Targeted Modeling Skill (TMS): component-built hero modelling

TMS formalizes the way the APEX GT was made. The car is never modelled as one sculpted shell. It is manufactured like the real thing: a chassis first, then every panel, insert, wheel, mechanical part and interior piece as its own named part. Each part is built, looked at and reviewed, then refined. The loop repeats until:

- the reference and the user are both satisfied
- the quadrant review (section 4b) scores at least 95% in every quadrant
- no key detail from the reference is missing
- the surface and form review (section 4c) passes: no panel overhangs or steps, no block-shaped inserts, smooth reflections
- the presentation sheet (section 4e) is made: the reference sheet's layout rebuilt on plain white from the Blender model, plus rear-left and front-right 3/4 views

1. Reference and blueprint first

- Collect the reference: a five-view sheet (front, back, left, top, bottom) plus any per-part sheets from the user. Keep them in `work/<asset>-build/ref/`.
- Pick the master view. For a car, the side view is the master for the profile. Treat perspective renders as approximate, not orthographic. Note which way the sheet's side view faces (front left or right) and set the compare camera to match.
- Take the scale from the tyre's horizontal width on the side view (or from the wheelbase), never from its height. The body and arches hide the tyre top, and a height-based scale put the monster truck's wheels visibly off.
- Extract measurements into a blueprint module. For the APEX GT this was `body_lines.py` with `silhouettes.json`.
  - It holds numbers only, no geometry: lengths, widths, crest lines, wheel centres, shut-line positions, the plan half-width table, the roof line, sheet crops and px-to-metre scales.
  - List the intended creases and character lines here too, so the fairness check (4c) knows which sharp edges are wanted.
  - Every part samples it, so the parts sit flush with each other.
- Fix the frame and write it at the top of every script. APEX example: front = -X, left = -Y, up = +Z, ground at z = 0. Record wheel centres, track and tyre radius.
- If the sheet shows a real production vehicle, build an original design at its proportions: no manufacturer badge, grille or lamp design copied, and invented markings only.
- **Detail inventory:** before modelling, list every visible feature in the reference, per view and per quadrant. Save it with the parts plan. It is the checklist the detail audit (4b) is scored against. For example:
  - lamps and their internal graphics
  - grille bars and badge
  - vents and creases
  - trims, handles, mirrors
  - extra windows and pillars
  - antennas and spotlight
  - plates, reflectors, fog lamps
  - skirts and sills, arch lips
  - wheel design, exhaust
  - light bar, push bar
- For each inventory item also note its **form**: rounded or flat, recessed or proud, and roughly how deep. "Grille: curved mesh, recessed ~3 cm in a gloss surround" is the target the blockiness check (4c) scores against.

2. Write the parts plan before modelling

Write a lettered, numbered plan (save it to the project as `<asset>-parts-plan.md`):

- **A Chassis:** tub/cabin cell, roll cage, crash structures, subframes, rails. This is what everything bolts to.
- **B Body panels:** nose, bonnet, wings, doors, roof, quarters, engine cover, bumpers, sills. Each panel is separate, split at real shut lines.
- **C Vents and intakes:** separate black inserts, louvres and meshes set into the panels.
- **D Aero and underbody:** splitter, canards, wing, endplates, uprights, diffuser, floor.
- **E Arch liners**
- **F Glazing**
- **G Lighting assemblies**
- **H Wheels and suspension:** tyre, rim, nut, disc, caliper, upright, wishbones, pushrod, steering.
- **I Visible mechanicals:** engine, gearbox, radiator, driveshafts, exhaust.
- **J Details:** mirrors, antennas, wiper, pins, fuel filler, tow hooks, number boards, latches.
- **K Interior:** seats, wheel, dash, net, extinguisher.
- **L Livery:** decals and paint layers, never geometry.

Give each part a code (A01, B05, H03 …, or an asset prefix such as TA01 or PB05) and a Blender collection per letter. Map every item in the detail inventory to a part code. Keep `00 REFERENCE` and `00 STUDIO` collections for references and lighting.

