

RunCode

```
StarSystemAPI sys = (StarSystemAPI) $playerFleet.getContainingLocation();
SectorEntityToken star = sys.getStar(); SectorEntityToken player = $playerFleet;
List<SectorEntityToken> entities = new
ArrayList<SectorEntityToken>(sys.getAllEntities()); for (SectorEntityToken entity :
entities) { if (!entity.equals(star) && !entity.equals(player)) { if (entity.getMarket() !=
null) { Global.getSector().getEconomy().removeMarket(entity.getMarket()); }
sys.removeEntity(entity); } }
float _solRad = 1000.0f;
StarSystemAPI system = (StarSystemAPI) $playerFleet.getContainingLocation(); if
(system != null) { PlanetAPI oldStar = system.getStar(); if (oldStar != null) { String starId
= oldStar.getId(); system.removeEntity(oldStar);
PlanetAPI newStar = system.initStar(starId, "star_yellow", _solRad, _solRad);
newStar.setName("Sol");
newStar.setLocation(0, 0); } }
SectorEntityToken _fleet = Global.getSector().getPlayerFleet();
StarSystemAPI _sys = (StarSystemAPI)_fleet.getContainingLocation();
SectorEntityToken _star = _sys.getStar();
_sys.setBackgroundTextureFilename("graphics/backgrounds/background4.jpg");
```

```
float _extCompressor = 8000f;
float _intCompressor = 0.8f;
float _moonExt = 1000.0f;
float _moonInt = 500.0f;
float _irrExt = 800.0f;
float _astfrequency = 10;
float _astsize = 15f;
float _earthRad = 121f;
float _lunarSystemDistBase = 600f;
float lunaDist = _earthRad + _lunarSystemDistBase;
```

```
PlanetAPI Mercury = _sys.addPlanet("Mercury", _star, "Mercury", "barren_castiron",
40, 57, _solRad + _extCompressor * (float) Math.log(_intCompressor * 0.387f + 1) -
lunaDist, 88);
_sys.addCustomEntity(null, null, "coronal_tap",
"neutral").setCircularOrbitPointingDown(_star, 40, _solRad + 200, 88);
```

```
MarketAPI marketMercury = _sys.getEntityById("Mercury").getMarket();
```

```
marketMercury.addCondition("very_hot");marketMercury.addCondition("ore_ultrarich");
marketMercury.addCondition("ruins_scattered");
marketMercury.addCondition("poor_light");
marketMercury.addCondition("no_atmosphere");
```

```
PlanetAPI Venus = _sys.addPlanet("Venus", _star, "Venus", "toxic", 261, 117,_solRad +
_extCompressor * (float) Math.log(_intCompressor * 0.723f + 1 ) - lunaDist , 224);
```

```
MarketAPI marketVenus = Venus.getMarket();
marketVenus.addCondition("toxic_atmosphere");
marketVenus.addCondition("very_hot");
marketVenus.addCondition("extreme_weather");
marketVenus.addCondition("decivilized_subpop");
marketVenus.addCondition("ruins_vast");
marketVenus.addCondition("tectonic_activity");
marketVenus.addCondition("poor_light");
marketVenus.addCondition("ore_moderate");
marketVenus.addCondition("organics_trace");
marketVenus.addCondition("dense_atmosphere");
```

```
SectorEntityToken iss = _sys.addCustomEntity("iss_station", "International Space
Station", "station_lowtech3", "neutral");
iss.setCircularOrbitPointingDown(Venus, 90, 240, 30);
MarketAPI mISS = Global.getFactory().createMarket("iss_market", "International Space
Station", 0);
mISS.setPrimaryEntity(iss);
mISS.setFactionId("neutral");
mISS.addCondition("abandoned_station");
mISS.addSubmarket("storage");
iss.setMarket(mISS);
Global.getSector().getEconomy().addMarket(mISS, true);
```

```
SectorEntityToken tiangong = _sys.addCustomEntity("tiangong_station", "Tiangong
347", "station_hightech1", "neutral");
tiangong.setCircularOrbitPointingDown(Venus, 270, 180, 30);
MarketAPI mTian = Global.getFactory().createMarket("tiangong_market", "Tiangong
347", 0);
mTian.setPrimaryEntity(tiangong);
mTian.setFactionId("neutral");
mTian.addCondition("abandoned_station");
```

```

mTian.addSubmarket("storage");
tiangong.setMarket(mTian);
Global.getSector().getEconomy().addMarket(mTian, true);

float earthAngle = 134;
float earthOrbitalPeriod = 365;
float lunaOrbitalPeriod = earthOrbitalPeriod / (365/30);
float earthDist = _solRad + _extCompressor * (float) Math.log(_intCompressor * 1.0f +
1);

float LPointOrbitDistance = lunaDist * 1.4f;
float vanAllenInner = (lunaDist-_earthRad) * 0.45f + _earthRad ;
float vanAllenOuter = (lunaDist-_earthRad) * 0.75f + _earthRad ;
float geoCommRelayDist = (lunaDist-_earthRad) * 0.1f + _earthRad+15;

PlanetAPI Earth = _sys.addPlanet("Earth", _star, "Earth", "terran", earthAngle,
_earthRad, earthDist, earthOrbitalPeriod);
MarketAPI mEarth = Earth.getMarket();
mEarth.addCondition("mild_climate");
mEarth.addCondition("habitable");
mEarth.addCondition("ruins_vast");
mEarth.addCondition("decivilized_subpop");
mEarth.addCondition("farmland_bountiful");
mEarth.addCondition("solar_array");
mEarth.addCondition("ore_moderate");
mEarth.addCondition("organics_plentiful");
mEarth.addCondition("rare_ore_sparse");
mEarth.addCondition("volatiles_diffuse");

SectorEntityToken earthField = _sys.addTerrain(Terrain.MAGNETIC_FIELD, new
com.fs.starfarer.api.impl.campaign.terrain.MagneticFieldTerrainPlugin.MagneticFieldPar
ams(200f, 350f, Earth, (int)vanAllenInner, vanAllenOuter, new Color(50, 20, 100, 50),
0.1f));
earthField.setCircularOrbit(Earth, 0, 0, 100);
earthField.setName("Van Allen Belts");

SectorEntityToken earthSensorArray = _sys.addCustomEntity("earth_sensor_array",
"Earth Sensor Array", "sensor_array", "neutral");
earthSensorArray.setCircularOrbit(_star, earthAngle - 60, earthDist, earthOrbitalPeriod);

```

```
SectorEntityToken earthRelay = _sys.addCustomEntity("earth_comm_relay", "GEO  
Comm Relay", "comm_relay", "neutral");  
earthRelay.setCircularOrbitPointingDown(Earth, 0, geoCommRelayDist, 5);
```

```
PlanetAPI Luna = _sys.addPlanet("Luna", Earth, "Luna", "barren", 0, 33, lunaDist,  
lunaOrbitalPeriod );  
MarketAPI marketLuna = Luna.getMarket();  
marketLuna.addCondition("no_atmosphere");  
marketLuna.addCondition("low_gravity");  
marketLuna.addCondition("ruins_vast");  
marketLuna.addCondition("ore_ultrarich");  
marketLuna.addCondition("rare_ore_sparse");  
marketLuna.addCondition("volatiles_trace");  
marketLuna.addCondition("hydroponics_complex");
```

```
SectorEntityToken lunaL4 = _sys.addCustomEntity("luna_l4", "Luna L4 Stable  
Location", "stable_location", "neutral");  
lunaL4.setCircularOrbit(Earth, 60, lunaDist, lunaOrbitalPeriod );  
SectorEntityToken lunaL5 = _sys.addCustomEntity("luna_l5", "Luna L5 Stable  
Location", "stable_location", "neutral");  
lunaL5.setCircularOrbit(Earth, 300, lunaDist, lunaOrbitalPeriod );  
SectorEntityToken lunaL3 = _sys.addCustomEntity("luna_l5", "Sol Gate",  
"inactive_gate", "neutral");  
lunaL3.setCircularOrbit(Earth, 180, lunaDist, lunaOrbitalPeriod );
```

```
_sys.addCustomEntity(null, "Luna Mirror Alpha", "stellar_mirror",  
"neutral").setCircularOrbitPointingDown(Luna, 0, 100, 5);  
_sys.addCustomEntity(null, "Luna Mirror Beta", "stellar_mirror",  
"neutral").setCircularOrbitPointingDown(Luna, 120, 100, 5);  
_sys.addCustomEntity(null, "Luna Mirror Gamma", "stellar_mirror",  
"neutral").setCircularOrbitPointingDown(Luna, 240, 100, 5);
```

```
_sys.addCustomEntity(null, "Lagrange Shade Alpha", "stellar_shade",  
"neutral").setCircularOrbitPointingDown(Earth, earthAngle + 176, LPointOrbitDistance,  
earthOrbitalPeriod);  
_sys.addCustomEntity(null, "Lagrange Shade Beta", "stellar_shade",  
"neutral").setCircularOrbitPointingDown(Earth, earthAngle + 180, LPointOrbitDistance,  
earthOrbitalPeriod);
```

```
_sys.addCustomEntity(null, "Lagrange Shade Gamma", "stellar_shade",
"neutral").setCircularOrbitPointingDown(Earth, earthAngle + 184, LPointOrbitDistance,
earthOrbitalPeriod);

_sys.addCustomEntity(null, "Lagrange Mirror Alpha", "stellar_mirror",
"neutral").setCircularOrbitPointingDown(Earth, earthAngle - 4f, LPointOrbitDistance,
earthOrbitalPeriod);
_sys.addCustomEntity(null, "Lagrange Mirror Beta", "stellar_mirror",
"neutral").setCircularOrbitPointingDown(Earth, earthAngle, LPointOrbitDistance,
earthOrbitalPeriod);
_sys.addCustomEntity(null, "Lagrange Mirror Gamma", "stellar_mirror",
"neutral").setCircularOrbitPointingDown(Earth, earthAngle + 4f, LPointOrbitDistance,
earthOrbitalPeriod);
```

```
com.fs.starfarer.api.impl.campaign.terrain.AsteroidFieldTerrainPlugin.AsteroidFieldParams earthL4FieldParams = new
com.fs.starfarer.api.impl.campaign.terrain.AsteroidFieldTerrainPlugin.AsteroidFieldParams(100f, 100f, 2, 2, 5f, 15f, "Earth L4 Trojans");
SectorEntityToken earthL4Field = _sys.addTerrain(Terrain.ASTEROID_FIELD,
earthL4FieldParams);
earthL4Field.setCircularOrbit(_star, earthAngle + 62, earthDist, earthOrbitalPeriod);

SectorEntityToken earthNavBuoy = _sys.addCustomEntity("earth_nav_buoy", "Earth
Nav Buoy", "nav_buoy", "neutral");
earthNavBuoy.setCircularOrbit(_star, earthAngle + 60, earthDist, earthOrbitalPeriod);
```

```
float cruithneOrbitDistance = _solRad + _extCompressor * (float)
Math.log(_intCompressor * 0.998f + 1);
float cruithneOscillationDist = 400;
float cruithneOscillationPeriod = earthOrbitalPeriod / (365/400);

SectorEntityToken EarthCruithneCenter = _sys.addCustomEntity(null, null,
"sensor_ghost", "neutral");
EarthCruithneCenter.setCircularOrbit(_star, earthAngle + 200, cruithneOrbitDistance,
earthOrbitalPeriod);

PlanetAPI Cruithne = _sys.addPlanet("Cruithne", EarthCruithneCenter, "Cruithne",
"barren", 0, 2, cruithneOscillationDist, cruithneOscillationPeriod);
```

```
MarketAPI marketCruithne = Cruithne.getMarket();
marketCruithne.addCondition("ruins_widespread");
marketCruithne.addCondition("hot");
marketCruithne.addCondition("low_gravity");
marketCruithne.addCondition("no_atmosphere");
marketCruithne.addCondition("ore_sparse");
```

```
PlanetAPI Eros = _sys.addPlanet("Eros", _star, "Eros", "barren", 266, 10,_solRad +
_extCompressor * (float) Math.log(_intCompressor * 1.458f + 1), 643);
```

```
MarketAPI marketEros = Eros.getMarket();
marketEros.addCondition("ore_moderate");
marketEros.addCondition("ruins_scattered");
marketEros.addCondition("cold");
marketEros.addCondition("no_atmosphere");
marketEros.addCondition("low_gravity");
```

```
PlanetAPI Phaethon = _sys.addPlanet("Phaethon", _star, "Phaethon", "barren", 300,
4,_solRad + _extCompressor * (float) Math.log(_intCompressor * 1.27f + 1), 524);
```

```
JumpPointAPI _jp2 = Global.getFactory().createJumpPoint("jp_phaethon", "Phaethon
Jump Point");
```

```
MarketAPI marketPhaethon = Phaethon.getMarket();
marketPhaethon.addCondition("no_atmosphere");
marketPhaethon.addCondition("low_gravity");
marketPhaethon.addCondition("hot");
marketPhaethon.addCondition("volatiles_plentiful");
```

```
_jp2.setStandardWormholeToHyperspaceVisual();
_jp2.setCircularOrbit(Phaethon, 0, 20, 20);
_sys.addEntity(_jp2);
```

```
float marsRad = _solRad + _extCompressor * (float) Math.log(_intCompressor * 1.524f
+ 1);
```

```
PlanetAPI Mars = _sys.addPlanet("Mars", _star, "Mars", "barren-desert", 277, 77,
marsRad, 779);
```

```
MarketAPI marketMars = Mars.getMarket();
marketMars.addCondition("cold");
marketMars.addCondition("thin_atmosphere");
marketMars.addCondition("volatiles_diffuse");
marketMars.addCondition("pollution");
```

```

marketMars.addCondition("ruins_vast");
marketMars.addCondition("organics_trace");

PlanetAPI Phobos = _sys.addPlanet("Phobos", Mars, "Phobos", "barren", 153, 10, 140,
0.319f);
MarketAPI marketPhobos = Phobos.getMarket();
marketPhobos.addCondition("ruins_extensive");
marketPhobos.addCondition("volatiles_trace");
marketPhobos.addCondition("low_gravity");
marketPhobos.addCondition("cold");
marketPhobos.addCondition("no_atmosphere");

PlanetAPI Deimos = _sys.addPlanet("Deimos", Mars, "Deimos", "barren", 153, 10, 200,
1.262f);
MarketAPI marketDeimos = Deimos.getMarket();
marketDeimos.addCondition("ruins_widespread");
marketDeimos.addCondition("low_gravity");
marketDeimos.addCondition("no_atmosphere");
marketDeimos.addCondition("cold");

SectorEntityToken marsL5 = _sys.addTerrain(Terrain.ASTEROID_FIELD, new
com.fs.starfarer.api.impl.campaign.terrain.AsteroidFieldTerrainPlugin.AsteroidFieldPara
ms(100, 100, 5, 2, 4f, 10f, "Mars L5 Trojans")); marsL5.setCircularOrbit(_star, 217,
marsRad, 779);

