DIRECTOR Guide: Turn Claude into a video studio with 8 free repos

By the end, Claude will be making finished MP4 films on your own computer: launch films, explainers, cartoons and lyric videos, with 10 copy-paste prompts and 5 ways to get paid for it this week.

Want more AI workflows like this?
Join our free community for the best AI tools, prompts, and shortcuts.
Join here: skool.com/viral-ads

How this actually works

Here is the part most people miss. Claude Opus 5.5 does not spit out a video file. It writes code that draws every single frame, like a flip book where the AI draws each page, and then a couple of free programs turn those pages into a real MP4.

1. **Opus writes the film as a web page**
   It uses HTML canvas, p5.js, three.js or Remotion. Think of these as different pens: canvas and p5.js draw flat and hand-drawn pictures, three.js draws 3D, and Remotion lets you build a video the way you build a website.
2. **A headless browser photographs every frame**
   Headless means Google Chrome runs invisibly in the background with no window. It opens the page, jumps to frame 1, takes a picture, jumps to frame 2, and so on.
3. **ffmpeg stitches the frames and the music together**
   ffmpeg is a free program that almost every video app quietly uses under the hood. It joins the pictures into an MP4 and lays the soundtrack under them.
4. **The music is code too**
   Opus writes the soundtrack and the sound effects as code on the same timeline, so every hit lands exactly on the cut.

THE KEY INSIGHT

> Because every frame is code, nothing is random. Claude can change one word, one colour or one second and render the exact same film again. That is why this beats AI video generators for brand work: the logo, the colours and every word of text come out perfect every time.

What it is great at: motion graphics, explainers, product launch films, hand-drawn and cartoon animation, lyric videos, data stories and 3D product shots.

What it is not for: photoreal people. When you need a real-looking human, Opus directs a video model such

as Seedance, Runway or Higgsfield and does the editing around it.

WHERE THE MONEY IS

> One research library studied 1,119 Claude video files and found that 71.6% of the ad and launch films were motion graphics or UI animation. That is exactly the lane where code-made video already looks like an agency made it.

Three things nobody tells you

1. The same prompt never makes the same film twice. You will get a different film on every run. The polished films you saw online were made, reviewed and revised, usually two rounds or more.

2. Effort changes the result more than wording. Effort is how hard Claude thinks before it acts. On high or xhigh it storyboards, checks its own frames and adds far more detail. On low you get something that looks like a template.

3. Ask for the storyboard first. Almost every great film in these repos started with "show me the storyboard before you write any code". You fix the plan in two minutes instead of re-rendering a whole film.

KNOW THE COST

> A one-line launch film can cost a few dollars of usage. A multi-minute music video can eat a big slice of a weekly plan in one session. Start with clips of 10 to 30 seconds and go longer once you like the style.

One-time setup (about 10 minutes)

You need four free tools plus Claude Code. Install them once and every repo in this guide works. You do not need to know how to code: you will only copy, paste and press Enter.

Step 1: Install Claude Code

Claude Code is Claude that lives in a terminal window on your computer, so it can create files and run programs for you. It comes with a paid Claude plan (Pro or Max). For daily film work I would go Max, because long renders use a lot of usage.

Windows: open the Start menu, type PowerShell, open it, paste this and press Enter.

```
irm https://claude.ai/install.ps1 | iex
```

Mac: open the Terminal app (press Cmd + Space, type Terminal), paste this and press Enter.

```
curl -fsSL https://claude.ai/install.sh | bash
```

Then type **claude**, press Enter and log in with your Claude account. Product page: claude.com/product/claude-code

Step 2: Install Node.js

Node.js is the engine that runs the render scripts. Go to nodejs.org, download the LTS version (22 or newer), open the file and click Next until it is done. Same on Windows and Mac.

Step 3: Install Google Chrome

Chrome is the invisible camera that photographs each frame. If you do not have it, get it at www.google.com/chrome/.

