

ALGODEQ

ALGOrithmic Design Quest
international programming competition

for architecture, cities, and general design
2013/2014

- QUEST:** Deep thought and path-breaking methods.
- Assignment:** Creative work to enable solutions to problems, or astonishing forms, or contributions to society and human culture.
- Compensation:** Recognition and prize money as appropriate.

Competition guidelines ver.131106b

The above invitation was inspired by the newspaper advertisement placed by Ernest Shackleton, polar explorer, calling for participants in his historic expedition to the South Pole. We hope it will be answered by individuals with the courage to venture into the unknown and the ability to open up new paths.

This competition is a call for methods and forms that inspire hope and dreams through new technology, creative logic, and aesthetic intuition.

Its purpose is to encourage the development of new design methods for better architecture and better cities (and, broadly, better design in general), and to recognize groups and individuals who have taken up this challenge.

By introducing outstanding achievements to a wider public, we hope to encourage the further development of new methods in this field.

To that end, this international competition will recognize computer programs that make outstanding contributions to algorithmic design, and outstanding works of architecture created by means of such programs.

We look forward to innovative entries that meet and surpass our expectations.

I - Description of the competition

What is algorithmic design?

Algorithmic design is not simply the use of computers to design architecture.

Algorithmic design is the use of algorithms to generate designs that are good and beyond the capabilities of humans acting alone, and the use of computers to execute such algorithms.

When is a design good and beyond the capabilities of humans alone?

For example, consider a complex curved surface

Regardless of its complexity, any form can be modeled. With enough time, it might be possible to create several models.

Of course, time is a limited resource. When a design is sufficiently complex, modeling it might be feasible only when a computer program is used to create the model more quickly.

But this is still not beyond the capabilities of humans. With a lifetime to devote to the task, a human designer could probably model the form without relying on computers.

How about building this complex form?

Fabricating and machining the many different parts of various shapes and sizes would be a formidable task. But if the model is sufficiently precise, it would not be impossible. Simply create the model and measure it. There were no computers in the age of Gaudí. Of course, this method is also costly and time-consuming. In real life, it would probably be infeasible. But perhaps it should not be called impossible.

From more efficient to higher quality

Both of these paths (design and construction), use computer programs to become more efficient.

Greater efficiency is a common requirement. Greater efficiency expands the range of what can be accomplished in the same time and at the same cost. An expanded range of choices makes it possible to select higher quality.

Sufficient quantity provides the opportunity to improve quality. Depending on how it is used, greater efficiency can open the way to higher quality. (Although of course it can also degrade quality.)

When viewed in this light, it can be said the design and construction of complex forms both become possible only through the use of computer programs, and that this is also part of algorithmic design. Whether it is true to say that depends on whether these methods are used to achieve good designs (high quality).

What about creating form by defining rules?

Consider rolling a pair of dice and moving forward by the number that comes up, turning left or right depending on whether it is an even or odd number. A unicursal figure (drawn with without lifting the pen) can be plotted by executing this rule. A work of architecture could be designed by using this figure as the plan.

It would be a design made with an algorithm.

If we changed the rule to increase the number of dice and drew the line in a 3D matrix, the resulting form would probably be impossible to plot by hand. This would be a plan that would not be possible without a computer program.

However, a plan made this way would be meaningless. It might be a good plan, or not. It would be a design beyond the capabilities of humans, but there is no guarantee that it would be a good one.

If rolls of the dice are meaningless because they are inherently random, consider another method.

For example, we could take the distribution of street noise, overlay the distribution of ultraviolet sunlight at that point, and make a plan from fluctuations in the waveform. But there is still no way to judge whether the plan is a good one. We cannot obtain good architecture simply by using algorithms.

Now what happens if we define an objective for the game?

Suppose we made a plan such that bumps and depressions on the facade created pleasant breezes in the neighborhood. Imagine a garden where pleasant breezes blow during hot weather while cold winds are blocked during the winter.

It would not be easy to design an algorithm to accomplish this. Beyond rules to generate forms, evaluation circuits are needed to generate only forms with the required performance.

And to create the evaluation circuits, we must define values and decide what will be accepted as good.

But it is only then that design becomes algorithmic design.

Some might say that this could be simulated.

It could be simulated in a wind tunnel or by using CFD (computational fluid dynamics). But there are almost innumerable variations in architectural shapes to produce the desired wind.

To find the shape that produces the best wind, we need a computer and an algorithm to look for it.

This would be hard to accomplish manually with wind tunnel tests. It might not be possible by spending more time. Changing the shape of one part affects the others, so that the answer cannot be found by trying one part after another.