```

```

float beltCenterRad = _solRad + _extCompressor * (float) Math.log(_intCompressor *
2.7f + 1);
system.addAsteroidBelt(system.getStar(), 150, beltCenterRad, 2200, 1500, 2500,
Terrain.ASTEROID_BELT, "Sol Asteroid Belt");
system.addAsteroidBelt(system.getStar(), 300, beltCenterRad, 1600, 1600, 2400,
Terrain.ASTEROID_BELT, "Sol Asteroid Belt");
system.addAsteroidBelt(system.getStar(), 500, beltCenterRad, 1000, 1700, 2300,
Terrain.ASTEROID_BELT, "Sol Asteroid Belt");
system.addAsteroidBelt(system.getStar(), 800, beltCenterRad, 400, 1800, 2200,
Terrain.ASTEROID_BELT, "Sol Asteroid Belt");

PlanetAPI Ceres = _sys.addPlanet("Ceres", _star, "Ceres", "rocky_ice", 185, 35,
_solRad + (_extCompressor * (float) Math.log(_intCompressor * 2.768f + 1)), 1460);
JumpPointAPI _jp1 = Global.getFactory().createJumpPoint("jp_ceres", "Ceres Jump
Point");

```

```

_jp1.setStandardWormholeToHyperspaceVisual();
_jp1.setCircularOrbit(Ceres, 0, 100, 30);
_sys.addEntity(_jp1);
MarketAPI marketCeres = Ceres.getMarket();
marketCeres.addCondition("volatiles_plentiful");
marketCeres.addCondition("cold");
marketCeres.addCondition("ore_moderate");
marketCeres.addCondition("rare_ore_sparse");
marketCeres.addCondition("ruins_vast");
marketCeres.addCondition("low_gravity");
marketCeres.addCondition("no_atmosphere");

SectorEntityToken antiopeBarycenter = _sys.addCustomEntity(null, null,
"sensor_ghost", "neutral");
antiopeBarycenter.setCircularOrbit(_star, 140, _solRad + (_extCompressor * (float)
Math.log(_intCompressor * 3.13f + 1)), 2022);
PlanetAPI AntiopeA = _sys.addPlanet("AntiopeA", antiopeBarycenter, "Antiope A",
"rocky_ice", 0, 9, 10, 1);
PlanetAPI AntiopeB = _sys.addPlanet("AntiopeB", antiopeBarycenter, "Antiope B",
"rocky_ice", 180, 8, 10, 1);

PlanetAPI Pallas = _sys.addPlanet("Pallas", _star, "Pallas", "rocky_ice", 305, 26,
_solRad + (_extCompressor * (float) Math.log(_intCompressor * 2.773f + 1)), 1679);
PlanetAPI Vesta = _sys.addPlanet("Vesta", _star, "Vesta", "barren", 58, 27, _solRad +
(_extCompressor * (float) Math.log(_intCompressor * 2.362f + 1)), 1314);
PlanetAPI Hygiea = _sys.addPlanet("Hygiea", _star, "Hygiea", "rocky_ice", 165, 23,
_solRad + (_extCompressor * (float) Math.log(_intCompressor * 3.142f + 1)), 2080);
PlanetAPI Intermanmia = _sys.addPlanet("Intermanmia", _star, "Intermanmia",
"rocky_ice", 226, 19, _solRad + (_extCompressor * (float) Math.log(_intCompressor *
3.062f + 1)), 1989);
PlanetAPI Juno = _sys.addPlanet("Juno", _star, "Juno", "barren", 140, 18, _solRad +
(_extCompressor * (float) Math.log(_intCompressor * 2.67f + 1)), 1595);
PlanetAPI Iris = _sys.addPlanet("Iris", _star, "Iris", "barren", 15, 16, _solRad +
(_extCompressor * (float) Math.log(_intCompressor * 2.39f + 1)), 1346);
PlanetAPI Flora = _sys.addPlanet("Flora", _star, "Flora", "barren", 340, 12, _solRad +
(_extCompressor * (float) Math.log(_intCompressor * 2.20f + 1)), 1193);
PlanetAPI Eunomia = _sys.addPlanet("Eunomia", _star, "Eunomia", "barren", 161, 18,
_solRad + (_extCompressor * (float) Math.log(_intCompressor * 2.643f + 1)), 1571);
PlanetAPI Euphrosyne = _sys.addPlanet("Euphrosyne", _star, "Euphrosyne",
"rocky_ice", 81, 18, _solRad + (_extCompressor * (float) Math.log(_intCompressor *
3.149f + 1)), 2041);

```

```

PlanetAPI Cybele = _sys.addPlanet("Cybele", _star, "Cybele", "rocky_ice", 36, 17,
_solidRad + (_extCompressor * (float) Math.log(_intCompressor * 3.433f + 1)), 2319);
PlanetAPI Hungaria = _sys.addPlanet("Hungaria", _star, "Hungaria", "barren", 180, 8,
_solidRad + (_extCompressor * (float) Math.log(_intCompressor * 1.94f + 1)), 991);
PlanetAPI Astraea = _sys.addPlanet("Astraea", _star, "Astraea", "barren", 20, 11,
_solidRad + (_extCompressor * (float) Math.log(_intCompressor * 2.57f + 1)), 1507);
PlanetAPI Hebe = _sys.addPlanet("Hebe", _star, "Hebe", "barren", 85, 14, _solidRad +
(_extCompressor * (float) Math.log(_intCompressor * 2.42f + 1)), 1380);
PlanetAPI Davida = _sys.addPlanet("Davida", _star, "Davida", "rocky_ice", 146, 19,
_solidRad + (_extCompressor * (float) Math.log(_intCompressor * 3.168f + 1)), 2055);

PlanetAPI Sylvia = _sys.addPlanet("Sylvia", _star, "Sylvia", "rocky_unstable", 81, 18,
_solidRad + (_extCompressor * (float) Math.log(_intCompressor * 3.485f + 1)), 2372);
MarketAPI marketSylvia = Sylvia.getMarket();
PlanetAPI Remus = _sys.addPlanet("Remus", Sylvia, "Remus", "barren", 346, 5, 80, 1);
PlanetAPI Romulus = _sys.addPlanet("Romulus", Sylvia, "Romulus", "barren", 43, 6,
140, 4);
PlanetAPI Kleopatra = _sys.addPlanet("Kleopatra", _star, "Kleopatra", "rocky_metallic",
88, 14, _solidRad + (_extCompressor * (float) Math.log(_intCompressor * 2.79f + 1)),
1705);
PlanetAPI Alexhelios = _sys.addPlanet("Alexhelios", Kleopatra, "Alexhelios", "barren",
45, 4, 60, 3);
PlanetAPI Cleoselene = _sys.addPlanet("Cleoselene", Kleopatra, "Cleoselene",
"barren", 225, 3, 40, 2);

PlanetAPI Lutetia = _sys.addPlanet("Lutetia", _star, "Lutetia", "rocky_metallic", 15, 10,
_solidRad + (_extCompressor * (float) Math.log(_intCompressor * 2.43f + 1)), 1374);
PlanetAPI Kalliope = _sys.addPlanet("Kalliope", _star, "Kalliope", "rocky_metallic", 220,
14, _solidRad + (_extCompressor * (float) Math.log(_intCompressor * 2.91f + 1)), 1818);
PlanetAPI Linus = _sys.addPlanet("Linus", Kalliope, "Linus", "barren", 0, 4, 40, 4);
PlanetAPI Antigone = _sys.addPlanet("Antigone", _star, "Antigone", "rocky_metallic",
145, 11, _solidRad + (_extCompressor * (float) Math.log(_intCompressor * 2.87f + 1)),
1773);
PlanetAPI Quintilla = _sys.addPlanet("Quintilla", _star, "Quintilla", "rocky_metallic", 330,
11, _solidRad + (_extCompressor * (float) Math.log(_intCompressor * 3.17f + 1)), 2060);

PlanetAPI Psyche = _sys.addPlanet("Psyche", _star, "Psyche", "rocky_metallic", 61,
18, _solidRad + (_extCompressor * (float) Math.log(_intCompressor * 2.924f + 1)), 1825);
_sys.addCustomEntity(null, "NeoNorilsk", "station_mining",
"neutral").setCircularOrbitPointingDown(Psyche, 90, 150, 45);
MarketAPI marketPsyche = Psyche.getMarket();
marketPsyche.addCondition("low_gravity");