3. Build live, one part at a time

- Work in the user's open Blender (MCP `execute_blender_code`) so they can watch. Keep one `.blend` per asset (e.g. `APEX_GT_PARTS.blend`). Don't touch the shipping game files until export. Only one Blender can hold the MCP port, and the add-on needs *Allow Online Access* turned on.
- Use a shared kit module (APEX: `scripts/art/apex/apex_kit.py`) with:
  - a material table (`mat(key)`, named `<ASSET> <material>`)
  - a bmesh `Geo` helper: `box`, `beam`, `tube`, `revolve`, `prism`, `quad_grid`, `cap`, `obj`
  - a part registry: `@part(code, title, coll)`
  - a driver: `build(code, frame='part', snap=True, save=True)` that clears the part, rebuilds it, frames the viewport, saves a snapshot PNG to `work/<asset>-build/progress/`, and saves the file
  - `look()`, `snapshot()` and `part_sheet()` for per-part views
  - a post-build hook that reverses any painted shell whose paint faces point inward. Inside-out panels mirror their decals and vanish in single-sided glTF.
- Use a shared form kit for every non-panel part:
  - `rbox`: rounded box
  - `rtube`: rounded-rectangle tube, with `bend=` fillets
  - `ctube`, `pill`
  - `loft` and `sweep`: seats, dashes, mirror housings, mouldings
  - `sheet`
  - `tray`: recessed lamp pocket with a flush lens
  - plain `g.box` is only for tiny emitters
- Group parts into stage modules (`parts_stage1.py` chassis + wheels, `parts_body.py`, `parts_aero.py`, `parts_inserts.py`, `parts_lights.py`, `parts_interior.py` …) so any part can be rebuilt on its own.
- Later correction passes may append overriding definitions at the end of a module, and the last definition wins. Other modules must then call shape helpers through the module (`B.helper()`) so they pick up the overrides.
- Build order:
  1. Rolling chassis (A + H). Get the user's approval on stance and ride height before any bodywork.
  2. Body panels.
  3. Inserts and liners.
  4. Aero and underbody.
  5. Glass and lights.
  6. Mechanicals, details and interior.
  7. Livery.
  8. Export.

How a panel is made (panel kit)

- Painted outer skin, cut to its shut lines with 2.5 mm per side (5 mm gaps).
- Arches trimmed with a flared lip. Apertures cut for lamps, vents and glass.

- A real back: carbon laminate thickness, a return flange on every edge, stiffening ribs. Leave bolt tabs off curved visible skins: they poked through the police sedan's wings as black spikes.
- **Smooth blueprint curves.** Interpolate every blueprint table (plan, nose, tail, bonnet and roof lines) with a monotone cubic (PCHIP), never straight segments. Straight segments put a kink at every table point, and the zebra review showed broken reflections on every panel.
- **Rolls at least 2× the skin thickness** (18 mm or more with 8 mm skins). A tighter roll folds the back through the skin, and back faces show as red triangles.
- **No zero-width rows.** Every grid row needs a real width, at least a 6 mm lip. When rows collapse to one point, the normals are garbage and the return flange flips up as a visible flap.
- **Three-panel junctions share one edge function.** Examples: bonnet / wing / fascia, trunk lid / quarter / C-pillar, and A-pillar / wing / door. Compute one panel's edge from the other's edge (for example, a table of the fascia's top edge that both the bonnet front and the wing front read), never as two separately tuned curves. Each wing and quarter wraps over a shoulder (a roll plus a ledge), and the bonnet or lid sits between the ledges. Pillars land on the panel below them, never flying over it. Mouldings and trims end where the panel under them ends; an end cap in open air reads as a floating plate.
- **Frames and pillars are thin skins.** Keep returns 10 mm or less and set the glass 6 mm or less under the skin. With 35 mm returns and deep-set glass, every pillar reads as a plank.
- Shared sculpt features live in the blueprint's `sculpt()` or shape helpers, and every panel samples them so seams stay tight. These include arch flares, scoops, character ridges and creases, haunch crests, the shoulder roll and tumblehome.
- **Build round, not square**, but don't overdo it. Boxy sections were the first complaint on the police sedan. Use:
  - tumblehome above the belt
  - a rounded shoulder and a tucked sill
  - crowned plan curves at the nose and tail
  - crowned, rolled edges on the bonnet and boot deck
- The rear face of a sedan stays broad. Keep rear corner radii at about a quarter of the half-width and ease the tumblehome towards the tail. Big corner radii plus full tumblehome made the police car's rear "curve up and in too much".
- **Panels meet; they never float.** Neighbouring panels must sample the same edge curve on both sides of a shut line. The bonnet's side edge sits on the wing's top line (flush, or lapping just under it), never above or outside it. The return flange turns down out of sight, so no slab of panel thickness shows from the game camera.
- **Smooth surfaces.** Use enough grid density on painted skins that reflections stay smooth, shade smooth with weighted or auto-smooth normals, and keep sharp edges only on the creases listed in the blueprint.
- Build window and lamp openings with slanted edges as their own warped (trapezoid) panels so the edges are exact. Never make them from skipped grid cells, which stair-step.
- Pass explicit material keys for every asset (wrap the shared panel function), so one asset never inherits another asset's paint.
- Never fall back to a one-piece master skin that gets sliced up. The user rejected that approach.

