## Pathplanning

- + AnglesForLinearPath(Vector2d& angles, Vector2d path):Vector2d&
- + AnglesForCircularPath(Vector2d& angles):Eigen:Vector2d
- + AnglesForContinuousPath(Vector2d& angles):Eigen::Vector2d

## Robotsimulator

- + robot:Robot
- + planPlath:Pathplanning
- + drawOuterCircle(int centerx, int centery, int r):int
- + drawInnerCircle(int centerx, int centery, int r):int
- + drawJoint(int centerx, int centery):int
- + drawLinkBtnJoints(float cx1, float cy1, float cx2, float cy2):int
- + drawEndEffector(float centerx, float centery):int
- + displayArm(void):void
- + reshape(int w, int h):void
- + display(void):void
- + drawPathCircle(void):int
- + drawTarget(float x, float y):int
- + runSimulation(int argc, char \* argv[]):void
- + AnglesForLinearPath(Vector2d& angles, Vector2d path): virtual Vector2d&
- + AnglesForCircularPath(Vector2d& angles): virtual Eigen:Vector2d+ AnglesForContinuousPath(Vector2d& angles): virtual Eigen::Vector2d