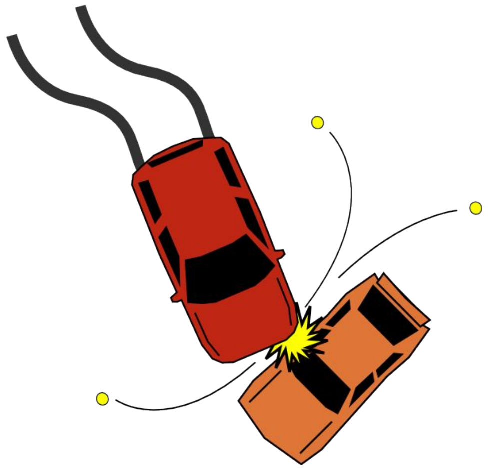
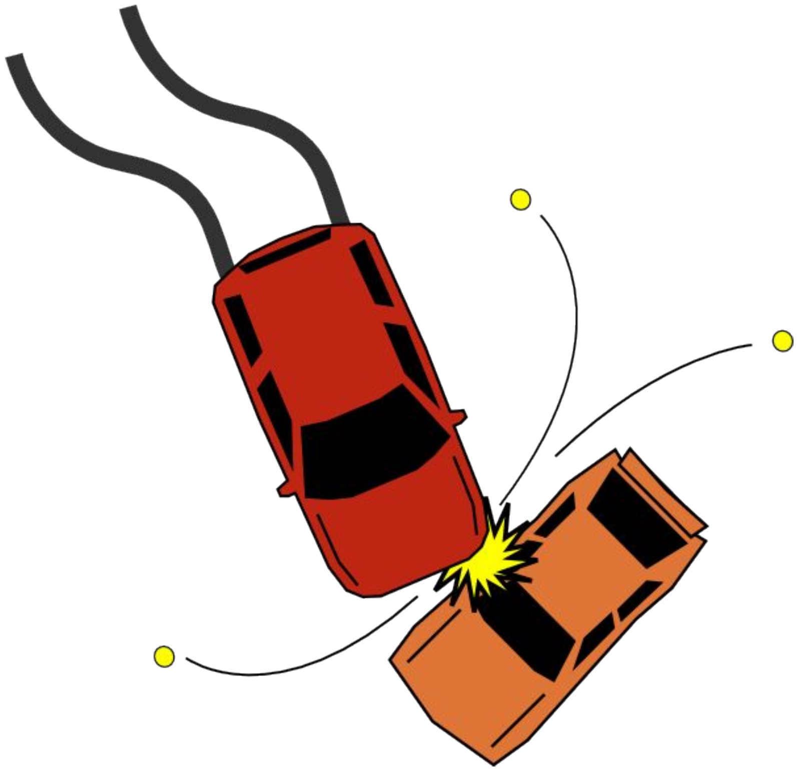