Lamps, grilles and inserts

- Make lamp housings open trays (back plate and walls, no front face), or proud shaped units laid on the panel surface with a bilinear outline. A solid box hides the LEDs.
- Modern lamps have projector cups with emitters, an LED light guide tracing the outline, an amber signal, a dark or chrome inner bezel, and a thin clear lens. A flat clear sheet with one LED stroke is not a modern lamp.
- **No blocks.** A grille, bumper insert or vent is never a box. Give it a real section: a curved or angled face following the fascia, a surround or bezel, a mesh or bars recessed 2–4 cm behind the opening, and a dark cavity behind that. A grille guard or push bar is bent tube with a rounded section, ribbed pads and integrated LED heads, never stacked boxes.
- **Light guides are flat ribbons** (about 3 × 10 mm) set back behind a smoked lens, with chrome reflector bays for depth. Round glowing tubes read as "pen lines". Never run an LED stripe across a body panel.
- **Light bars** are one product:
  - a segmented tinted lens (red | clear takedown | blue)
  - reflector heads with LED rows inside

  - a gloss-black top cover with segment joints
  - an aluminium base and end caps
  - never loose modules strung along a plate
- **Exhausts and valances:** exhaust tips stand 30–40 mm proud, with a rolled lip and a dark inside. Valance and diffuser inserts sit flush, in gloss, with fins, never as a matt plate standing proud.
- **Mirrors:** a lofted aero housing on a sail mount, with a glass face and a repeater. Size and place it from the front view; the sheet's mirrors decide quadrant E.
- **Interiors are sculpted, not boxes:**
  - lofted bucket seats with bolsters and headrests on posts
  - a moulded rear bench
  - a dash swept along the car, with screens and vents
  - a round wheel with spokes and a hub
  - a console, door cards, carpet and headliner
- Make glass and lenses see-through with material alpha (blended), not transmission, so solid view, EEVEE and glTF all agree.

4. Review loops (the part that makes TMS effective)

4a. Silhouette compare and review stills

- **Silhouette compare:** render orthographic cameras locked to the blueprint framing at the sheet's own pixel scale (`compare.py` / `cmp_render`). Score each view by overlap = shared area ÷ combined area. Log the scores per pass in a table and fix the worst view first. Also run a gap report for holes and open edges.
- **Side by side:** render the textured model (EEVEE) at the sheet's pixel scale for the side, front and top views (`match_render`), and lay each render next to the matching sheet crop. Send this to the user each loop.
- **Review PNG (the user's quality image):** render the exported game model in the game's own renderer settings on a light neutral background. One page with front, back, left, top, bottom and a chase-cam 3/4 view. The user may pass it to another reviewer. Use a pure white studio in Blender for stills.

4b. Quadrant review and detail audit (required before calling the model done)

Split each reference view into close-up quadrants and compare the sheet with the model area by area.

Quadrants:
- **Side view** (quadrants overlap by about 10% so the seams are covered):
  - A front-upper: bonnet, windscreen, A-pillar, mirror
  - B front-lower: bumper, lamp, wheel, arch
  - C rear-upper: roof, C-pillar, backlight, deck
  - D rear-lower: quarter, tail, bumper, wheel
- **Front view:**
  - E upper: roof, windscreen, bonnet, lamps
  - F lower: grille, bumper, push bar, tyres
- Add rear and top quadrants when the sheet has those views.

Each quadrant gets one row:
- sheet crop, model crop at the same scale, and an outline overlay (sheet edges red, model edges cyan), all at 3× zoom
- a caption with the quadrant's silhouette overlap, plus the model-extra and sheet-missing areas in m²

Sheet mask:
- Use the pixels darker than the white background.
- Fill only small holes (under about 1,500 px) so white livery areas count.
- Exclude the ground shadow below the tyre contact line, so it never counts as missing body.