This is why a solution becomes possible only when a computer program is used.

Instead of simply setting up rules, generating forms, and selecting the one you like the best, another method is to define an objective, such as what you want to do or what you want to happen.

Then develop a program to generate designs with the specified performance.

In the example above, the performance we want would be pleasant breezes. When it is possible to create an algorithm to obtain the required performance, that method can be called algorithmic design.

The above definition assumes that a computer program is needed to run the algorithm, but this is not an absolute requirement. If it were possible to create an effective algorithm that could be processed by our own brains, that would be a great achievement.

But in most cases, because of the large number of elements and their mutual influence, the problem becomes too complex for our brains to handle.

We will probably need a computer to execute the algorithm.

Algorithms to generate designs that better meet the objective

Let's try a different definition of algorithmic design.

Algorithmic design is the description of an objective, the definition of an algorithm to generate designs that meet the objective, and design that (when necessary) uses a computer program to execute the algorithm.

In the above example, the objective was a space where pleasant breezes blow.

The objective can be set freely. It can be set freely because of the wide variety of conditions in architectural design. It would be impossible to list all of the conditions that architecture is expected to meet.

But there is one expression that sums them up.

In a word, architects are expected to design good architecture.

To create better architecture. The demands made of architecture can be expressed in this single phrase.

Unfortunately, "good" cannot be defined.

There are as many varieties of good as there are designers and users.

Good means values.

Under the heading of good, which is vague enough to cover anything, anyone is free to discover value wherever they please. Of course, it would be possible to list a set of values and offer them as choices, to compare the number of people who agree with each value. But this method would not allow us to compare values that are not in the list (latent values).

Algorithmic design and ALGODEQ

Designs beyond the capabilities of humans alone, and algorithms to generate them

Let us summarize once more.

ALGODEQ is looking for algorithmic design which defines an objective or problem for the sake of realizing better architecture, and which creates an algorithm to meet the objective by generating designs that are beyond the capabilities of humans alone, and which uses a computer program to execute the algorithm.

"Better architecture" is the fundamental premise.

The "objective" is a declaration of where the entry wants to go.

(Architecture can also be taken to mean cities or man-made objects.)

The specified objective is also a "problem" to be solved.

"Beyond the capabilities of humans alone" is the reason for using a program.

"Generating" refers to design concepts that methods that are somewhat different from conventional design.

Algorithms and programs are described separately because we believe that the most important point is the creation of an algorithm to meet an objectives or solve a problem. An algorithm should be defined before being implemented using specific programming tools or techniques, and it should not depend on the program type.

However, ALGODEQ allows a certain amount of leeway in these definitions.

Rather than strict definitions and distinctions between entries that qualify and entries that do not, we think that interpretation should be left up to applicants. We look forward to a wide range of outstanding entries

and hope that among them will be entries that fit the definitions.

In short, we want applicants to apply on the basis of their own interpretation of the definitions of ALGODEQ.

Algorithmic design as flexible science

The approach described here aims to incorporate scientific perspectives and methods into architectural design.

To allow science to be applied to design, a field with amorphous outlines, science may need to be flexible enough to handle ambiguity.

The act of designing itself may also change.

In the methodology of algorithmic design, what designers select is the method, and not the final shape or arrangement.

Forms and arrangements are states, which are generated by the method under certain conditions.

The act of selection itself may change.

Selection by the many – crowds, markets, or democracy – may become one area of activity for algorithmic design, along with network effects that propagate themselves after being touched off by some incident.

Or, rather than saying that the act of designing will change, it might be better to say that the mode of designing will change.

Of course, what is wanted here is not automated design, where the design can be completed with a single click. (Which is not possible anyway, at least not for some time.)

What is wanted is a collaboration between humans and computers, with each side contributing what it does best to generate better architecture.

Through cooperation between computers, which do what human brains cannot (high-speed processing of diverse data), and human minds, which do what computers cannot (dream), we hope to reach heights that neither could reach alone.

Two categories: QP / QA

ALGODEQ is a two-stage competition with two categories:

the QP category, for program entries, and the QA category, for architecture completed by using programs.

In the first stage, all applicants should submit presentation materials that explain their entries.

In the second stage, they should submit the programs.

Presentation materials for the first stage should explain the following three points (according to the following and the competition guidelines).

Points for competition entries to explain – objectives, method, results

a. Objectives

First, provide an easy-to-understand statement of what the entry wants to accomplish, what value it tries to provide, what problem it tries to solve, and what it considers to be a good solution.

From among the many questions posed to and conditions imposed upon architecture and the city (and the world of design in general), which does the entry try to answer or solve?