```

```
marketPsyche.addCondition("no_atmosphere");
marketPsyche.addCondition("poor_light");
marketPsyche.addCondition("ruins_scattered");
marketPsyche.addCondition("ore_ultrarich");
marketPsyche.addCondition("rare_ore_ultrarich");

_sys.addCustomEntity(null, "Salyut 3 Anti-Air Gun Module", "weapons_cache_low",
"neutral").setCircularOrbit(_star, 135, beltCenterRad, 2200);
```

```
float jupOrbitRadius = _solRad + _extCompressor * (float) Math.log(_intCompressor *
5.203f + 1);
float _jupRad = 376f;
PlanetAPI Jupiter = _sys.addPlanet("Jupiter", _star, "Jupiter", "gas_giant", 226, 450,
jupOrbitRadius, 4015);
MarketAPI marketJupiter = Jupiter.getMarket();
marketJupiter.addCondition("irradiated");
marketJupiter.addCondition("high_gravity");
marketJupiter.addCondition("dense_atmosphere");
marketJupiter.addCondition("volatiles_plentiful");
marketJupiter.addCondition("meteor_impacts");
marketJupiter.addCondition("cold");
marketJupiter.addCondition("extreme_weather");
```

```
SectorEntityToken jupiterField = _sys.addTerrain(Terrain.MAGNETIC_FIELD, new
com.fs.starfarer.api.impl.campaign.terrain.MagneticFieldTerrainPlugin.MagneticFieldPar
ams(300, 1, Jupiter, 400, 1300, new Color(80, 10, 110, 60), 1));
jupiterField.setCircularOrbit(Jupiter, 0, 0, 100); jupiterField.setName("Great Jovian
Magnetosphere");
```

```
Float _jupRing = _jupRad + (_moonExt * (float) Math.log(_moonInt * 0.00086f + 1));
_sys.addTerrain(Terrain.RING, new
com.fs.starfarer.api.impl.campaign.terrain.RingSystemTerrainPlugin.RingParams(100,
_jupRing-60, Jupiter, "Halo Ring")).setCircularOrbit(Jupiter, 0, 0, 100);

_sys.addRingBand(Jupiter, "misc", "rings_dust0", 256, 1, new Color(75, 65, 60, 55), 40,
_jupRad + (_moonExt * (float) Math.log(_moonInt * 0.00086f + 1)), 32);
```

```
_sys.addTerrain(Terrain.RING, new
com.fs.starfarer.api.impl.campaign.terrain.RingSystemTerrainPlugin.RingParams(40,
_jupRing, Jupiter, "Main Ring")).setCircularOrbit(Jupiter, 0, 0, 100);

_sys.addTerrain(Terrain.RING, new
com.fs.starfarer.api.impl.campaign.terrain.RingSystemTerrainPlugin.RingParams(100f,
40+ _jupRing , Jupiter, "Gossamer Ring")).setCircularOrbit(Jupiter, 0, 0, 100);
```

```
PlanetAPI Metis = _sys.addPlanet("Metis", Jupiter, "Metis", "barren", 153, 5, _jupRad +
(_moonExt * (float) Math.log(_moonInt * 0.00085f + 1)), 0.294f);
PlanetAPI Adrastea = _sys.addPlanet("Adrastea", Jupiter, "Adrastea", "barren", 237, 5,
_jupRad + (_moonExt * (float) Math.log(_moonInt * 0.00086f + 1)), 0.298f);
PlanetAPI Amalthea = _sys.addPlanet("Amalthea", Jupiter, "Amalthea", "barren", 264,
10, _jupRad + (_moonExt * (float) Math.log(_moonInt * 0.0012f + 1)), 0.498f);
PlanetAPI Thebe = _sys.addPlanet("Thebe", Jupiter, "Thebe", "barren", 125, 5, _jupRad
+ (_moonExt * (float) Math.log(_moonInt * 0.0015f + 1)), 0.675f);
```

```
PlanetAPI Io = _sys.addPlanet("Io", Jupiter, "Io", "lava_minor", 97, 44, _jupRad + (750f
* (float) Math.log(_moonInt * 0.0028f + 1)), 4);
MarketAPI marketIo = Io.getMarket();
marketIo.addCondition("ruins_scattered");
marketIo.addCondition("extreme_tectonic_activity");
marketIo.addCondition("cold");
marketIo.addCondition("low_gravity");
marketIo.addCondition("thin_atmosphere");
marketIo.addCondition("toxic_atmosphere");
marketIo.addCondition("ore_ultrarich");
marketIo.addCondition("rare_ore_ultrarich");
marketIo.addCondition("poor_light");
```

```
SectorEntityToken ioField = _sys.addTerrain(Terrain.MAGNETIC_FIELD, new
com.fs.starfarer.api.impl.campaign.terrain.MagneticFieldTerrainPlugin.MagneticFieldPar
ams(100f, 250f, Io, 48, 100, new Color(30, 4, 44, 23), 1.0f));
ioField.setCircularOrbit(Io, 0, 0, 100);
ioField.setName("Io Plasma Torus");
```

```
PlanetAPI Europa = _sys.addPlanet("Europa", Jupiter, "Europa", "cryovolcanic", 292,
34, _jupRad + (750f * (float) Math.log(_moonInt * 0.0045f + 1)), 8);
MarketAPI marketEuropa = Europa.getMarket();
marketEuropa.addCondition("irradiated");
```

```
marketEuropa.addCondition("tectonic_activity");
marketEuropa.addCondition("cold");
marketEuropa.addCondition("volatiles_plentiful");
marketEuropa.addCondition("ruins_widespread");
marketEuropa.addCondition("ore_moderate");
marketEuropa.addCondition("rare_ore_sparse");
marketEuropa.addCondition("no_atmosphere");
marketEuropa.addCondition("low_gravity");
marketEuropa.addCondition("poor_light");
```

```
PlanetAPI Ganymede = _sys.addPlanet("Ganymede", Jupiter, "Ganymede",
"rocky_ice", 197, 68, _jupRad + (750f * (float) Math.log(_moonInt * 0.0072f + 1)), 16);
MarketAPI marketGanymede = Ganymede.getMarket();
marketGanymede.addCondition("cold");
marketGanymede.addCondition("low_gravity");
marketGanymede.addCondition("thin_atmosphere");
marketGanymede.addCondition("volatiles_plentiful");
marketGanymede.addCondition("organics_trace");
marketGanymede.addCondition("ruins_widespread");
marketGanymede.addCondition("poor_light");
```

```
SectorEntityToken ganymedeField = _sys.addTerrain(Terrain.MAGNETIC_FIELD, new
com.fs.starfarer.api.impl.campaign.terrain.MagneticFieldTerrainPlugin.MagneticFieldPar
ams(50f, 100f, Ganymede, 120f, 100f, new Color(40, 5, 60, 40), 0.5f));
ganymedeField.setCircularOrbit(Ganymede, 0, 0, 100);
ganymedeField.setName("Ganymede's Magnetosphere");
```

```
PlanetAPI Callisto = _sys.addPlanet("Callisto", Jupiter, "Callisto", "rocky_ice", 161, 62,
_jupRad + (750f * (float) Math.log(_moonInt * 0.0126f + 1)), 32);
MarketAPI marketCallisto = Callisto.getMarket();
marketCallisto.addCondition("cold");
marketCallisto.addCondition("low_gravity");
marketCallisto.addCondition("no_atmosphere");
marketCallisto.addCondition("volatiles_plentiful");
marketCallisto.addCondition("ore_sparse");
marketCallisto.addCondition("ruins_extensive");
marketCallisto.addCondition("poor_light");
```

```
PlanetAPI Himalia = _sys.addPlanet("Himalia", Jupiter, "Himalia", "barren", 45, 6,
_jupRad + (500f * (float) Math.log(_moonInt * 0.076f + 1)), 250);
```

```

PlanetAPI Elara = _sys.addPlanet("Elara", Jupiter, "Elara", "barren", 120, 4, _jupRad +
(500f * (float) Math.log(_moonInt * 0.078f + 1)), 260);
PlanetAPI Lysithea = _sys.addPlanet("Lysithea", Jupiter, "Lysithea", "barren", 290, 3,
_jupRad + (500f * (float) Math.log(_moonInt * 0.078f + 1)), 259);
PlanetAPI Ananke = _sys.addPlanet("Ananke", Jupiter, "Ananke", "barren", 210, 3,
_jupRad + (500f * (float) Math.log(_moonInt * 0.142f + 1)), -613);
PlanetAPI Carme = _sys.addPlanet("Carme", Jupiter, "Carme", "barren", 60, 4, _jupRad +
(500f * (float) Math.log(_moonInt * 0.156f + 1)), -702);
PlanetAPI Pasiphae = _sys.addPlanet("Pasiphae", Jupiter, "Pasiphae", "barren", 330, 5,
_jupRad + (500f * (float) Math.log(_moonInt * 0.157f + 1)), -708);
PlanetAPI Sinope = _sys.addPlanet("Sinope", Jupiter, "Sinope", "barren", 15, 3,
_jupRad + (500f * (float) Math.log(_moonInt * 0.160f + 1)), -724);

```

```

_sys.addCustomEntity(null, "Jupiter L4 Stable Location", "stable_location",
"neutral").setCircularOrbit(_star, 286, _jupOrbitRadius, 4015);

com.fs.starfarer.api.impl.campaign.terrain.AsteroidFieldTerrainPlugin.AsteroidFieldPara
ms greekCenterP = new
com.fs.starfarer.api.impl.campaign.terrain.AsteroidFieldTerrainPlugin.AsteroidFieldPara
ms(1500f, 500f, (int)(_astfrequency * 1.6f), (int)(_astfrequency * 1.6f), 5f, _astsize, "Greek
Camp");
com.fs.starfarer.api.impl.campaign.terrain.AsteroidFieldTerrainPlugin.AsteroidFieldPara
ms greekSideP = new
com.fs.starfarer.api.impl.campaign.terrain.AsteroidFieldTerrainPlugin.AsteroidFieldPara
ms(1250f, 250f, (int)(_astfrequency * 1), (int)(_astfrequency * 1.0f), 5f, _astsize * 0.7f,
"Greek Camp");

SectorEntityToken greekCamp = _sys.addTerrain(Terrain.ASTEROID_FIELD,
greekCenterP);
greekCamp.setCircularOrbit(_star, 286, _jupOrbitRadius, 4015);
SectorEntityToken greekHead = _sys.addTerrain(Terrain.ASTEROID_FIELD,
greekSideP);
greekHead.setCircularOrbit(_star, 294, _jupOrbitRadius, 4015);
SectorEntityToken greekTail = _sys.addTerrain(Terrain.ASTEROID_FIELD, greekSideP);
greekTail.setCircularOrbit(_star, 278, _jupOrbitRadius, 4015);

SectorEntityToken patroclusBarycenter = _sys.addCustomEntity(null, null,
"sensor_ghost", "neutral");
patroclusBarycenter.setCircularOrbit(_star, 286, _jupOrbitRadius - 200, 4015);

```

```
PlanetAPI Patroclus = _sys.addPlanet("Patroclus", patroclusBarycenter, "Patroclus",
"barren", 0, 15, 50, 4);
PlanetAPI Menoetius = _sys.addPlanet("Menoetius", patroclusBarycenter, "Menoetius",
"barren", 180, 14, 50, 4); // Slightly smaller radius visually\
SectorEntityToken lucy = _sys.addCustomEntity("lucy_probe", "Lucy",
"generic_probe", "neutral");
lucy.setCircularOrbit(Patroclus, 50, 60, 120);
```

```
PlanetAPI Achilles = _sys.addPlanet("Achilles", _star, "Achilles", "barren", 292, 15,
jupOrbitRadius - 150, 4015);
PlanetAPI Mentor = _sys.addPlanet("Mentor", _star, "Mentor", "barren", 280, 15,
jupOrbitRadius + 150, 4015);
```

```
_sys.addCustomEntity(null, "Jupiter L5 Stable Location", "stable_location",
"neutral").setCircularOrbit(_star, 166, jupOrbitRadius, 4015);
```

```
com.fs.starfarer.api.impl.campaign.terrain.AsteroidFieldTerrainPlugin.AsteroidFieldPara
ms trojanCenterP = new
com.fs.starfarer.api.impl.campaign.terrain.AsteroidFieldTerrainPlugin.AsteroidFieldPara
ms(1500f, 500f, (int)(_astfrequency * 0.3f), (int)(_astfrequency * 1.0f), 5f, _astsize * 0.9f,
"Trojan Camp");
com.fs.starfarer.api.impl.campaign.terrain.AsteroidFieldTerrainPlugin.AsteroidFieldPara
ms trojanSideP = new
com.fs.starfarer.api.impl.campaign.terrain.AsteroidFieldTerrainPlugin.AsteroidFieldPara
ms(1250f, 250f, (int)(_astfrequency * 0.15f), (int)(_astfrequency * 0.6f), 5f, _astsize * 0.6f,
"Trojan Camp");
```

```
SectorEntityToken trojanCamp = _sys.addTerrain(Terrain.ASTEROID_FIELD,
trojanCenterP);
trojanCamp.setCircularOrbit(_star, 166, jupOrbitRadius, 4015);
SectorEntityToken trojanHead = _sys.addTerrain(Terrain.ASTEROID_FIELD,
trojanSideP);
trojanHead.setCircularOrbit(_star, 174, jupOrbitRadius, 4015);
SectorEntityToken trojanTail = _sys.addTerrain(Terrain.ASTEROID_FIELD, trojanSideP);
trojanTail.setCircularOrbit(_star, 158, jupOrbitRadius, 4015);
```

```
PlanetAPI Hektor = _sys.addPlanet("Hektor", _star, "Hektor", "barren", 162, 20,
jupOrbitRadius + 300, 4015);
PlanetAPI Skamandrios = _sys.addPlanet("Skamandrios", Hektor, "Skamandrios",
"barren", 48, 6, 70, 3);
```

```

PlanetAPI Agamemnon = _sys.addPlanet("Agamemnon", _star, "Agamemnon",
"barren", 160, 15, jupOrbitRadius + 100, 4015);

float hildaRadius = _solRad + _extCompressor * (float) Math.log(_intCompressor * 3.97f
+ 1);
PlanetAPI Hilda = _sys.addPlanet("Hilda", _star, "Hilda", "barren", 46, 15, hildaRadius,
2889);

com.fs.starfarer.api.impl.campaign.terrain.AsteroidFieldTerrainPlugin.AsteroidFieldPara
ms hildaCenterP = new
com.fs.starfarer.api.impl.campaign.terrain.AsteroidFieldTerrainPlugin.AsteroidFieldPara
ms(1000f, 400f, (int)(_astfrequency * 0.3f), (int)(_astfrequency * 1.2f), 5f, _astsize * 0.8f,
"Hilda Triangle");
com.fs.starfarer.api.impl.campaign.terrain.AsteroidFieldTerrainPlugin.AsteroidFieldPara
ms hildaSideP = new
com.fs.starfarer.api.impl.campaign.terrain.AsteroidFieldTerrainPlugin.AsteroidFieldPara
ms(800f, 200f, (int)(_astfrequency * 0.15f), (int)(_astfrequency * 0.6f), 5f, _astsize * 0.5f,
"Hilda Triangle");

SectorEntityToken hildaL3Center = _sys.addTerrain(Terrain.ASTEROID_FIELD,
hildaCenterP);
hildaL3Center.setCircularOrbit(_star, 46, hildaRadius, 2889);
SectorEntityToken hildaL3Head = _sys.addTerrain(Terrain.ASTEROID_FIELD,
hildaSideP);
hildaL3Head.setCircularOrbit(_star, 54, hildaRadius, 2889);
SectorEntityToken hildaL3Tail = _sys.addTerrain(Terrain.ASTEROID_FIELD, hildaSideP);
hildaL3Tail.setCircularOrbit(_star, 38, hildaRadius, 2889);

SectorEntityToken hildaL4Center = _sys.addTerrain(Terrain.ASTEROID_FIELD,
hildaCenterP);
hildaL4Center.setCircularOrbit(_star, 286, hildaRadius, 2889);
SectorEntityToken hildaL4Head = _sys.addTerrain(Terrain.ASTEROID_FIELD,
hildaSideP);
hildaL4Head.setCircularOrbit(_star, 294, hildaRadius, 2889);
SectorEntityToken hildaL4Tail = _sys.addTerrain(Terrain.ASTEROID_FIELD, hildaSideP);
hildaL4Tail.setCircularOrbit(_star, 278, hildaRadius, 2889);

SectorEntityToken hildaL5Center = _sys.addTerrain(Terrain.ASTEROID_FIELD,
hildaCenterP);
hildaL5Center.setCircularOrbit(_star, 166, hildaRadius, 2889);