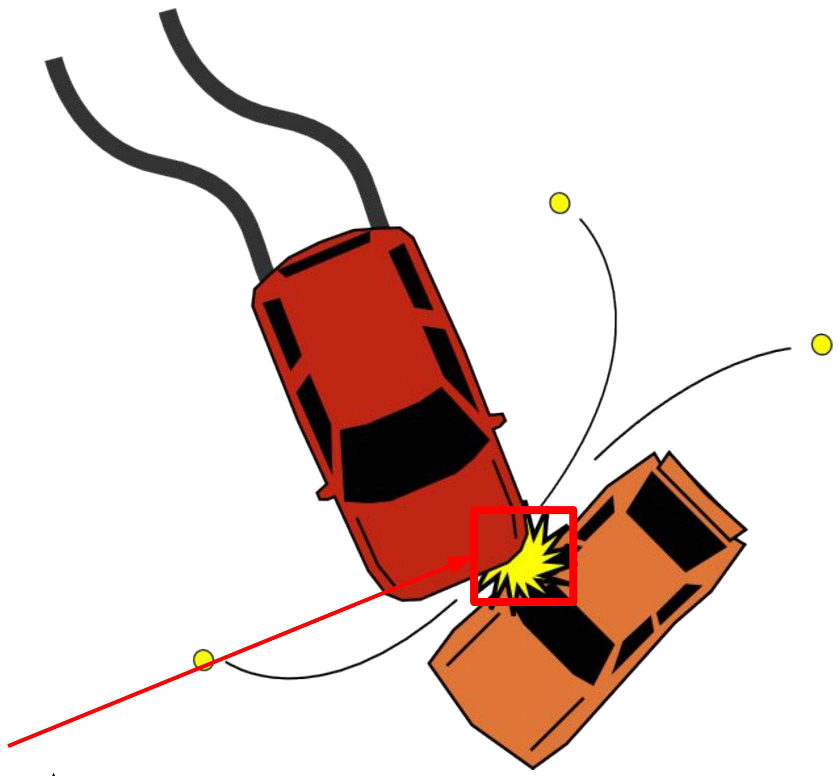
Collusion Detection

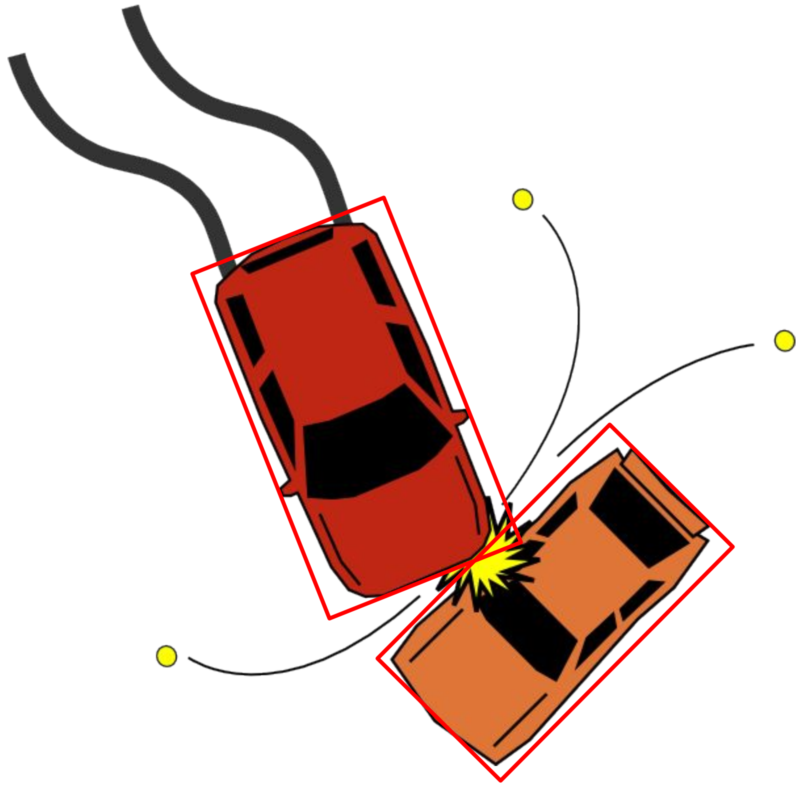
event handling

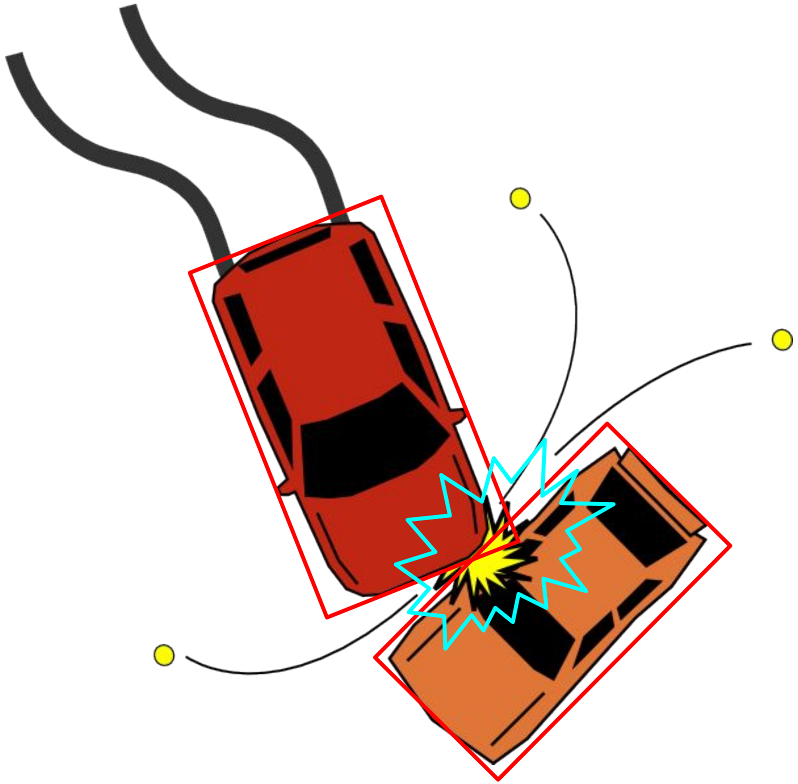
Chouri Soulaymen
https://praisethemoon.org