As in the example of the garden above, what is wanted here is not purely functional.

The value of the solution could be beauty, at what seems to be the opposite pole of function.

Some may object that beauty is personal, arbitrary, and not public enough.

This is unfounded, because there are as many values as there are people.

If a program can generate beautiful forms, it should be commended.

Objectives that seem only loosely related may not be so different in algorithmic design – for example, high-density housing that aims to have efficient privacy and also aims to have a beautiful form.

It may be possible to achieve both objectives with the same (or similar) methods and techniques.

Different worlds are driven by similar mechanisms. This is one of the characteristic properties of algorithmic design.

In addition to function and beauty, there is no limit to the problems that architecture is called upon to solve.

Adapt to user requests, conform to the look of the neighborhood. Lower the cost, use local materials, build it quickly.

Connect people to each other, reduce the number of disputes, make it so that people of different beliefs and customs can live together.

Make it dramatic, astonishing, moving. Make it fun, happy, new.

Respond to every type of demand. Trigger creativity. Offer encouragement.

Even the example above where the plan was decided by rolling dice could be an objective, if the concept is to establish such a design procedure and follow it. This would be close in character to art.

However, not all objectives and values have the same importance.

Whether its values can be shared with others is an important point that will figure in the evaluation of an entry. The value of values.

There is no single standard for judging values.

There are various objectives, and various values.

Function, structure, productivity.

Form, aesthetics, concept. Spatial richness, response to history and to the environment.

Social worth, public value, general applicability.

Arrangement, configuration, hierarchy, order.

Effect, action, consonance, conversion.

Transformation of spatial concepts, stretching of physical limits, removal of temporal restrictions.

We want to be convinced that the objectives and values you chose to pursue are important.

b. Method

Next is an explanation of the thinking behind your attempt to meet the objectives (values, conditions), and the method employed.

More than an explanation of programming technique, this is how the program was constructed.

It is an explanation of the design of the program, or in a broad sense an explanation of the algorithm.

We would like to see a clear explanation of the originality and effectiveness of your algorithm and method.

c. Results

Further, provide a visual commentary using the results of the program that explains how the program was used to meet the specified objectives (values, conditions).

At the same time, explain how the results are superior when compared to results that would be obtained without using this method.

From this world, and for a new world

We look forward to excellent entries that demonstrate creativity and an understanding of the perspective envisaged by the organizers, but also to unexpected results that break through limits and barriers with new concepts and a flexible approach.

II - Summary

1- Type :

- ▶ Two-stage competition

2- Categories :

- ▶ Entries should be submitted in one of the following categories.

ALGODEQ_P (= ALGOarithmic Design Program) category (called “QP” category below) :

- ▶ Category for original computer programs created by the applicant that embody algorithmic design.

ALGODEQ_A (= ALGOarithmic Design Award) category (called “QA” category below):

- ▶ Category for outstanding works of actually built architecture, designed by using an original computer program that embodies algorithmic design and submitted together with the program.

3- Eligible computer programs (both QP and QA categories) :

- ▶ Computer programs embodying algorithmic design (called simply “programs” below), in any programming language or format. (However, see the requirements described below in “17- Entry programs”)
- ▶ Here and below, "program" refers to programs written in general-purpose programming languages (C++, etc.) and also to programs written in visual programming languages (Grasshopper, Processing, etc.) as well as to programs that depend on specific applications.

4- Eligible works of architecture (QA category only) :

- ▶ Built works of architecture to which algorithmic design has made a significant contribution to realization of one or more principal concepts, which would have been difficult to realize without the use of the relevant computer program. Interiors, temporary installations, and furnishings are not eligible in this category. (They should be submitted in the QP category.)

5- Entry qualifications :

- ▶ The competition is open to researchers, architects, designers, planners, system engineers, students, and all others with an interest in the field of algorithmic design worldwide.
- ▶ Copyright to a program entered in the QP category belongs to the applicant.
- ▶ The architect of the entered project shall be the entry representative for entries in the QA category. The holder of the copyright to the relevant program shall be included among the applicants.
- ▶ Entries will be accepted from both individuals and groups (joint entry). (Joint entries shall specify one individual as representative of the group. Entries may not be submitted in the name of companies, universities, laboratories, etc.)
- ▶ Committee members and the juries cannot enter the competition. Students, graduate students, researchers, assistants, staff members, and collaborators who are associated with the company or university department of a jury member are eligible to enter, but in that case the jury member will not participate in the discussion of their entry.