Detail audit (key missing details):
- For every quadrant, walk the detail inventory and mark each item **present / wrong / missing**, looking at the 3× crops, not only the silhouette score.
- Look especially for features that leave no trace in the outline:
  - lamp graphics and size
  - grille bars and badge

  - creases and character lines
  - extra side windows
  - arch lips and black sills
  - mirror and spotlight shape and position
  - fog lamps, reflectors, plates
  - trims, handles, antennas
  - wheel design
  - exhaust
  - light-bar and push-bar proportions
- List the **missing** and **wrong** items at the top of the report, ranked by how visible they are from the game camera. For each, give its size or position from the sheet: "third side window behind the rear door, about 25 cm wide", "tail lamp 13 cm tall, wraps 35 cm onto the side", "push-bar uprights 7 cm thick, ±0.45 m apart".

Findings and loop:
- Write shape findings as concrete, measurable corrections, e.g. "bonnet front 3–5 cm low", "cabin wider than the sheet's A-pillars".
- Rank shape findings and missing details together, fix the top items, and re-run.
- Re-run 4b after every 4c fix too. Detail changes such as mirrors, spotlights and the rear corner radius moved quadrant E by ±0.01.

Gate:
- Keep looping until every quadrant's overlap is at least 0.95.
- No detail-inventory item is missing, and any remaining "wrong" items are explained, for example by the sheet's perspective.
- Record each loop's quadrant scores and the audit table in `<asset>-review-loops.md`.

4c. Surface and form review (required; the outline can't see these)

The silhouette and quadrant scores measure the outline only. The police sedan scored 95%+ everywhere while:
- its bonnet overhung the wings with a visible slab edge
- its grille guard was a stack of blocks
- a wing sliver stuck out at the A-pillar
- its bonnet reflections were lumpy

Orthographic front and side views hide all of that. Run this review every loop after 4b.

Beauty close-ups:
- Render EEVEE with a reflective studio (HDRI or strip lights) and gloss paint, from the angles the game shows:
  - front 3/4 low (bumper height, about 1.5–2.5 m out)
  - front 3/4 high
  - rear 3/4 low
  - side at wheel height
  - a straight-on rear view
  - a tight close-up of each lamp, the grille and the bumper corners
- If the sheet has a perspective or hero render, match its camera and lay the two side by side. Send the close-ups to the user each loop.

Reflection (zebra) test:
- Render the same close-ups with a striped environment or a zebra matcap.
- Stripes must flow smoothly across each panel and carry on across shut lines. A kink or wobble marks faceting or a dent. A jump at a shut line marks a step between panels.

Tools (a shared surface-review module, generic for any asset):
- `run(tag, out_dir, cams, paint, back, thru, skip_colls, form_colls)` renders, for each camera, `<tag>_<cam>_gloss.png`, `_zebra.png` and `_flags.png`, and writes `<tag>_checks.json`.
  - In the flags image, red means a back face is visible (overhangs, slabs, holes, flaps) and yellow means a step between painted panels.
  - The JSON holds the per-camera counts with locations, faceting per panel and the hard-edge share per part.
  - Rays that pass through glass don't count, so the interior is not flagged.

- `panel_map(tag, out_dir, cams)` renders each object in a flat colour with back faces in pure red. Hide everything except the panels at one junction to see exactly which piece overhangs or leaves a hole. It is the fastest diagnosis at the A-pillar base, the C-pillar base, the bonnet corners and the lamp surrounds.
- To name a mystery piece in the user's screenshot, cast rays from their view angle over the suspect area and report which object each ray hits.
- Standard cameras:
  - front 3/4 low and high
  - rear 3/4 low and high
  - straight rear
  - side
  - nose, tail and cabin close-ups
  - the light bar from above
  - the user's own viewport angles when they send screenshots

Automatic checks (one script, prints a pass/fail table; every failure is a finding with its location):
- **Flush / step:** sample every shut line every 5 cm and measure the offset between the two panels along the surface normal. Fail if a step is over 3 mm, or if one panel's edge stands proud of or outside its neighbour. This check catches a bonnet riding over the wing.
- **Overhang:** in plan, every panel's outline must stay inside the blueprint plan line (plus any flare). In section, the bonnet's side edge must meet the wing's top line, not sit above it.
- **Visible thickness:** from the game cameras, no panel may show its back or laminate edge (a dark slab under a lip).
- **Fairness:** on painted skins, the angle between neighbouring face normals stays under about 8°, except on the creases listed in the blueprint. Report the count and location of the rest, plus any concave dent where the surface should be convex.
- **Blockiness:** for every insert, trim and interior part (grille, bumper inserts, lamps, vents, push bar, mirrors, light bar, seats, dash), compute the hard-edge share: the length of edges over 65° as a share of total edge length. A plain box scores 1.0 and rounded forms score about 0. Over 0.3 means rebuild it with the form kit to the form noted in the detail inventory.
- **Stray geometry:** fail any piece that pokes outside the body skin by more than a few millimetres, such as a wing sliver at the A-pillar.

