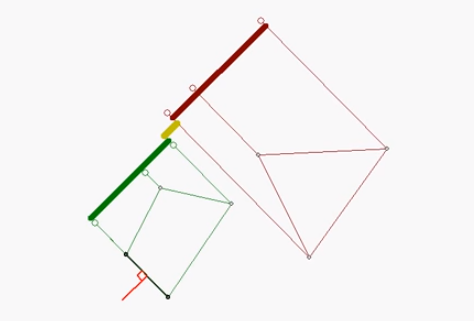
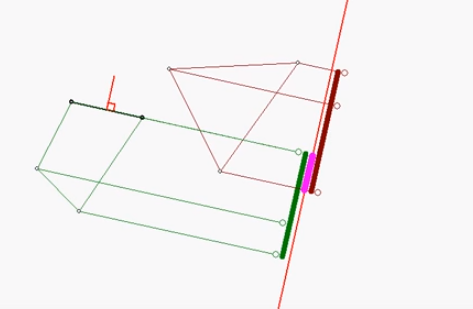
**Separating Axis Theorem(SAT)**

**Theorem** **stated**:

* “**If two convex objects are not penetrating, there exists an axis for which the projection of the objects will not overlap.**”

(NOTE: This is an implementation for 2D only. 3D using the same concept but with more steps)

This theorem is useful in computer programing, particularly in shape collision detection or physic simulation… What this theorem stated, gave us a way - an algorithm to determine if two **convex** polygons are intersecting. The concept is by checking two projections of the two polygons along the perpendicular vector of the edge and see if they overlap. Doing this, we are accentually checking for overlap along an axis, and if there is no overlap in one of the axis, we can safely assume the two polygons are not intersect.



**Implementation**:

1. Find all the edge of the polygons.
2. Find the edge’s normal, then find the unit vector of the normal edge.
3. Project each vertex of the polygon to the normal vector to find the scalar projection.
4. Check for the min and max projection scalars of each vertex along the normal edge for 2 polygons.
5. Check for overlap: if overlapped, then repeat step 2; if not overlap then the two polygons have an axis that is not overlap, therefor no intersection.

**Explanation**:

How to find normal edge vector:

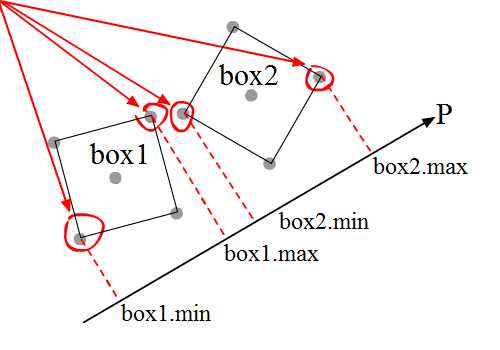
* First we need to know the vector of the edge.
* To find vector along 2 points (a,b), we use V = <bx – ax, by - ay>
* Then to find the perpendicular vector, we can reorder X and Y and multiply by negative for y.
* <x, y> => <-y, x>
* After we got the normal edge vector, we need to turn it into a unit vector.
* By apply the formula U = <x / ||v||, y / ||v||>
* The reason we need the unit vector of the vector is because we will need to find the projection of each vertex onto that vector.

How to find the projection scalar of a vertex onto a vector:

* Formula: Comp­­­ab =
* In this case, we will consider vertices of the shape to be vector a^.
* Vector a will be <ax – 0, ay - 0>
* Since we already have the unit vector of b (normal edge vector), we can just take a dot b and find the scalar projection by that way.

Checking for min and max for overlap:

* The reason we are checking for min and max scalar projections is we want to later on compare the overlap projections of the two polygons



* We can see that if box1.max >= min1 and box1.min >= box2.max, then we know that the two polygons are overlap.

You can see the demonstration on the attached file: SeparatingAxisTheoremDemo.jar, or <https://www.youtube.com/watch?v=WBy6AveIRRs>

**Reflection of codes with the theorem:**

The codes I implemented work 90% of the times. In some cases, when checking for collision using this algorithm, it the 2 polygons sometimes behave like they are intersecting, but in reality they are not (\*see the SeparatingAxisTheoremDemo.jar bugged test case). To fix this, all I have to do is to flip the name of the codes,

Ex:

**if (poly3.isOverlap(poly1)) {// this work}**

**if (poly1.isOverlap(poly3)) {// this won’t work}**

This behavior is weird because, with property of the dot product, it doesn’t matter if a.b or b.a, the value should be the same.

What I’m suspecting is that when finding the projection scalar, some values became negative which make the code not work properly.

\*\*\*I will tackle on the solution once I have time or if anyone want to look at the codes and figure out the problem, I also included the whole project just in case. The java framework I’m using is [**Libgdx**](https://libgdx.badlogicgames.com/)if anyone interested\*\*\*