```

```
SectorEntityToken hildaL5Head = _sys.addTerrain(Terrain.ASTEROID_FIELD,
hildaSideP);
hildaL5Head.setCircularOrbit(_star, 174, hildaRadius, 2889);
SectorEntityToken hildaL5Tail = _sys.addTerrain(Terrain.ASTEROID_FIELD, hildaSideP);
hildaL5Tail.setCircularOrbit(_star, 158, hildaRadius, 2889);
```

```
float _satRad = 446f;
PlanetAPI Saturn = _sys.addPlanet("Saturn", _star, "Saturn", "gas_giant", 17, 346,
_satRad + _extCompressor * (float) Math.log(_intCompressor * 9.58f + 1), 10585);
MarketAPI marketSaturn = Saturn.getMarket();
```

```
SectorEntityToken saturnInnerField = _sys.addTerrain(Terrain.MAGNETIC_FIELD, new
com.fs.starfarer.api.impl.campaign.terrain.MagneticFieldTerrainPlugin.MagneticFieldPar
ams(400f, 600f, Saturn, 346, 700, new Color(25, 50, 25, 20), .5f));
saturnInnerField.setCircularOrbit(Saturn, 0, 0, 100);
saturnInnerField.setName("Saturn's Magnetosphere");

_sys.addRingBand(Saturn, "misc", "rings_special0", 256, 0, Color.RED, 110, 485, 30);
_sys.addTerrain(Terrain.RING, new
com.fs.starfarer.api.impl.campaign.terrain.RingSystemTerrainPlugin.RingParams(110,
485, Saturn, "C Ring")).setCircularOrbit(Saturn, 0, 0, 100);

_sys.addRingBand(Saturn, "misc", "rings_dust0", 256, 1, Color.RED, 5, 540, 30);

_sys.addRingBand(Saturn, "misc", "rings_special0", 256, 1, Color.RED, 200, 640, 30);
_sys.addTerrain(Terrain.RING, new
com.fs.starfarer.api.impl.campaign.terrain.RingSystemTerrainPlugin.RingParams(200,
640, Saturn, "B Ring")).setCircularOrbit(Saturn, 0, 0, 100);

_sys.addRingBand(Saturn, "misc", "rings_special0", 50, 1, Color.RED, 10, 745, 30);

_sys.addRingBand(Saturn, "misc", "rings_special0", 100, 3, Color.RED, 30, 770, 30);
_sys.addTerrain(Terrain.RING, new
com.fs.starfarer.api.impl.campaign.terrain.RingSystemTerrainPlugin.RingParams(30,
770, Saturn, "Inner A Ring")).setCircularOrbit(Saturn, 0, 0, 100);

_sys.addRingBand(Saturn, "misc", "rings_special0", 50, 9, Color.RED, 20, 800, 30);
```

```
_sys.addTerrain(Terrain.RING, new
com.fs.starfarer.api.impl.campaign.terrain.RingSystemTerrainPlugin.RingParams(20,
800, Saturn, "Outer A Ring")).setCircularOrbit(Saturn, 0, 0, 100);

_sys.addRingBand(Saturn, "misc", "rings_dust0", 256, 1, Color.RED, 5, 835, 30);
_sys.addTerrain(Terrain.RING, new
com.fs.starfarer.api.impl.campaign.terrain.RingSystemTerrainPlugin.RingParams(5, 835,
Saturn, "F Ring")).setCircularOrbit(Saturn, 0, 0, 100);
```

```
PlanetAPI Pan = _sys.addPlanet("Pan", Saturn, "Pan", "rocky_ice", 125, 5, _satRad +
340, 0.575f);
PlanetAPI Daphnis = _sys.addPlanet("Daphnis", Saturn, "Daphnis", "rocky_ice", 216, 5,
_satRad + 341, 0.594f);
PlanetAPI Atlas = _sys.addPlanet("Atlas", Saturn, "Atlas", "rocky_ice", 136, 5, _satRad
+ 342, 0.601f);
PlanetAPI Pandora = _sys.addPlanet("Pandora", Saturn, "Pandora", "rocky_ice", 351,
7, _satRad + 375, 0.628f);
PlanetAPI Epimetheus = _sys.addPlanet("Epimetheus", Saturn, "Epimetheus",
"rocky_ice", 181, 7, _satRad + (_moonExt * (float) Math.log(_moonInt * 0.001f + 1)),
0.694f);
PlanetAPI Janus = _sys.addPlanet("Janus", Saturn, "Janus", "rocky_ice", 338, 7,
_satRad + (_moonExt * (float) Math.log(_moonInt * 0.001f + 1)), 0.694f);

PlanetAPI Mimas = _sys.addPlanet("Mimas", Saturn, "Mimas", "rocky_ice", 32, 10,
_satRad + (_moonExt * (float) Math.log(_moonInt * 0.0012f + 1)), 4);
MarketAPI marketMimas = Mimas.getMarket();

PlanetAPI Enceladus = _sys.addPlanet("Enceladus", Saturn, "Enceladus",
"cryovolcanic", 127, 12, _satRad + (_moonExt * (float) Math.log(_moonInt * 0.0016f +
1)), 5);
SectorEntityToken enceladusField = _sys.addTerrain(Terrain.MAGNETIC_FIELD, new
com.fs.starfarer.api.impl.campaign.terrain.MagneticFieldTerrainPlugin.MagneticFieldPar
ams(50f, 100f, Enceladus, 70, 70, new Color(40, 40, 40, 20), 0.05f));
enceladusField.setCircularOrbit(Enceladus, 0, 0, 100);
enceladusField.setName("Enceladus Plasma Dynamo");
MarketAPI marketEnceladus = Enceladus.getMarket();
marketEnceladus.addCondition("cryovolcanic");
marketEnceladus.addCondition("volatiles_plentiful");
marketEnceladus.addCondition("low_gravity");
marketEnceladus.addCondition("no_atmosphere");
```

```

marketEnceladus.addCondition("cold");
marketEnceladus.addCondition("poor_light");
marketEnceladus.addCondition("ore_rich");
marketEnceladus.addCondition("rare_ore_moderate");
marketEnceladus.addCondition("tectonic_activity");

PlanetAPI Tethys = _sys.addPlanet("Tethys", Saturn, "Tethys", "frozen", 70, 20,
_satRad + (_moonExt * (float) Math.log(_moonInt * 0.0019f + 1)), 7);
MarketAPI marketTethys = Tethys.getMarket();

PlanetAPI Telesto = _sys.addPlanet("Telesto", Saturn, "Telesto", "rocky_ice", 10, 5,
_satRad + (_moonExt * (float) Math.log(_moonInt * 0.0019f + 1)), 7); PlanetAPI Calypso
= _sys.addPlanet("Calypso", Saturn, "Calypso", "rocky_ice", 130, 5, _satRad +
(_moonExt * (float) Math.log(_moonInt * 0.0019f + 1)), 7);

PlanetAPI Dione = _sys.addPlanet("Dione", Saturn, "Dione", "frozen", 130, 20, _satRad
+ (_moonExt * (float) Math.log(_moonInt * 0.0025f + 1)), 11);
MarketAPI marketDione = Dione.getMarket();

PlanetAPI Helene = _sys.addPlanet("Helene", Saturn, "Helene", "rocky_ice", 190, 5,
_satRad + (_moonExt * (float) Math.log(_moonInt * 0.0025f + 1)), 11);
PlanetAPI Polydeuces = _sys.addPlanet("Polydeuces", Saturn, "Polydeuces",
"rocky_ice", 70, 3, _satRad + (_moonExt * (float) Math.log(_moonInt * 0.0025f + 1)), 11);

PlanetAPI Rhea = _sys.addPlanet("Rhea", Saturn, "Rhea", "frozen", 130, 30, _satRad +
(_moonExt * (float) Math.log(_moonInt * 0.0035f + 1)), 20);
MarketAPI marketRhea = Rhea.getMarket();

PlanetAPI Titan = _sys.addPlanet("Titan", Saturn, "Titan", "frozen", 130, 60, _satRad +
(_moonExt * (float) Math.log(_moonInt * 0.0081f + 1)), 30);
Titan.getSpec().setTexture("graphics/planets/toxic_atmosphere.jpg");
Titan.getSpec().setAtmosphereColor(new Color(210, 140, 70, 255));
Titan.getSpec().setCloudColor(new Color(200, 130, 60, 150));
Titan.getSpec().setIconColor(new Color(210, 140, 70, 255));
Titan.applySpecChanges();
MarketAPI marketTitan = Titan.getMarket();
marketTitan.addCondition("very_cold");
marketTitan.addCondition("poor_light");
marketTitan.addCondition("organics_plentiful");
marketTitan.addCondition("volatiles_abundant");
marketTitan.addCondition("ore_moderate");
marketTitan.addCondition("rare_ore_moderate");

```

```

marketTitan.addCondition("ruins_vast");

PlanetAPI Hyperion = _sys.addPlanet("Hyperion", Saturn, "Hyperion",
"rocky_unstable", 130, 10, _satRad + (_moonExt * (float) Math.log(_moonInt * 0.0099f +
1)), 35);
MarketAPI marketHyperion = Hyperion.getMarket();

PlanetAPI Iapetus = _sys.addPlanet("Iapetus", Saturn, "Iapetus", "rocky_ice", 130, 30,
_satRad + (_moonExt * (float) Math.log(_moonInt * 0.023f + 1)), 20);

PlanetAPI Phoebe = _sys.addPlanet("Phoebe", Saturn, "Phoebe", "barren", 130, 5,
_satRad + (_irrExt * (float) Math.log(_moonInt * 0.086f + 1)), -550);
JumpPointAPI _jp3 = Global.getFactory().createJumpPoint("jp_phoebe", "Phoebe Jump
Point");
_jp3.setStandardWormholeToHyperspaceVisual();
_jp3.setCircularOrbit(Phoebe, 0, 100, 30);
_sys.addEntity(_jp3);
MarketAPI marketPhoebe = Phoebe.getMarket();
marketPhoebe.addCondition("very_cold");
marketPhoebe.addCondition("poor_light");
marketPhoebe.addCondition("low_gravity");
marketPhoebe.addCondition("no_atmosphere");
marketPhoebe.addCondition("ruins_widespread");
marketPhoebe.addCondition("rare_ore_ultrarich");
marketPhoebe.addCondition("ore_rich");
marketPhoebe.addCondition("volatiles_diffuse");
marketPhoebe.addCondition("inimical_biosphere");

PlanetAPI Ymir = _sys.addPlanet("Ymir", Saturn, "Ymir", "barren", 300, 5, _satRad +
(_irrExt * (float) Math.log(_moonInt * 0.154f + 1)), -1315);
PlanetAPI Siarnaq = _sys.addPlanet("Siarnaq", Saturn, "Siarnaq", "barren", 45, 8,
_satRad + (_irrExt * (float) Math.log(_moonInt * 0.117f + 1)), 896);
PlanetAPI Albiorix = _sys.addPlanet("Albiorix", Saturn, "Albiorix", "barren", 200, 6,
_satRad + (_irrExt * (float) Math.log(_moonInt * 0.108f + 1)), 783);

PlanetAPI UO14 = _sys.addPlanet("2019 UO14", _star, "2019 UO14", "rocky_ice", 317,
12, _solRad + _extCompressor * (float) Math.log(_intCompressor * 9.58f + 1), 10585);

