# CS 04.114

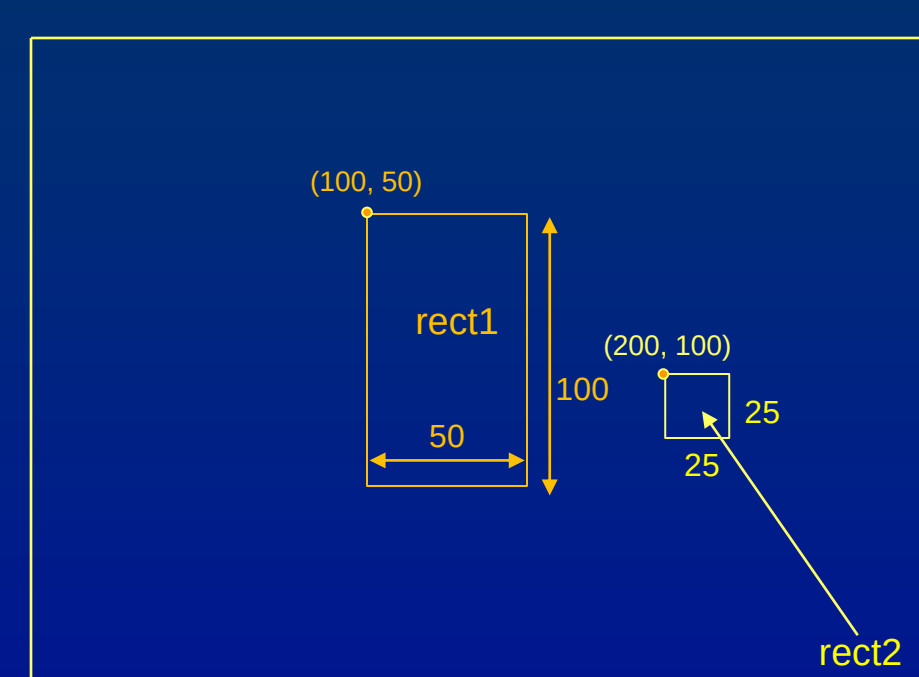
## Object Oriented Programming and Data Abstraction (OOPDA)