6- Prizes :

- ▶ ALGODEQ Grand Prix (one): 1,000,000- Yen
- ▶ Outstanding achievement prizes by category, and subcategory prizes
- ▶ Sub-categories will be created by the jury on the basis of the distribution of item selection in the registered entries. Judging will be carried out for each sub-category.

7- Announcements :

- ▶ The competition will be announced on the competition website, on competition announcement sites, in magazines, and elsewhere.
- ▶ Winning programs will be published, time to be determined.

8- Schedule :

- | | |
|-------------------------|---|
| November 4, 2013 | Competition announced and registration starts |
| December 9, 2013 | End of inquiry period |
| December 23, 2013 | Application period starts |
| February 3, 2014 | Registration deadline |
| March 31, 2014 | First stage application deadline |
| May 26, 2014 | Notification to entries passing first stage selection, application period starts for second stage |
| June 23, 2014 | Second stage application deadline |
| September 8, 2014 | Publication of award candidates short list |
| October 20, 2014 | Award announcements |
- ▶ Uploading of entries must be complete by the end of the specified date. (Coordinated Universal Time)

9- Organizers :

ALGODEQC (ALGOrithmic Design Quest international programming competition Committee)

10- Cooperation :

ETH: Swiss Federal Institute of Technology Zurich, CAAD (Switzerland)
 Graz University of Technology, Institute of Architecture and Media (IAM) (Austria)
 Tamkang University, Department of Architecture (Taiwan)
 Tokyo City University, Faculty of Urban Life Studies (Japan)
 Architectural Institute of Japan (Japan)

11- Jury evaluation criteria :

► Algorithmic design program evaluation criteria: common for QP and QA categories:

- Overall merit:** Objectives, problem resolution strategies, and results are all on a high level.
- Innovation:** Viewpoint, methodology, and results are new.
- Elegance:** The methodology is elegant. For the same function, simpler.
- Technical merit:** Creative use of programming technology and techniques.
- Ease of use:** Users can achieve the desired results with simple operations.
- Genetic influence:** Likelihood that the demonstrated method will have successor programs (descendants).
- Appeal:** The approach is appealing and makes one want to use it.

► Architecture evaluation criteria: QA category:

- Merit:** The overall merit of the design as a work of architecture
- Use of algorithmic design:** The importance of those aspects of the design that use algorithmic design, and the quality of the results.

► Evaluation criteria will not be weighted equally, but according to the judgment of the jury members.

12- Application procedure :

▶ Visit the ALGODEQ website <http://algodeq.org> (called “the website” below) to complete all procedures related to the competition.

1. Download the competition guidelines.
2. Register for the competition, and receive your registration code by e-mail.
3. Complete the required registration information, and upload your first-stage submission file.
4. If your entry is selected for the second-stage short list, upload your second-stage submission file.

▶ Registration is free, and there are no charges to enter the competition.

13- Materials to be submitted :

▶ First-stage presentation materials:

1. Presentation file
2. Short movies

▶ Second-stage program materials (second-stage short list entries only):

3. Program (executable file, etc.)
4. Data
5. Required documents

III - Submission material details

14- First-stage presentation materials :

1: Presentation file

One PowerPoint presentation, containing up to 30 slides. File format: .pptx

The maximum file size is 20 MB. Include your registration code on each slide.

Still pictures only (do not use movies, animation or transition effects).

The presentation should contain at least one slide on each of the following five topics, in the order indicated (a to e). Include the topic title on each slide, and include a series number when a topic is explained with a series of slides, (Example: a-Objectives 1, a-Objectives 2, ...)

Use as many slides as you wish for each topic.

a-Objectives, b-Method, c-Results, d-Technical, e-Operation

Upload a single ZIP file containing item 1.

2: Short movies

► The following two movies.

Include your registration code in the first and last scene of the movie.

2- (1) : An easy-to understand explanation of how to use the program and the program's operating states.

2- (2) : An easy-to-understand description of the results obtained with the program and their meaning (how results differ depending on program parameters, etc.).

For the file format, length (time), data volume, upload method, etc., see the notice to be posted on the competition website.

15- Explanation of the entry :

► First-stage presentation materials should explain items “a” to “e” below in an easy-to-understand fashion. Images and illustrations may be used as desired.

Explanations should be designed to convey the content and value of the program to members of the jury even when the program itself cannot be started or operated.

For more information, see the “Description of the competition” at the start of these guidelines and “17-Entry programs”.

a. Objectives

1. Declare the problem that the program is intended to solve.

Problem areas may be chosen freely by applicants from any field that interests them, including society, the environment, aesthetics, history, regional culture or economy, urbanism, personal interests, etc.

