# Launch Control — NASA Space Apps Challenge 2026 submission pack

**Team:** sterlings · **Project:** Launch Control · **Challenge:** Space Mission Design Game
**Compiled:** 2026-11-15 from repository revision `13ac270` (`docs/hackathon/`). Single-document version of the pack files; the separate files remain the editable source.

> Draft for the team. It is not a completed submission or an eligibility certification.

## Contents


1. Overview, pending information and submission checklist
2. Challenge alignment
3. Project documentation and portal copy
4. Resources, provenance and AI disclosure
5. Presentation structure and production criteria
6. Portal Project Demo script (29 seconds)

# 1. Overview, pending information and submission checklist

*Source file:* `docs/hackathon/README.md`

Team: **sterlings**. Event: **NASA Space Apps Challenge 2026 — 15th Edition**.

Prepared in English on 15 November 2026. Evidence reviewed: project revision `157a6b9` (updated from `f52c774`). This is a submission draft, not a completed submission or an eligibility certification.

## Files

**Single document:** launch-control-space-apps-2026.md and PDF join all the files below. They are generated copies: edit the separate files and regenerate them.

Start with the presentation structure and production criteria, the latest specification for creating the presentation later.

1. Challenge alignment: requirement mapping and official NASA resources.
2. Project documentation and portal copy.
3. NASA resources, third-party credits, and AI disclosure.
4. Separate 29-second portal demo.

The four-minute talk is for a local presentation if the local organizer allows that duration. **Do not submit its recording as the portal's short Project Demo.** This pack provides a separate short-video plan and a seven-slide structure. Export the portal deck as seven English slides with still images; the local talk may use the longer gameplay clip.

## Information still needed

| Field | Required action |
|---|---|
| Event edition and local event | 2026, 15th Edition; Ensenada, Baja California. Exact UABC venue remains to be confirmed. |
| Official team name | sterlings, confirmed by the requester. Launch Control is the project title. |
| Team member names | All six names supplied by the requester are recorded in the project documentation. Confirm their accepted event registration separately from repository activity. |
| Selected challenge | Space Mission Design Game. Official brief supplied by Gael on 15 November 2026; mapping done in challenge-alignment.md. |
| NASA resource required by the challenge | Official Resources tab supplied. The project relates to *Design and Develop Science Missions* and *Mission Design Lab*, as described in challenge-alignment.md §4. The four earlier NASA URLs are additional references, not official challenge resources. |

## Final handoff checklist

☑ Record the supplied edition, team name, challenge, city and six member names.
☑ Record and review the four NASA URLs supplied by Gael.
☐ Confirm the exact UABC venue and accepted event roster.
☐ Verify public names with the team.

☐ Verify the current official submission guide and local presentation rules.
☑ Document the team-confirmed analysis of NASA references, gameplay adaptations and limitations.
☑ Verify the documented resource use against the official challenge Resources tab (challenge-alignment.md §4).
☐ Decide whether to close any quick gap from challenge-alignment.md §5 before recording.
☐ Finish asset/library credits and project distribution rights. No root project license was found in the inspected revision; do not assume the team's code is MIT-licensed.
☐ Disclose AI assistance honestly, including any contribution to ideas or design. Check the user's supplied restriction on core ideation against the applicable rules; do not certify human-only ideation without evidence.
☐ Apply the supplied media policy: visible AI labels on generated images/video, descriptive and metadata attribution for generated audio/code/data. Review gameplay showing AI-assisted assets with the organizer; use a clear disclosure overlay in the meantime.
☐ Use gameplay and diagrams only. Exclude minors' faces, voices, photos and representations under the supplied policy; use an adult narrator or text-only captions. Do not assume a team member is an adult.
☐ Do not generate or alter NASA logos, insignia or mission marks. This pack uses text-only attribution.
☐ Export either a video under 30 seconds or at most seven English slides for Project Demo. Inspect the exported duration/page count, not only the editor timeline.
☐ Add English captions to the video. Add English annotations/subtitles to Spanish gameplay; for the submitted slide deck, use English-only labels or annotated visuals with Spanish UI text covered in the presentation export.
☐ If uploading portal images, follow the supplied JPG/PNG/GIF and 10 MB limits; verify current portal limits.
☐ A team member reviews originality and terms, accepts the portal checkboxes and clicks **Submit for Judging**. Do not treat a saved draft as submitted.
☐ Confirm the deadline, timezone, resubmission procedure and certificate conditions in the actual event portal. The supplied 11:59 p.m. cutoff needs its date and local timezone confirmed.
☐ Save the submission confirmation.

## Rule verification notes

The submission guide and legal pages returned HTTP 502 during this review. The official `nasa/spaceapps` discussion on hosting and demos supports separate public project/demo links and short-video/seven-slide formats. A Global Organizing Team answer specifies a NASA source from the challenge resources and AI attribution requirements. These are useful guidance, not a substitute for the edition's full terms.