# Lecture 14

# Agenda

Simplified general Rectangle collision detection

# Collision of two rectangles



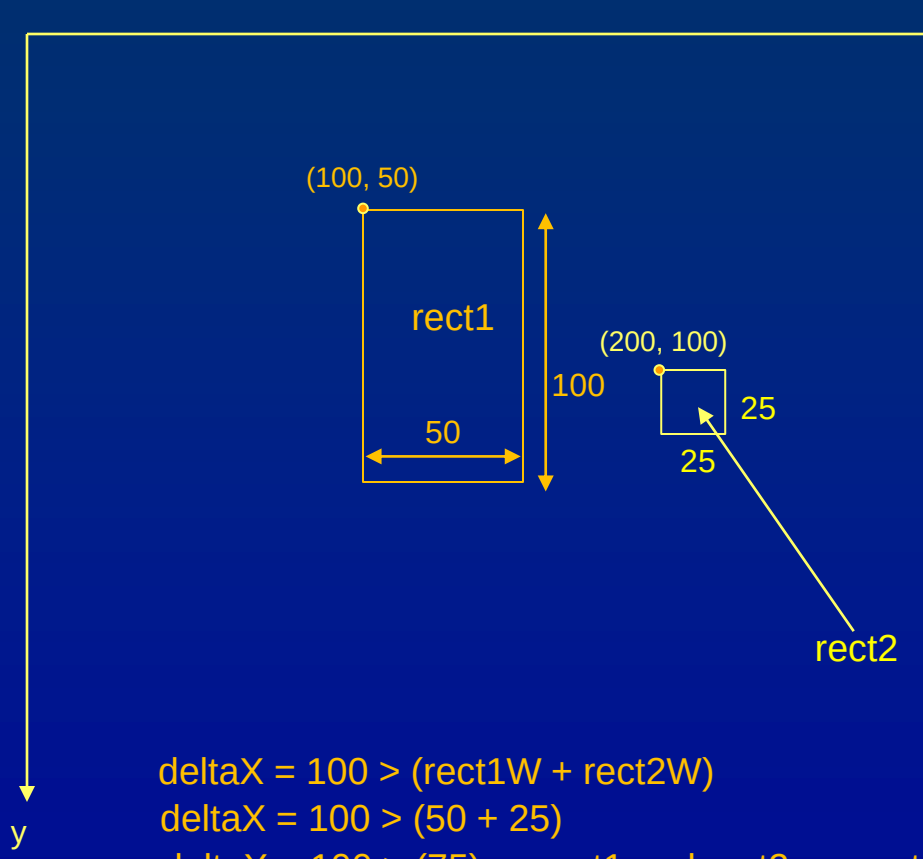

rect1X = 100
rect1Y = 50
rect1W = 50
rect1H= 100

rect2X = 200
rect2Y = 100
rect2W = 25
rect2H = 25

deltaX = abs(rect1X – rect2X) = abs(100-200) =100

deltaY = abs(rect1Y – rect2Y) = abs(50 – 100) = 50

# Collision of two rectangles

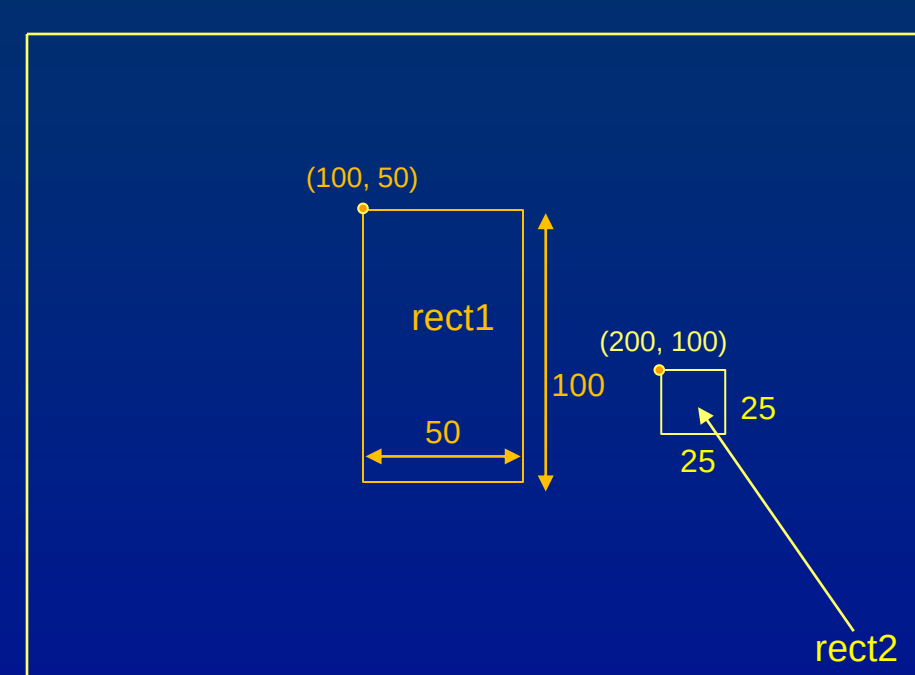



rect1X = 100
rect1Y = 50
rect1W = 50
rect1H= 100

rect2X = 200
rect2Y = 100
rect2W = 25
rect2H = 25

deltaX = 100 > (rect1W + rect2W)
deltaX = 100 > (50 + 25)
deltaX = 100 > (75) so rect1 and rect2 can not possibly collide

# Collision of two rectangles




rect1X = 100
rect1Y = 50
rect1W = 50
rect1H= 100

rect2X = 200
rect2Y = 100
rect2W = 25
rect2H = 25

deltaX = 100 > (rect1W + rect2W)
deltaX = 100 > (50 + 25)
deltaX = 100 > (75) so rect1 and rect2 can not possibly collide

# Collision of two rectangles

- Using these facts, we can test two intervals for intersection in x and test two intervals for intersection in y
- If one of the interval intersections fails, then the rectangles can not possibly intersect
- Test two x intervals for intersection

```
I1 = [rect1X, rect1X + rec1W]
I2 = [rect2X, rect2X + rec2W]
```

- Test two x intervals for intersection

```
I3 = [rect1Y, rect1Y + rec1W]
I4 = [rect2Y, rect2Y + rec2W]
```

# Collision of two rectangles

- Using these facts, we can test two intervals for intersection in x and test two intervals for intersection in y
- If one of the interval intersections fails, then the rectangles can not possibly intersect
- Test two x intervals for intersection

```
I1 = [rect1X, rect1X + rec1W]
I2 = [rect2X, rect2X + rec2W]
intervalIntersect(rect1X, rect1X + rec1W, rect2X, rect2X + rec2W)
```

- Test two x intervals for intersection

```
I3 = [rect1Y, rect1Y + rec1W]
I4 = [rect2Y, rect2Y + rec2W]
```

- `intervalIntersect(rect1Y, rect1Y + rec1H, rect2Y, rect2Y + rec2H)`

# Interval intersection algorithm

```
// If both of these are true the rectangles intersect
intervalIntersect(rect1X, rect1X + rec1W, rect2X, rect2X + rec2W)
intervalIntersect(rect1Y, rect1Y + rec1H, rect2Y, rect2Y + rec2H)


boolean intervalIntersect(int a, int b, int c, int d)
{
   boolean intersect = true;
   if (a > d ) || (c > b)
   {
       intersect = false;
   }
   return intersect;
}
```

# Project 5

- Build an object-oriented 2D-graphics system
  - Build it using skills you have learned in OOPDA
  - Built using object composition, object inheritance, and polymorphism
  - Build a little test a little (rapid prototyping)
- Goals
  - Create a nice picture
  - Add time in the form of  a game using the event loop
  - Chance for creativity in both OOP, Graphics, and gaming