Example



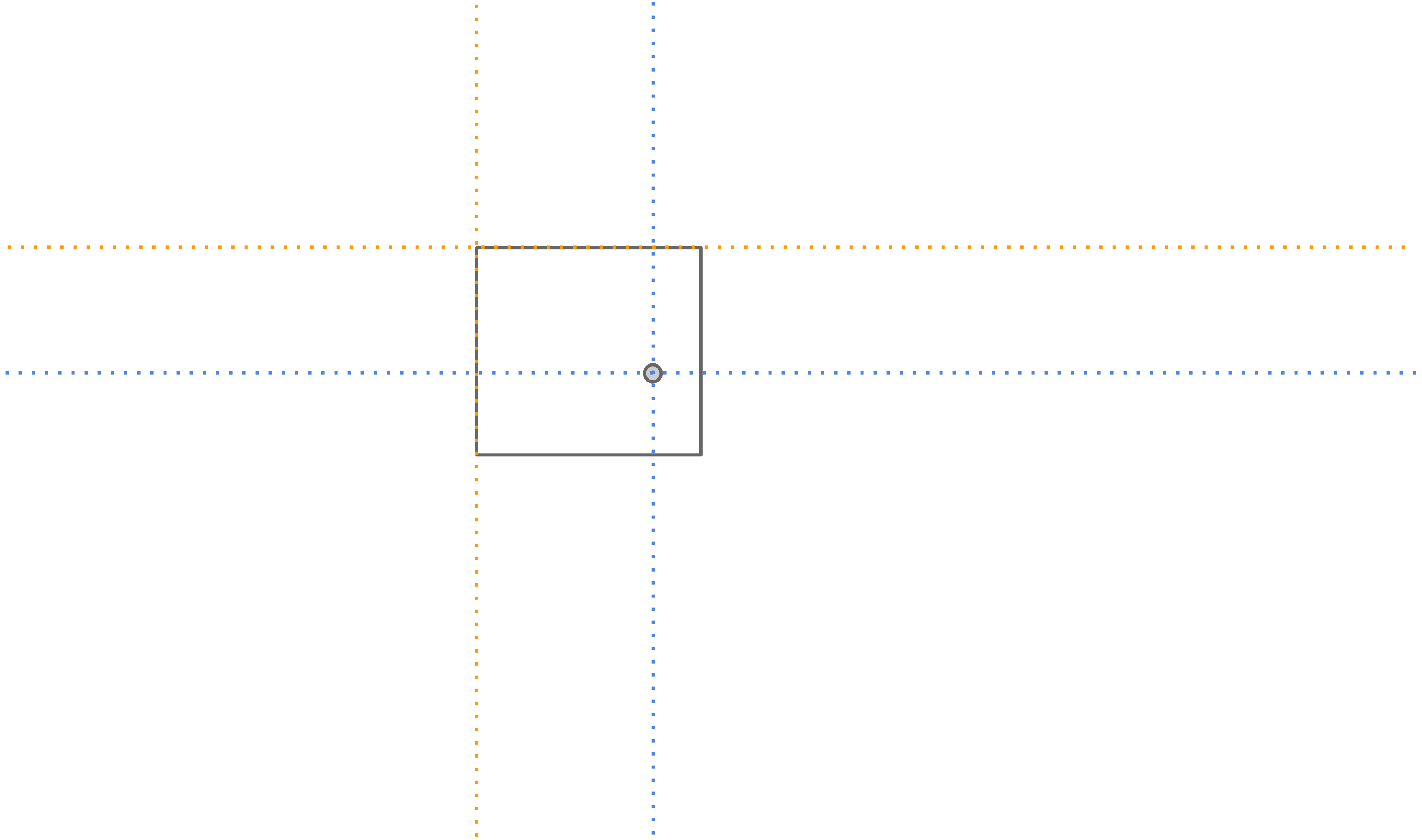

Collision Area





Different type of collusion

1. Existence of point in a box



Function:

