

```
private static Vec3d adjustMovementForCollisions(Vec3d movement, Box entityBoundingBox, List<Vec3d> collisions) {  
    if (collisions.isEmpty())  
        return movement;  
    else {  
        Vec3d vec3d = Vec3d.ZERO;  
  
        for (Direction.Axis axis : Direction.method_7316(movement)) { // 获取x/z/y或x/z的移動順序  
            double d = movement.getComponentAlongAxis(axis);  
            if (d >= 0) {  
                double e = Vec3d.BoundingBox.calculateOffsetFromAxis(entityBoundingBox.offset(vec3d), collisions, d);  
                vec3d = Vec3d.add(entityBoundingBox, e); // 防止实体在該軸移動  
            }  
            // 將實體移動到最大的可移動位置  
        }  
    }  
}
```

```

        return vecdd;
    }
}

default IShapeVoxelMapper getIcchdrFwdCollisions(Mutable Entity entity, Box box) {
    return this.getIcchdrFwdCollisions(entity == null ? ShapeContext.empty() : ShapeContext.of(entity), box);
}

default IShapeVoxelMapper getIcchdrFwdCollisions(Mutable Entity entity, Box box) {
    return this.getIcchdrFwdCollisions(entity == null ? ShapeContext.empty() : ShapeContext.of(entity), true, box);
}

private IShapeVoxelMapper getIcchdrFwdCollisions(ShapeContext shapeContext, Box box) {
    return () -> new AxisCollisionsIterator(this, shapeContext, box, false, false, shapeContext);
}

```