The strict minors restriction, ban on AI core ideation, precise image-size limit, automatic disqualification language and certificate condition were supplied by the requester and were not independently confirmed in the full current terms. They are retained as preparation constraints, not represented as newly verified official quotations. Confirm them at the official resources page and legal page.

## Venue clarification

The requester named UABC Campus El Sauzal - Gimnasio Universitario, with FIAD / UABC Campus Punta Morro as an alternative depending on the local announcement. These are not treated as one confirmed venue. Use **Ensenada, Baja California, Mexico** in the presentation until the local confirmation resolves the building/campus. The official Ensenada event page was found in search, but its full contents could not be retrieved.

# 2. Challenge alignment

*Source file:* `docs/hackathon/challenge-alignment.md`

Project: **Launch Control**. Team: **sterlings**. Prepared 15 November 2026 from the official challenge text and Resources tab supplied by Gael (copied from the challenge page), against repository revision `157a6b9`.

Status labels used below:

- **Implemented** — in the integrated game on `main`.
- **Partial** — present, but narrower than the challenge describes.
- **Designed** — specified in the team's expansion guide (`GUIA_IMPLEMENTACION_MINIJUEGOS_CODEX.md`, "Protocolo CML"), **not implemented**. Never present it as working.
- **Not covered** — neither implemented nor designed.

## 1. What the challenge asks

Space Mission Design Game asks for an interactive game in which participants **design, manage and simulate a complete space mission**: define an objective, configure a spacecraft and instruments, choose a launch vehicle and orbit, and manage competing constraints (mass, power, budget, data, communications, lifetime). The emphasis is on **trade-offs**: each decision should have consequences, and players should learn *why* a mission succeeds or fails. Single-player or teams of up to four, with different roles, are suggested. Difficulty levels, feedback, scoring, mission reports and post-mission analysis are listed as possible features ("may, but are not required to").

## 2. Honest positioning

Launch Control implements the **operations end** of the mission lifecycle, not the full concept-to-operation design process. Two players with different information — an astronaut in the **space segment** and mission control in the **ground segment** — must coordinate, under a shared countdown, to bring the spacecraft's systems to a launch-ready state. The result is a launch or a comic failure.

What it teaches today: complementary roles, procedures, communication and verification before launch. What it does not teach yet: design-time trade-offs between mass, power, budget, orbit and launch vehicle. The team's "Protocolo CML" expansion is designed to bring those resource trade-offs into the same cooperative format, but it is not built.

**Recommended framing for the portal and talk:** "We built the *operations and launch-readiness* stage of a mission as a cooperative game, and designed how the next stage brings mission-design trade-offs into it." Do not describe the current build as a complete mission-design simulator.

## 3. Requirement-to-feature mapping

| Challenge element | Launch Control | Status | Evidence |
|---|---|---|---|
| Interactive game with a clear interface | Two 3D workstations (cockpit and mission-control room), host/join lobby, persistent HUD | Implemented | `scenes/main.gd`, `scenes/cockpit/`, `scenes/tower/`, `ui/hud/` |

| Challenge element | Launch Control | Status | Evidence |
|---|---|---|---|
| Collaboration with different roles | Astronaut (sees and operates the panel) and mission control (reads manual, uses calculator); neither can succeed alone | Implemented | `docs/qa/antenna-round.md`, `docs/qa/t031-integracion-dificultad.md` |
| Teams of up to four participants | Exactly two players | Partial | Network session supports one host and one client |
| Different difficulty levels | Easy (antenna) and normal (antenna + numeric code + calculator) | Implemented | `resources/difficulties/`, T030/T031 |
| Clear objectives and feedback on success/failure | Shared countdown, pending-module count, per-attempt feedback (wrong/retry), launch animation on success, comic failure on timeout, "connection lost" on disconnect | Implemented | `scenes/outcomes/`, #53 |
| Explain *why* a mission succeeds or fails | Outcome shows *what* happened (launched / timed out); no causal explanation | Partial | — |
| Simulate multiple times / iterate | Authoritative rematch logic exists; in-game rematch button still pending (restart required) | Partial | T035 (#49), issue #55 |
| Define a mission objective first | Fixed objective: make the spacecraft launch-ready before the countdown ends | Partial | — |
| Spacecraft components and instruments | Cockpit modules (antenna alignment, numeric code) as abstract subsystems | Partial | `scenes/modules/` |
| Resource management: **time** | Shared host-authoritative countdown | Implemented | `autoload/server_clock.gd` |
| Resource management: **power** | 35 % power-reserve fuse bank | Designed | Guide §3, minigame 2 |
| Resource management: **communications** | Deep-space link tuner (+3 dB margin) | Designed | Guide §3, minigame 1 |
| Orbit / launch vehicle / **delta-V** | Propulsion and delta-V budget injector; stage-separation sequencer | Designed | Guide §3, minigames 5 and 7 |
| Instruments, **data** and science value | Science-traceability data bay | Designed | Guide §3, minigame 6 |
| Risk and trade-off decisions | 5×5 technical-risk mitigation matrix | Designed | Guide §3, minigame 3 |
| Mass, budget, mission lifetime | — | Not covered | — |
| Scoring, ranking, mission report, post-mission analysis | Win/lose only | Not covered | — |
| Accessible to non-experts | Plain-language manual, icons + text + color for every state, no prior knowledge needed | Implemented | `docs/art/visual-guide.md` §2.2, `ui/manual/` |