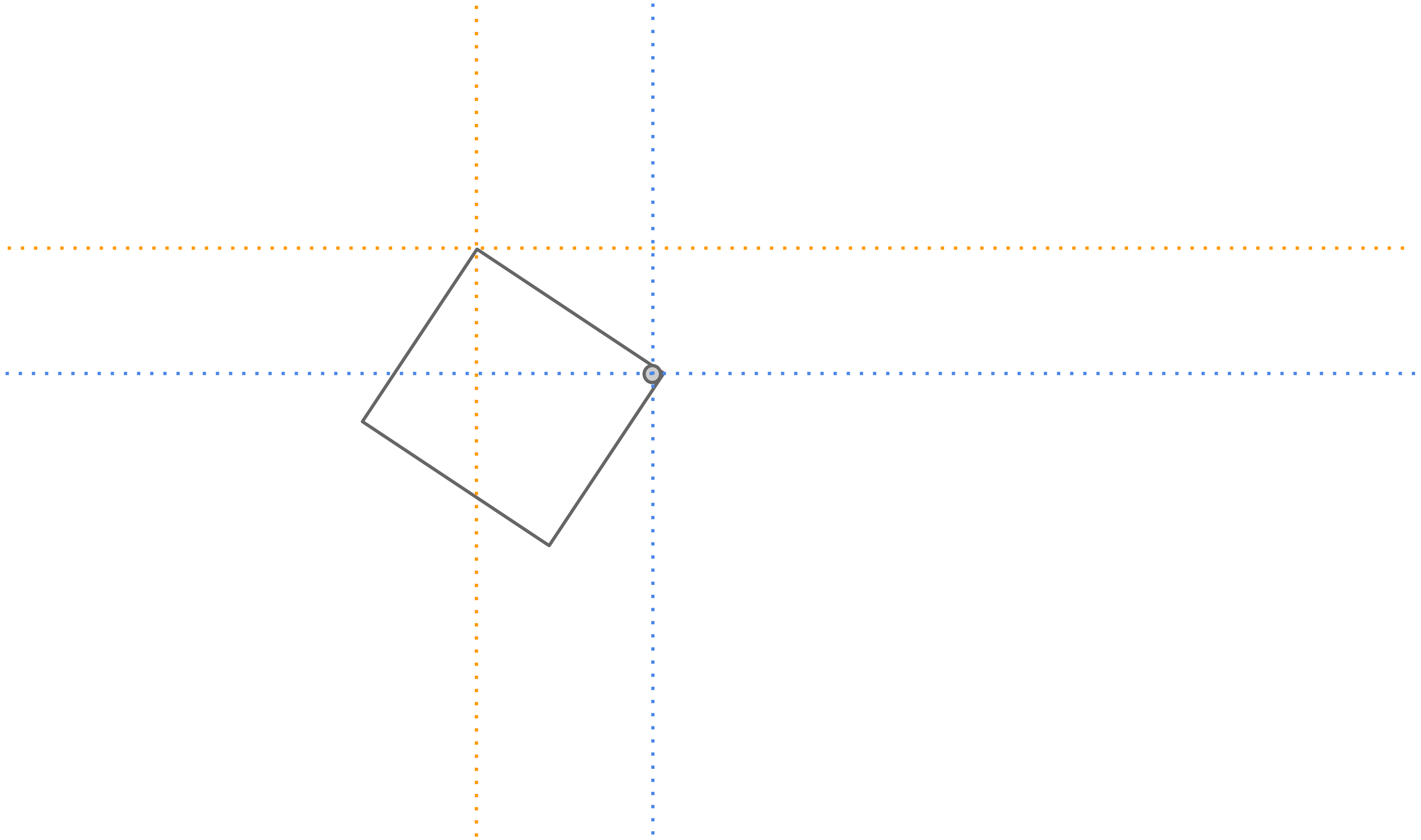
```lua
-- Checks the existence of a point in
-- x1, y1, w1, h1 respectively x, y coordinates of the box, and it's width and height
-- x2, y2 are point's coordinates
function pointInBox(x1,y1,w1,h1, x2,y2)
  return x2 > x1 and
         x2 < x1+w1 and
         y2 > y1 and
         y2 < y1+h1
end
```