Step 4: Install ffmpeg

Windows: paste this into PowerShell and press Enter.

```
winget install Gyan.FFmpeg
```

Mac: first install Homebrew, a free app installer, from brew.sh (copy the one line on that page into Terminal). Then paste this.

```
brew install ffmpeg
```

Other options are on the official page: ffmpeg.org/download.html

Step 5: Check that everything works

Close the terminal, open a new one (new installs only show up in a fresh window), and paste these two lines. Each one should print a version number, not an error.

```
node -v
```

```
ffmpeg -version
```

Optional extras: some repos also want Python for the soundtrack (www.python.org/downloads/, on Windows tick "Add python.exe to PATH" on the first screen) and Git for downloading repos (git-scm.com/downloads). No Git? On any GitHub page click the green Code button, then Download ZIP, and unzip it.

Step 6: Open a folder in Claude Code and paste a prompt

1. **Make an empty folder**
   For example a folder called films on your Desktop. Every film gets its own folder.
2. **Open a terminal inside that folder**
   Windows: open the folder in File Explorer, right-click on the empty space and choose Open in Terminal. Mac: in Terminal type cd and a space, drag the folder into the window, and press Enter.
3. **Start Claude**
   Type claude and press Enter.
4. **Pick the model and the effort**
   Type /model, choose Opus 5.5 and set the effort to high or xhigh.
5. **Paste a prompt from this guide and press Enter**
   Claude will ask permission to create files and run commands in that folder. Say yes. When it finishes, the MP4 is in the folder (usually inside an out folder).

LAZY MODE

> Skip steps 2 to 4 entirely: open Claude Code and paste the prompt below. It checks your machine and installs whatever is missing.

```
Check whether Node.js, Google Chrome, ffmpeg, Python and Git are
installed on this computer.

Install anything that is missing, then show me the version of each one.
```

The 8 free repos

These are in the same order as the carousel. For each one: what it is, what you can make or sell with it, exactly how to use it, and the link.

1. Lemo-Opuscar: 59 film styles in one Claude Code skill

A skill you install into Claude Code with 59 ready film styles, each with a demo film made fully in code. You pick a style, bring your story, and Claude directs the whole thing: picture, music and narration. 6.3K stars, MIT

licence for the code.

The styles: watercolor, ink wash, whiteboard explainer, dark tech keynote, glass product render, broadcast news package, 1930s rubber-hose cartoon, 80s cel anime, 16-bit pixel RPG, brick toy, Swiss motion graphics, blueprint, data storytelling and many more.

What you can sell with it: app launch films in the Dark Tech Keynote style, product reveals in Glass Product Render, client explainers in Whiteboard Explainer, a weekly news recap in Broadcast News Package. You can also turn any style into a brand style for a client (their colours, logo and characters) and reuse it on every film you make them.

How to use it: paste these two lines into your terminal (not inside Claude) to install the skill.

```
claude plugin marketplace add lemomo-ai/lemo-opuscar

claude plugin install lemo-opuscar@lemolab
```

Then open any empty folder in Claude Code and just say what you want. This is the repo's own example:

```
Make a 45-second film in the watercolor style about the coffee farm my
family runs. Warm female narrator.
```

Not sure which style? Type **/lemo-opuscar:lemo-opuscar** and nothing else. It asks your language, whether you have your own voice or music, and whether you want to see a storyboard first. Say yes to the storyboard.

To build a client's brand style: say it in one line, like the repo shows.

```
I want to change the Ukiyo-e style into my own version.
```

GOOD TO KNOW

> One film takes the agent about 30 to 60 minutes and a fair amount of usage. It needs Node 20 or newer, ffmpeg and Python 3.11 or newer, and it installs the rest itself. Default output is 1920x1080 at 24 fps, so ask for 1080x1920 when you want a vertical film.

Link: github.com/lemomo-ai/lemo-opuscar (gallery: lemomo-ai.github.io/lemo-opuscar/)

2. mg-styles-15: 15 motion-design styles with the full prompt behind each

Fifteen 10-second motion-design films, one per style, each published with its style notes, the complete prompt

that made it, and the full source. No image or video generation: the picture is code, and the music and sound effects are synthesised in code too. MIT licence.

