72nd International Astronautical Congress (IAC), Dubai, United Arab Emirates, 25-29 October 2021.

IAC-21-B3.7.10

Using Computational Techniques for the Optimal Design of Evolving Habitats

Thomas J.-R. Lagarde[a*]

[a] *Space Architecture Technical Committee, American Institute of Aeronautics and Astronautics, 12700 Sunrise Valley Dr #200, Reston, VA 20191, United States*, info@spacearchitect.org, thomas.lagarde@outlook.com
* Corresponding Author

Abstract

The goal of this paper is to demonstrate the possibility of using existing solutions and concepts developed and used for earth applications as a design architecture for outer-space habitats. The future habitats/cities will need to evolve constantly, fixing a form, a system or a program is not the solution to adapt to an environment that we will learn a lot from when we get there. The design for a habitat and its systems will require constant modifications to adapt to changes in the environment, our knowledge of it and/or our reaction to it. Interior and exterior organizations will certainly change rapidly depending on new requirements. To produce an optimal design at a fast pace and correctly we need to use computational techniques such as parametric design or topology optimization. The new design solution should be the best according to a chosen set of conditions. For example: well-being, comfort, ease of operation and construction. With the help of software such as Rhino/Grasshopper and SIMOC we can demonstrate the practicality and the necessity of this approach for future human settlements in any extreme environment.

Keywords: Computational design, SIMOC, Space Architecture, Mars, Human Exploration, Site Planning

Acronyms/Abbreviations

SIMOC	= Scalable Interactive Model of an Off-world Community
ECLS	= Environmental Control Life Support System
RECLS	= Regenerative Environmental Control Life Support System
CAD	= Computer Aided Design
Rhino	= Rhinoceros 3D
Grasshopper	= Rhino visual scripting objects
CO_2	= Carbon dioxide
O_2	= Oxygen
H_2O	= Water
PV	= Photovoltaics
3D	= Three spatial dimensions
4D	= Three spatial dimensions + Time dimension
.json	= JavaScript Object Notation Format
.epw	= EnergyPlus Weather Data Format

1. Introduction to computational design and Scalable Interactive Model of an Off-world Community (SIMOC) software

1.1 Introduction to computational design

Computational design is a field of computer aided design (CAD) that includes many different concepts: designing with data, processing power, parameter setting, generative design, 3D modelling and visualization tools [1]. The 3D modelling and visualization tools have been used extensively in the last 40 years and the use of CAD software for creative tasks is now a standard across many industries such as architectural design or mechanical design.

The design process with CAD software requires external data in order to create correct and code compliant drawings. Different tools such as Excel sheets, standard tables or pre-configured blocks are used by the designers to correctly size their shapes and volumes. Over the years, a library of external standards has been created and the designers currently need to rely on literature to properly draw their parts or spaces. An example of that external set of standards is the international building code [6].

This paper will explain the reasons and the necessary steps for creating a direct link between external data sets and open-source weather, energy and forming tools. After such a link is established, it becomes possible to draw lines, shapes, volumes and dimensions directly and automatically into the design software. The designer doesn't need to translate and import external raw data. The code should also be totally free and modifiable so as to be improved and scaled up.

We need to first explain the different components that will help us in this automatic process. The first one is parametric design, it starts with a set of data. This data set is linked to parameter settings, which can scale up or down a design and influence the result of scripts that will modify in turn the shapes created. This process is still based on the designer input. It gives more flexibility to designing objects since parameter settings can allow a project to be adapted to changing conditions (for example more crew, less energy, etc)

The second component, generative design is based on certain constraints in which the scripts will create multiple propositions, analyse them and find out which ones responds best to the limits set by the designer. It is an intensive computing process. This process allows the creation of a shape or object that is the optimal solution based on the input from the user.

The increase in processing power in the last decade has allowed those processes to be more efficient, even if Moore's law is slowing down because of physical limitations, a threshold has been reached that allows scripts to help the design decisions in an efficient way, many designers have used those tools to achieve good results especially with weather analysis [2].

