

ITerrain. Интерфейс модуля запросов к земле.

[Abstract](#)

[Использование](#)

[Система координат](#)

[class ITerrain](#)

[getRoadNetwork](#)

[Преобразование координат](#)

[class IRoadNetwork](#)

Abstract

ITerrain реализуется с использованием библиотек: osg, edCore.

ITerrain не оперирует графическими объектами. Для рендера используется интерфейс ITerrainGraphics.

В настоящее время ITerrain частично реализован для:

- Terrain31 (старая земля: Кавказ)
- Terrain41 (новая земля: Невада, Корсика, Азербайджан, Форт Ракер, Персидский залив, ...)

Использование

Территория моделируется в UTM проекции. Система координат описана ниже.

Для преобразования в географические координаты (широта, долгота) и MGRS координаты используются соответствующие методы.

ITerrain обеспечивает следующую функциональность:

- Запрос высоты, типа поверхности и нормали в заданной точке
- Преобразование координат
- Поиск точки пересечения поверхности с отрезком
- Поиск объектов сцены в заданном объеме или по отрезку
- Поиск путей в дорожных сетях
- Поиск путей по РД и свойств стоянок
- Поиск путей по поверхности с заданными ограничениями

ITerrain инициализируется по пути к конфигу территории.

Каждая территория представляет собой каталог с основным скриптом в корне [terrain.cfg.lua]. Все ссылки на ресурсы представлены путями относительно каталога территории.

Система координат

Система координат мира (далее world).

Ноль выбирается для каждой территории свой исходя из удобства производства.

Направление осей:

- ось X - на север;
- ось Y - в зенит;
- ось Z - на восток

class ITerrain

ITerrain инициализируется по пути к конфигу территории с помощью обертки

ITerrainEntryPoint. Обертка отвечает и за удаление ITerrain.

getRoadNetwork

IRoadNetwork* getRoadNetwork(const char* name)

Вернет требуемый IRoadNetwork

name = "roads"	Автомобильные дороги
name = "railroads"	Железные дороги
name = airfield name	Рулежки аэродрома

Запросы к дорожной сети: читай [class IRoadNetwork](#).

Преобразование координат

toLatLon(double x, double y, double& lat, double& lon);

Из [world](#) в широту и долготу

fromLatLon(double lat, double lon, double& x, double& y)const;

Из широты и долготы в [world](#).

ed::string fromLockonToMGRS(double lockon_z, double lockon_x, int prec=5)const;

Из [world](#) в MGRS

osg::Vec3d fromUTMtoLockon(const osg::Vec2d& utm)const;

Из UTM в [world](#).

class IRoadNetwork

Дорожная сеть представляет из себя граф: перекрестки связанные дорогами.

Слова перекресток и дорога используются исключительно для удобства.

IRoadNetwork позволяет находить пути и запрашивать информацию о перекрестках и дорогах.

Дорожные сети разных типов существуют независимо друг от друга: автомобильные дороги, железные дороги, рулежки аэродромов.

TODO

```
virtual int roadsCount() { return 0; }
virtual IRoute* getRoad(int i) { return nullptr; }

virtual int crossroadsCount() { return 0; }
virtual ICrossroad* getCrossroad(int i) { return nullptr; }

// Точка в пространстве по точке на дороге
virtual bool getPoint(const PointOnRoad& pointOnRoad, osg::Vec3d& point) {return
false;}

// Ближайшая точка
virtual bool findClosestPoint(const osg::Vec3d& point, PointOnRoad& closestPoint,
    ICheckRoadSink* sink = NULL, double startDistance = 10000.0){return false;}

// Найти путь
virtual IRoute* findPath(const PointOnRoad& startPoint, const PointOnRoad& endPoint,
    double velocity, double error = 0.01, ICheckRoadSink* sink = nullptr) {return
nullptr;}
```