The styles: frame-by-frame line boil, isometric 2.5D, flat vector, line art, 3D render, shape morph, sticker explainer, cyberpunk HUD, collage cut-out, aurora glass, Bauhaus, 80s synthwave, pixel art, liquid motion, and a vertical 9:16 variety-show caption style.

What you can sell with it: a style menu for clients. Send them the watch page, let them pick a look, and make their message in that style. Agencies love this because the client chooses from finished films instead of vague mood boards.

How to use it: this is the repo's own method.

1. **Watch the films with sound**
   Pick the style you like on the watch page.
2. **Open the prompts folder on GitHub and open that style's file**
   The instructions around the prompt are in Chinese, but the prompt itself is in English. Copy the whole prompt block.
3. **Paste it into Claude Code in an empty folder**
   You need Node 22.12 or newer, Chrome, ffmpeg and Python 3.11 or newer.
4. **Make it yours**
   Change only the Creative seed part (story, brand name, copy, colours) and the Output part (for example vertical 1080x1920). Keep the Signature features part, because that is what makes the style recognisable.
5. **Push it further**
   When the film is done, have a second AI review it with the repo's rubric.md and paste the review back for another round. The original films went through make, review, revise twice.

Link: github.com/Vincentwei1021/mg-styles-15 (watch all 15: vincentwei1021.github.io/mg-styles-15/)

3. ClaudeAnimationBase: hand-painted cartoon animation with a ready character

A starter kit for hand-painted cartoon animation in p5.js and p5.brush (digital brushes that paint watercolour and pencil strokes in code). It comes with a ready character, Clawd, who has 31 acted emotions, plus the rules Claude follows to animate well. MIT licence.

What you can sell with it: a cartoon channel, animated stories for a kids' brand, a cartoon intro for a podcast, or short animated ads starring a brand mascot. The repo itself suggests asking Claude to swap the character, invent new emotions or costumes, or work from your own reference images.

How to use it: download it, open the folder in Claude Code and paste the starter prompt from the README.

```
git clone https://github.com/JohnHeibel/ClaudeAnimationBase.git
```

```
cd ClaudeAnimationBase

claude
```

```
Read ANIMATION_GUIDE.md, then make a 15-second video of Clawd trying to
catch a butterfly.
```

Claude storyboards first, builds shot by shot, renders contact sheets (a grid of still frames) to check its own work, and writes the finished film to out/video.mp4. Swap the butterfly for your own story.

To render the included demo yourself: this makes an 11-second test film. Open studio.html in Chrome to scrub through it.

```
npm install

node render.mjs --clip --out=out/video.mp4
```

PRO MOVE

> All of the repo's test videos were made with Opus 5.5 on xhigh effort. The README says the effort level decides how extravagant and detailed the scene gets.

Link: github.com/JohnHeibel/ClaudeAnimationBase

4. pdoom-video: a karaoke lyric music video rendered in code

The complete source of a lyric music video where every word of the song appears on screen in sync with the singer. Every frame is calculated from the song's time, so the live preview and the final export are identical. It exports at 1080p60 or 4K60. Built with three.js. MIT licence for the code.

Why it is in sync: Claude split the vocals out of the song, matched each word to the exact moment it is sung, and mapped every beat. That timing data drives the whole video, which is why the words hit like a professional lyric video.

What you can sell with it: lyric videos for independent musicians who release songs every month and cannot afford a motion designer each time.

How to use it: use it as the blueprint for your own song. Download it, put the song and the lyrics in the folder,

start Claude Code there, and paste my prompt.

```
Study this repo, especially docs/ENGINE.md and docs/TREATMENT.md.
Then build the same kind of word-synced karaoke lyric video for my song.
The song is song.mp3 and the lyrics are in lyrics.txt.

Show me the treatment and the storyboard before you write any code.
```

To preview the original: it uses bun, a fast alternative to Node (bun.sh). Then open http://localhost:5173 in Chrome.

```
cd app
bun install

bunx vite
```

To render it: add --samples 4 instead of --samples auto for a quick draft.

```
cd app

bun scripts/render.ts video --samples auto --shutter 0.2 --out
../out/pdoom.mp4
```

Link: github.com/mexicat/pdoom-video (watch it in 4K: www.youtube.com/watch?v=5EoO5413dBY)

5. PDoomVideo: a full music video Opus directed with a crew of subagents

Another complete music video for the same song, this one hand-painted with p5.brush. Opus directed it with parallel subagents, which are helper copies of Claude that each build different scenes at the same time, like a director with a crew. The only brief was to use the Clawd character and give each lyric interesting visuals and transitions.

