The AI Video Repo Vault

Five free GitHub repos that turn Claude Code into a motion designer, with the install steps and the prompts I run.

Why I put this list together

I kept hitting the same wall with AI video. I would describe a product film, get back something that looked almost right, and then notice the logo had changed shape halfway through and the headline said something close to what I typed but not quite. Re-prompting gave me a different video, not a fixed one.

All five repos in here take the other road. Instead of generating pixels, they hand Claude Code a method and a rendering engine, and the agent writes React, TypeScript, HTML or p5.js that draws the frames. The text stays the text. The hex code stays the hex code. And when you want the third shot two seconds longer, you change a number instead of rolling the dice again.

Everything below comes from the repos themselves. Star counts and licences were checked on 20 November 2026. Nothing here needs a paid plan beyond the Claude subscription you already have.

01

How it actually works

1. You install the repo as an agent skill, which is a folder with a SKILL.md file in it. The folder goes where your agent reads skills, and that is the whole install. Nothing runs as a background service.
2. You describe the video in plain English in a new chat. The skill triggers on the request itself, so there is no slash command to remember. The agent reads SKILL.md, then the reference files it needs for the kind of video you asked for.
3. The agent writes code, renders it to MP4 with Remotion or headless Chrome, extracts frames with ffmpeg and looks at them, fixes what is wrong and re-renders. You get the finished file plus the source project, so the next revision is an edit rather than a new roll.

02

The one thing almost everyone gets wrong

The folder name has to match the **name:** field inside that repo's SKILL.md. If you clone **claude-remotion-skill** and drop the whole repo folder into your skills directory, the skill is called **remotion-motion-graphics** and your folder is called **claude-remotion-skill**, so nothing ever triggers. You then sit there asking for a logo sting and watching Claude write plain Remotion code with no motion rules applied, wondering why the repo is overrated.

WATCH OUT

Several of these repos keep the skill in a subfolder, so you copy the subfolder, not the repo root. For claude-remotion-skill it is remotion-motion-graphics/. For paint-mv-skills it is the four directories under skills/. Check what SKILL.md calls itself before you copy anything, and test with one small request straight after.

The fastest proof that a skill loaded: ask for something tiny and watch what the agent reads first. If it starts writing code without opening SKILL.md, the skill is not in the path.

03

The five repos

SOURCE CODE

The biggest of the five, 21,805 stars and Apache-2.0 licensed. Start here if you are selling a product:

https://github.com/Vincentwei1021/video-shotcraft

ALL FIVE REPOS FROM THE VIDEO

- video-shotcraft, product and promo films from your real pages
  https://github.com/Vincentwei1021/video-shotcraft
- anything2explainer, narrated explainers with voiceover and subtitles
  https://github.com/Vincentwei1021/anything2explainer
- bang-motion, openers and explainers as one HTML file
  https://github.com/bangtutorial/bang-motion
- claude-remotion-skill, the motion rules that fix cheap animation
  https://github.com/haidrrrry/claude-remotion-skill
- paint-mv-skills, hand-painted watercolour music videos from a song
  https://github.com/lintsinghua/paint-mv-skills

REPO	MAKES	OUTPUT	LICENCE
video-shotcraft	Product promos from your pages	1920x1080 MP4 at 30fps plus the Remotion project	Apache-2.0
anything2explainer	Narrated topic explainers	1280x720 MP4 with subtitles and chapter cards	PolyForm Noncommercial
bang-motion	Openers, promos, kinetic type, explainers	One index.html that plays in a browser	MIT
claude-remotion-skill	Motion design rules for Remotion	Rules, 17 patterns, a theme template	MIT
paint-mv-skills	Watercolour music videos	MP4 painted frame by frame with p5.brush	MIT

video-shotcraft captures your real pages, picks from 124 shot recipe cards covering 150 shots in 10 categories, storyboards the film, animates it in Remotion, cuts it to the beat and sources sound design from Freesound. After delivery it opens Motion Workbench, a CapCut-style editor in the browser where every shot, transition, caption and sound sits on its own track. It can also export a JianYing draft if you edit there.

anything2explainer takes a topic rather than a product. It researches with sources, writes the narration, generates voiceover with per-word boundaries, builds a frame-accurate timeline, storyboards every shot, then dispatches parallel build agents that each write one Remotion component. It stops at four checkpoints: length and language, narration sign-off, voice engine, and the first 30 seconds.