2. Explain specifically why the problem is important.

3. Explain approaches to the problem taken up to now, and the results that these approaches have produced or failed to produce.

b. Method

1. Explain the method selected to solve the problem, and the thinking behind the method.

2. Explain the original features of the method and why it is superior.

c. Results

1. Present results generated by the program (designs, etc.). Note when images are photos of the actual work (as opposed to computer graphics).

2. Explain how results obtained with the program are superior to results obtained without the program, giving concrete examples.

► In the QA category, in addition to the above items, explain items 3, 4, and 5 below.

3. Explain what is interesting and appealing about the entry project as a work of architecture.

4. Explain where the program was used in the design process and on which parts of the design, and how the program contributed to realization of important and appealing features in the entry’s design.

5. Explain specific aspects of the entry project that are superior when compared to an attempt to achieve the same objectives without using the program.

d. Technical

1. List the programming technologies, libraries, etc. used by the entry program.

2. If the program uses original algorithms or techniques, explain them.

3. If the program is based on an existing program, explain that.

e. Operation

1. A normal user should be able to use the program with no explanation other than the presentation and movies.

2. When a special procedure is required to run the program, it should be explained.

16- Second-stage program materials (second-stage short list entries only) :

- ▶ Upload a single ZIP file containing items 3, 4 and 5 listed below.
For the data volume, etc., see the notice to be posted on the competition website.
- ▶ The deadline for submission of second-stage entries is soon after notification of first-stage selection results, so applicants should complete their programs by the time of their first-stage application.
- ▶ Programs may be disqualified if they do not perform as explained in the first-stage presentation materials.

3. Program

- ▶ The program file and all required supporting files, stored in a single directory
OS: Windows (Windows7, Windows 8, Windows 8.1), Mac OS X.

4 : Results data

- ▶ Save the following in one folder.

4- (1) : QP category only:

- ▶ When the entry program is of the type that produces architectural or other 3D objects, one or more 3D printer data files produced by the program must be included. File format: .stl

4- (2) : QA category only:

- ▶ Design data for entry project.
For the file format, etc., see the notice to be posted on the competition website.
3D original CG data, main CAD drawing data, images (site plan, plan of each floor, main sections, main elevations, functional and other explanatory diagrams, etc.)
- ▶ 3D printer data: One or more data files. File format: .stl
- ▶ Specifications of entry project.
For the file format, etc., see the notice to be posted on the competition website.

5. Copyright declaration file

- ▶ This is a personally signed document declaring that the applicant is the holder of the copyrights to the program, entry project, and other materials, and that the program does not infringe on the copyright of any other person. This document should not be encrypted.
Download the copyright declaration file from the competition website. File format: .pdf

17- Entry programs (second-stage short list entries only) :

- ▶ Entry programs should be easy to operate by anyone. During evaluation by the jury, it should be possible for normal users without special programming skills to operate the program.
- ▶ Operating procedures should be explained by the presentation and short movies, without requiring users to consult an operation manual.
- ▶ If for any reason the program cannot be started or operated during evaluation by the jury, then judging will be based solely on other application materials, and in some cases the entry may be disqualified.
- ▶ There should be no user registration or verification and no expiration period. No information should be collected about users.
- ▶ There are no restrictions on the language or tools used by entry programs.

Programs may be developed from scratch or may make use of other software, including commercial libraries, development kits, and authoring tools. However, programs must run in a standard PC machine environment. Programs that run only on a supercomputer, mainframe computer, game machine, smart phone, tablet, or other special execution environment are not eligible.

- ▶ Pseudocode (or flow chart) to explain the original logic of the program is required.

(Regarding pseudocode, see the description and links on the competition website)

When data files are required to use a program, the data files must be submitted with the program.

When a special procedure is required to run the program, it should be explained in the presentation file.

- ▶ The program format shall be as follows.

1: Standalone software (Executable file)

Software that can be executed as a standalone file, without the installation of any other software or an execution platform. Source code is not required.

As far as possible, this format is the most desirable. (Example: Executable files produced by C++, C#, etc.)

2: Software requiring installation

When the entry program requires the installation of a platform or other software, the entry program (as an executable file) and all of the required software should be included. (Example: Processing, Python, Java, etc.)

3: Plug-in software, etc.

When the entry program requires a commercial or free software application that cannot be installed in the entry program folder, this should be noted in the presentation, together with the application information (name, version, author credits). (Example: Grasshopper/Rhinoceros, etc.)

However, as far as possible, the required software should be included in the entry folder.

At the time of judging, it may not be possible to evaluate the entry on the external application specified by the applicant.