What you can learn from it: how little direction Opus needs. Open STORYBOARD.md (the shot-by-shot plan Opus wrote) and ANIMATION_GUIDE.md (the style guide Opus wrote to brief its crew). Copy that structure for your own long projects.

What you can sell with it: animated music videos, and longer animated pieces for clients where one prompt has to produce many scenes.

How to render it: the song is included. The first line paints every frame (you can stop and resume it), the second joins the frames and the song.

```
git clone https://github.com/JohnHeibel/PDoomVideo.git
cd PDoomVideo
npm install
node render.mjs --frames=0:156.6 --workers=4

node render.mjs --encode --out=out/pdoom.mp4
```

For scale: this film is 156.6 seconds, about 3,760 frames, and it reportedly used about 10% of a Max 5x weekly allowance.

Link: github.com/JohnHeibel/PDoomVideo (watch it: youtu.be/8j-hR4fJywU)

6. opus-video-prompts: 18 full prompts, 54 cases, 9 ways to build

A collection of 18 complete prompts, 54 public cases and 9 technical routes, each linked to the original post, with a column for how long it took and what it cost. The README is in Chinese: open it in Chrome, right-click anywhere on the page and choose Translate to English.

What is inside: the shortest test prompt that produced a launch film in about a minute for about $2, a fill-in-the-blanks 30-second business explainer, Remotion app promo prompts, a premium product film built from real footage, and a list of writing tips that I use on every film.

The shortest test prompt: paste this word for word as your very first test.

```
make a modern slick and punchy video for a modern startup that works on
inference
```

What you can sell with it: treat the prompts list as your service menu. Launch films, explainers, recipe explainers and app promos are all in there with the exact wording that worked.

Link: github.com/joeseesun/opus-video-prompts

7. awesome-claude-5-5-videos: the research library

A library of 424 source posts of videos made with Claude 5.5 (301 of them with Opus), each linked to the original, grouped by type, with research on what keeps showing up in the best ones. CC BY 4.0 licence.

What you can use it for: pitching. Before you message a client, find three examples of the format you are selling and send them as references. The "Make your own" table maps each task (2D animation, explainer, 3D scene, turning existing footage into animation) to an example and a guide.

Two findings worth knowing: 70% of game and interactive videos were 3D, and 71.6% of ad and launch

videos were motion graphics or UI. Motion-graphics films split almost evenly between dark and light looks, so there is no single winning palette.

Link: github.com/athemeroy/awesome-claude-5-5-videos

8. awesome-opus5-5-videos: the biggest prompt library (703 prompts)

The biggest library of its kind: 703 prompts behind viral Opus 5.5 videos, sorted into motion graphics, explainers, 3D scenes and games. Every entry links to the original post, and 418 of them have a live remake you can watch side by side with the original. 10.1K stars, MIT licence.

What you can use it for: a swipe file. When a client asks for something, search the prompts folder for it (launch, explainer, recipe, vertical) and start from a prompt that already went viral.

1. **Open a prompt file and copy the block under Prompt**
   Some creators only shared part of their prompt, and those files say so.
2. **Paste it into Claude Code running Opus 5.5**
   Swap in your brand, product or story.
3. **Ask it to render an MP4**
   Add: "Render it to an MP4 with ffmpeg in headless Chrome."

Link: github.com/yihui-dev/awesome-opus5-5-videos (watch the remakes: skillry.dev/ai-videos/opus-5-5)

---

10 prompts I would start with

All ten are copied word for word from the repos above, typos included, because that is the exact wording that worked. Replace anything in [brackets] with your own details. Paste each one into Claude Code in a fresh, empty folder.