## 4. NASA resources from the challenge's Resources tab

The four NASA web pages already cited elsewhere in this pack (Artemis II, Missions, A to Z of NASA Missions, Rocket Systems Area) are **not** in the challenge's Resources tab. Use the official ones below for the portal's NASA-resource field, and describe their use exactly as stated here.

| Official resource | How it relates to Launch Control | Use |
|---|---|---|
| **Design and Develop Science Missions** — the four mission segments | The game's structure follows the segments: **launch segment** = the rocket on the pad and its launch/failure outcome; **space segment** = the astronaut's cockpit and its subsystems; **ground segment** = the mission-control room, manual and communication; **user segment** = not represented (the designed science-traceability minigame would introduce it). | Used to explain and frame the existing roles; the segments were not a source for the authored puzzle rules. |
| **Mission Design Lab (MDL)** — collaborative, discipline-based mission studies | Supports the core idea that a mission is evaluated by specialists with different knowledge working together. The designed minigames map to MDL disciplines: communications, power, propulsion, risk. | Framing and roadmap; no MDL data or study process is reproduced. |
| **Concept Maturity Levels (CMLs, JPL)** | The designed expansion uses a playful "CML" progress indicator. | Inspiration only. The guide states that the game's indicator **does not** measure real concept maturity. |
| **A Learning Game About Telescopes** | Comparable approach: constrained choices, a launch animation, and results tied to the player's decisions. Its link from results to real missions is a model for the missing post-mission explanation. | Reference for planned improvements; not used in the current build. |
| GMAT, NASA Mission Viz, CubeSat 101, 2023 VR challenge, CSA games | Not used. | Do not cite as used. |

Portal-field draft:

> We used NASA's *Design and Develop Science Missions* overview to structure the game around mission segments: the astronaut works in the space segment, mission control in the ground segment, and both drive the launch segment's outcome. NASA's *Mission Design Lab* informed our emphasis on specialists collaborating, and our designed expansion maps new minigames to its disciplines (communications, power, propulsion, risk). Puzzle rules, numbers and the countdown are authored game mechanics, not NASA procedures or data.

## 5. Closing the most visible gaps in the remaining time

Ordered by value for judging and by effort. None of these is done; they are options.

1. **Mission briefing card** (small): before the round, one screen that states the objective in segment terms ("Ground and space segments must bring the vehicle to launch-ready in 3:00"). Addresses "define a mission objective first".
2. **Post-mission explanation** (small): on failure, list which subsystems stayed pending and what that would mean in a real mission; on success, name the segments that cooperated. Addresses "explain

why".
3. **Rematch button** (Jorge, issue #55): enables "simulate multiple times".
4. **One designed resource minigame** (medium): implementing the power-reserve or link-margin minigame would turn one "Designed" row into "Implemented" and show a real trade-off.

## 6. What not to claim

- A complete mission-design simulator, or mass/budget/orbit trade-offs in the current build.
- Support for four players.
- That NASA resources supplied the game's rules or numbers.
- That the "CML" indicator measures real concept maturity.

# 3. Project documentation and portal copy

*Source file:* `docs/hackathon/project-documentation.md`

**Project:** Launch Control
**Team:** sterlings
**Registered members:** Gael Flores Cañedo, Jorge Roel Vasquez Sanchez, Kevin Humberto Soto Ortiz, Israel Gomez Ahumada, Jorge Abraham Garcia Sojo, Adrian Balderas Rosas
**Event:** NASA International Space Apps Challenge — 2026, 15th Edition / Ensenada, Baja California, Mexico
**Selected challenge:** Space Mission Design Game — https://www.spaceappschallenge.org/2026/challenges/space-mission-design-game/

Prepared from repository revision `157a6b9`. Complete the resource audit before copying into the portal. Field character limits were not available; shorten to the actual form limits if necessary.

## 1. High-level project summary — portal draft

Launch Control is a cooperative 3D game prototype that introduces players to communication and shared problem-solving in space operations. One player works inside a spacecraft cockpit while another operates from mission control. Each receives different information: the astronaut observes and operates a panel, while mission control consults a manual and communicates instructions. They must combine their knowledge before a shared countdown expires.