```

```
_sys.addCustomEntity(null, null, "derelict_cryosleeper",
"neutral").setCircularOrbit(_star, 77, _solRad + _extCompressor * (float)
Math.log(_intCompressor * 9.58f + 1), 10585);
```

```
float _uraRad = 225f;
PlanetAPI Uranus = _sys.addPlanet("Uranus", _star, "Uranus", "ice_giant", 336,
_uraRad,_solRad + _extCompressor * (float) Math.log(_intCompressor * 19.19f + 1),
30660);
Uranus.getSpec().setTexture("graphics/planets/volturn.jpg");
Uranus.getSpec().setAtmosphereColor(new Color(160, 210, 255, 160));
Uranus.getSpec().setCloudColor(new Color(0, 0, 0, 0));
Uranus.getSpec().setIconColor(new Color(160, 210, 255, 160));
Uranus.applySpecChanges();
```

```
SectorEntityToken uranusInnerField = _sys.addTerrain(Terrain.MAGNETIC_FIELD, new
com.fs.starfarer.api.impl.campaign.terrain.MagneticFieldTerrainPlugin.MagneticFieldPar
ams(250f, 400f, Uranus, 225, 450, new Color(5, 15, 50, 30), 0.5f));
uranusInnerField.setCircularOrbit(Uranus, 0, 0, 100);
uranusInnerField.setName("Uranus' Magnetosphere");

_sys.addTerrain(Terrain.RING, new
com.fs.starfarer.api.impl.campaign.terrain.RingSystemTerrainPlugin.RingParams(30,
_uraRad + 60, Uranus, "Zeta Ring")).setCircularOrbit(Uranus, 0, 0, 100);

_sys.addRingBand(Uranus, "misc", "rings_dust0", 256, 1, Color.RED, 4, _uraRad + 85,
30);
_sys.addRingBand(Uranus, "misc", "rings_dust0", 256, 1, Color.RED, 4, _uraRad + 90,
30);
_sys.addRingBand(Uranus, "misc", "rings_dust0", 256, 1, Color.RED, 4, _uraRad + 95,
30);
_sys.addTerrain(Terrain.RING, new
com.fs.starfarer.api.impl.campaign.terrain.RingSystemTerrainPlugin.RingParams(15,
_uraRad + 90, Uranus, "Inner Rings")).setCircularOrbit(Uranus, 0, 0, 100);

_sys.addRingBand(Uranus, "misc", "rings_dust0", 256, 1, Color.RED, 5, _uraRad + 110,
30);
_sys.addRingBand(Uranus, "misc", "rings_dust0", 256, 1, Color.RED, 5, _uraRad + 118,
30);
```

```

    _sys.addTerrain(Terrain.RING, new
com.fs.starfarer.api.impl.campaign.terrain.RingSystemTerrainPlugin.RingParams(15,
_uraRad + 114, Uranus, "Alpha-Beta Rings")).setCircularOrbit(Uranus, 0, 0, 100);

    _sys.addRingBand(Uranus, "misc", "rings_dust0", 256, 1, Color.RED, 4, _uraRad + 130,
30);
    _sys.addRingBand(Uranus, "misc", "rings_dust0", 256, 1, Color.RED, 4, _uraRad + 136,
30);
    _sys.addRingBand(Uranus, "misc", "rings_dust0", 256, 1, Color.RED, 4, _uraRad + 142,
30);
    _sys.addTerrain(Terrain.RING, new
com.fs.starfarer.api.impl.campaign.terrain.RingSystemTerrainPlugin.RingParams(16,
_uraRad + 136, Uranus, "Delta Rings")).setCircularOrbit(Uranus, 0, 0, 100);

    _sys.addRingBand(Uranus, "misc", "rings_dust0", 256, 1, Color.RED, 3, _uraRad + 150,
30);
    _sys.addRingBand(Uranus, "misc", "rings_special0", 256, 1, Color.RED, 5, _uraRad +
162, 30);
    _sys.addTerrain(Terrain.RING, new
com.fs.starfarer.api.impl.campaign.terrain.RingSystemTerrainPlugin.RingParams(12,
_uraRad + 162, Uranus, "Epsilon Ring")).setCircularOrbit(Uranus, 0, 0, 100);

    _sys.addRingBand(Uranus, "misc", "rings_dust0", 256, 1, Color.RED, 5, _uraRad + 205,
30);
    _sys.addTerrain(Terrain.RING, new
com.fs.starfarer.api.impl.campaign.terrain.RingSystemTerrainPlugin.RingParams(5,
_uraRad + 205, Uranus, "Nu Ring")).setCircularOrbit(Uranus, 0, 0, 100);

```

```

PlanetAPI Portia = _sys.addPlanet("Portia", Uranus, "Portia", "rocky_ice", 45, 5,
_uraRad + (_moonExt * (float) Math.log(_moonInt * 0.00044f + 1)), 0.513f);
PlanetAPI Juliet = _sys.addPlanet("Juliet", Uranus, "Juliet", "rocky_ice", 90, 4, _uraRad
+ (_moonExt * (float) Math.log(_moonInt * 0.00043f + 1)), 0.493f);
PlanetAPI Belinda = _sys.addPlanet("Belinda", Uranus, "Belinda", "rocky_ice", 135, 4,
_uraRad + (_moonExt * (float) Math.log(_moonInt * 0.00050f + 1)), 0.623f);
PlanetAPI Cressida = _sys.addPlanet("Cressida", Uranus, "Cressida", "rocky_ice", 180,
4, _uraRad + (_moonExt * (float) Math.log(_moonInt * 0.00041f + 1)), 0.623f);
PlanetAPI Rosalind = _sys.addPlanet("Rosalind", Uranus, "Rosalind", "rocky_ice", 225,
3,
_uraRad + (_moonExt * (float) Math.log(_moonInt * 0.00047f + 1)), 0.463f);

```

```
PlanetAPI Desdemona = _sys.addPlanet("Desdemona", Uranus, "Desdemona",  
"rocky_ice", 270, 3, _uraRad + (_moonExt * (float) Math.log(_moonInt * 0.00042f + 1)),  
0.474f);  
PlanetAPI Bianca = _sys.addPlanet("Bianca", Uranus, "Bianca", "rocky_ice", 315, 3,  
_uraRad + (_moonExt * (float) Math.log(_moonInt * 0.00039f + 1)), 0.435f);  
  
PlanetAPI Puck = _sys.addPlanet("Puck", Uranus, "Puck", "rocky_ice", 322, 5,  
_uraRad + (_moonExt * (float) Math.log(_moonInt * 0.00057f + 1)), 0.762f);  
  
PlanetAPI Miranda = _sys.addPlanet("Miranda", Uranus, "Miranda", "rocky_ice", 172,  
10, _uraRad + (_moonExt * (float) Math.log(_moonInt * 0.00086f + 1)), 8);  
  
PlanetAPI Ariel = _sys.addPlanet("Ariel", Uranus, "Ariel", "rocky_ice", 89, 20, _uraRad  
+ (_moonExt * (float) Math.log(_moonInt * 0.00127f + 1)), 8);  
MarketAPI marketAriel = Ariel.getMarket();  
  
PlanetAPI Umbriel = _sys.addPlanet("Umbriel", Uranus, "Umbriel", "rocky_ice", 345,  
22, _uraRad + (_moonExt * (float) Math.log(_moonInt * 0.00178f + 1)), 16);  
  
PlanetAPI Titania = _sys.addPlanet("Titania", Uranus, "Titania", "rocky_ice", 233, 30,  
_uraRad + (_moonExt * (float) Math.log(_moonInt * 0.00292f + 1)), 32);  
MarketAPI marketTitania = Titania.getMarket();  
marketTitania.addCondition("very_cold");  
marketTitania.addCondition("dark");  
marketTitania.addCondition("no_atmosphere");  
marketTitania.addCondition("low_gravity");  
marketTitania.addCondition("ore_ultrarich");  
marketTitania.addCondition("rare_ore_ultrarich");  
marketTitania.addCondition("volatiles_plentiful");  
marketTitania.addCondition("ruins_vast");  
  
PlanetAPI Oberon = _sys.addPlanet("Oberon", Uranus, "Oberon", "rocky_ice", 23, 28,  
_uraRad + (_moonExt * (float) Math.log(_moonInt * 0.00390f + 1)), 52);  
MarketAPI marketOberon = Oberon.getMarket();  
marketOberon.addCondition("very_cold");  
marketOberon.addCondition("dark");  
marketOberon.addCondition("no_atmosphere");  
marketOberon.addCondition("low_gravity");  
marketOberon.addCondition("ore_rich");  
marketOberon.addCondition("rare_ore_moderate");  
marketOberon.addCondition("volatiles_diffuse");
```

```
marketOberon.addCondition("ruins_widespread");

PlanetAPI Sycorax = _sys.addPlanet("Sycorax", Uranus, "Sycorax", "barren", 330, 6,
_uraRad + (_irrExt * (float) Math.log(_moonInt * 0.081f + 1)), -1288);
PlanetAPI Caliban = _sys.addPlanet("Caliban", Uranus, "Caliban", "barren", 15, 4,
_uraRad + (_irrExt * (float) Math.log(_moonInt * 0.048f + 1)), -579);
PlanetAPI Prospero = _sys.addPlanet("Prospero", Uranus, "Prospero", "barren", 120, 3,
_uraRad + (_irrExt * (float) Math.log(_moonInt * 0.108f + 1)), -1978);
PlanetAPI Setebos = _sys.addPlanet("Setebos", Uranus, "Setebos", "barren", 240, 3,
_uraRad + (_irrExt * (float) Math.log(_moonInt * 0.116f + 1)), -2225);
PlanetAPI Stephano = _sys.addPlanet("Stephano", Uranus, "Stephano", "barren", 60,
2, _uraRad + (_irrExt * (float) Math.log(_moonInt * 0.053f + 1)), -677);
```

```
PlanetAPI QF99 = _sys.addPlanet("2011 QF99", _star, "2011 QF99", "rocky_ice", 36,
14, _solRad + _extCompressor * (float) Math.log(_intCompressor * 19.1f + 1), 30660);
PlanetAPI YX49 = _sys.addPlanet("2014 YX49", _star, "2014 YX49", "rocky_ice", 38,
16, _solRad + _extCompressor * (float) Math.log(_intCompressor * 19.1f + 1), 30660);
```

```
_sys.addCustomEntity(null, "Uranus L5 Stable Location", "stable_location",
"neutral").setCircularOrbit(_star, 276, _solRad + _extCompressor * (float)
Math.log(_intCompressor * 19.19f + 1), 30660);
```

```
float _nepRad = 260f;
float nepOrbitRadius = _extCompressor * (float) Math.log(_intCompressor * 30.07f + 1);
PlanetAPI Neptune = _sys.addPlanet("Neptune", _star, "Neptune", "ice_giant", 127,
200, nepOrbitRadius, 49860);
MarketAPI marketNeptune = Neptune.getMarket();
marketNeptune.addCondition("very_cold");
marketNeptune.addCondition("dark");
marketNeptune.addCondition("high_gravity");
marketNeptune.addCondition("dense_atmosphere");
marketNeptune.addCondition("extreme_weather");
marketNeptune.addCondition("volatiles_plentiful");
```

```

SectorEntityToken neptuneInnerField = _sys.addTerrain(Terrain.MAGNETIC_FIELD, new
com.fs.starfarer.api.impl.campaign.terrain.MagneticFieldTerrainPlugin.MagneticFieldPar
ams(250f, 400f, Neptune, 200, 400, new Color(5, 15, 50, 30), 0.5f));
neptuneInnerField.setCircularOrbit(Neptune, 0, 0, 100);
neptuneInnerField.setName("Neptune's Magnetosphere");

_sys.addRingBand(Neptune, "misc", "rings_dust0", 256, 0, Color.RED, 10, _nepRad +
(_moonExt * (float) Math.log(_moonInt * 0.00028f + 1)), 30);
_sys.addTerrain(Terrain.RING, new
com.fs.starfarer.api.impl.campaign.terrain.RingSystemTerrainPlugin.RingParams(60,
_nepRad + (_moonExt * (float) Math.log(_moonInt * 0.00028f + 1)), Neptune, "Galle
Ring")).setCircularOrbit(Neptune, 0, 0, 100);

_sys.addRingBand(Neptune, "misc", "rings_dust0", 256, 1, Color.RED, 10, _nepRad +
(_moonExt * (float) Math.log(_moonInt * 0.00036f + 1)), 30);
_sys.addTerrain(Terrain.RING, new
com.fs.starfarer.api.impl.campaign.terrain.RingSystemTerrainPlugin.RingParams(20,
_nepRad + (_moonExt * (float) Math.log(_moonInt * 0.00036f + 1)), Neptune, "Le Verrier
Ring")).setCircularOrbit(Neptune, 0, 0, 100);

_sys.addRingBand(Neptune, "misc", "rings_dust0", 256, 1, Color.RED, 10, _nepRad +
50, 30);
_sys.addRingBand(Neptune, "misc", "rings_dust0", 256, 1, Color.RED, 10, _nepRad +
50, 30);
_sys.addTerrain(Terrain.RING, new
com.fs.starfarer.api.impl.campaign.terrain.RingSystemTerrainPlugin.RingParams(10,
_nepRad + 50, Neptune, "Lassell Ring")).setCircularOrbit(Neptune, 0, 0, 100);

_sys.addRingBand(Neptune, "misc", "rings_dust0", 256, 1, Color.RED, 10, _nepRad +
(_moonExt * (float) Math.log(_moonInt * 0.00042f + 1)), 30);
_sys.addRingBand(Neptune, "misc", "rings_dust0", 256, 1, Color.RED, 20, _nepRad +
(_moonExt * (float) Math.log(_moonInt * 0.00042f + 1)), 30);
_sys.addTerrain(Terrain.RING, new
com.fs.starfarer.api.impl.campaign.terrain.RingSystemTerrainPlugin.RingParams(60,
_nepRad + (_moonExt * (float) Math.log(_moonInt * 0.00042f + 1)), Neptune, "Adams
Ring")).setCircularOrbit(Neptune, 0, 0, 100);

```

```

PlanetAPI Naiad = _sys.addPlanet("Naiad", Neptune, "Naiad", "rocky_ice", 245, 4,
_nepRad + (_moonExt * (float) Math.log(_moonInt * 0.00032f + 1)), 0.294f);
PlanetAPI Thalassa = _sys.addPlanet("Thalassa", Neptune, "Thalassa", "rocky_ice", 15,