- ▶ In the QA category, an entry can be submitted without the program when the applicant does not wish to submit it, when special steps must be taken to run or operate the program in a PC environment, or for other reasons. Submission of the program is required in the QP category.

18- Common guidelines for first-stage and second-stage entries :

Languages:

- ▶ English

Registration code notation:

- ▶ Include your registration code in presentation screens, movies, data files, submission documents, file names, and folder names.

File names and folder names:

- ▶ Use the following format for file and folder names (RC=registration code, SN=serial number, NAME=name indicating file content, as required)

Programs: RC_prg_SN_NAME
 Presentation files: RC_pst
 Short movies: algodeq_RC_mvi_SN
 Results data: RC_daa_SN_NAME
 Copyright declaration file: RC_dcu_SN_NAME

Example: When the registration code of the applicant is “ab1234”, the short movie folder should be named “ab1234_mvi” and the short movies should be named “ab1234_mvi_01” and “ab1234_mvi_02”.

Submission files

Do not password protect submission files. Upload files that can be opened without permissions.

19- Registration :

- ▶ Register at the competition website. Upon completion of entry registration, a unique registration code will be mailed to the applicant’s e-mail address.
- ▶ The content of entry registration can be changed before submission of the entry (uploading of the submission file), but cannot be changed after submission.

20- Inquiries :

- ▶ Inquiries in the specified format will be accepted on the website for a specified period. Responses to selected inquiries only will be posted for a specified period on the website.
- ▶ It will not be possible to respond to individual inquiries. Inquiries in other than the specified format will not be accepted.

21- How to enter :

- ▶ Apply by following the instructions at the competition website. After confirming your registration information, fill in the entry form and enter the competition by uploading your submission file. Entries will not be return.

IV - Organizers

22- ALGODEQC: ALGORithmic Design Quest International Programming Competition Committee :

(arbitrary order)

Makoto Sei Watanabe (Chairman)

Architect / Prof., Space-Generating Lab, Faculty of Urban Life Studies, Tokyo City University, Tokyo

Michael Hansmeyer

Architect / Director, Postgraduate Studies Program, Chair for Computer Aided Architectural Design (CAAD), Swiss Federal Institute of Technology (ETH), Zurich

Urs Hirschberg

Architect / Prof., Vice-Dean, Faculty of Architecture, Director, TU Graz Field of Expertise Sustainable Systems, Director Institute of Architecture and Media (IAM), Graz University of Technology, Graz

Chen-Cheng Chen

Prof., Department of Architecture, Tamkang University, New Taipei

Theodore Spyropoulos

Director, Design Research Lab, Architectural Association (AA) School of Architecture, London

Dillenburger Benjamin

Assistant Prof., John H. Daniels Faculty of Architecture, Landscape, and Design, University of Toronto

Antonino Saggio

Prof., Department of Architecture and Project DiAP, University of La Sapienza, Rome

Christian Girard

Prof., Chair, the Digital Knowledge Department, Ecole Nationale Supérieure d'Architecture Paris-Malaquais, Paris

Hani Rashid

Architect / Prof., Director, Studio Hani Rashid, University of Applied Arts, Vienna

Marcos Novak

Prof., Director, transLAB, Vice Chair, Media Arts and Technology Program (MAT), University of California, Santa Barbara

Sean Hanna

Reader, Space and Adaptive Architectures, Space Group, Director of MSc/MRes Adaptive Architecture and Computation, Academic director of EngD Virtual Environments, Imaging and Visualization, Bartlett School of Graduate Studies, University College London

Wolfgang Höhl

Assistant Prof., Faculty of Informatics, Chair of Media Informatics, Ludwig-Maximilians University Munich,
Assistant Prof., Faculty for Building, Master Course of Architecture, FH JOANNEUM University of Applied
Sciences, Graz

Mike Xie

Prof., Director, Centre for Innovative Structures and Materials, Royal Melbourne Institute of Technology,
Melbourne

Biao Li

Architect / Prof., Director, Lab. of Architectural Algorithms & Applications, School of Architecture,
Southeast University, Nanjing

Aaron Sprecher

Architect / Assistant Prof., Research Director LIPHE, McGill University School of Architecture, Montreal

Shen-Guan Shih

Prof., Group of Architectural Synergy, Chair of the Department of Architecture, National Taiwan University
of Science and Technology, Taipei

Kas Oosterhuis

Architect/Prof., Director, Hyperbody and the Protospace Laboratory for Collaborative Design and
Engineering , Faculty of Architecture, Delft University of Technology, Delft

Philip F. Yuan

Associate Prof., Director of Archi-Union Architects, the Architecture and Planning Institute of Tongji
University, Shanghai