1. The one-line launch film

The fastest way to see what Opus can do. The person who posted it reported about a minute and about $2. Swap "a modern startup that works on inference" for your own business.

```
make a modern slick and punchy video for a modern startup that works on
inference
```

From: opus-video-prompts (prompt 01), also in awesome-opus5-5-videos.

2. A launch film for a real website

This one pulls real images and branding from the site, so the film looks like the client's own. Put the client's

website where it says [website].

```
I want you to create a highly professional SaaS product launch video. Go
and look up [website] Pick that, and then make sure to get actual assets
and images and all of that stuff from the internet. Turn it into these
typical, very professionally edited, motion-graphics-styled product
launch videos that you see people making on Twitter when they launch new
SaaS products (which are showing off the features, the benefits, and all
of these things). It should show what an incredible motion designer you
are, like it's your showreel for a resume. Go all out.
```

From: awesome-opus5-5-videos.

3. The 15-second motion-graphics showreel

Gives Opus total creative freedom. Run it on the highest effort. It is the best way to show a client what the tool can do before you touch their brand.

```
make a dynamic 15-second motion graphics video that shows what an
incredible motion designer you are, like it's your showreel for a résumé.
go all out.
```

From: opus-video-prompts (prompt 02).

4. The 30-second business explainer (fill in the blank)

Five scenes: the problem, what you do, how it works in 3 steps, one proof point, your name. Fill in the last line with what you sell, who it is for and your colours. Afterwards, give one change at a time ("slower", "more energy", "add a pricing scene").

```
Adopt the role of an expert motion designer. Build a 30-second animated
explainer for my business as a single HTML page. 5 scenes. The customer's
problem, what I do, how it works in 3 steps, one proof point, and my name
at the end. Bold text, smooth transitions, my brand colours. My business
[DESCRIBE WHAT YOU SELL, WHO IT'S FOR AND YOUR COLOURS]
```

From: opus-video-prompts (prompt 07). Add "and render it to an MP4" if you want a file instead of a page.

5. The recipe explainer (bars, restaurants, food brands)

The original made a Negroni explainer with a step counter, measurements and a progress bar. Swap the

cocktail for any recipe, product setup or how-to.

```
We're going to try a little test. Do you think you could render a recipe
motion graphic animation using javascript or html (w/e you think will
produce the best) to show the full recipe from start to finish (empty
glass to completed cocktail) - Explainer video style - Showing the recipe
ingreidents + measurements as they're going into the cup. Should be a 30s
video.
```

From: opus-video-prompts (prompt 08).

6. The lyric video

Put the song file in the folder first. Opus syncs the words to the vocals and the beat.

```
make a dynamic lyrics motion graphics video based on the attached music,
perfectly syncing the lyrics and animation to the vocals and beat. make
it feel like an incredible motion designer’s showreel for a résumé. go
all out.
```

From: awesome-opus5-5-videos. For a deeper build, use the pdoom-video repo as the blueprint (my prompt is in repo 4 above).

7. The cartoon scene

Run it inside the ClaudeAnimationBase folder. Change the scene to anything: Clawd opening your product, Clawd stuck in traffic, Clawd learning to cook.

```
Read ANIMATION_GUIDE.md, then make a 15-second video of Clawd trying to
catch a butterfly.
```

From: ClaudeAnimationBase README.

8. The styled story film with narration

Run it with the Lemo-Opuscar skill installed. Swap "watercolor" for any of the 59 styles and the coffee farm for your client's story.

```
Make a 45-second film in the watercolor style about the coffee farm my
family runs. Warm female narrator.
```

From: Lemo-Opuscar README.