1.2 Introduction to SIMOC

SIMOC is a research and educational platform for the simulation of a hybrid mechanical and biological regenerative life support system or RECLSS, in a Mars habitat. It is hosted by the National Geographic organization and was created by an international team headed by Kai Staats. SIMOC is freely available online [7] and can be used by anybody, including specialists and non-specialists. The data collected through SIMOC will help to balance the required elements to ensure the survival and well-being of a human crew on Mars. It also aims to determine the minimum amount of cargo to be shipped and the minimum energy consumption for the duration of the mission.

The team behind SIMOC has developed a comprehensive system based on published research [3]. SIMOC is an agent-based model, which is characterized by the simulation of the actions and interactions of autonomous agents. These independent agents exchange currencies such as Oxygen (O_2), Carbon Dioxide (CO_2) or Water (H_2O). There are nine general categories for these agents, sub-divided into smaller individual categories. These categories are Inhabitants, ECLSS, Agriculture, ISRU, Structure, Fabrication, Power, Mobility, and Communication. More information can be obtained by reading the free and downloadable guide [3].

The SIMOC interface is web-based and can be accessed from anywhere on any device [7]. An example of the interface and the first page can be seen in Figure 1.

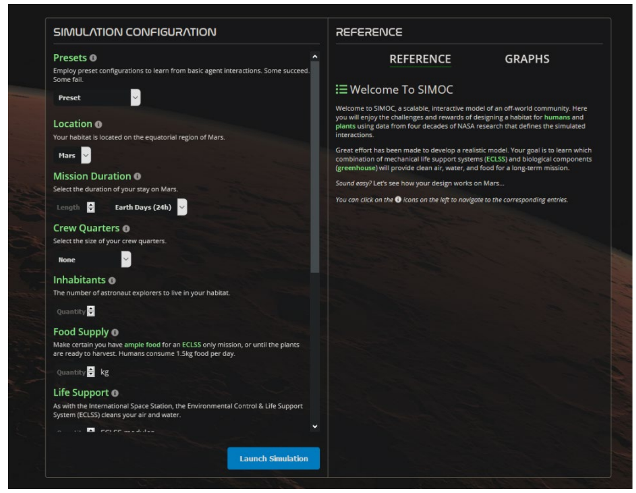


Figure 1. The first SIMOC page where a mission can be configured.

1.3 The integration of the results in a 4D design

The main interface of SIMOC, after configuring the mission, shows numerical and graphical representations of the raw data being created, see Figure 2 for reference. Those numbers and graphs can be used by a designer to build a 3D representation, for example the amount of CO^2 stored can be used to size the storage areas.

Figure 2. A full SIMOC interface. New panels can be added to the interface to have a good view of the ongoing results of the simulation.

After a simulation is complete, the data file can be exported. The data is saved as a .json file type. It is a standard text format of storage for data storage. This result file is the most important data set for this project, we will use it as the foundation to build the different elements of the base.

The initial parameters of the simulation are the dimension of the greenhouse, the dimension of the habitat, the number of PV arrays, the number of battery packs, the type and number of ECLSS modules, the size of the storage tanks and the type and quantity of edible plants.

We can import data from the SIMOC website into numbers and letters that a computational design software can understand. In this instance, we will Grasshopper from McNeel but others such as Dynamo from Autodesk could be used.

To accomplish that task, we are using elements from a plugin for Grasshopper called Jswan, as shown in Figure 3. Those components can extract and store the data needed from the main .json file into the software.

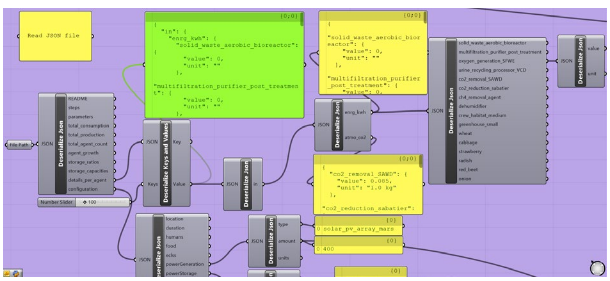


Figure. 3. The SIMOC .json extraction and utilization inside Grasshopper with the JSWAN plugin.

2. Data-driven design with SIMOC and Grasshopper

The goal of this research project is to make available to current and future architects, system engineers, designers and others, the tools to correctly size their buildings and ECLSS components.