Built in Godot with GDScript, the prototype turns a space-themed procedure into an approachable teamwork activity. Its intended value is to help learners experience why clear observations, useful instructions and coordinated action matter. It is an educational abstraction, not flight-training software or a physically accurate mission simulator.

**Challenge focus:** Space Mission Design Game asks for a game covering a mission from concept to operation, with resource trade-offs. Launch Control implements the **operations and launch-readiness stage** as a cooperative game: the astronaut (space segment) and mission control (ground segment) must coordinate to make the vehicle launch-ready before the countdown ends. Design-time trade-offs (power, communications, propulsion, risk, science data) are specified in the team's expansion design but **not implemented**. See the full challenge alignment.

## 2. Detailed project description — portal draft

### What we built

Launch Control is a two-player desktop prototype with complementary astronaut and mission-control roles. The astronaut uses a modular cockpit panel. The mission-control player enters a room with rows of consoles and a large screen displaying a rendered view of the project's rocket. An interactive laptop opens the instruction manual, while a persistent heads-up display keeps the round information visible.

Two difficulty levels are integrated. **Easy** uses the antenna task; **normal** adds a numeric-code module in a second cockpit slot, a matching manual page and a calculator tool for mission control. The antenna task is the simplest example: the astronaut observes whether a light is fixed or blinking and reports it. Mission control reads the matching instruction in the manual and tells the astronaut which symbol to select. A

correct submitted action resolves the component; an incorrect action leaves it pending for another attempt. The host decides the result, so the visual interface cannot award itself success.

### How it works

Players create or join a session, select different roles, and confirm readiness. The host controls round progression and validates submitted attempts. Role-specific views separate the astronaut's clue from the mission-control manual. A shared countdown provides urgency; success, timeout and interrupted sessions have distinct outcomes. Spoken coordination takes place in person or through an external communication service; built-in voice chat is not claimed.

The mission-control wall display is a second rendering camera, not a live NASA camera or external video feed. It observes the same in-game rocket used by the visual outcome scenes. The three-dimensional environments use simple original geometry and a restrained palette to keep the controls recognizable.

### Why it matters

Space exploration involves people with different responsibilities working toward a common result. Launch Control makes that dependency playable: seeing a problem is different from knowing the procedure, and knowing the procedure is different from operating the controls. The intended learning outcomes are precise reporting, instruction-following, communication and collaborative reasoning.

Potential settings include facilitated STEM demonstrations and introductory space-themed learning activities. These are intended uses, not validated deployments. No classroom study or measured learning improvement has been completed in the evidence reviewed.

### Technical implementation

- **Engine and language:** Godot 4.7.2 standard edition, typed GDScript and `.tscn` scenes.
- **Rendering:** Compatibility renderer, with a 1280×720 baseline and a secondary viewport for the mission-control launch display.
- **Networking:** Godot high-level multiplayer using ENet, with a host and one client.
- **State:** `SessionService`, `ServerClock` and `RoundController` separate session management, time display and authoritative round state.
- **Presentation:** cockpit, mission-control room, manual, HUD and module views communicate through explicit APIs and signals.
- **Content:** original low-poly meshes and SVG icons; reference photographs are analysis material, not packaged textures.
- **Development and checks:** Git/GitHub, PowerShell and shell scripts, automated GDScript checks and Windows/Linux CI.
- **Hardware actually documented:** a Windows 11 laptop with Intel Core i5-1135G7, 12 GB RAM and integrated Intel Iris Xe graphics. No special controller or VR headset is required by the current design.

### NASA connection and scientific scope

The companion resource notes use NASA's public descriptions of mission control to explain the relationship between specialized ground support and crew communication. These sources were added to

the documentation during submission preparation; they are not evidence that the original gameplay rules were derived from NASA data.

The team supplied four official references: Artemis II, Missions, A to Z of NASA Missions, and Rocket Systems Area - Design and Construction. Their content is summarized in the resource register: mission and vehicle context, mission discovery, and historical ground-test infrastructure. The team analyzed these real-world concepts and adapted them into a friendly, shortened mission-inspired experience. Preparation, observation, communication, action and feedback become a small set of cooperative tasks. Two roles stand in for a larger operations team, while a compressed timeline and simple controls serve gameplay needs. The source-to-design analysis records how the research informed this adaptation. No NASA numerical dataset, live telemetry service or orbital model was found in the inspected implementation. The light-to-symbol mapping, countdown and module puzzles are authored game rules. We do not describe them as authentic NASA procedures. Membership in the selected challenge's official resource list and compliance with its requirements still need verification before final submission.