9. The 9:16 app ad

For anyone selling an app. It records the real app in a simulator, then builds a vertical ad around the footage and keeps text clear of the TikTok and Instagram buttons.

```
Make a 45-50 second vertical video for my app: a glance inside the app.
Show every feature, use motion, be creative. I want it good enough for
your CV.

- Real, not mocked: record each feature being used in the simulator (tap
a filter, open a result, add it).
- Stretch your imagination: let each feature break out of the phone and
change the world around it.
- The bar: a motion designer's showreel piece. The best motion you can
do. Go all out.
- Nothing made up: every number and label on screen comes from the
recordings.
- No stock sound effects; I'll add the music.
- Keep text clear of the TikTok/Instagram interface.

- Let viewers follow: each feature stays on screen for at least 2.5
seconds.
```

From: awesome-opus5-5-videos.

10. The 9:16 faceless story short

A vertical story with audio made in code. Change the topic in the last line and you have a repeatable format for a faceless channel.

```
Create a pure javascript animation. 30-60s, vertical 9:16, paper cut-out
shadow theatre style with gentle, fitting audio (made in code) on the
topic: What makes a place feel like home?
```

From: awesome-opus5-5-videos.

---

5 ways to make money with it this week

THE PITCH THAT WORKS

> Make the film before you ask for anything. A free 10-second film made with the prospect's own logo and website gets a reply far more often than any message that says "I do video".

1. 15-second launch films for local brands and startups

What you deliver: one 15-second film in 16:9, a 9:16 cut for Reels and TikTok, and three still frames they can post.

How to find the first client: look for anyone launching something this week: a new menu, a new product, an app on Product Hunt, a startup announcing on LinkedIn. Run prompt 2 with their website, send them the result, and offer the full version.

What I'd charge to start: a few hundred dollars per film while you build a portfolio, then raise it.

2. White-label motion-design packages for agencies

What you deliver: a 30-second explainer in two styles plus a 9:16 cutdown, with no branding of yours, so the agency resells it as their own.

How to find the first client: message ten small marketing agencies that already sell social media or ads. Send them the mg-styles-15 watch page as your style menu and offer to make one sample for their best client.

3. Lyric videos for independent musicians

What you deliver: a full lyric video in 1080p and a 15-second 9:16 clip of the chorus for promotion.

How to find the first client: find artists releasing a single in the next few weeks (Instagram, TikTok and SoundCloud are full of them) who have no lyric video yet. Offer the chorus clip free. If they like it, the full video is the paid job.

RIGHTS

> Only make lyric videos for songs the client owns or has the rights to. Never upload a famous artist's song.

4. A faceless cartoon or explainer channel

What you deliver: to yourself, three shorts a week in one niche: history in line art, how everyday things work, kids' bedtime stories, or explainers for your own industry.

How to start: pick one style from Lemo-Opuscar or the Clawd kit and keep it for every video so the channel is recognisable. Use prompt 10 or prompt 7 as the template and change only the topic.

How it pays: platform creator programs once you qualify, sponsors, and clients who see the channel and want the same thing for their business.

5. 9:16 social ads

What you deliver: three versions of the same ad with three different hooks, captions kept out of the area the TikTok and Instagram buttons cover, so the client can test which one wins.

How to find the first client: search the Meta Ad Library (www.facebook.com/ads/library/) for businesses in your city that are already running ads. They already pay for ads, so fresh ad creative is an easy yes. Make one free version with their logo and offer the set of three.

PRO MOVE

> Build the client's brand style once (Lemo-Opuscar can save it), then every new ad is one prompt. That is how one person delivers like a small studio.

Pro moves most people miss

- **Brief like a director.** Give the total length, the number of scenes and what each scene says. A reference video beats a pile of adjectives.
- **Use camera words.** "Dramatic cuts", "match cut", "kinetic typography" and "orbiting camera" get you a far more professional edit.
- **Give a banned list.** Most of the "template" look comes from particle bursts, RGB split, lens flares, neon glows and bouncy easing. Tell Claude not to use them.
- **Check stills before the full render.** Ask for a contact sheet of key frames first and fix anything crowded, overlapping or hard to read.
- **One change at a time.** After the first version, give one note per message ("slower", "more energy", "bigger logo at the end").
- **Hide your API keys.** If you plug in a voice or music service, put the key in a file called .env and add the line below to .claude/settings.json in that folder, so Claude can use the service without ever reading the key.

```
"deny": ["Read(./.env)"]
```

These tips come from the writing notes in opus-video-prompts and the repos' own READMEs.