Problem 1

Existence of a point inside a box

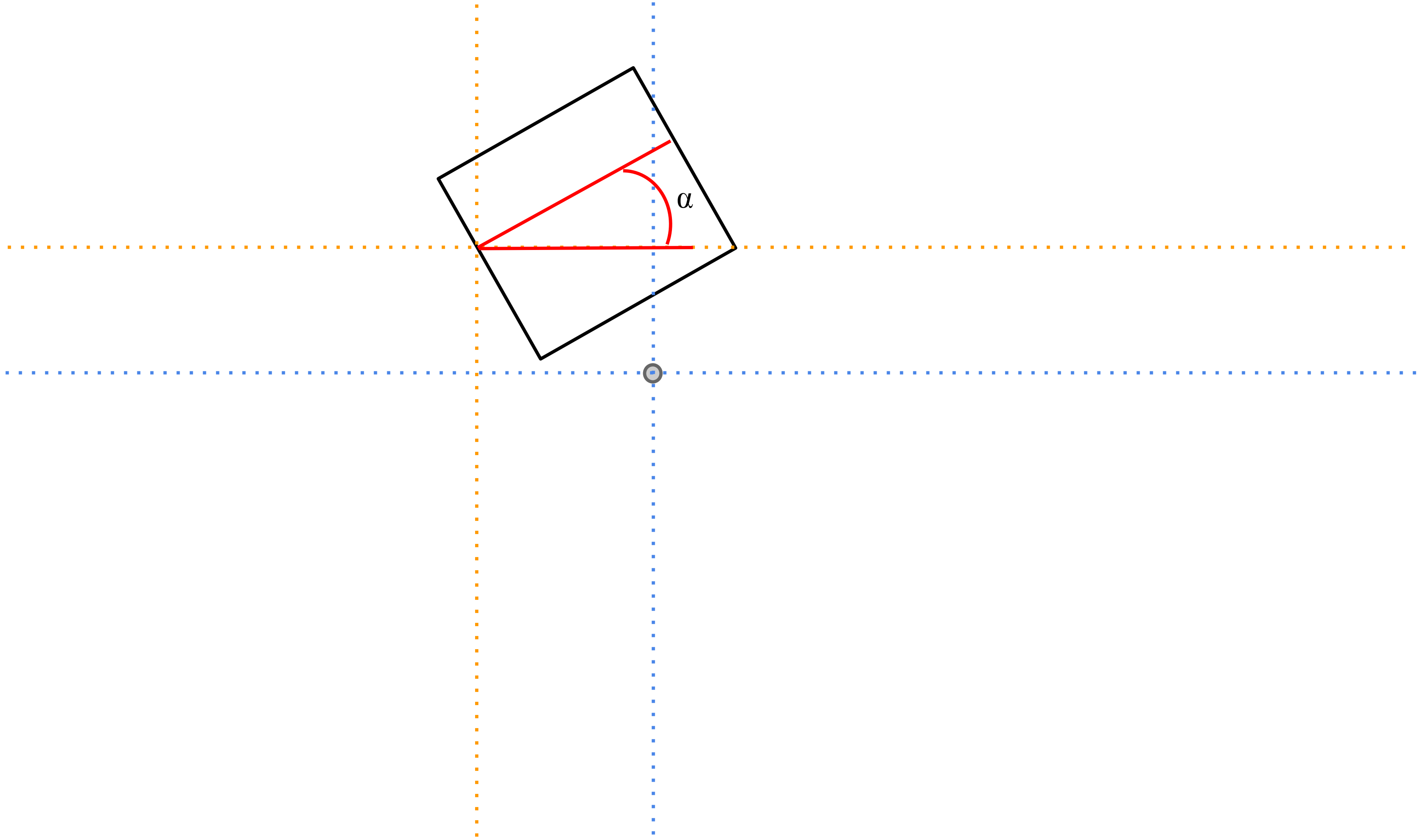
NOTE:

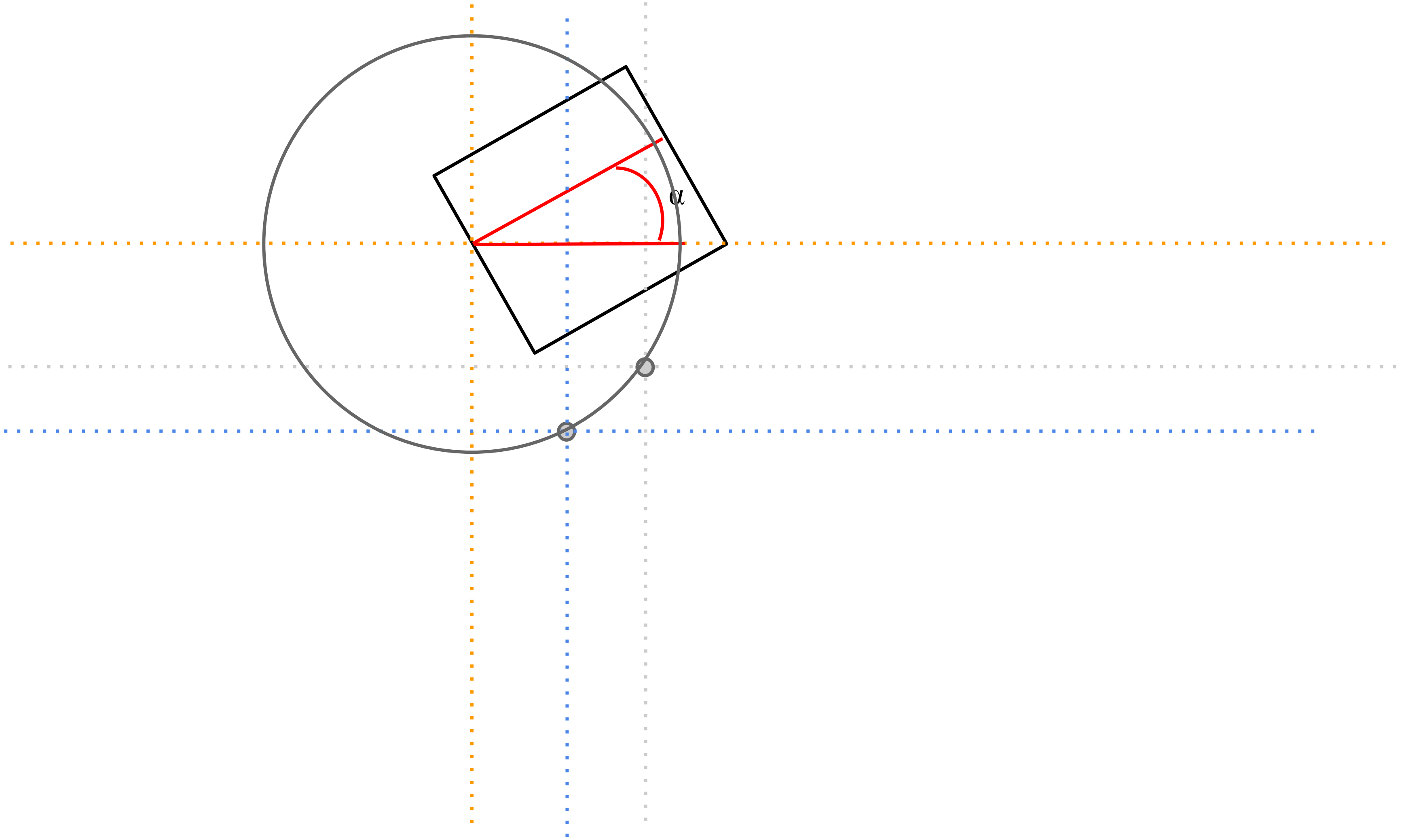
Rotation origin matters

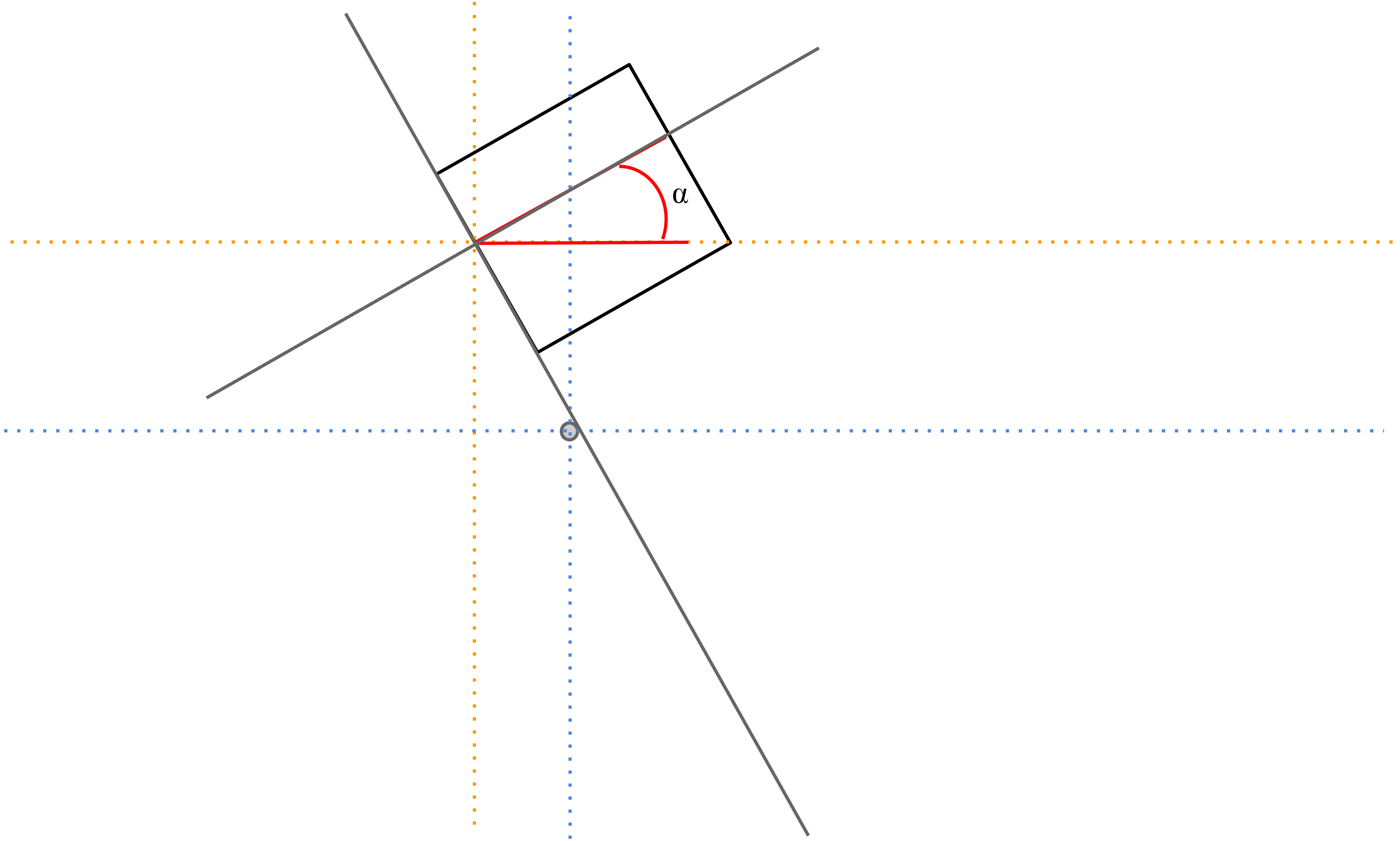


Solution

Rotate the point by the opposite angle value of the box


α

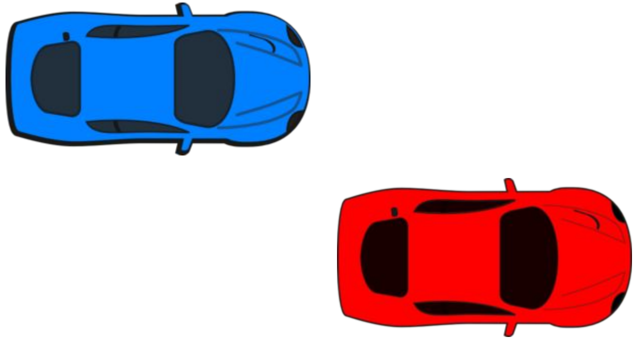

α

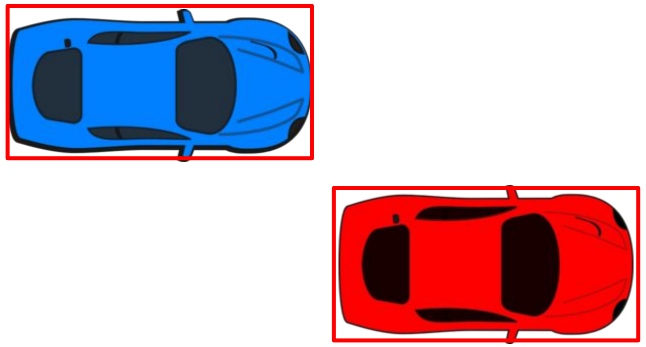

α

Solving the problem

Method1: Bounding Box

Transform objects into circles or polygons