This project is open-source, modifiable and capable of integrating new environments and parameters. It is possible therefore to simulate other extreme environments such as the Moon, Europa or others with any amount of crew or storage for any amount of time.

The first step in any design project is to know your environment. For Mars, the temperature, wind speed, cloud cover and pressure data are extracted from a .json file created by Amber Thomas [13]. The .json file was built using a daily minimum and maximum value inside Gale crater.

We will need to create an .epw file with that data, it will be the second most impotant component of this project.

In order for that data to be correctly interpreted by the other tools, it is necessary to convert all the data points into a continuous list of hourly data. The EnergyPlus Weather Data Format (EPW) [10] that we will use requires 8,760 data points, which corresponds to the number of hours in a year. This operation is accomplished by using the different values mentioned before as upper and lower limits and populating the rest with a set of linear values that gradually progress from one limit to the other.

An .epw file requires other entries to be used, such as solar radiation or humidity. We can get those data points from different published papers or websites such as Seasonal and interannual variability of solar radiation at Spirit, Opportunity and Curiosity landing sites [4]. The wind direction and speed are from a paper called Gale surface wind characterization based on the Mars Science Laboratory REMS dataset [5] and humidity levels can be extracted from a NASA website [14].

All the data sets are combined and configured to create the required .epw file, this action is accomplished in Grasshopper by using components such as the ones showed in Figure 4

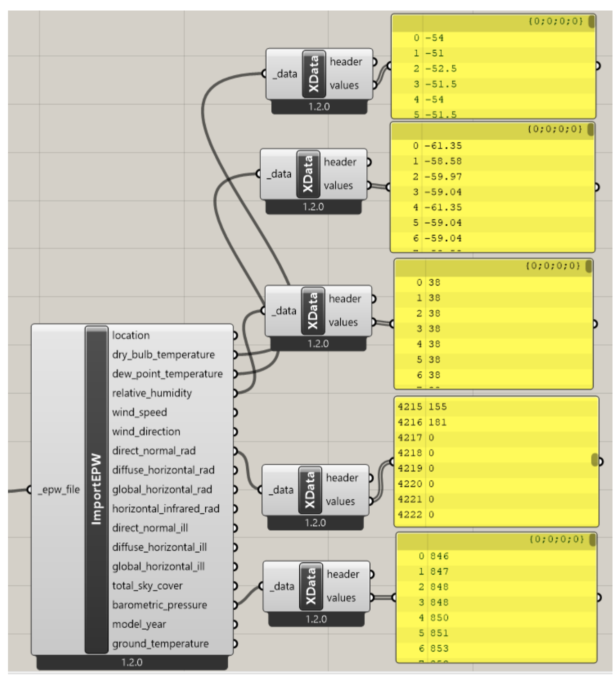


Figure.4 Creation of an .epw file that is as close as possible to the Martian environment.

With both .epw and .json files loaded in the software and correctly integrated in the code, a topography file can be imported. The project uses a topographic file that represents Gale Crater because the Curiosity rover from NASA is currently collecting weather data there. In addition, this site has many different layers of interesting sediments. A perfect environment for a Human exploration mission.

We need to import a CAD file of the terrain in the code, this file can be built of meshes or breps. After importing a terrain file we can start to run a terrain analysis process using a specific part of this code, it will show the user many important visual information about the area selected for the base.

Water flow analysis can show where to find the most interesting samples that were close to water for a long period, see Figure 5.

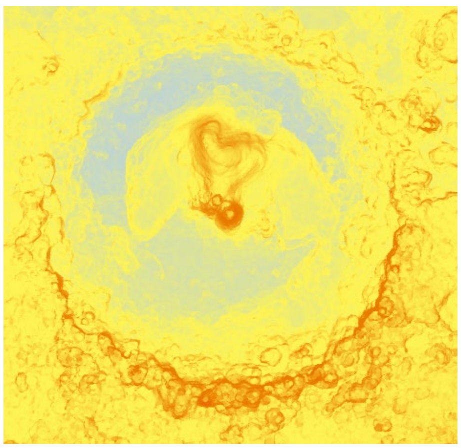


Figure. 5. Water flow analysis of the terrain

An elevation analysis as shown in Figure 6, should be used in combination with pressure and travel time requirements.