Troubleshooting

The render is painfully slow. Without a dedicated graphics card, the watercolour fills in p5.brush take

seconds per frame. Paste this, and render drafts at half resolution until you like the film.

```
Replace the watercolour fills with flat colour or simple brush strokes,
turn off motion blur for now, and render a half-resolution draft first.
```

It says Chrome was not found. Some repos look for Chrome in one standard place. ClaudeAnimationBase and PDoomVideo accept --chrome= with the path, ClaudeAnimationBase also reads CHROME_PATH, and mg-styles-15 reads MG_CHROME. The easy way is to tell Claude: "Find where Chrome is installed on this computer and use that path."

```
Windows: C:\Program Files\Google\Chrome\Application\chrome.exe
Mac: /Applications/Google Chrome.app/Contents/MacOS/Google Chrome
```

It says ffmpeg is not recognised. Close the terminal and open a new one, then run ffmpeg -version. Still nothing? Run the install line from Step 4 again, or ask Claude to install it for you.

The film looks generic. Raise the effort to xhigh, add a banned list, give a reference video, and ask for the storyboard before any code.

Text hides under the app buttons on vertical videos. Add "keep all text inside the 9:16 safe zone, clear of the TikTok/Instagram interface" to the prompt.

KEEP CLIPS SHORT

> Keep each film between 10 and 30 seconds while you learn. Render time and usage grow with every second, and a 2-minute film is a few thousand frames.

Stay safe

BEFORE YOU INSTALL ANYTHING

> Lemo-Opuscar, mg-styles-15, ClaudeAnimationBase, pdoom-video and awesome-opus5-5-videos are MIT licensed, and awesome-claude-5-5-videos is CC BY 4.0. PDoomVideo and opus-video-prompts show no licence file, so study them and borrow ideas, but do not copy their code into paid client work. Most of these repos are only a few weeks old and several have under 5,000 stars, so install only from the exact links in this guide, read the README first, and run each one in a fresh, empty folder. Music: only use songs and sounds you have the rights to. The two P(doom) repos both say the song is not

theirs.

Your first week

1. **Day 1: Setup and the one-line test**
   Do the 10-minute setup, then run prompt 1. You will have your first MP4 the same day.
2. **Day 2: Install Lemo-Opuscar**
   Browse the gallery, pick one style and make a 30-second film about your own business with the storyboard step on.
3. **Day 3: Make a cartoon**
   Clone ClaudeAnimationBase and run prompt 7 with your own scene.
4. **Day 4: Build your style menu**
   Pick three styles from mg-styles-15 that fit your clients and make your own logo film in each one.
5. **Day 5: Make a free spec film**
   Choose one real business, run prompt 2 with their website and send it to them.
6. **Day 6: Go vertical**
   Make a 9:16 version with prompt 9 or 10 and post it.
7. **Day 7: Package the offer**
   Pick one of the five money plays above, put your three best films on one page, and send it to ten prospects.

Resources and Links

- Lemo-Opuscar (59 film styles skill)
- Lemo-Opuscar gallery
- Lemo-Opuscar style index
- mg-styles-15 (15 motion-design styles)
- mg-styles-15 watch page
- mg-styles-15 review rubric
- ClaudeAnimationBase (cartoon animation kit)
- ClaudeAnimationBase animation guide
- pdoom-video (code-rendered lyric video)
- pdoom-video in 4K on YouTube
- PDoomVideo (music video directed with subagents)
- PDoomVideo on YouTube
- opus-video-prompts (18 prompts, 54 cases)
- awesome-claude-5-5-videos (research library)
- awesome-opus5-5-videos (703 prompts)
- Skillry: originals and remakes side by side
- Claude Code
- Node.js
- Google Chrome
- ffmpeg downloads

- Homebrew (Mac installer)
- Python
- Git
- bun
- Meta Ad Library

Want more AI workflows like this?

Join our free community for the best AI tools, prompts, and shortcuts.

Join here: skool.com/viral-ads