```

```

4, _nepRad + (_moonExt * (float) Math.log(_moonInt * 0.00033f + 1)), 0.311f);
PlanetAPI Despina = _sys.addPlanet("Despina", Neptune, "Despina", "rocky_ice", 53,
5, _nepRad + (_moonExt * (float) Math.log(_moonInt * 0.00035f + 1)), 0.335f);
PlanetAPI Galatea = _sys.addPlanet("Galatea", Neptune, "Galatea", "rocky_ice", 8, 5,
_nepRad + (_moonExt * (float) Math.log(_moonInt * 0.00041f + 1)), 0.428f);
PlanetAPI Larissa = _sys.addPlanet("Larissa", Neptune, "Larissa", "rocky_ice", 320, 5,
_nepRad + (_moonExt * (float) Math.log(_moonInt * 0.00049f + 1)), 0.555f);
PlanetAPI Hippocamp = _sys.addPlanet("Hippocamp", Neptune, "Hippocamp",
"rocky_ice", 90, 3, _nepRad + (_moonExt * (float) Math.log(_moonInt * 0.00070f + 1)),
0.940f);
PlanetAPI Proteus = _sys.addPlanet("Proteus", Neptune, "Proteus", "frozen", 318, 10,
_nepRad + (_moonExt * (float) Math.log(_moonInt * 0.00078f + 1)), 1.122f);

PlanetAPI Triton = _sys.addPlanet("Triton", Neptune, "Triton", "cryovolcanic", 128, 50,
_nepRad + (_moonExt * (float) Math.log(_moonInt * 0.00237f + 1)), -5.877f);
MarketAPI marketTriton = Triton.getMarket();
marketTriton.addCondition("very_cold");
marketTriton.addCondition("dark");
marketTriton.addCondition("thin_atmosphere");
marketTriton.addCondition("low_gravity");
marketTriton.addCondition("tectonic_activity");
marketTriton.addCondition("volatiles_plentiful");
marketTriton.addCondition("ore_rich");
marketTriton.addCondition("rare_ore_rich");
marketTriton.addCondition("ruins_vast");

PlanetAPI Nereid = _sys.addPlanet("Nereid", Neptune, "Nereid", "frozen", 306, 10,
_nepRad + (_moonExt * (float) Math.log(_moonInt * 0.0369f + 1)), 360);
_sys.updateAllOrbits();
PlanetAPI Halimede = _sys.addPlanet("Halimede", Neptune, "Halimede", "barren", 140,
6, _nepRad + (_irrExt * (float) Math.log(_moonInt * 0.111f + 1)), -1879);
PlanetAPI Sao = _sys.addPlanet("Sao", Neptune, "Sao", "barren", 45, 5, _nepRad +
(_irrExt * (float) Math.log(_moonInt * 0.148f + 1)), 2914);
PlanetAPI Laomedeaia = _sys.addPlanet("Laomedeaia", Neptune, "Laomedeaia", "barren",
300, 5, _nepRad + (_irrExt * (float) Math.log(_moonInt * 0.157f + 1)), 3167);
PlanetAPI Psamathe = _sys.addPlanet("Psamathe", Neptune, "Psamathe", "barren",
120, 4, _nepRad + (_irrExt * (float) Math.log(_moonInt * 0.313f + 1)), -9115);
PlanetAPI Neso = _sys.addPlanet("Neso", Neptune, "Neso", "barren", 240, 6, _nepRad
+ (_irrExt * (float) Math.log(_moonInt * 0.329f + 1)), -9373);

```

```
_sys.addCustomEntity(null, "Neptune L4 Stable Location", "stable_location",
"neutral").setCircularOrbit(_star, 187, nepOrbitRadius, 49860);

com.fs.starfarer.api.impl.campaign.terrain.AsteroidFieldTerrainPlugin.AsteroidFieldPara
ms nepL4CoreP = new
com.fs.starfarer.api.impl.campaign.terrain.AsteroidFieldTerrainPlugin.AsteroidFieldPara
ms(0f, 2000f, (int)(_astfrequency * 0.8f), (int)(_astfrequency * 2.0f), 5f, _astsize,
"Neptune L4 Trojans");
com.fs.starfarer.api.impl.campaign.terrain.AsteroidFieldTerrainPlugin.AsteroidFieldPara
ms nepL4MidP = new
com.fs.starfarer.api.impl.campaign.terrain.AsteroidFieldTerrainPlugin.AsteroidFieldPara
ms(0f, 1600f, (int)(_astfrequency * 0.6f), (int)(_astfrequency * 1.6f), 5f, _astsize * 0.9f,
"Neptune L4 Trojans");
com.fs.starfarer.api.impl.campaign.terrain.AsteroidFieldTerrainPlugin.AsteroidFieldPara
ms nepL4OuterP = new
com.fs.starfarer.api.impl.campaign.terrain.AsteroidFieldTerrainPlugin.AsteroidFieldPara
ms(0f, 1300f, (int)(_astfrequency * 0.4f), (int)(_astfrequency * 1.2f), 5f, _astsize * 0.8f,
"Neptune L4 Trojans");
com.fs.starfarer.api.impl.campaign.terrain.AsteroidFieldTerrainPlugin.AsteroidFieldPara
ms nepL4TipP = new
com.fs.starfarer.api.impl.campaign.terrain.AsteroidFieldTerrainPlugin.AsteroidFieldPara
ms(0f, 1000f, (int)(_astfrequency * 0.2f), (int)(_astfrequency * 0.8f), 5f, _astsize * 0.7f,
"Neptune L4 Trojans");

SectorEntityToken nepL4F1 = _sys.addTerrain(TerrainASTEROID_FIELD, nepL4CoreP);
nepL4F1.setCircularOrbit(_star, 187, nepOrbitRadius, 49860);

SectorEntityToken nepL4F2 = _sys.addTerrain(TerrainASTEROID_FIELD, nepL4MidP);
nepL4F2.setCircularOrbit(_star, 190, nepOrbitRadius, 49860);
SectorEntityToken nepL4F3 = _sys.addTerrain(TerrainASTEROID_FIELD, nepL4MidP);
nepL4F3.setCircularOrbit(_star, 184, nepOrbitRadius, 49860);

SectorEntityToken nepL4F4 = _sys.addTerrain(TerrainASTEROID_FIELD,
nepL4OuterP);
nepL4F4.setCircularOrbit(_star, 193, nepOrbitRadius, 49860);
SectorEntityToken nepL4F5 = _sys.addTerrain(TerrainASTEROID_FIELD,
nepL4OuterP);
nepL4F5.setCircularOrbit(_star, 181, nepOrbitRadius, 49860);

SectorEntityToken nepL4F6 = _sys.addTerrain(TerrainASTEROID_FIELD,
nepL4OuterP);
nepL4F6.setCircularOrbit(_star, 196, nepOrbitRadius, 49860);
```

```

SectorEntityToken nepL4F7 = _sys.addTerrain(Terrain.ASTEROID_FIELD,
nepL4OuterP);
nepL4F7.setCircularOrbit(_star, 178, nepOrbitRadius, 49860);

SectorEntityToken nepL4F8 = _sys.addTerrain(Terrain.ASTEROID_FIELD, nepL4TipP);
nepL4F8.setCircularOrbit(_star, 199, nepOrbitRadius, 49860);
SectorEntityToken nepL4F9 = _sys.addTerrain(Terrain.ASTEROID_FIELD, nepL4TipP);
nepL4F9.setCircularOrbit(_star, 175, nepOrbitRadius, 49860);

PlanetAPI WG157 = _sys.addPlanet("2011 WG157", _star, "2011 WG157", "rocky_ice",
191, 20, nepOrbitRadius + 120, 49860);
PlanetAPI Otrera = _sys.addPlanet("Otrera", _star, "Otrera", "rocky_ice", 182, 18,
nepOrbitRadius - 150, 49860);
PlanetAPI Clete = _sys.addPlanet("Clete", _star, "Clete", "rocky_ice", 195, 18,
nepOrbitRadius + 200, 49860);
PlanetAPI QR322 = _sys.addPlanet("2001 QR322", _star, "2001 QR322", "rocky_ice",
179, 16, nepOrbitRadius - 80, 49860);
PlanetAPI RC158 = _sys.addPlanet("2013 RC158", _star, "2013 RC158", "rocky_ice",
198, 15, nepOrbitRadius + 100, 49860);
PlanetAPI TT191 = _sys.addPlanet("2010 TT191", _star, "2010 TT191", "rocky_ice",
176, 14, nepOrbitRadius - 180, 49860);

_sys.addCustomEntity(null, "Neptune L5 Stable Location", "stable_location",
"neutral").setCircularOrbit(_star, 67, _extCompressor * (float) Math.log(_intCompressor *
30.07f + 1), 49860);

PlanetAPI KY18 = _sys.addPlanet("2013 KY18", _star, "2013 KY18", "rocky_ice", 66,
15, _extCompressor * (float) Math.log(_intCompressor * 30.07f + 1), 49860); PlanetAPI
LC18 = _sys.addPlanet("2008 LC18", _star, "2008 LC18", "rocky_ice", 55, 14,
_extCompressor * (float) Math.log(_intCompressor * 29.85f + 1), 49860); PlanetAPI
KV18 = _sys.addPlanet("2004 KV18", _star, "2004 KV18", "rocky_ice", 78, 14,
_extCompressor * (float) Math.log(_intCompressor * 30.25f + 1), 49860); PlanetAPI
HM102 = _sys.addPlanet("2011 HM102", _star, "2011 HM102", "rocky_ice", 61, 15,
_extCompressor * (float) Math.log(_intCompressor * 30.02f + 1), 49860);

PlanetAPI EN65 = _sys.addPlanet("2010 EN65", _star, "2010 EN65", "rocky_ice", 307,
16, _extCompressor * (float) Math.log(_intCompressor * 30.07f + 1), 49860);

```

```

SectorEntityToken plutoBarycenter = _sys.addCustomEntity(null, null, "sensor_ghost",
"neutral");
plutoBarycenter.setCircularOrbit(_star, 245, _extCompressor * (float)
Math.log(_intCompressor * 39.48f + 1), 90560);

PlanetAPI Pluto = _sys.addPlanet("Pluto", plutoBarycenter, "Pluto", "cryovolcanic", 0,
50, 30, 12);
MarketAPI marketPluto = Pluto.getMarket();
marketPluto.addCondition("very_cold");
marketPluto.addCondition("dark");
marketPluto.addCondition("low_gravity");
marketPluto.addCondition("thin_atmosphere");
marketPluto.addCondition("volatiles_plentiful");
marketPluto.addCondition("ore_moderate");
marketPluto.addCondition("rare_ore_moderate");
marketPluto.addCondition("ruins_widespread");

PlanetAPI Charon = _sys.addPlanet("Charon", plutoBarycenter, "Charon", "frozen",
180, 24, 220, 12);
MarketAPI marketCharon = Charon.getMarket();
marketCharon.addCondition("very_cold");
marketCharon.addCondition("dark");
marketCharon.addCondition("low_gravity");
marketCharon.addCondition("no_atmosphere");
marketCharon.addCondition("volatiles_diffuse");
marketCharon.addCondition("ore_moderate");
marketCharon.addCondition("ruins_scattered");

JumpPointAPI _jp4 = Global.getFactory().createJumpPoint("jp_charon_l3", "Charon L3
Jump Point");
_jp4.setStandardWormholeToHyperspaceVisual();
_jp4.setCircularOrbit(plutoBarycenter, 0, 180, 12);
_sys.addEntity(_jp4);

PlanetAPI Styx = _sys.addPlanet("Styx", plutoBarycenter, "Styx", "rocky_ice", 18, 8,
450, 20);
PlanetAPI Nix = _sys.addPlanet("Nix", plutoBarycenter, "Nix", "rocky_ice", 89, 10, 550,
25);