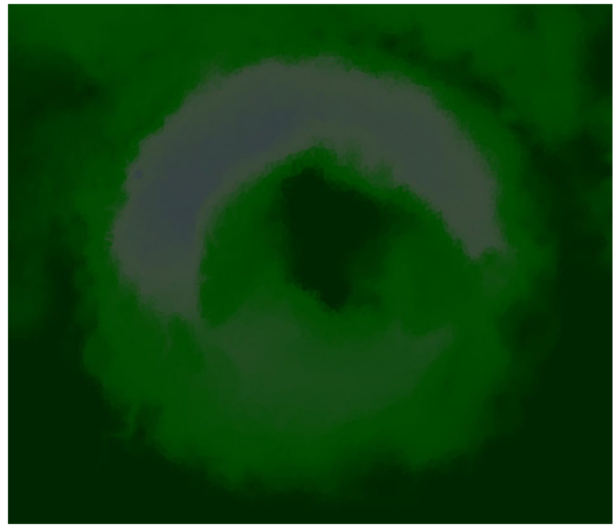


Figure. 6. Elevation analysis of Gale crater.

Finally, a slope analysis is useful when selecting a terrain for the location of the base. A flat terrain will be less difficult to build on than a sloped terrain. In Figure 7, the areas where the slope is too strong are represented in red and should be avoided for the construction of a base, except for radiation shielding measures.

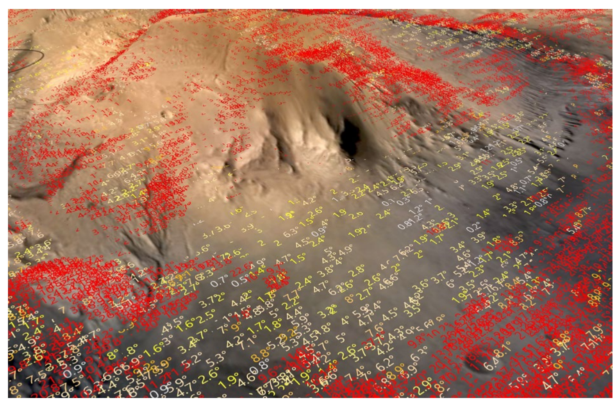


Figure. 7. Slope analysis of the terrain.

After processing and examining all that visual information, the designer can select a site and start laying out the base. The user needs to select CAD geometries (points, lines, volumes, or surfaces) already present in the model space and move them around the topography. The code will automatically place the different structures on the topography and adapt the modules to the terrain where the geometries are located. An example of this automatic layout function is shown in Figure 8.

