beamplan

A command-line tool to determine Starlink beam planning.

💾 Technology stack

The technology stack used for `beamplan` is a Python (3.7+) environment, with a standard `venv` Python virtual environment. The command-line tool can be used on Linux (tested for Ubuntu 20.04), Windows (tested for Windows 10), and macOS (tested for Catalina).

For the best, most intuitive use (via UNIX shell), the use of Linux is recommended.

Category	Technology	Description
Language	Python (3.7+)	An interactive programming language
Virtual environment	venv	A lightweight virtual environment
OS	Linux (Ubuntu), Windows and macOS	Tested for all three (Windows 10 and Catalina)

🛠 Setting up your environment

Before any installation of repository-specific content, ensure that your environment is set up per instruction below.

If you do not have your developmental environment set up already...no worries! Instructions for Linux set up will be provided here, and reference guides will be provided for the other operating systems as well.

Linux (Ubuntu 20.04+)

These instructions were tested for Ubuntu 20.04+, and can likely be applied to most Debian-based systems.

Install Python

First, the latest version of Python (3.X) must be installed in order to to get the rest of the environment set up.

1. Open a terminal on your Linux machine.
2. Update your Ubuntu package manager, apt, with the following command:

```
$ sudo apt-get update
```

3. Once your package manager is updated, install the latest version of Python, and it's respective package manager, pip, with the following command:

```
$ sudo apt-get install python3.9 python3-pip
```

To verify your installation, run the following command:

```
$ python --version
```

If it reverberates with the latest version you installed, your set up for Python is complete. If not, please refer to this in-depth guide with troubleshooting and other methods of installation.

Install venv

Once Python is installed, it is **highly** recommended to install venv, a lightweight Python virtual environment package. It is simple to set up and very useful for controlling a development environment.

1. Open a terminal on your Linux machine.
2. Run the following package install command to install the venv module

```
$ sudo apt-get install python3-venv
```

Other operating systems

For other operating systems, the download(s) are all still the same, they just need to be done through different means.

For Windows, download Python following this guide, and install venv the same way as presented for Linux.

For macOS, download Python following this guide, and venv will automatically be installed with it.

The rest of the instructions for the package follow the same idea as the rest of the document.

🚀 Building and running for production

After following and verifying the instructions to set up your environment, clone the repository.

```
$ git clone https://github.com/rdekovich/beamplan.git
```

Navigate into the repository.

```
$ cd beamplan/
```

Create a virtual environment (in this case, it's named `env`) with Python `venv`, and source it open.

```
$ python3 -m venv env
$ source env/bin/activate
```

Your terminal window should now show the sourced environment.

```
$ (env) rdekovich@ubuntu: -
```

Now, run the `setup.py` setuptools file to install all site-packages and scripting aliases to the local virtual environment.

```
$ pip install -e .
```

The package is now set up. You can now invoke the `beamplan` package as an executable, as such.

```
$ beamplan var/tests/00_example.txt
```

The following table is all invocation options, with examples using them.

Argument	Required	Description	Example
INFILE	Yes	Input file to the beamplan tool	`$ beamplan infile.txt`
--debug, -d	No	Routes stdout to an output file `INFILE.out`	`$ beamplan infile.txt --debug`
--help, -h	No	Package help string for this table	`$ beamplan --help`

🏆 Heuristic coverages

The following table outlines the heuristic coverages achieved with the given version(s). All constraints are met, and each test allocates less than 1GB memory and runs less than 15 minutes.

Test	v0
00_example.txt	100%
01_simplest_possible.txt	100%
02_two_users.txt	100%
03_five_users.txt	80%
04_one_interferer.txt	0%
05_equatorial_plane.txt	99.8%
06_partially_fullfillable.txt	76.68%
07_eighteen_planes.txt	97.8%
08_eighteen_planes_northern.txt	78.92%
09_ten_thousand_users.txt	90.55%
10_ten_thousand_users_geo_belt.txt	81.45%
11_one_hundred_thousand_users.txt	29.25%