Jose Pinto Duarte

Prof., Faculty of Architecture, University of Lisbon

Neil Leach

Adjunct Prof., the University of Southern California, Los Angeles,
Visiting Prof., Harvard GSD, Cambridge,
Visiting Prof., Tongji University, Shanghai, Prof., the European Graduate School, Saas-Fee,

Branko Kolarevic

Prof., Faculty of Environmental Design, University of Calgary

Marco Brizzi

Prof., School of Architecture, California State University, International Program Florence

Peter Cachola Schmal

Director, Deutsches Architektur Museum (DAM), Frankfurt

Toru Hasegawa

Architect / Adjunct Assistant Prof., Graduate School of Architecture, Planning and Preservation (GSAPP),
Columbia University, NY

Taro Narahara

Assistant Prof., Digital Design, College of Architecture and Design, New Jersey Institute of Technology,
Newark

Kazuo Hiramoto

Prof., City planning Lab, Faculty of Urban Life Studies, Tokyo City University, Tokyo

Shohei Matsukawa

Architect / Assistant Prof., Faculty of Environment and Information Studies, Keio University SFC,
Fujisawa

Atsushi Takizawa

Associate Prof., Department of Applied Mathematics, Osaka City University, Osaka

Tomohiro Fukuda

Associate Prof., Environmental Design and Information Technology lab, Division of Sustainable Energy
and Environmental Engineering, Graduate School of Engineering, Osaka University, Osaka

Shuichi Asayama

Prof., Computer Science and Design Technology Lab, Department of Architecture and Building
Engineering, School of Science and Technology for Future Life, Tokyo Denki University, Tokyo

Gakuhito Hirasawa

Associate Prof., Graduate School of Engineering, Division of Architecture and Urban Science, Chiba
University, Chiba

Ryusuke Naka

Prof., Chair, Workplace Design Lab, Faculty of Design Engineering & Management,
Kyoto Institute of Technology, Kyoto

Yasushi Ikeda

Architect / Prof., Keio University SFC, Graduate School of Media and Governance, Fujisawa

Yusuke Obuchi

Project Associate Prof., School of Engineering, Director of Global 30 Architecture and Urban Design
Course, the University of Tokyo

Taichi Sunayama

Adjunct Prof., Department of Architecture, Faculty of Engineering, the University of Tokyo

Akira Wakita

Associate Prof., Metamorphic Architecture Lab, Faculty of Environment and Information Studies,
Keio University, Fujisawa

Takashi Chiba

Software Engineer, representative, archi pivot, Tokyo

To be joined by other international leaders in the field.

(arbitrary order)

23- Jury :

Members of the jury will be appointed from among the members of the competition committee and other specialists. To be announced later.

V - Entry conditions

24- Anonymity :

- ▶ Entries in the QP category must be anonymous. Do not include anything in the presentation, short movies, or program itself that could identify the applicant.
- ▶ Entries in the QA category do not need to be anonymous.

25- Multiple entries :

- ▶ There is no limit to the number of entries that can be submitted by an applicant. However, the same program cannot be entered in both the QP and QA categories. Enter in one category or the other.
- ▶ When the same applicant submits multiple entries, and when an applicant in a joint entry is an applicant in another entry, all registration codes must be entered when registering.
- ▶ When multiple entries have been made, and the jury chooses to regard them as belonging to a single proposal, the jury may decide which entry to consider.

26- Entry categories :

- ▶ There are no requirements relating to completed architecture (or temporary installations, furniture, etc.) for programs entered in the QP category. However, when work has been completed by using a program, that may be a plus in the evaluation of the program.
Programs that have been used to design built architecture may be entered in the QP category instead of the QA category. However, they may not be entered in multiple categories.
- ▶ Regardless of the entry category chosen by the applicant at the time of registration, the entry may be assigned to a different category at the discretion of the jury.

27- First publication :

- ▶ There are no requirements regarding the year of creation of entry programs, and no requirement that they be unpublished. However, the fact that a program is unpublished may be regarded as a plus during evaluation.
- ▶ When an entry program has been previously published (or when it uses a previously published program), its publication history must be indicated. (Excepting publication within the applicant's university or company.)
- ▶ There are no requirements relating to the year of completion of entries in the QA category, or whether they are published or unpublished.