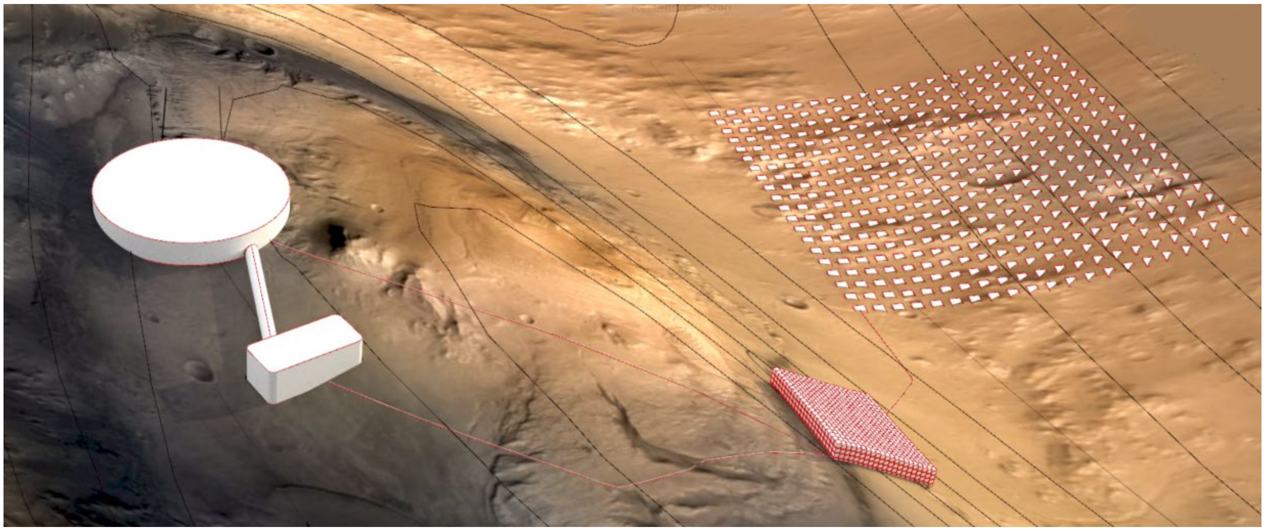


Figure. 8. Automatic placement of the different parts of the base according to the slope of the terrain.

Numerous other components of the base are also automatically placed on the topography such as electrical cables or the tube connecting the habitat and the greenhouse. The electric cables are a good example of the benefits of using this type of code for site planning. It will give the designer the shortest path between two points and therefore optimize the length of the cables. This planning process can help optimize the storage and transportation problems for setting up a base.

We will demonstrate the advantages of using this code for general planning purposes by discussing the case study of the greenhouse.

3. Case study: Greenhouse design

Due to the low amount of solar radiation that reaches Mars, having a transparent Greenhouse is not sufficient to properly grow earth-based plants. The solution requires supplemental artificial lighting.

The code created can be used to optimally orient the greenhouse, it will use the solar radiation data and the sun vectors coming from the .epw file (Figure 9).

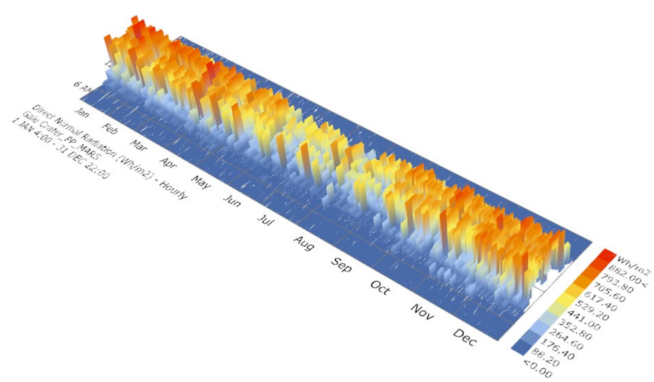


Figure. 9. A 3D representation of the annual solar radiation intensity at Gale Crater.

With the .epw data set, the code will orient the building at different angles and will calculate which orientation will provide the maximum solar exposure with the least amount of heat collected.

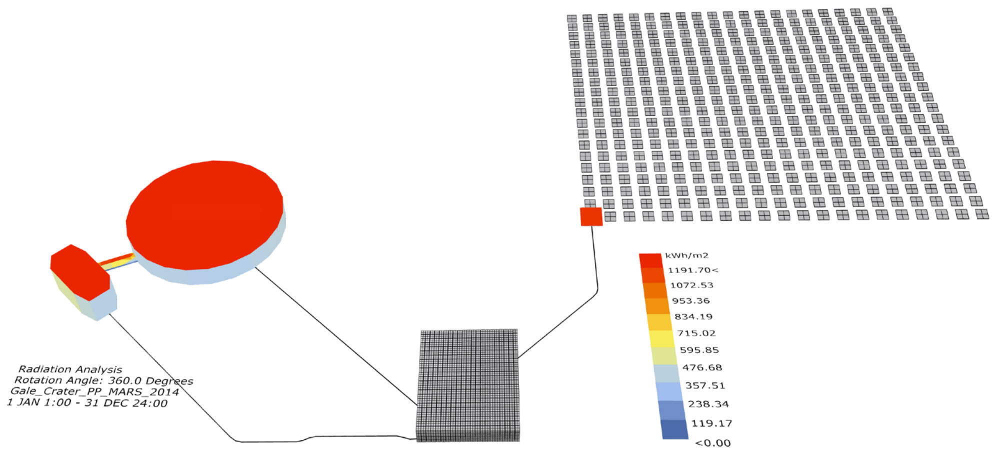


Figure 11. Amount of solar radiation that is reaching each structure in a year

Once the correct orientation is determined and the location is chosen. The greenhouse is placed in the model window is size being determined by the .json file/ After being placed, it can be populated with the plant trays that were used in the SIMOC simulation file.