bang-motion is the quick one. Every deliverable is a single index.html with autoplay and loop, and ?debug=1 adds a scrub bar. It ships 10 style directions and 19 concepts, and it opens with a style brief: three concept candidates, one recommendation, and a palette where every colour states where it came from. Nothing needs npm install to be watched.

claude-remotion-skill is a rulebook rather than a builder. Ten rules, 17 copy-paste motion patterns and a theme.ts template. The rules are the useful part: springs and bezier curves instead of linear interpolation, entrances that animate two or three properties at once, staggered timing in 3 to 6 frame offsets, exits faster than entrances, Ken Burns on every still, all timing derived from fps, and render then inspect frames then fix before shipping.

paint-mv-skills turns a song and its lyrics into a watercolour music video. p5.js and p5.brush lay down ink and wash in headless Chrome, and every frame is a pure function of song time. Four skills split the work: a director that scaffolds and times the beat, a storyboard skill that builds the cast and sets out of the lyrics, an animation skill with the engine API, and a render skill for the frame loop and ffmpeg encode.

Set it up in Claude Code or Codex

1. Decide where the skill lives. In Claude Code, ~/.claude/skills/<name>/ makes it available in every project, and .claude/skills/<name>/ inside a project keeps it with that repo so your team gets it too. In Codex, use your Codex skills folder or ~/.codex/skills/<name>/, and add a pointer line in AGENTS.md such as: For motion graphics and explainers, read and follow bang-motion/SKILL.md.
2. Clone the repo you want. git clone https://github.com/Vincentwei1021/video-shotcraft.git, or swap in any of the other four URLs from the list above.
3. Put the skill folder in place. video-shotcraft and anything2explainer want a symlink of the repo root. claude-remotion-skill wants the remotion-motion-graphics subfolder copied in. paint-mv-skills wants the four directories under skills/, and the repo already has symlinks set up if you work inside it.
4. Check the folder name matches the name: field in SKILL.md. This is the step that silently breaks everything else.
5. Install the shared dependencies: Node.js 18 or newer and ffmpeg. Remotion downloads its own headless Chrome on first render, so you do not need to install that separately.
6. Install the per-repo extras only if you are using that repo. anything2explainer wants a Python virtualenv with edge-tts pinned, plus kokoro and espeak-ng for English narration. paint-mv-skills wants Google Chrome, git and patch installed, plus uv if your lyrics are plain text and need aligning. video-shotcraft wants uv only for beat analysis and puppeteer only for page capture, which the capture script installs when it needs it.
7. If a repo asks for an API key, ask Claude to add it to your shell profile for you, then open a brand new terminal window so the value is actually loaded. None of these five require a key for the core render, which is why I have not listed one.
8. Run the first test. Ask claude-remotion-skill for a three second logo sting, or ask bang-motion for a 15 second opener. A successful first run means the agent reads SKILL.md before writing code, renders a file, extracts frames with ffmpeg, looks at them, and only then tells you it is done. If it hands you a render with no frame check, the skill did not load.

INSTALL-THE-FIVE.SH

```
# 1. video-shotcraft
git clone https://github.com/Vincentwei1021/video-shotcraft.git
cd video-shotcraft
ln -s "$(pwd)" ~/.claude/skills/video-shotcraft   # Claude Code
ln -s "$(pwd)" ~/.codex/skills/video-shotcraft    # Codex
cd ..
# or, in one line:  npx skills add Vincentwei1021/video-shotcraft

# 2. anything2explainer
git clone https://github.com/Vincentwei1021/anything2explainer.git
ln -s "$PWD/anything2explainer" ~/.claude/skills/anything2explainer

# 3. bang-motion  (manual install for any agent)
git clone https://github.com/bangtutorial/bang-motion
cp -r bang-motion ~/.claude/skills/bang-motion
# or as a Claude Code plugin, inside Claude Code:
#   /plugin marketplace add bangtutorial/bang-motion
#   /plugin install bang-motion@bang-motion

# 4. claude-remotion-skill  (copy the SUBFOLDER, not the repo root)
git clone https://github.com/haidrrrry/claude-remotion-skill.git
mkdir -p ~/.claude/skills
cp -r claude-remotion-skill/remotion-motion-graphics ~/.claude/skills/

# 5. paint-mv-skills  (pulls the upstream engine at a pinned commit)
git clone https://github.com/lintsinghua/paint-mv-skills.git
cd paint-mv-skills
node skills/paint-mv/scripts/fetch_upstream.mjs
```