```

```
PlanetAPI Kerberos = _sys.addPlanet("Kerberos", plutoBarycenter, "Kerberos",
"rocky_ice", 4, 8, 680, 32);
PlanetAPI Hydra = _sys.addPlanet("Hydra", plutoBarycenter, "Hydra", "rocky_ice", 306,
10, 800, 38);
```

```
PlanetAPI Arrokoth = _sys.addPlanet("Arrokoth", _star, "Arrokoth", "rocky_ice", 243, 4,
_extCompressor * (float) Math.log(_intCompressor * 60.6f + 1), 110000);
_sys.addCustomEntity("neohorizons_probe", "Neo Horizons", "generic_probe",
"neutral").setCircularOrbit(Arrokoth, 45, 100, 100);
```

```
PlanetAPI Chariklo = _sys.addPlanet("Chariklo", _star, "Chariklo", "barren", 235, 18,
_extCompressor * (float) Math.log(_intCompressor * 15.7f + 1), 22987);
_sys.addRingBand(Chariklo, "misc", "rings_dust0", 256, 0, Color.RED, 10, 190, 4);
_sys.addTerrain(Terrain.RING, new
com.fs.starfarer.api.impl.campaign.terrain.RingSystemTerrainPlugin.RingParams(10,
190, Chariklo, "Oiapoque Ring")).setCircularOrbit(Chariklo, 0, 0, 100);
_sys.addRingBand(Chariklo, "misc", "rings_dust0", 256, 1, Color.RED, 5, 205, 4);
_sys.addTerrain(Terrain.RING, new
com.fs.starfarer.api.impl.campaign.terrain.RingSystemTerrainPlugin.RingParams(5, 205,
Chariklo, "Chuí Ring")).setCircularOrbit(Chariklo, 0, 0, 100);
```

```
PlanetAPI Chiron = _sys.addPlanet("Chiron", _star, "Chiron", "barren", 160, 16,
_extCompressor * (float) Math.log(_intCompressor * 13.7f + 1), 18523);
_sys.addRingBand(Chiron, "misc", "rings_dust0", 256, 1, Color.RED, 5, 195, 4);
_sys.addRingBand(Chiron, "misc", "rings_dust0", 256, 0, Color.RED, 5, 205, 4);
SectorEntityToken chironRing = _sys.addTerrain(Terrain.RING, new
com.fs.starfarer.api.impl.campaign.terrain.RingSystemTerrainPlugin.RingParams(20,
200, Chiron, "Chiron Rings"));
chironRing.setCircularOrbit(Chiron, 0, 0, 100);
```

```
PlanetAPI Haumea = _sys.addPlanet("Haumea", _star, "Haumea", "frozen", 95, 36,
_extCompressor * (float) Math.log(_intCompressor * 43.1f + 1), 103410);
_sys.addRingBand(Haumea, "misc", "rings_dust0", 256, 1, Color.RED, 8, 300, 4);
_sys.addTerrain(Terrain.RING, new
com.fs.starfarer.api.impl.campaign.terrain.RingSystemTerrainPlugin.RingParams(8, 300,
Haumea, "Haumea Ring")).setCircularOrbit(Haumea, 0, 0, 100);
PlanetAPI Hiiaka = _sys.addPlanet("Hi'iaka", Haumea, "Hi'iaka", "rocky_ice", 306, 12,
600, 50);
```

```
PlanetAPI Namaka = _sys.addPlanet("Namaka", Haumea, "Namaka", "rocky_ice", 306, 8, 350, 20);
```

```
PlanetAPI Quaoar = _sys.addPlanet("Quaoar", _star, "Quaoar", "rocky_ice", 5, 30, _extCompressor * (float) Math.log(_intCompressor * 43.7f + 1), 105410);  
_sys.addRingBand(Quaoar, "misc", "rings_dust0", 256, 1, Color.RED, 10, 220, 10);  
_sys.addTerrain(Terrain.RING, new  
com.fs.starfarer.api.impl.campaign.terrain.RingSystemTerrainPlugin.RingParams(10, 220, Quaoar, "Quaoar Ring")).setCircularOrbit(Quaoar, 0, 0, 100);  
PlanetAPI Weywot = _sys.addPlanet("Weywot", Quaoar, "Weywot", "rocky_ice", 306, 8, 280, 20);
```

```
PlanetAPI Orcus = _sys.addPlanet("Orcus", _star, "Orcus", "frozen1", 125, 26, _extCompressor * (float) Math.log(_intCompressor * 39.17f + 1), 89410);  
PlanetAPI Vanth = _sys.addPlanet("Vanth", Orcus, "Vanth", "rocky_ice", 306, 14, 250, 9);
```

```
PlanetAPI Makemake = _sys.addPlanet("Makemake", _star, "Makemake", "frozen", 310, 34, _extCompressor * (float) Math.log(_intCompressor * 45.3f + 1), 111410);  
PlanetAPI Mk2 = _sys.addPlanet("Mk2", Makemake, "Mk2", "rocky_ice", 306, 8, 160, 12);
```

```
PlanetAPI Ixion = _sys.addPlanet("Ixion", _star, "Ixion", "frozen3", 35, 21, _extCompressor * (float) Math.log(_intCompressor * 39.8f + 1), 114410);
```

```
PlanetAPI Salacia = _sys.addPlanet("Salacia", _star, "Salacia", "frozen2", 45, 24, _extCompressor * (float) Math.log(_intCompressor * 42.0f + 1), 100410);  
PlanetAPI Actea = _sys.addPlanet("Actea", Salacia, "Actea", "rocky_ice", 306, 12, 160, 5);
```

```
PlanetAPI Varda = _sys.addPlanet("Varda", _star, "Varda", "frozen1", 15, 22, _extCompressor * (float) Math.log(_intCompressor * 45.6f + 1), 114410);  
PlanetAPI Ilmare = _sys.addPlanet("Ilmare", Varda, "Ilmare", "rocky_ice", 306, 12, 140, 5);
```

```
PlanetAPI Albion = _sys.addPlanet("Albion", _star, "Albion", "rocky_ice", 290, 6, _extCompressor * (float) Math.log(_intCompressor * 44.2f + 1), 120801); // 1992 QB1
```

```
PlanetAPI TX300 = _sys.addPlanet("2002 TX300", _star, "2002 TX300", "rocky_ice",
```

```
110, 10, _extCompressor * (float) Math.log(_intCompressor * 43.3f + 1), 120801);

PlanetAPI Chiminigagua = _sys.addPlanet("Chiminigagua", _star, "Chiminigagua",
"frozen", 358, 24, _extCompressor * (float) Math.log(_intCompressor * 59.0f + 1),
163660);

PlanetAPI Varuna = _sys.addPlanet("Varuna", _star, "Varuna", "rocky_ice", 165, 21,
_extCompressor * (float) Math.log(_intCompressor * 42.9f + 1), 101980);

SectorEntityToken lempoBarycenter = _sys.addCustomEntity(null, null, "sensor_ghost",
"neutral");
lempoBarycenter.setCircularOrbit(_star, 55, _extCompressor * (float)
Math.log(_intCompressor * 39.3f + 1), 90572);
PlanetAPI Lempo = _sys.addPlanet("Lempo", lempoBarycenter, "Lempo", "rocky_ice",
0, 16, 40, 2);
PlanetAPI Hiisi = _sys.addPlanet("Hiisi", lempoBarycenter, "Hiisi", "rocky_ice", 180, 14,
40, 2);
PlanetAPI Paha = _sys.addPlanet("Paha", lempoBarycenter, "Paha", "rocky_ice", 292,
10, 180, 50);

PlanetAPI Aya = _sys.addPlanet("Aya", _star, "Aya", "frozen2", 350, 23,
_extCompressor * (float) Math.log(_intCompressor * 47.4f + 1), 117848);

PlanetAPI Mani = _sys.addPlanet("Máni", _star, "Máni", "frozen1", 105, 24,
_extCompressor * (float) Math.log(_intCompressor * 41.9f + 1), 98429);

PlanetAPI Achlys = _sys.addPlanet("Achlys", _star, "Achlys", "frozen3", 290, 23,
_extCompressor * (float) Math.log(_intCompressor * 39.4f + 1), 90202);
PlanetAPI AchlysMoon = _sys.addPlanet("S/2005 Achlys I", Achlys, "S/2005 Achlys I",
"rocky_ice", 292, 8, 100, 5);

PlanetAPI OF201 = _sys.addPlanet("2017 OF201", _star, "2017 OF201", "frozen", 300,
15, _extCompressor * (float) Math.log(_intCompressor * 44.3f + 1), 8833000);

PlanetAPI Goibniu = _sys.addPlanet("Goibniu", _star, "Goibniu", "frozen1", 175, 22,
_extCompressor * (float) Math.log(_intCompressor * 42.2f + 1), 100034);

PlanetAPI Ritona = _sys.addPlanet("Ritona", _star, "Ritona", "frozen2", 270, 21,
_extCompressor * (float) Math.log(_intCompressor * 41.6f + 1), 97151);
```

```
PlanetAPI Uni = _sys.addPlanet("Uni", _star, "Uni", "frozen1", 155, 20, _extCompressor
* (float) Math.log(_intCompressor * 42.7f + 1), 101157);
PlanetAPI Tinia = _sys.addPlanet("Tinia", Uni, "Tinia", "rocky_ice", 292, 8, 150, 50);
```

```
PlanetAPI JJ43 = _sys.addPlanet("2007 JJ43", _star, "2007 JJ43", "frozen", 205, 21,
_extCompressor * (float) Math.log(_intCompressor * 48.0f + 1), 120801);
```

```
PlanetAPI Chaos = _sys.addPlanet("Chaos", _star, "Chaos", "rocky_ice", 25, 18,
_extCompressor * (float) Math.log(_intCompressor * 45.7f + 1), 113199);
```

```
SectorEntityToken silaBarycenter= _sys.addCustomEntity(null, null, "sensor_ghost",
"neutral");
silaBarycenter.setCircularOrbit(_star, 330, _extCompressor * (float)
Math.log(_intCompressor * 44.0f + 1), 107026);
PlanetAPI Sila = _sys.addPlanet("Sila", silaBarycenter, "Sila", "rocky_ice", 0, 14, 40,
10);
PlanetAPI Nunam = _sys.addPlanet("Nunam", silaBarycenter, "Nunam", "rocky_ice",
180, 12, 40, 10);
```

```
PlanetAPI Rumina = _sys.addPlanet("Rumina", _star, "Rumina", "frozen3", 21, 20,
_extCompressor * (float) Math.log(_intCompressor * 35.0f + 1), 309834);
```

```
PlanetAPI Huya = _sys.addPlanet("Huya", _star, "Huya", "rocky_ice", 320, 15,
_extCompressor * (float) Math.log(_intCompressor * 28.5f + 1), 90318);
PlanetAPI Huya_Moon = _sys.addPlanet("Huya I", Huya, "Huya I", "rocky_ice", 292, 8,
80, 4);
```

```
PlanetAPI Eris = _sys.addPlanet("Eris", _star, "Eris", "frozen", 280, 42, _extCompressor
* (float) Math.log(_intCompressor * 96.0f + 1), 205410);
PlanetAPI Dysonomia = _sys.addPlanet("Dysonomia", Eris, "Dysonomia", "rocky_ice",
306, 16, 400, 15);
```

```
PlanetAPI JO179 = _sys.addPlanet("2010 JO179", _star, "2010 JO179", "frozen2", 85,
22, _extCompressor * (float) Math.log(_intCompressor * 78.0f + 1), 255466);
```

```
PlanetAPI DR15 = _sys.addPlanet("2021 DR15", _star, "2021 DR15", "frozen3", 115,
22, _extCompressor * (float) Math.log(_intCompressor * 67.0f + 1), 201115);
```

```
PlanetAPI OG19 = _sys.addPlanet("2008 OG19", _star, "2008 OG19", "frozen2", 225,
```

```
18, _extCompressor * (float) Math.log(_intCompressor * 38.0f + 1), 197137);

PlanetAPI TC302 = _sys.addPlanet("2002 TC302", _star, "2002 TC302", "frozen2", 200,
14, _extCompressor * (float) Math.log(_intCompressor * 55.3f + 1), 120801);

PlanetAPI Gkun = _sys.addPlanet("G!kunll'homdima", _star, "G!kunll'homdima",
"frozen1", 215, 20, _extCompressor * (float) Math.log(_intCompressor * 37.5f + 1),
226512);
PlanetAPI GoeHu = _sys.addPlanet("G!o'e!Hu", Gkun, "Glo'e!Hu", "rocky_ice", 292, 8,
100, 11);

PlanetAPI Dziewanna = _sys.addPlanet("Dziewanna", _star, "Dziewanna", "rocky_ice",
260, 16, _extCompressor * (float) Math.log(_intCompressor * 32.5f + 1), 216416);

PlanetAPI Gonggong = _sys.addPlanet("Gonggong", _star, "Gonggong", "frozen2",
340, 32, _extCompressor * (float) Math.log(_intCompressor * 89.0f + 1), 200410);
PlanetAPI Xiangliu = _sys.addPlanet("Xiangliu", Gonggong, "Xiangliu", "rocky_ice",
306, 8, 200, 5);

PlanetAPI TL66 = _sys.addPlanet("1996 TL66", _star, "1996 TL66", "rocky_ice", 175, 9,
_extCompressor * (float) Math.log(_intCompressor * 83.4f + 1), 20541000);

PlanetAPI Sedna = _sys.addPlanet("Sedna", _star, "Sedna", "frozen3", 185, 28,
_extCompressor * (float) Math.log(_intCompressor * 84.0f + 1), 20541000);

PlanetAPI TheGoblin = _sys.addPlanet("The Goblin", _star, "The Goblin", "rocky_ice",
60, 8, _extCompressor * (float) Math.log(_intCompressor * 80.0f + 1), 20541000);
MarketAPI marketTheGoblin = TheGoblin.getMarket();
marketTheGoblin.addCondition("dark");
marketTheGoblin.addCondition("low_gravity");
marketTheGoblin.addCondition("organics_abundant");
marketTheGoblin.addCondition("inimical_biosphere");

PlanetAPI DeeDee = _sys.addPlanet("DeeDee", _star, "DeeDee", "frozen2", 75, 22,
_extCompressor * (float) Math.log(_intCompressor * 92.0f + 1), 407913);

PlanetAPI Biden = _sys.addPlanet("Biden", _star, "2012 Vice President Biden",
"frozen1", 145, 20, _extCompressor * (float) Math.log(_intCompressor * 80.0f + 1),
20541000);
```

```
_sys.addCustomEntity(null, null, "warning_beacon", "neutral").setCircularOrbit(Biden, 0, 100, 30);
_sys.addCustomEntity(null, "Brilliance of Biden", "fusion_lamp", "neutral").setCircularOrbitPointingDown(Biden, 45, 350, 779);
MarketAPI marketBiden = Biden.getMarket();
marketBiden.addCondition("low_gravity");
marketBiden.addCondition("volatiles_diffuse");
marketBiden.addCondition("ruins_vast");
marketBiden.addCondition("rare_ore_rich");
marketBiden.addCondition("ore_sparse");
```

```
PlanetAPI Farout = _sys.addPlanet("Farout", _star, "Farout", "frozen", 195, 15,
_extCompressor * (float) Math.log(_intCompressor * 120.0f + 1), 265888);
```

```
PlanetAPI Farfarout = _sys.addPlanet("Farfarout", _star, "Farfarout", "frozen", 65, 15,
_extCompressor * (float) Math.log(_intCompressor * 132.0f + 1), 295000);
_sys.addCustomEntity(null, "Skylab 47", "station_research_remnant",
"neutral").setCircularOrbitPointingDown(Farfarout, 65, 1000, 295000);
```

```
_sys.addCustomEntity("voyager1_probe", "Voyager 1", "generic_probe",
"neutral").setCircularOrbit(_star, 178, _extCompressor * (float) Math.log(_intCompressor
* 3843f + 1), 200000);
_sys.addCustomEntity("voyager2_probe", "Voyager 2", "generic_probe",
"neutral").setCircularOrbit(_star, 2, _extCompressor * (float) Math.log(_intCompressor *
3316f + 1), 180000);
_sys.addCustomEntity("pioneer10_probe", "Pioneer 10", "generic_probe",
"neutral").setCircularOrbit(_star, 182, _extCompressor * (float) Math.log(_intCompressor
* 2674f + 1), 160000);
_sys.addCustomEntity("pioneer11_probe", "Pioneer 11", "generic_probe",
"neutral").setCircularOrbit(_star, 358, _extCompressor * (float) Math.log(_intCompressor
* 2454f + 1), 150000);
```

```
_sys.updateAllOrbits();
_sys.autogenerateHyperspaceJumpPoints(true, true);
```

```
Metis.getMarket().addCondition("irradiated");
Adrastea.getMarket().addCondition("irradiated");
Amalthea.getMarket().addCondition("irradiated");
Thebe.getMarket().addCondition("irradiated");
Io.getMarket().addCondition("irradiated");
Europa.getMarket().addCondition("irradiated");
Mimas.getMarket().addCondition("irradiated");
Enceladus.getMarket().addCondition("irradiated");
Tethys.getMarket().addCondition("irradiated");
Dione.getMarket().addCondition("irradiated");
Rhea.getMarket().addCondition("irradiated");
Miranda.getMarket().addCondition("irradiated");
Ariel.getMarket().addCondition("irradiated");
Umbriel.getMarket().addCondition("irradiated");
Titania.getMarket().addCondition("irradiated");
Oberon.getMarket().addCondition("irradiated");
Naia.getMarket().addCondition("irradiated");
Thalassa.getMarket().addCondition("irradiated");
Despina.getMarket().addCondition("irradiated");
Galatea.getMarket().addCondition("irradiated");
Larissa.getMarket().addCondition("irradiated");
Proteus.getMarket().addCondition("irradiated");

Haumea.getMarket().addCondition("thin_atmosphere");
Haumea.getMarket().removeCondition("no_atmosphere");
Eris.getMarket().addCondition("thin_atmosphere");
Eris.getMarket().removeCondition("no_atmosphere");
Orcus.getMarket().addCondition("thin_atmosphere");
Orcus.getMarket().removeCondition("no_atmosphere");
Sedna.getMarket().addCondition("thin_atmosphere");
Sedna.getMarket().removeCondition("no_atmosphere");

Hyperion.getMarket().removeCondition("tectonic_activity");
Hyperion.getMarket().removeCondition("extreme_tectonic_activity");
Sylvia.getMarket().removeCondition("tectonic_activity");
Sylvia.getMarket().removeCondition("extreme_tectonic_activity");

java.util.List<PlanetAPI> airlessBodies = new java.util.ArrayList<PlanetAPI>();

airlessBodies.add(Ceres); airlessBodies.add(Pallas); airlessBodies.add(Vesta);
airlessBodies.add(Hygiea); airlessBodies.add(Intermanmia); airlessBodies.add(Juno);
```

```
airlessBodies.add(Iris); airlessBodies.add(Flora); airlessBodies.add(Eunomia);
airlessBodies.add(Euphrosyne); airlessBodies.add(Cybele);
airlessBodies.add(Hungaria);
airlessBodies.add(Astraea); airlessBodies.add(Hebe); airlessBodies.add(Davida);
airlessBodies.add(Sylvia); airlessBodies.add(Remus); airlessBodies.add(Romulus);
airlessBodies.add(Kleopatra); airlessBodies.add(Alexhelios);
airlessBodies.add(Cleoselene);
airlessBodies.add(Lutetia); airlessBodies.add(Kalliope); airlessBodies.add(Linus);
airlessBodies.add(Antigone); airlessBodies.add(Quintilla); airlessBodies.add(Psyche);
airlessBodies.add(Patroclus); airlessBodies.add(Menoetius);
airlessBodies.add(Achilles);
airlessBodies.add(Mentor); airlessBodies.add(Hektor); airlessBodies.add(Skamandrios);
airlessBodies.add(Agamemnon); airlessBodies.add(Hilda);
airlessBodies.add(Cruithne); airlessBodies.add(Eros); airlessBodies.add(Phaethon);
airlessBodies.add(Metis); airlessBodies.add(Adrastea); airlessBodies.add(Amalthea);
airlessBodies.add(Thebe); airlessBodies.add(Ganymede); airlessBodies.add(Callisto);
airlessBodies.add(Himalia); airlessBodies.add(Elara); airlessBodies.add(Lysithea);
airlessBodies.add(Ananke); airlessBodies.add(Carme); airlessBodies.add(Pasiphae);
airlessBodies.add(Sinope);
airlessBodies.add(Pan); airlessBodies.add(Daphnis); airlessBodies.add(Atlas);
airlessBodies.add(Pandora); airlessBodies.add(Epimetheus); airlessBodies.add(Janus);
airlessBodies.add(Mimas); airlessBodies.add(Enceladus); airlessBodies.add(Tethys);
airlessBodies.add(Telesto); airlessBodies.add(Calypso); airlessBodies.add(Dione);
airlessBodies.add(Helene); airlessBodies.add(Polydeuces); airlessBodies.add(Rhea);
airlessBodies.add(Hyperion); airlessBodies.add(Iapetus); airlessBodies.add(Phoebe);
airlessBodies.add(Ymir); airlessBodies.add(Siarnaq); airlessBodies.add(Albiorix);
airlessBodies.add(UO14);
airlessBodies.add(Portia); airlessBodies.add(Juliet); airlessBodies.add(Belinda);
airlessBodies.add(Cressida); airlessBodies.add(Rosalind);
airlessBodies.add(Desdemona);
airlessBodies.add(Bianca); airlessBodies.add(Puck); airlessBodies.add(Miranda);
airlessBodies.add(Ariel); airlessBodies.add(Umbriel); airlessBodies.add(Titania);
airlessBodies.add(Oberon); airlessBodies.add(Sycorax); airlessBodies.add(Caliban);
airlessBodies.add(Prospero); airlessBodies.add(Setebos); airlessBodies.add(Stephano);
airlessBodies.add(QF99); airlessBodies.add(YX49);
airlessBodies.add(Naiad); airlessBodies.add(Thalassa); airlessBodies.add(Despina);
airlessBodies.add(Galatea); airlessBodies.add(Larissa); airlessBodies.add(Hippocamp);
airlessBodies.add(Proteus); airlessBodies.add(Nereid); airlessBodies.add(Halimede);
airlessBodies.add(Sao); airlessBodies.add(Laomedea); airlessBodies.add(Psamathe);
airlessBodies.add(Neso);
airlessBodies.add(WG157); airlessBodies.add(Otrera); airlessBodies.add(Clete);
airlessBodies.add(QR322); airlessBodies.add(RC158); airlessBodies.add(TT191);
```

```
airlessBodies.add(KY18); airlessBodies.add(LC18); airlessBodies.add(KV18);
airlessBodies.add(HM102); airlessBodies.add(EN65);

airlessBodies.add(Charon); airlessBodies.add(Styx); airlessBodies.add(Nix);
airlessBodies.add(Kerberos); airlessBodies.add(Hydra); airlessBodies.add(Arrokoth);
airlessBodies.add(Chariklo); airlessBodies.add(Chiron);
airlessBodies.add(Hiiaka); airlessBodies.add(Namaka);
airlessBodies.add(Quaoar); airlessBodies.add(Weywot);
airlessBodies.add(Vanth);
airlessBodies.add(Makemake); airlessBodies.add(Mk2);
airlessBodies.add(Ixion);
airlessBodies.add(Salacia); airlessBodies.add(Actea);
airlessBodies.add(Varda); airlessBodies.add(Ilmare);
airlessBodies.add(Albion); airlessBodies.add(TX300); airlessBodies.add(Chiminigagua);
airlessBodies.add(Varuna);
airlessBodies.add(Lempo); airlessBodies.add(Hiisi); airlessBodies.add(Paha);
airlessBodies.add(Aya); airlessBodies.add(Mani);
airlessBodies.add(Achlys); airlessBodies.add(AchlysMoon);
airlessBodies.add[OF201]; airlessBodies.add(Goibniu); airlessBodies.add(Ritona);
airlessBodies.add(Uni); airlessBodies.add(Tinia);
airlessBodies.add(JJ43); airlessBodies.add(Chaos);
airlessBodies.add(Sila); airlessBodies.add(Nunam);
airlessBodies.add(Rumina);
airlessBodies.add(Huya); airlessBodies.add(Huya_Moon);
airlessBodies.add(Dysonomia);
airlessBodies.add(JO179); airlessBodies.add(DR15); airlessBodies.add(OG19);
airlessBodies.add(TC302);
airlessBodies.add(Gkun); airlessBodies.add(GoeHu);
airlessBodies.add(Dziewanna);
airlessBodies.add(Gonggong);
airlessBodies.add(Xiangliu);
airlessBodies.add(TL66);
airlessBodies.add(DeeDee);
airlessBodies.add(Farout);
airlessBodies.add(Farfarout);

for (PlanetAPI planet : airlessBodies) {
    if (planet != null) {
        MarketAPI m = planet.getMarket();
        if (m != null) {
            m.addCondition("no_atmosphere");
            m.removeCondition("thin_atmosphere");
        }
    }
}
```

```

        m.removeCondition("dense_atmosphere");
        m.removeCondition("toxic_atmosphere");
        m.removeCondition("extreme_weather");
    }
}
}

java.util.List<PlanetAPI> coldBodies = new java.util.ArrayList<PlanetAPI>();

coldBodies.add(Ceres); coldBodies.add(Pallas); coldBodies.add(Vesta);
coldBodies.add(Hygaea); coldBodies.add(Intermania); coldBodies.add(Juno);
coldBodies.add(Iris); coldBodies.add(Flora); coldBodies.add(Eunomia);
coldBodies.add(Euphrosyne); coldBodies.add(Cybele); coldBodies.add(Hungaria);
coldBodies.add(Astrea); coldBodies.add(Hebe); coldBodies.add(Davida);
coldBodies.add(Sylvia); coldBodies.add(Remus); coldBodies.add(Romulus);
coldBodies.add(Kleopatra); coldBodies.add(Alexhelios); coldBodies.add(Cleoselene);
coldBodies.add(Lutetia); coldBodies.add(Kalliope); coldBodies.add(Linus);
coldBodies.add(Antigone); coldBodies.add(Quintilla); coldBodies.add(Psyche);
coldBodies.add(Patroclus); coldBodies.add(Menoetius); coldBodies.add(Achilles);
coldBodies.add(Mentor); coldBodies.add(Hektor); coldBodies.add(Skamandrios);
coldBodies.add(Agamemnon); coldBodies.add(Hilda);
coldBodies.add(Metis); coldBodies.add(Adrastea); coldBodies.add(Amalthea);
coldBodies.add(Thebe); coldBodies.add(Io); coldBodies.add(Europa);
coldBodies.add(Ganymede); coldBodies.add(Callisto);
coldBodies.add(Himalia); coldBodies.add(Elara); coldBodies.add(Lysithea);
coldBodies.add(Ananke); coldBodies.add(Carme); coldBodies.add(Pasiphae);
coldBodies.add(Sinope); coldBodies.add(AntiopeA); coldBodies.add(AntiopeB);

for (PlanetAPI planet : coldBodies) {
    if (planet != null) {
        MarketAPI m = planet.getMarket();
        if (m != null) {
            m.removeCondition("very_cold");
            m.addCondition("cold");
            m.removeCondition("poor_light");
        }
    }
}

java.util.List<PlanetAPI> saturnMoonsLite = new java.util.ArrayList<PlanetAPI>();