The SIMOC uses plant trays [3] that measures one meter by one meter. In the SIMOC simulation used for this paper, it was determined that the crew needed 20 trays of wheat, 30 of cabbage, 10 of strawberries, 50 of radish, 50 of red beet and 50 of onions.

Those trays are directly imported into the greenhouse with manual location selection, see Figure 10. Future research will aim to automate the process and provide an optimal placement of the different plant trays.

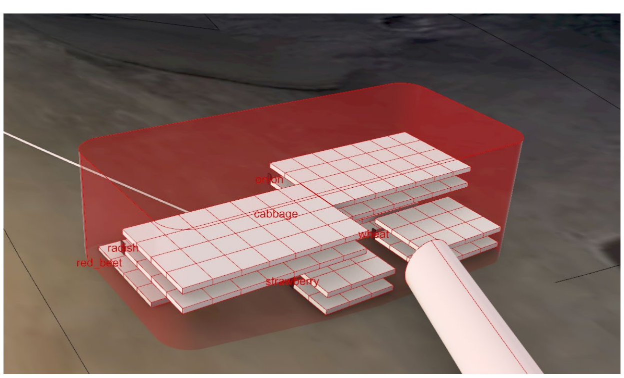


Figure. 10. Manual layout of the greenhouse with automated amount of plant trays.

The number of PV arrays and the number of batteries required is calculated by the SIMOC simulation for the needs of the greenhouse and for general use. The code can optimally orient and place the PV arrays and the batteries on the topography and can calculate the amount of power that will hit each structure, see Figure 11.

This information on a building is useful for cooling and heating purposes, for the PV array it will tell the designer how much power each panel will produce a year. Future evolution of the code will calculate cooling and heating loads for the module.

The footprint of the greenhouse is also an important information, the code will calculate how much material needs to be removed for the greenhouse to lay flat on the topography. This gives the designer helpful information and can improve his decisions regarding the layout of the base.

The greenhouse is a good example for the utility of the code in the planning phase. The designer is given the tools he/she needs to correctly size a base and to choose the best location for its different parts(best implies, the most economical in terms of setup time, storage transported, heating loads and other parameters)

4. Lessons Learned and future evolution

The existing tools for earth-based energy simulation can and should be used for Martian or other extreme environments. The creation of a practical system to link the SIMOC project and different 3D design software is a necessary step in exploring new pathways of simulation and planification.

The possibility of using existing open-source software and data sets allows future designers to start their project with a strong basis, validated by previous research and successful simulations.

The current code is the result of the involvement of just one individual, see Figure12, but it is hoped that by releasing the code to the public, it will be improved upon and will benefit future designers and planners.

The author hope that this project and this code will help to connect the world of computational design and extreme environment habitats.

The free flow of information is essential to the future of space exploration.

