Khronos Group
Request for Proposals

ANARI CTS
Pilot Project

May 2022

Notice

Trademarks

1. Background

The ANARI Working group was established in 2020 to create an open, royalty-free API standard for interfacing applications to existing rendering engines via a portable C API. In late 2021, the provisional 1.0 specification[1] was ratified and released to the public.

This is a pilot project to investigate and complete the design of the ANARI 1.0 Conformance Test Suite (CTS), estimate the effort needed and create a full project plan to implement that design. The project work estimate and plan will inform the funding of a follow-on project to implement the full ANARI 1.0 CTS.

2. Methodology

Khronos has a fixed budget for this project, with payment due on project completion.

This RFP is being circulated to all Khronos members, and any interested company is welcome to respond.

Khronos will establish a project email list and Slack channel for communications regarding this project that any interested Khronos member may join. A short weekly status and progress report should be prepared for the weekly ANARI working group meetings.

All code development shall take place within the Khronos public ANARI SDK on GitHub.

3. Scope

The overall goal of the ANARI CTS is to maximally cover observable behavior from an ANARI device implementation, which are split between visual and non-visual behaviors. The ANARI CTS is expected to be a tool that implements each of the following, using an arbitrary ANARI device determined at runtime:

1. Render a fixed set of images from a collection of test scenes
2. Compare images from two different devices rendering the same set of test scenes
3. Verification of object/parameter info metadata
4. Verification of known object properties
5. List core extensions implemented by a device
6. Aggregated test result reports

The remainder of this section will describe desirable details of each item of the above list.

3.1 Render a set of known test scenes

The ANARI CTS needs a command-line application that can render one or more test scenes that are arranged in a hierarchy for logical grouping. This task itemizes into two parts:

- Defining and constructing test scenes
- Running tests and reporting images/outcomes

Test scenes should favor procedural generation in order to minimize the need for scenes stored on the filesystem. While not a hard requirement, it is suggested that scene generation be done in a scripting language (e.g. Python) bound to the ANARI API to make test scene definitions external to a deployed ANARI SDK.

The work outlined in this document only covers the establishment of such a test scene system: the ANARI Working Group is responsible for defining the complete set of scenes that represent conformance of the ANARI API. Work to generate a comprehensive list of tests may come in a future RFP.

The executable which uses the API to render the test scenes ought to both create an output image file and output file of messages reported by the device. The severity level of outputs captured ought to be configurable at runtime.

3.2 Image comparison “smoke tests”

The ANARI CTS must implement a tool to take two sets of images from the tool listed in 3.1, and generate output image diffs with corresponding text statistics about the differences found. This tool is intended to be used as the core component in determining conformance of tests according to WG accepted ground truth images.

Ground truth image comparisons for judging conformance are only for tests that are considered “objective”. Given that the ANARI API *intentionally permits* diverse interpretations of shading, conformance tests should only detect the existence of some features in an image. Some features (lighting, materials, etc.) will only be qualitatively tested outside the CTS.

An example of an objective test would be to conservatively detect if a triangle is correctly visible in the final image (i.e. not the background color), using any or all of the different indexing modes for triangle geometry. How such a triangle is shaded would not be relevant to the outcome of the test.

3.3 Verification of object/parameter info metadata

The ANARI API provides functions which enable applications to query which object subtypes are available to create, as well as associated parameter metadata of understood parameters. The ANARI CTS must implement a tool which queries all known objects and parameters from core features and core extensions found in the specification.

3.4 Verification of known object properties

The ANARI API provides functions which enable applications to query values from instantiated objects. The ANARI CTS must implement a tool which creates objects in such a fashion that it can validly query properties and verify their correct value. This tool may need to draw from the library of test scenes used to feed the tool outlined in 3.1.

3.5 List core extensions implemented by a device

The ANARI specification lists extensions with known interfaces, called *core extensions*. The ANARI CTS must implement a tool which instantiates a device and outputs a table of whether the device implements each extension found in the ANARI specification.

3.6 Aggregated test results

The ANARI CTS must be able to orchestrate the execution of all previously listed tools and consolidate their results into a single, portably viewable report (e.g. HTML, PDF).

4. Deliverables

The following deliverables are in priority order.

4.1. ANARI 1.0 CTS Design Document

A design document outlining the design of the ANARI CTS framework changes in Scope needed to accommodate ANARI 1.0 testing.

Acceptance Criteria: signoff after discussions with the ANARI Working Group that the design is feasible and practical.

4.2. Full Project Work Estimate and Plan

An estimate of the amount of effort required to implement the above design and a full project plan for completion by May 27th, 2022.

Acceptance Criteria: signoff after discussions with the ANARI Working Group that the project work estimate and plan is feasible and practical.

4.3. Initial CTS framework, Platform and Test Updates

If there is budgeted effort remaining after completing the above deliverables, work should start on implementing the full project plan with agreement from the Working Group.

5. Schedule and Budget

Khronos has a budget of $20k USD for this project and expects work to be complete within two months of project initiation, with payment due on completion and acceptance.

6. Selection Process

Khronos shall designate a Khronos RFP Manager and will use an RFP email list (anari_rfp@lists.khronos.org) that can be used to contact the RFP Manager and all other members involved in the bid selection process. No member making a bid shall be on the RFP list. Any company considering making a bid in response to the RFP should notify the RFP list as soon as possible. Any potential bidder may request additional information and submit questions directly to the RFP manager or on the RFP email list. Any additional Khronos information and RFP clarifications will be distributed equally to all potential bidders.

All bidders should provide the following information in the format of their choice:

- Proposed schedule, assuming work starts in June 2022.
- Confirmation that if your bid is accepted, you are willing to work under the terms of the Khronos Contractor Agreement[2].
- Any issues or risk factors that they wish to highlight.
- Supporting materials, including background materials about their company, highlighting experience and expertise relevant to this project.

RFP responses are requested by **5PM PT on Friday, May 27th, 2022** and should be sent to the RFP list. Bidders may update their bid as they wish before the submission deadline. In exceptional circumstances a requested submission deadline extension may be issued to all bidders at Khronos' discretion.

Khronos will evaluate all bids and select the winning bid based on timescales, and relevant experience and expertise.

Khronos expects to announce the selected bid one week after the submission deadline and will immediately notify all bidders and enter into contract negotiations with the selected bidder to finalize deliverables and payment schedule. Khronos will immediately notify all other bidders once contract negotiations are complete. In the case contractual agreement cannot be reached, Khronos may select an alternative bidder and re-enter negotiations.

Work can start immediately when the contract is negotiated and executed by both parties.

7. Contractors Agreement

The selected contractor will be required to execute the Khronos Contractors Agreement with Milestones and Costs entered into Exhibit B and Contractor Disclosures entered into Exhibit C.

No work shall begin, and Khronos shall be liable for no costs or expenses, until the selected contractor is in receipt of a mutually executed Contractor's Agreement.

It is important that contractors understand that, under the terms of the Contractors Agreement, Khronos will assess progress on a regular basis and reserves the right to terminate or renegotiate the contract in the event of insufficient progress or other issues.

8. Project Continuation

If this project is completed satisfactorily, the selected bidder will be invited to bid on executing the full project plan.

9. References

[1] The ANARI Provisional 1.0 Specification
https://www.khronos.org/registry/ANARI/

[2] Khronos Contractors Agreement template
https://members.khronos.org/document/dl/23303