28- Copyright :

- ▶ Applicants must hold the copyright to entry programs and entry projects.
- ▶ Entry in the competition will be regarded as a declaration by the applicant that the submitted entry does not violate the copyright of another copyright holder.
- ▶ When a competition entry includes programs or works (including music, video, images, and text, etc.) copyrighted by other than the copyright holder of the competition entry (called “copyrighted material” below), or when the competition entry was created by modifying or reworking copyrighted material, the applicant shall as required obtain the consent of the copyright owner of the copyrighted material before entering the competition, and shall indicate in the application that such consent has been obtained, and as required shall also credit the copyright holder and indicate the name of the copyrighted material in the presentation file. The above applies to all copyrighted material, regardless of whether it is commercial or free software, libraries, etc.
- ▶ In the QA category, when the applicant (the entry representative = the architect) is not the copyright holder of the entry program, the copyright holder of the entry program shall be included in the applicants for the entry.
- ▶ When the copyright holder of the entry program is not the developer of the entry program, the role of the copyright holder in development of the program shall be indicated in the presentation.
- ▶ Information about prize candidate entries will be published on the website for a certain period after their selection.

If any copyright-related objection is received during this period, and the objection is deemed valid, the corresponding entry will be removed from the prize candidate list and another entry will be promoted to the list.

- ▶ Copyright to competition entries belongs to competition applicants.
- ▶ The organizers reserve the right to publish and reproduce the content of competition entries (on the website, in print, lectures, symposia, exhibitions, etc.) without the consent of competition applicants.
- ▶ Corporate co-sponsors of the competition may use some of the competition entry materials in their publicity activities. However, they may not use the entry program.
- ▶ The organizers will not sell any part of competition entries to third parties.
- ▶ The organizers will not reverse engineer, modify, or commercialize any program submitted in this competition without the express permission of the applicant.

29- Sole responsibility :

- ▶ Applicants shall have sole responsibility in all matters related to the copyright status of their entries. When a prize winning entry or any other entry violates the copyright of a third party, and when objections are made related to such violations, the ALGODEQ competition and persons involved in the competition(including the organizers, committee members and jury members) shall not be a party to the dispute and shall bear no responsibility.

30- Submission of supplementary materials, etc. :

- ▶ Applicants may be requested to provide supplementary materials or data for evaluation, publishing, exhibition, etc.
- ▶ Prize winners may be requested to attend a symposium or other events.

31- Payment of prize money :

- ▶ Prize winners will be notified separately regarding the date of payment of prize money.
- ▶ Taxes and other charges applicable to prize money will be included in the prize money (tax-inclusive).
- ▶ Bank commissions and other fees related to transfer of prize money will be included in the prize money.

32- Cancellation of awards :

- ▶ In the case that the rules of the competition are violated, awards may be cancelled, even after they are announced.
- ▶ Awards may be cancelled in the event that a copyright or other dispute arises over a competition entry and the copyright claim or other objection is deemed to be valid.
- ▶ When an award is rescinded pursuant to the above provisions, award winners may be requested to refund the amount of the award.

33- Other notes :

- ▶ When there are differences between the English and Japanese versions of the guidelines or statements on the website, the Japanese version shall take precedence.
- ▶ Unless otherwise noted, statements in the guidelines and on the website apply to both the QP and QA categories.
- ▶ Unless otherwise noted, “program” in the guidelines and on the website refers to a computer program. Likewise, unless otherwise noted, “entry” refers to the program or architectural project of an entry. Also, unless otherwise noted, “guidelines” includes statements on the website.

34- Changes :

- ▶ The content of these guidelines and the website are subject to change without notice.

In the case of changes to rules or guidelines, the most recent version posted on the website will replace all previous versions.

- ▶ The number of prizes, names of prizes, prize money (and its distribution), and these guidelines and other descriptions are subject to change for the purpose of enabling more appropriate administration of the competition. Changes will be made by the jury on the basis of an evaluation of submitted entries and other factors, with the approval of the organizers.
- ▶ Depending on the opinion of the jury and organizers regarding the level of entries, some or all prizes may not be awarded.
- ▶ The content and schedule announced in these guidelines and on the website and elsewhere are subject to postponement or cancellation in the event of force majeure or other unforeseeable events.

35- Administration :

- ▶ Registration, inquiries, and entry procedures will be handled on the website only. The organizers cannot respond to any other inquiries. Prize winners will be contacted separately by the organizers

- ▶ Contact for media organizations and news about ALGODEQ:

ALGODEQ Administration Office:

info@algodeq.org

ALGODEQ Administration Office , AGLOBE, Mejiro –zaka -ST building, 2-3-3, Sekiguchi, Bunkyo-ku, Tokyo, 112-0014, Japan

Note: The above information is for media organizations only.

The Administration Office cannot answer questions regarding participation in the competition.

Persons wishing to participate should contact the organizers via the competition website only.