/motion-system/SKILL.md`

Revision #2

Created 2026-05-11 21:18:33 UTC by thymon

Updated 2026-05-11 21:36:18 UTC by thymon