DEPENDENCIES.SH

```
# shared by all five
brew install ffmpeg
node --version        # needs 18 or newer

# anything2explainer only
python3 -m venv ~/.venvs/a2e && source ~/.venvs/a2e/bin/activate
pip install 'edge-tts==7.2.8' numpy pillow scipy
pip install kokoro soundfile && brew install espeak-ng   # English narration

# paint-mv-skills only: Google Chrome, git and patch must be installed.
# Plain-text lyrics also need uv, for the auto-alignment pass.
```

💡 QUICK TIPS

- Install one copy of each skill only. A Claude Code plugin and a manual folder sharing a skill name will conflict, and the plugin wins.
- Start every new video in a fresh chat. These skills are read when first invoked, and a long chat keeps old decisions in context that quietly shape the new film.
- Say which style you want, not which scenes. Naming scenes from a reference video gets you a copy of that video. Naming a style gets you something of your own.

04 The prompts I actually run

PRODUCT PROMO WITH VIDEO-SHOTCRAFT

Use video-shotcraft to make a [LENGTH] second promo for [PRODUCT NAME]. The site is [URL]. Capture the real pages, use the Ink Press template, and keep the palette from our logo. Audience is [WHO IT IS FOR].

NARRATED EXPLAINER WITH ANYTHING2EXPLAINER

Make me an explainer video about [TOPIC]. [LENGTH] minutes, English, dot-field background. Research it with sources first and show me the narration before you generate any voiceover.

FAST SOCIAL OPENER WITH BANG-MOTION

Build a [LENGTH] second opener for [PRODUCT NAME] in the [STYLE NAME] style, [9:16 or 16:9]. Logo and screenshots attached. Give me three concept candidates and the palette before you write any code.

FIX A FLAT RENDER WITH CLAUDE-REMOTION-SKILL

My Remotion video at [PATH] looks generic and flat. Apply the motion rules: springs instead of linear, three-property entrances, staggered timing, Ken Burns on the stills. Extract frames and show me before and after.

MUSIC VIDEO WITH PAINT-MV-SKILLS

Make an MV from this song and lyrics: [PATH TO song.mp3], [PATH TO lyrics.lrc]. Title is [SONG TITLE]. Write the storyboard from the lyrics first and show me the cast sheet before you paint any chapters.

When it breaks

ERROR	FIX
The skill never triggers and Claude writes plain code	The folder name does not match the name: field in SKILL.md. Rename the folder, restart the session, test with one small request.
Two versions of the same skill fight each other	You have a plugin and a manual folder with the same name. Delete one. The plugin wins, so remove the manual copy unless you want the opposite.
Maximum for --concurrency is 2	A low-core machine rendering with Remotion. Pass --concurrency=1.
Remotion cannot launch the system Chrome	Recent Chrome dropped the old headless mode. Point Remotion at a chrome-headless-shell binary with --browser-executable=<path>, or set REMOTION_BROWSER_EXECUTABLE for anything2explainer.
edge-tts stops working after an upgrade	It tracks a Microsoft endpoint and breaks across versions. Pin it: pip install 'edge-tts==7.2.8'.
kokoro will not install on ARM or Python 3.13	It pins an old numpy and pulls spaCy, which has no aarch64 wheels. Use TTS_ENGINE=kokoro_onnx with the kokoro-onnx model files, or TTS_ENGINE=piper.
paint-mv-skills is missing engine files like core.js	The upstream PDoomVideo source is not bundled, by design. Run node skills/paint-mv/scripts/fetch_upstream.mjs once in the repo.
Parallel builds die partway through a long explainer	Several agents bundle Remotion at once. Keep at least 5GB free, and dispatch in waves past about 12 agents.
It delivered a render with no frame checks	The skill is not loaded. Every one of these renders, extracts frames and inspects them before handing over, so a render without that step means the agent never read SKILL.md.

Ready to go deeper?

Want the videos made for you? Paste any link or idea into Content Engine and get a published video in under 3 minutes, with DM automation armed to capture leads.

Meet Content Engine →

https://builds.digicuratoragency.com/content-engine/