Docs

Introduction

In order to present a solution for the challenge **Your Home in Space: The Habitat Layout Creator** a design software that leverages AI models was created. In this document we will mainly focus on the scientific justification behind said modelling of data, the AI that tells the player whether their layout is suitable for habitation or not, and how this AI was created. Also, we will mention how the design of workflow allows for easy optimization and escalation of the AI. This aspect of the design software works as a strong tool for scientific research.

The story behind the data

We began by understanding the story that can be built from the provided resources. From this research we identified characteristics that place the human, science and sustainability in the first place. We need to think of a habitat as more than a piece of metal and pressurised air to keep astronauts alive. We are building a home: psychological well-being is a necessity. It is vital that comfort, privacy and recreation are provided. Furthermore, the habitat will work as a scientific space to drive innovation and research forward. A scientist needs to have an optimized and easy to access environment in order to work. A clean space is a happy space. Lastly, our habitat should not depend completely on the Earth. A sustainable design allows us to progress towards the future and build a permanent presence wherever we go.

"Deep Space Habitability Design Guidelines Based on the NASA NextSTEP Phase 2 Ground Test Program" emphasized the importance of real feedback by astronauts in designing meaningful metrics, to quantize apparently subjective data. "A Tool for Automated Design and Evaluation of Habitat Interior Layouts" by M. Simon and A. Wilhite provided a solid basis for defining how a habitat is evaluated by building more explicit formulas that can be operated mathematically and correspond to the feedback given in the previously mentioned document.

"Overview of NASA's Moon to Mars Planetary Autonomous Construction Technology (MMPACT)" by R. Clinton, J. Edmunson, M. Fiske, M. Effinger, E. Jensen, and J. Ballard showcased how NASA is not only concerned with reaching a destination, but rather they aim to drive research forward. This is further emphasized in papers such as "NASA's Moon to Mars (M2M) Transit Habitat Refinement Point of Departure Design"that specified guidelines on volume and mass to optimize the use of the space.

The papers "Defining the Net Habitable Volume for Long Duration Exploration Missions" by C. Stromgren, C. Burke, J. Cho, R. Calderon, M. Rucker and *Moon to Mars Architecture Definition Document* guided us in identifying the elements that make up Habitat and how those can be modeled.

Our key goals with the model were to:

1. Drive exploration forward: By automating the design evaluation process we are reducing the time it requires to be completed, since a model can evaluate the layout in a few seconds. This allows us to easily try more combinations and find optimal solutions to prepare for space exploration.

2. Create better homes: We make sure that the habitats built provide an enjoyable experience for the humans living in them.
3. Sharing knowledge: While our tool works as a design software, it can also be used by people without a technical or design background. The visualization and processing of the model allows for a sort of game to be played where people can develop an intuition for designing habitats and preparing them for the future.

With this tool, we aim to create a compass to guide humanity in its future exploration of the cosmos.

<u>Main metrics</u>

The main metrics to determine whether the final built plot was any good were based on the following modules:

- Social module: Module that pertains towards allowing our crew members to de-stress and talk between themselves.
- Private quarters: Module that is focused solely on giving each crew member their own set of space in order to de-stress and rest.
- Hygiene module: Module focused on allowing the crewmates to partake in regular hygiene routines.
- Waste module: Module that solely takes care of managing both biological and artificial waste.
- Exercise module: Module that allows crew members to de-stress by way of doing exercise.
- Food module: Module focused on tackling all things pertinent to food, whether it be preparation or consumption of the same.
- Maintenance module: Module relative to all things that have to do with maintenance of the plot that has been built.
- Science module: Module that focuses on bringing equipment and appropriate technology to advance in fields of scientific research.
- Medical module: Module related to the application of basic, advanced first aid, as well as other inconveniences that might happen during the mission.
- Logistics module: Module relative to all that has to do with information and storage collection.
- Airlock module: Module that is representative of a real-life airlock.
- Mission planning module: Module that concentrates on keeping technology and equipment to keep the mission the crewmates have been sent to going.

Considering this, we set up some rules that would then allow us to calculate the final score, such as:

- Desirability of proximity: There are some modules that we wish would be closer to each other

We calculated these rules since they work as tensors. Because they model the relationships of the elements they are not dependant of specific placing, which might skew the model, but rather abstract the main ideas and concepts behind a good layout design philosophy.

<u>Model</u>

Workflow

The data was gathered, defined and processed in the following way. First we understood what was necessary to properly tell the story a good layout conveys. Then we defined how the data would be represented by the objects on the frontend in order to satisfy the metrics defined and implemented functions that represent the relationship between those elements. It is valuable to represent the relationship

In order to start training the data, we first needed to create an algorithm that allowed us to create randomized layouts in order to be able to grade them based on efficiency. We also printed out all of the scores to be able to get an idea of how the algorithm saw the decisions it took.

We ran said algorithm about a 100 times, and each time we graded each one of the attempts that were made by the algorithm itself based on the following parameters:

Software design

To actually be able to build a habitat, a specialized software was implemented. This software includes placing set modules, an isometric and top view, as well as the ability to process habitat layouts, such as how clean it is, how much space is wasted, and other key considerations of this challenge.

This software was developed in python using mainly the pygame library. This approach gives us full customization on what we want the software to use.

Isometric view

The main way to view a habitat. The reason for making the view a fixed isometric one, is to include every angle of the habitat, while maintaining a pleasant visual experience. You can select which module you would like to place by pressing any number from 1 - 9.

For data processment, a top view was implemented with a different set of functionalities.

This code implements the main way to view a habitat: a fixed isometric scene that shows every face of the structure while keeping the experience clean and readable. The camera converts grid coordinates to isometric screen space, and all tiles (ground, floors, walls, and assets) are drawn in a single layered sprite group so depth looks correct. You can select which module to place by pressing any number from 1–9; a translucent preview (“ghost” block) follows the mouse and is tinted blue for placement and red for removal. Clicking applies the current action, stacking blocks by column unless a floor tile should share the same height, and deletion always removes the topmost element. Terrain is generated as purely visual tiles at very low layers and is safely regenerated (the previous terrain is cleared) when switching themes like Mars or Moon. A simple background renderer draws a subtle sky/ground or a lunar night with twinkling stars, independent of the grid, so the habitat always stands out.

For data processing and layout inspection, a separate top view can be used with a different set of interactions, making it easier to analyze coverage, adjacency, and module counts without visual occlusion. Together, the isometric and top-down modes balance clarity and control: one is ideal for presentation and spatial intuition; the other is optimized for measurement and organization.

Top view

To better understand the relationship between modules and to better process data, we decided to create a top view of the habitat. The software includes a top bar with various widgets, such as a mode selector, a size selector, and a category selector.

Each area category is color coded to visualize each module and its placement in the grid. The module placement is based on a size system, ranging from 1x1 to 4x4, and a placed module can not be overwritten by a newly placed module, so no overlapping can occur.

This software includes a budget system, where once you run out of budget, you can't place any more modules. To get that money back, just remove a placed module from the grid with the mode selector button. Modes include draw and remove.

One key aspect of the AI model is its ability to measure distances. The main reason for creating the top view is so the distance between each module is based on the grid mechanic, making the calculation easier in terms of an X and Y coordinate system.

In order to pass the grid data to the AI model, first we must export the grid into a valid format. The export button included on the topbar does just that, so that a .json file is ready to introduce to the AI model. As well as an export option, an import option is included to visualize set habitat layouts, in case you would like to revisit an optimal layout.