

Learning OMPL+RRT*

1. ompl::base::State

ob::State是OMPL定义的许多State Space的基类, 以下为创建一种State Space, 并为其实例进行内存分配的两种方法:

- ```
ompl::base::StateSpacePtr space(new T()); //实例化一个Space指针
ompl::base::ScopedState<> state(space); //创建这个Space下的State指针
```
- ```
ompl::base::StateSpacePtr space(new T());
ompl::base::SpaceInformationPtr si(space);
ompl::base::ScopedState<T> state(si);
```

一个Space指针对应一个State指针, 接下来可以用State指针继续操作。

2. ompl::base::RealVectorStateSpace

ompl的实向量空间, 初始化时可以指定空间的维数:

```
ob::StateSpacePtr space(new ob::RealVectorStateSpace(3));
```

再设定空间每个维度的范围:

```
ob::RealVectorBounds bounds(3);
bounds.setLow(0, -_x_size * 0.5);
bounds.setLow(1, -_y_size * 0.5);
bounds.setLow(2, 0.0);

bounds.setHigh(0,+_x_size * 0.5);
bounds.setHigh(1,+_y_size * 0.5);
bounds.setHigh(2, _z_size);

space->as<ob::RealVectorStateSpace>()->setBounds(bounds);
```

该Space对应的State为一个double数组values:

```
ob::ScopedState<> start(space);
start->values = {start_pt[0], start_pt[1], start_pt[2]};
```

3. ompl::base::ProblemDefinition

ProblemDefinition定义了要求解的问题, 包括初始State和目标State, 如果求解成功后还会包含解的结果。创建方法如下:

```
ProblemDefinition & ProblemDefinition (SpaceInformationPtr si);
pdef->setStartAndGoalStates(start, goal); //定义初始和终止状态
pdef->setOptimizationObjective(getPathLengthObjective(si)); //定义优化目标对象 (度量)
```

4. ompl::base::Planner

Planner是求解器类，用求解方法比如RRT*来初始化，再设置要求解的问题、求解时间等：

```
ob::ProblemDefinitionPtr *pdef(new ob::ProblemDefinition(si));
pdef->setStartAndGoalStates(start, goal);
pdef->setOptimizationObjective(getPathLengthObjective(si));
ob::PlannerPtr optimizingPlanner(new og::RRTstar(si));
optimizingPlanner->setProblemDefinition(pdef);
optimizingPlanner->setup();
ob::PlannerStatus solved = optimizingPlanner->solve(1.0);
```