Quality rating:
- For each close-up, rate every region 1–5 against the sheet on:
  - shape: rounded where the sheet is rounded
  - panel fit
  - surface smoothness
  - detail depth
  - lamp and grille modernity
- Any score under 4 is a finding with a concrete fix, e.g. "grille is a flat 6-face box: rebuild as a curved mesh insert recessed 3 cm in a gloss surround" or "bonnet side edge 2 cm above and 1.5 cm outside the wing: re-sample the wing top line".

Gate:
- No failures in the automatic checks.
- Every region is rated 4 or higher.
- The close-ups have been sent to the user next to the sheet.
- Record the check table and ratings in `<asset>-review-loops.md` alongside the quadrant scores.

4d. Back and forth with the user

- Show snapshots after each meaningful step. Apply the user's review sheet as a named correction pass. Record every pass (what changed, with numbers) in the project doc `<asset>-review-loops.md`.
- When the user sends a viewport screenshot, reproduce that exact angle as a review camera and keep it in the standard set for later loops.
- Don't call a part done until it has been seen in context on the whole car, from the game's own camera angles, not only the orthographic views.

4e. Presentation sheet (the deliverable image)

Rebuild the reference sheet itself from the model: the same layout on a plain white background, with the reference photos replaced by Blender renders.

- **Layout** (matches the reference):
  1. Title block: vehicle name, subtitle and tagline, plus the reference's small right-hand headings.
  2. FRONT VIEW, SIDE VIEW and TOP VIEW of the whole vehicle: orthographic, framed tight, front on the left in side and top.
  3. **ADDITIONAL VIEWS**: a **rear-left 3/4** and a **front-right 3/4** perspective. These are always added, even when the reference doesn't have them.
  4. MAJOR COMPONENTS CATALOG: the same numbered panels and part list as the reference, each part isolated and shown from the views the reference shows. Label the views OUTER / TOP / INNER, FRONT / SIDE / BOTTOM and so on. When a reference part isn't a separate part in the model (for example door glass and frames that live in the glazing and greenhouse parts), render what exists and note it.
- **Tools:**
  - A sheet renderer's `render_sheet(out_dir, SPEC)` writes transparent PNGs and `manifest.json` to `work/<asset>-build/sheet/`.
    - The SPEC holds the title, subtitle, tagline, `extra_views` (eye, target, lens) and `parts`.
    - Each part has a title and rows of (label, object-name prefixes, view), where the view is one of front, rear, left, right, top or bottom. Only the listed objects render.
    - See `POLICE_SPEC` for a worked example.
  - A sheet-composition script lays the renders out. It uses Poppins, with soft contact shadows and light panel borders.
- **Look:**
  - a white world at about 0.3 strength with key, fill, top and under area lights
  - AgX with the Medium High Contrast look, so black paint stays black; at a higher world strength the car renders washed-out grey
  - a transparent film, composited on white
- **Delivery:**
  - Save to `work/<asset>-build/sheet/<asset>_sheet.png`, send it into the chat and write it into the user's folder.
  - Regenerate it after every major loop.
  - It sits alongside the review PNG (4a) as the user's main image for judging the model against the reference.

5. Export for the game

- Keep damageable parts as separate named nodes: bumpers, bonnet, doors, wing, mirrors, splitter, light bar.
- Join meshes per material only inside each part.
- Align the ground contact with the game's contact height. Build near + LOD levels.
- Check triangles per part before export. Rounded boxes cost about 600 triangles each at n=3, so use `n=1` for small plates and plain boxes for tiny LED emitters. The police light bar went from 112k to 15k triangles this way. A hero car should land around 180k near and 75k LOD.
- Compress (meshopt), then verify in game with a capture. Keep new cars behind a dev flag (e.g. `?candidate=parts`) until approved; then make them the default.

Rules

- Only original designs and invented brands. No real manufacturer marks, tyre brands or copied models. Never download or rip third-party models.
- Fine detail comes from parts and inserts, not from texture tricks on a single shell.
- A high silhouette score is not a finished car. Outline (4b) and surface and form (4c) must both pass.
- Save progress to the project docs at each milestone: the parts plan with its detail inventory, review loops (with quadrant scores, the audit table and the 4c check table), review PNG notes and the latest presentation sheet.
- When sending files to the user's computer, commit from a fresh staged file name each time. Re-using a name can re-send a stale copy.