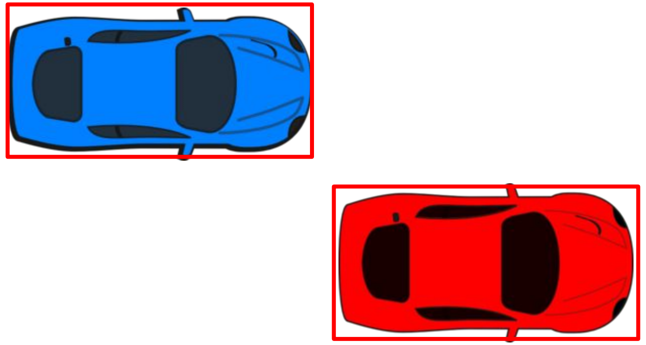




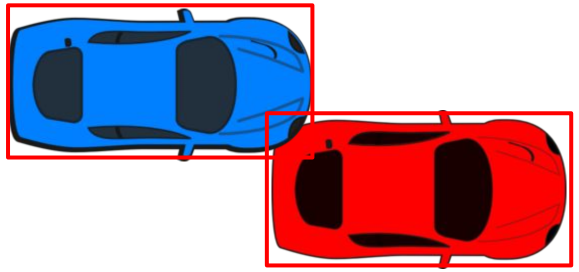
Rectangle collision is the easiest to calculate

Function: (http://love2d.org/wiki/BoundingBox.lua)

```
-- Collision detection function.
-- Returns true if two boxes overlap, false if they don't
-- x1,y1 are the left-top coords of the first box, while w1,h1 are its width and height