### Results and limitations

The repository includes automated checks for modules, session behavior, clocks, scene interactions and integrated role views. The recorded T025 workstation run used two local processes on one Windows laptop. It measured average frame rates between 59.8 and 59.9 FPS at 1280×720, with one-second minimum samples between 52 and 57 FPS. These are development measurements on one device, not a guarantee for other hardware or Internet conditions.

The integrated flow at revision `157a6b9` covers the lobby, role selection, easy and normal difficulty (antenna, plus the numeric-code module and calculator in normal), role-specific environments, the persistent countdown, and the result presentations: a launch animation on success, a comic failure on timeout and a "connection lost" message on disconnect, shown on both workstations. These presentations were connected in the last sprint and still need a recorded human playthrough. Authoritative rematch logic exists in the round controller, but the in-game rematch button is not connected yet. Do not edit separate scenes to imply one uninterrupted working session.

Remaining validation includes human-operated playtesting, a second physical computer, external-network testing and a verified distributable build. Until the rematch flow is connected in the main scene, a new round requires returning to the start screen and restarting the game; seamless replay is not claimed.

## 3. Challenge-to-feature mapping

The complete table, NASA resources from the official Resources tab and gaps are in challenge-alignment.md. Summary:

| Challenge element | Status in Launch Control |
|---|---|
| Interactive game, clear interface, accessible to non-experts | Implemented |
| Collaboration with different roles | Implemented (two roles) |
| Teams of up to four | Partial: two players |
| Difficulty levels | Implemented: easy and normal |
| Success/failure feedback | Implemented: launch, comic failure, connection lost |
| Explaining *why* a mission succeeds or fails | Partial: shows the result, not the cause |

| Challenge element | Status in Launch Control |
| --- | --- |
| Time as a managed resource | Implemented: shared countdown |
| Power, communications, delta-V, risk, science data | Designed in the expansion, not implemented |
| Mass, budget, lifetime, scoring, mission reports | Not covered |

The current build should be presented as the operations and launch-readiness stage of a mission, not as a complete mission-design simulator.

## 4. Final-project package

Prepare a public folder, release or site containing:

1. A short English landing page with the challenge and the team.
2. Source code at a named commit and clear license/third-party notices, if distributing source.
3. A tested Windows build, only if one has actually been exported and verified.
4. English setup/play instructions: host, join, roles, ready, antenna interaction, manual, outcomes and exit.
5. This documentation, resource provenance and AI disclosure.
6. The compliant short demo or seven-slide deck.
7. The longer gameplay clip as optional supporting material, distinctly labeled from Project Demo.

## 5. Evidence pointers

- `README.md`: current stack, setup and contributor responsibilities.
- `scenes/main.gd`: integrated composition by difficulty (antenna and code modules, manual, calculator, HUD) and current replay limitation.
- `docs/qa/t031-integracion-dificultad.md`: easy/normal integration evidence.
- `docs/art/asset-register.md`: asset and license register (T041).
- `autoload/session_service.gd`: ENet session and attempt handling.
- `docs/qa/workstations.md`: T025 measurements and limits, based on `d4cbdb3`.
- `docs/art/station-redesign.md`: visual provenance, screen implementation and isolated measurements.
- `docs/manual/integration.md`: manual/calculator API and evidence.
- `scenes/outcomes/README.md`: launch and comic-failure scenes, connected to the host result from the cockpit and control room.
- Resources and disclosure: external references and eligibility gaps.

# 4. Resources, provenance and AI disclosure

*Source file:* `docs/hackathon/resources-and-disclosure.md`

Project: Launch Control. Team: sterlings. Event: NASA Space Apps Challenge 2026 — 15th Edition.

## Official challenge resources (use these in the portal)

The challenge's Resources tab was supplied on 15 November 2026. The resources the project relates to are **Design and Develop Science Missions** (the launch, space, ground and user segments) and the **Mission Design Lab** (collaborative, discipline-based studies); **Concept Maturity Levels** only inspire a playful indicator in the designed expansion. Exactly how each is used, and which official resources are *not* used, is in challenge-alignment.md §4. Its portal-field draft replaces the older draft below.

## Additional NASA references used in this documentation

Accessed 15 November 2026. These sources support the educational explanation below. They are **not** listed in the challenge's Resources tab, so they complement, but do not replace, the official resources above.

| Resource | Concrete use in this package | What it does not establish |
|---|---|---|
| NASA, JSC Mission Control Center | Explains specialized ground roles and the Capcom connection to the crew. Used in the educational note and presentation's NASA-resource slide. | No downloaded telemetry, operational checklist or scientific dataset is implemented. |
| John Uri / NASA, Building on a Mission: The Houston Mission Control Center | Provides context for consoles and shared wall displays. Used to explain the room's visual organization. | The game's room is not an exact reconstruction; this citation does not establish the provenance or reuse rights of local reference images. |

### Educational note included in the final documentation

NASA describes mission control as a coordinated set of specialized roles, with Capcom communicating with the crew. Launch Control compresses this relationship into two players: one observes and acts, while the other interprets instructions. The simplification helps explain cooperation but should not be mistaken for operational training. NASA role reference.