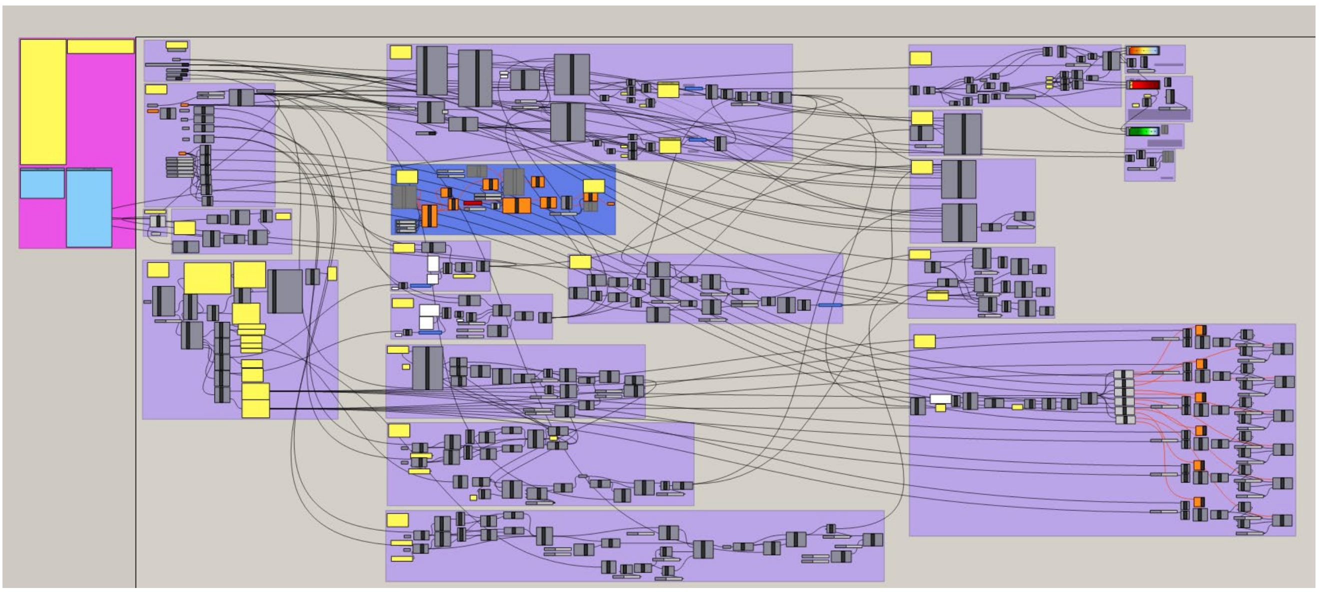


Figure. 12. Latest iteration of the code as of the date of publication of this paper.

Efforts have already been made to adapt the tools to extreme environments. The communities behind those projects have been contacted and are willing to collaborate on new versions of their projects [8, 9, 10]. The process of modifying and validating the existing tools will require the involvement of multiple communities working together [11, 12]. The author welcomes the involvement of everybody and is dedicated to offer his work and his results to the people that are interested in improving the design process for habitats in extreme environments.

Please refer to the websites that the author uses for communication. Type Thomas Lagarde in Researchgate, Academia, Linkedin or ISSUU. You will have access to the updated material, tutorials and a list of people and communities (including the author) you can contact to help improve this project.

Acknowledgements

The SIMOC project team (Kai Staats, Project Lead, Iurii Milovanov, Lead Server Developer, Ezio Melotti, Lead Front-end Developer, Sheri Klug Boonstra, Associate lead, Don Boonstra, Educational Developer, Bryan Versteeg, Space Habitat Architect & 3D Artist

The Grasshopper community

The Ladybug community

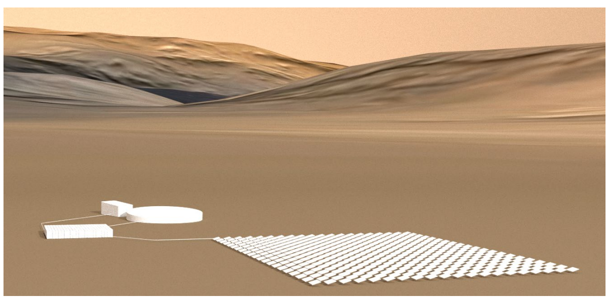
The Openstudio community

The Energy+ community

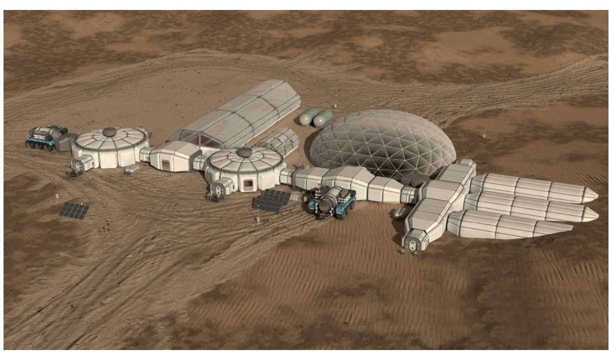
All the people that dedicated their time and energy to provide the community with new tools to improve their workflow and their design process

My family (Isabelle that helped me to go forward) and my colleagues in the industry

