Warp-in Points

A warp-in point is where your ship will land in space when warping to an object.

Many external factors may disrupt a warp, e.g. warp disruption fields, insufficient electrical capacity, warp jitter, etc.; these formulas do not account for these phenomenas, they assume perfect conditions.

Ordinary Objects

An ordinary object is any object that does not fall withing any of the other categories described below.

Let the 3D vectors p_d and p_s represent the object's position and the warp's origin, respectively; and $\vec{v}$ the directional vector from p_s to p_d. Let r be the object's radius.

The object's warp-in point is the vector $p_s + \vec{v} - r\hat{v}$.

Large Objects

A large object is any celestial body whose radius exceeds 90 kilometres (180 kilometres in diameter), except planets.

Let x, y, and z represent the object's coordinates. Let r be the object's radius.

The object's warp-in point is the vector $\begin{aligned}&(x + (r + 5000000)\cos r\\&, y + 1.3r - 7500\\&, z - (r + 5000000)\sin r\\&)\,.\end{aligned}$

Planets

The warp-in point of a planet is determined by the planet's ID, its location, and radius.

Let x, y, and z represent the planet's coordinates. Let r be the planet's radius.

The planet's warp-in point is the vector $\left(x + d\sin\theta, y + \frac{1}{2}r\sin j, z - d\cos\theta\right)$ where:

$$d = r(s + 1) + 1000000$$

$$\theta = \sin^{-1}\left(\frac{x}{|x|} \cdot \frac{z}{\sqrt{x^2 + z^2}}\right) + j$$

$$s|_{0.5 \leq s \leq 10.5} = 20\left(\frac{1}{40}\left(10\log_{10}\left(\frac{r}{10^6}\right) - 39\right)\right)^{20} + \frac{1}{2}$$

Now, j is a special snowflake. Its value is the Python equivalent of `(random.Random(planetID).random() - 1.0) / 3.0`.

Example Implementation

```
import math
import random

def warpin(id, x, y, z, r):
    j = (random.Random(id).random() - 1.0) / 3.0
    t = math.asin(x/abs(x) * (z/math.sqrt(x**2 + z**2))) + j
    s = 20.0 * (1.0/40.0 * (10 * math.log10(r/10**6) - 39))**20.0 + 1.0/2.0
    s = max(0.5, min(s, 10.5))
    d = r*(s + 1) + 1000000

    return (x + d * math.sin(t), y + 1.0/2.0 * r * math.sin(j), z - d * math.cos(t))
```