NASA's account of Houston mission control describes consoles and shared large displays. The game interprets that arrangement through original geometry and an in-game rocket camera. It does not display a real mission feed. NASA historical reference.

### Earlier portal draft (superseded by challenge-alignment.md §4)

The team used NASA's Artemis II page, Missions directory, A to Z of NASA Missions, and Rocket Systems Area - Design and Construction as research references for analyzing real mission concepts and adapting them into an approachable game. We examined mission context, the relationship between a spacecraft and its launch vehicle, and the importance of preparation and equipment checks. We translated this

research into a shortened mission-inspired flow: prepare, observe, communicate, act and receive a result. The cockpit and mission-control roles make cooperation visible, while readable controls, simple geometry and a limited set of tasks reduce the complexity for players.

The mission flow is deliberately compressed for gameplay. Real operations involve many specialists, extensive procedures and much longer timelines; our game concentrates that complexity into two complementary roles and short interactive challenges. The countdown, light-to-symbol puzzle and success/error feedback are authored mechanics, not NASA procedures or numerical models. The project uses NASA information as educational and design reference material, not live telemetry. The Mission Control Center articles above provide additional context added during documentation preparation.

### Source-to-design analysis

Sources supplied by Gael and reviewed on 15 November 2026. Gael confirmed that the team analyzed real-world material and adapted it for a friendly user experience and a reduced mission flow. The table describes that conceptual adaptation; it does not claim exact engineering reproduction.

| Resource | Real-world information analyzed | Adaptation for the player | Scope and limits |
|---|---|---|---|
| Artemis II | Crewed mission context and the distinct roles of SLS and Orion. | Grounds the spacecraft/launch setting in a recognizable mission context. A short cooperative sequence represents preparation and progress toward a launch outcome. | The game does not reproduce Artemis II's trajectory, duration, systems or operating procedures. |
| Missions | Mission categories, purposes and the organization of mission information. | Supports a mission-oriented framing: players work toward a shared objective through a small number of understandable tasks. | Used for background research; its catalogue is not imported into the game. |
| A to Z of NASA Missions | A reference index for finding and identifying real missions. | Supports the research process and mission vocabulary behind the simplified space setting. | A discovery/reference source, not a distinct gameplay system or numerical dataset. |
| Rocket Systems Area - Design and Construction | Ground-test facilities, instrumentation and equipment checkout before operation. | Informs the emphasis on preparation and checking equipment. The game expresses these ideas through observing a panel, consulting instructions and confirming an action. | Historical test-facility context, not a rocket-construction manual. The antenna puzzle is an original abstraction. |

### Transformation and visible result

1. **Analyze reality:** identify mission objectives, vehicle roles, ground support and verification from the cited material.
2. **Choose understandable concepts:** retain preparation, complementary responsibilities, communication and action feedback.
3. **Adapt to play:** compress the timeline, reduce the number of roles and systems, and replace complex procedures with clear interactive tasks.
4. **Present accessibly:** use a friendly visual style, recognizable cockpit and control-room spaces, readable labels, and state feedback combining text, icons and color.
5. **Show the result:** the integrated antenna and numeric-code scenarios in `scenes/main.gd`, the cockpit and tower scenes in `scenes/cockpit/` and `scenes/tower/`, and

`scenes/modules/antenna_view.gd` demonstrate this simplified interaction. Separate outcome scenes should still be presented according to their integration status in the demonstrated build.

This is the team's design rationale, not evidence that the game produces measured learning gains. No images, numerical values or instrument specifications from these four pages were imported by this documentation update.

### Challenge-resource verification

- Official challenge: Space Mission Design Game.
- Publisher of all four references: NASA; web-page titles and URLs are recorded above, reviewed 15 November 2026. They are live web resources rather than a versioned dataset.
- Use: qualitative analysis and adaptation, described above; no numerical data extraction or simulation validation is claimed.
- Remaining check: verify which references appear in the official challenge Resources tab and whether this application satisfies its requirements. Source usage is now documented; eligibility is not certified by this record alone.
- If NASA media is later reproduced, record the individual asset credit and applicable reuse terms before distribution.

See the organizing-team resource guidance.