-- x2,y2,w2 & h2 are the same, but for the second box
function CheckCollision(x1,y1,w1,h1, x2,y2,w2,h2)
  return x1 < x2+w2 and
         x2 < x1+w1 and
         y1 < y2+h2 and
         y2 < y1+h1
end
```

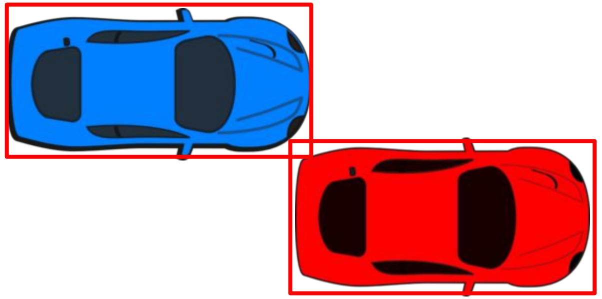


No collusion

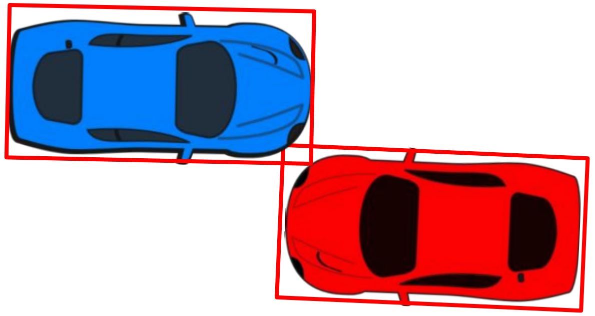


Collusion

Drawbacks



1. Not precise



2. Not generic