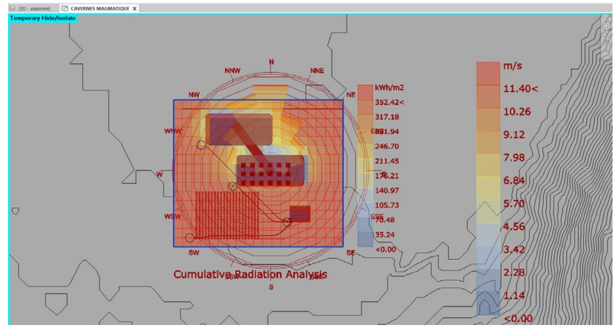
Appendix



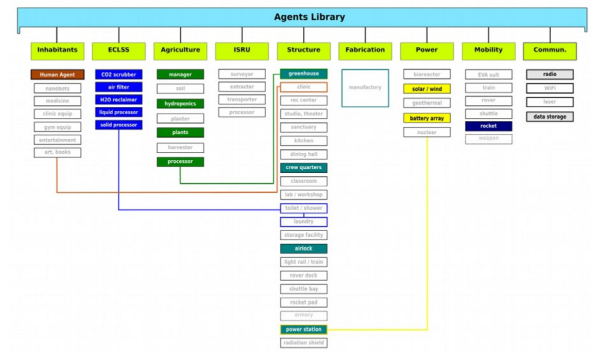
Correct amount of PV panels, battery packs and volume of pressurized environments for four crews and RECLSS according to one of the SIMOC simulation



Possible Mars base, presented by the SIMOC website. This base was designed by Bryan Versteeg Copyright © SIMOC



Integration of the code within Revit using Rhino.Inside.Revit



The list of the agents in the SIMOC project

References

[1] Ineˆs Caetano, Luı´s Santos, Anto´nio Leita, Computational design in architecture: Defining parametric, generative, and algorithmic design, Frontiers of Architectural Research (2020) 9, 287e300

[2] Sahar Soltani, Gabriela Dias Guimaraes, Pan Liao, Victor Calixto, Ning Gu, Computational Design Sustainability A Conceptual Framework for Built Environment Research, Conference: The 38th eCAADe Conference. eCAADe – Education and Research in Computer Aided Architectural Design in Europe

[3] Kai Staats, Project Lead, Iurii Milovanov, Lead Server Developer, Don Boonstra, Educational Developer, A SIMOC Technical Document, ASU, School of Earth & Space Exploration, Interplanetary Initiative Pilot Project

[4] Álvaro VICENTE-RETORTILLO, Mark T. LEMMON, Germán M. MARTÍNEZ, Francisco VALERO, Luis VÁZQUEZ, Mª Luisa MARTÍN, Seasonal and interannual variability of solar radiation at Spirit, Opportunity and Curiosity landing sites, Física de la Tierra 126 Vol. 28 (2016) 111-127

[5] D.Viúdez-Moreiras, J.Gómez-Elviraa, C.E.Newman, S.Navarroa, M.Marina, J.Torresa, M.de la Torre-Juárez, the MSL team, Gale surface wind characterization based on the Mars Science Laboratory REMS dataset. Part I: Wind retrieval and Gale's wind speeds and directions, Icarus Volume 319, February 2019, Pages 909-925

[6] International Building Code, https://codes.iccsafe.org/content/IBC2021P1

[7] WELCOME TO SIMOC (starting portal for SIMOC users), https://ngs.simoc.space/, (accessed 15.08.21).

[8] RADIANCE (Github for RADIANCE users and builders), https://github.com/NREL/Radiance/, (last accessed 15.08.21).

[9] UNMET HOURS (Energy Plus Forum) https://unmethours.com/questions/, (last accessed 10.08.21).

[10] EPW working group, https://github.com/building-energy/epw, (last accessed 08.08.21).

[11] Food for Rhino (Forum for Grasshopper users), https://www.food4rhino.com/fr, (last accessed 16.08.21).

[12] Ladybug homepage (Resource, tutorial and homepage), https://www.ladybug.tools/, (last accessed 17.08.21).

[13] Amber Thomas homepage for code, https://github.com/the-pudding/data/tree/master/mars-weather, (last accessed 15.08.21).

[14] Atmosphere relative humidity Gale crater, https://www.nasa.gov/mission_pages/msl/multimedia/pia16915.html, (last accessed 21.08.21).