## Third-party and original-material register

This is the documented minimum, not a completed distribution audit.

| Item | Use / provenance | Release action |
|---|---|---|
| Godot Engine 4.7.2 | Engine, renderer, networking and built-in facilities; official license information | Include required engine and bundled-component notices in the exported distribution. |
| Godot ENet multiplayer | Networking through the engine's `ENetMultiplayerPeer` | Preserve applicable bundled dependency notices; do not describe it as a NASA library. |
| Git and GitHub | Version control, repository hosting and CI | Identify as development tools, not project content. |
| Spec Kit templates | Planning templates; `specs/LICENSE.spec-kit` is tracked | Retain its license when distributing those templates. It does not license the whole game. |
| OpenAI Codex / ChatGPT and Anthropic Claude / Claude Code | AI-assisted coding, debugging, design elaboration and documentation evidenced by this work/history | Use the disclosure below; verify exact tool/model versions only if recorded. |

| Item | Use / provenance | Release action |
| --- | --- | --- |
| Team-authored geometry, SVG icons and game rules | Original project assets, some created through AI-assisted code | Preserve authorship and AI-assistance notices; confirm contributor distribution permissions. |
| `img diseno/Cohete` reference images | Cockpit layout references; authors and rights not established by their filenames | Do not package or reproduce the photographs without verified permission. |
| `img diseno/control/gettyimages-165602827-612x612.jpg` | Reference only, visibly marked stock image | Do not distribute, crop out its watermark, or treat it as a licensed game asset. |
| `img diseno/control/Mission_control_center.jpg` | Reference only; local filename does not establish ownership | Trace the original source before any reuse. Original game geometry can be shown instead. |
| Planning reference screenshots from the agents repository | Analyzed for interaction principles, not bundled | Keep third-party game screenshots out of the release. |
| Fonts, sound effects, music and any later plugins | Per the team's asset register (`docs/art/asset-register.md`, T041): the only font is Godot's built-in Open Sans (SIL OFL 1.1, covered by Godot's notices); no third-party textures, models, sounds or music are in the project. | Re-check the register before export; any later asset must be added there with its source and license. |

## AI tools field — draft disclosure

Yes. We used OpenAI Codex/ChatGPT and Anthropic Claude/Claude Code to assist with GDScript implementation, automated checks, debugging, documentation, English submission drafting and the elaboration of original scene geometry and interface layouts. The development history includes AI-assisted suggestions as well as code generation. The team selected and revised the project direction and requested specific changes; we do not claim that every design decision was made without AI assistance.

AI-generated code was evaluated through automated checks and rendered-scene inspection, with the limits documented in the repository. Existing commits use `AI-Assisted: true` where applicable. No NASA logo or insignia was generated or altered for this documentation package. Exact AI model versions and a complete prompt history are not claimed where records are unavailable.

**Before submission:** the team must verify the origin of the central idea and the extent of AI contributions against the applicable rules, particularly the requester's supplied core-ideation restriction. This disclosure does not resolve that eligibility question by itself.

## Media labels and privacy

- Use a persistent, readable overlay for generated imagery/video: **“AI-generated / AI-assisted prototype visuals.”** Keep it away from the countdown and controls. For ordinary gameplay capture of AI-assisted assets, ask the organizer how its media rule applies; the explicit overlay is the preparation choice for this pack.
- Video/audio descriptions and file metadata should identify AI tools and their role. State whether narration is human or synthetic; do not invent an AI voice credit.
- Example metadata description: “Launch Control by sterlings. Gameplay captured from the team's Godot prototype. Code, scene construction and documentation developed with AI assistance from OpenAI Codex/ChatGPT and Anthropic Claude/Claude Code. No live NASA telemetry.”
- This text pack creates no video, audio or slide export. Labels still need to be applied to the actual final media.
- Use gameplay without webcams, photographs of participants or personal desktop notifications. Under the supplied minors policy, use only a confirmed adult narrator or silent English captions, and exclude minors' representations.
- Do not use NASA branding as decoration or imply NASA endorsement. Textual source attribution is sufficient for these drafts.

See the organizing-team AI guidance and verify the final media against the current full terms.

# 5. Presentation structure and production criteria

*Source file:* `docs/hackathon/presentation-criteria.md`

Team: **sterlings**. Project: **Launch Control**.

This is a specification for later production, not a slide deck or video. All slides, narration and gameplay annotations must be in English. Use seven slides, 3 minutes 50 seconds of content and a 10-second buffer.

