Comp285 Coursework

Purpose

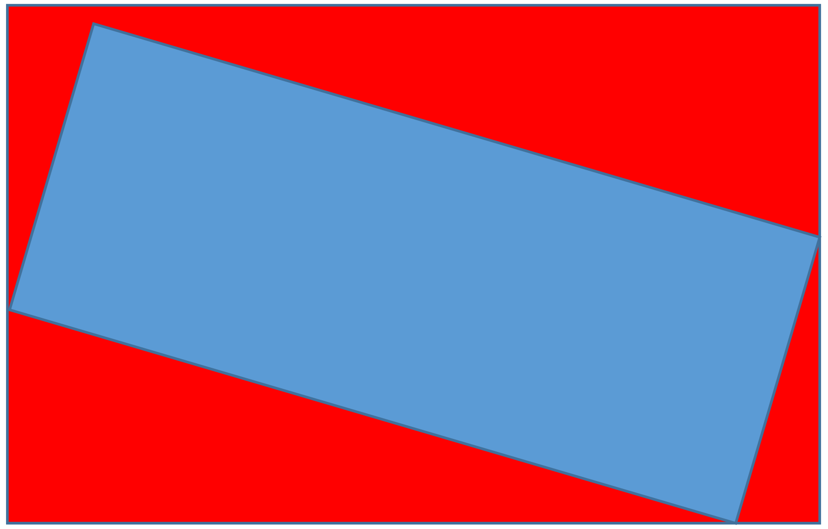
- To practice and demonstrate skills in
  - Test driven development
  - Numeric testing
  - Debugging
  - Development
  - JUnit testing
- Anything broken should be fixed

Classes/Interfaces

- IShape
  - float getArea();
  - Classes implement to calculate their area, you need to implement
  - getLowerLeftPoint();
    - Get minx and miny point but after transformation (for example rotation)
    - So this will be the minx of all the points and the miny of all the points after rotation
  - getUpperRightPoint():
    - Get maxx and maxy after rotation
  - doesCollide(IShape)
    - Determines if 2 shapes overlap and therefore collide
    - The simplest implementation involves checking if 2 shapes bounding rectangles overlap, it is possible using this architecture to implement more realistic collision detection

Bounding

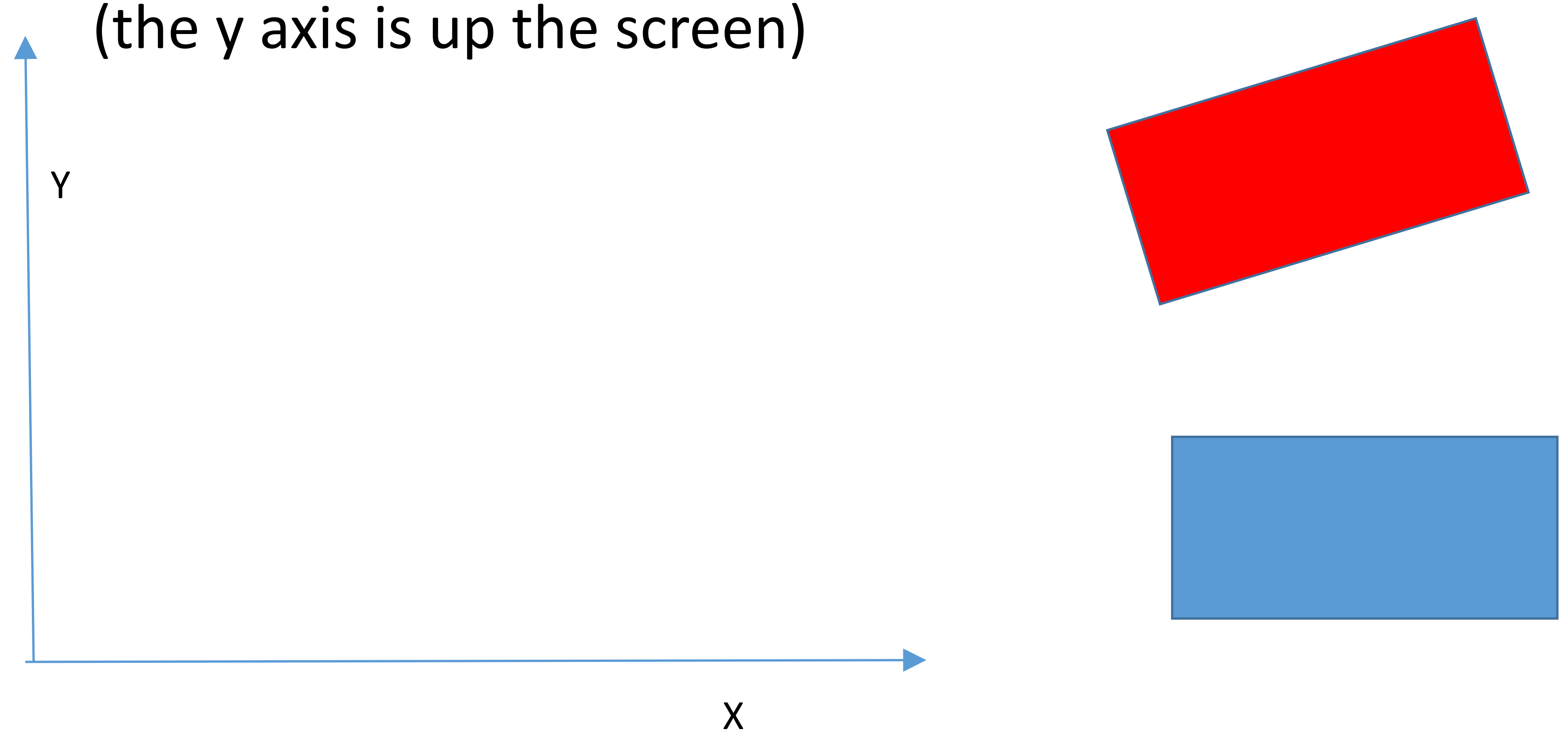
- The bounding rectangle is the smallest rectangle that contains the shape



- See the blue rectangle rotated, red rectangle is the bounding rectangle, we can see the lower left of the red

Rotation

- public void setRotation(Point p, double angle);
- For small rotations round the origin the rotation will be anti-clockwise in this picture you can see the blue shape being rotated round the origin (the y axis is up the screen)



Shape class

- Abstract class
- Implements a basic version of collision detection which could be overridden to implement a more complex and accurate algorithm

Testing rotation

- Notice rotation involves 2 trig functions sin and cos
- This in generate 4 equivalence partitions because of
- ASTC 4 quadrants in which sin, cos can act differently
- Also note rotations of 45 degrees are weak tests because
- cos(45)=sin(45)   so swapping functions would not change value
- So try and keep away from angles like 45, 45+90, 45+180 etc.

Summary

- Implement tests for all classes
  - Remember to put in tests first
- Complete development and debugging for all classes
  - Remember for a class to work and be complete, all dependent classes have to be debugged and complete (so all super classes and composite classes)
- Canvas clas
- Zip up final implementation