

```
template<typename S1, typename S2> bool GJKSolver_indep::shapeDistance  
(const S1& s1, const Transform3f& tf1, const S2& s2, const Transform3f& tf2,  
    FCL_REAL* distance, Vec3f* p1, Vec3f* p2) const  
    narrowphase/narrowphase.h
```

```
void ShapeDistanceTraversalNode::leafTesting(int, int) const  
    traversal/traversal_node_shapes.h
```

```
void distanceRecurse(DistanceTraversalNodeBase* node,  
    int b1, int b2, BVHFrontList* front_list)
```

```
void distance(DistanceTraversalNodeBase* node,  
    BVHFrontList* front_list, int qsize)
```

```
template<typename T_SH1, typename T_SH2, typename NarrowPhaseSolver>  
FCL_REAL ShapeShapeDistance(const CollisionGeometry* o1, const Transform3f& tf1,  
    const CollisionGeometry* o2, const Transform3f& tf2,  
    const NarrowPhaseSolver* nsolver, const DistanceRequest& request,  
    DistanceResult& result)
```

```
template<typename T_SH1, typename T_SH2, typename NarrowPhaseSolver>  
std::size_t ShapeShapeCollide(const CollisionGeometry* o1, const Transform3f& tf1,  
    const CollisionGeometry* o2, const Transform3f& tf2,  
    const NarrowPhaseSolver* nsolver, const CollisionRequest& request,  
    CollisionResult& result)
```