| Slide | Time | Required content | Production criteria |
|---|---|---|---|
| 1. Launch Control - sterlings | 0:00-0:20 | Project, team, six members, NASA Space Apps Challenge 2026 and Space Mission Design Game. | One original gameplay image; readable title; compact member list. Do not read every name aloud. |
| 2. Making Space Missions Approachable | 0:20-0:45 | Real mission complexity and the objective of making coordination understandable through play. | One problem and one intended benefit. Do not claim measured learning gains. |
| 3. Two Roles, One Shared Mission | 0:45-1:20 | Astronaut and mission control, their different information, and Prepare -> Observe -> Communicate -> Act -> Receive Feedback. | Side-by-side perspectives with labels explaining what each player sees and does. Explain cooperation before playback. |
| 4. Gameplay Demonstration | 1:20-2:35 | Observe the antenna clue, consult the manual, communicate, submit an action and show feedback. | Actual working gameplay with English annotations. Keep controls and countdown legible. Label separate previews; include a launch outcome only if integrated in the demonstrated build. |
| 5. NASA Mission Segments, Made Playable | 2:35-3:05 | Official resources: Design and Develop Science Missions (space, ground and launch segments mapped to cockpit, control room and outcome) and Mission Design Lab (specialists collaborating); designed next stage with trade-offs; simplification of roles, controls and mission stages. | Three columns: Real-world reference -> Adaptation -> In-game example. Readable source credits. Explain compressed timelines and authored mechanics. |
| 6. Implementation and Evidence | 3:05-3:30 | Godot, GDScript, two-player networking, host validation and verified tests. | One simple architecture diagram and one verified result. Distinguish local testing from Internet testing. |
| 7. Value and Next Steps | 3:30-3:50 | Communication, problem-solving, next validation steps and AI disclosure. | Close with "Two roles. One shared mission." |

## Gameplay sequence: 75 seconds

| Duration | Footage | What viewers should understand |
|---|---|---|
| 10 s | Both role perspectives | Each player has different information. |
| 15 s | Astronaut observes and reports | Accurate observation starts the interaction. |
| 20 s | Mission control consults the manual | The other role provides missing knowledge. |

| Duration | Footage | What viewers should understand |
|---|---|---|
| 20 s | Astronaut acts; feedback appears | Communication produces a visible result. Include error/retry only if it fits clearly. |
| 10 s | Result and transition | The task depended on both players. |

## Completion criteria for the future presentation

- Exactly seven slides, one main message per slide; no extra cover or credits page.
- Consistent friendly visual style, original gameplay images and readable labels.
- English throughout; annotate or cover Spanish interface text in presentation visuals.
- Sources and AI disclosures remain readable without covering gameplay.
- Rehearse with the actual clip; verify runtime after export. Planned timing is not a measured recording.
- Keep the four-minute talk separate from the portal's short video demo. For the portal slide option, use representative stills and check the official requirements.

Slide 2 is 25 seconds and slide 3 is 35 seconds; the total remains 230 seconds. See the resource record for source-to-design rationale.

# 6. Portal Project Demo script (29 seconds)

*Source file:* `docs/hackathon/portal-demo-29-seconds.md`

This is a production script, not an exported video. Target 29 seconds to leave margin under the supplied 30-second ceiling. Do not append a separate intro, logo animation or credits after 0:29.

Team: sterlings. Project: Launch Control. Event: NASA Space Apps Challenge 2026 — 15th Edition.

## Shot list

| Time | Picture | English on-screen text |
|---|---|---|
| 0:00–0:04 | Cockpit/control-room split screen. | "Launch Control — sterlings" and "Space Mission Design Game" |
| 0:04–0:10 | Astronaut observes a clearly visible light pattern. | "Astronaut: observe and report" |
| 0:10–0:16 | Mission control opens the antenna manual and finds the matching rule. | "Mission control: read the procedure" |
| 0:16–0:23 | Astronaut selects the matching symbol, confirms; both views show the real result. | "Communicate. Confirm. Solve together." |
| 0:23–0:29 | Stable room/result shot with a concise resource-use caption. | "NASA mission segments, made playable: space + ground → launch." |

The caption refers to the official resource *Design and Develop Science Missions* (see `challenge-alignment.md` §4). It frames the roles; it does not claim NASA data or procedures.

## English voiceover

"Launch Control, by sterlings, turns space operations into a cooperative game. The astronaut observes a panel. Mission control reads the procedure. They communicate, act, and solve the task before time runs out. Our Godot prototype separates their information and validates their actions. NASA research informed a simplified mission flow for approachable gameplay. Two roles. One shared mission."

This is approximately 56 words. Record and measure the result; word count alone does not prove a sub-30-second runtime. Use a confirmed adult narrator or English captions without narration under the supplied privacy policy.

## Production requirements

- Include English subtitles even when narration is English; translate or annotate relevant Spanish game labels.
- Keep the AI disclosure readable throughout: "AI-generated / AI-assisted prototype visuals." Include tool attribution in the description and applicable file metadata.
- Record the actual working antenna flow. Cuts are allowed in this production plan, but label time compression and never imply unimplemented behavior.
- The rocket ascent is connected to the round result since `157a6b9`; use it only from the same recorded round. Any isolated preview must be identified as such.

- Use original game capture, not stock cockpit photographs, reference-game screenshots, NASA logos or participant webcams.
- Inspect the exported video's final duration, captions and image legibility.
- Keep this video separate from the 3:50 local presentation and optional longer gameplay footage.