```

```
saturnMoonsLite.add(Pan); saturnMoonsLite.add(Daphnis);
saturnMoonsLite.add(Atlas);
saturnMoonsLite.add(Pandora); saturnMoonsLite.add(Epimetheus);
saturnMoonsLite.add(Janus);
saturnMoonsLite.add(Mimas); saturnMoonsLite.add(Enceladus);
saturnMoonsLite.add(Tethys);
saturnMoonsLite.add(Telesto); saturnMoonsLite.add(Calypso);
saturnMoonsLite.add(Dione);
saturnMoonsLite.add(Helene); saturnMoonsLite.add(Polydeuces);
saturnMoonsLite.add(Rhea);
saturnMoonsLite.add(Titan); saturnMoonsLite.add(Hyperion);
saturnMoonsLite.add(Iapetus);
saturnMoonsLite.add(Phoebe);
saturnMoonsLite.add(Ymir); saturnMoonsLite.add(Siarnaq);
saturnMoonsLite.add(Albiorix);
saturnMoonsLite.add(UO14); saturnMoonsLite.add(Chiron);
saturnMoonsLite.add(Chariklo);

for (PlanetAPI moon : saturnMoonsLite) {
    if (moon != null) {
        MarketAPI m = moon.getMarket();
        if (m != null) {
            m.removeCondition("dark");
            if (!m.hasCondition("poor_light")) {
                m.addCondition("poor_light");
            }
        }
    }
}
```