Scripting Languages

Python pygame Library

- using rectangles
- handling keys
- moving objects
- detecting collisions
- displaying texts

Rectangle as Object

- Create rectangle
  - from coordinates+size

```
r = pygame.Rect( position+size )
```

  - from existing surface

```
r = s.get_rect()
```

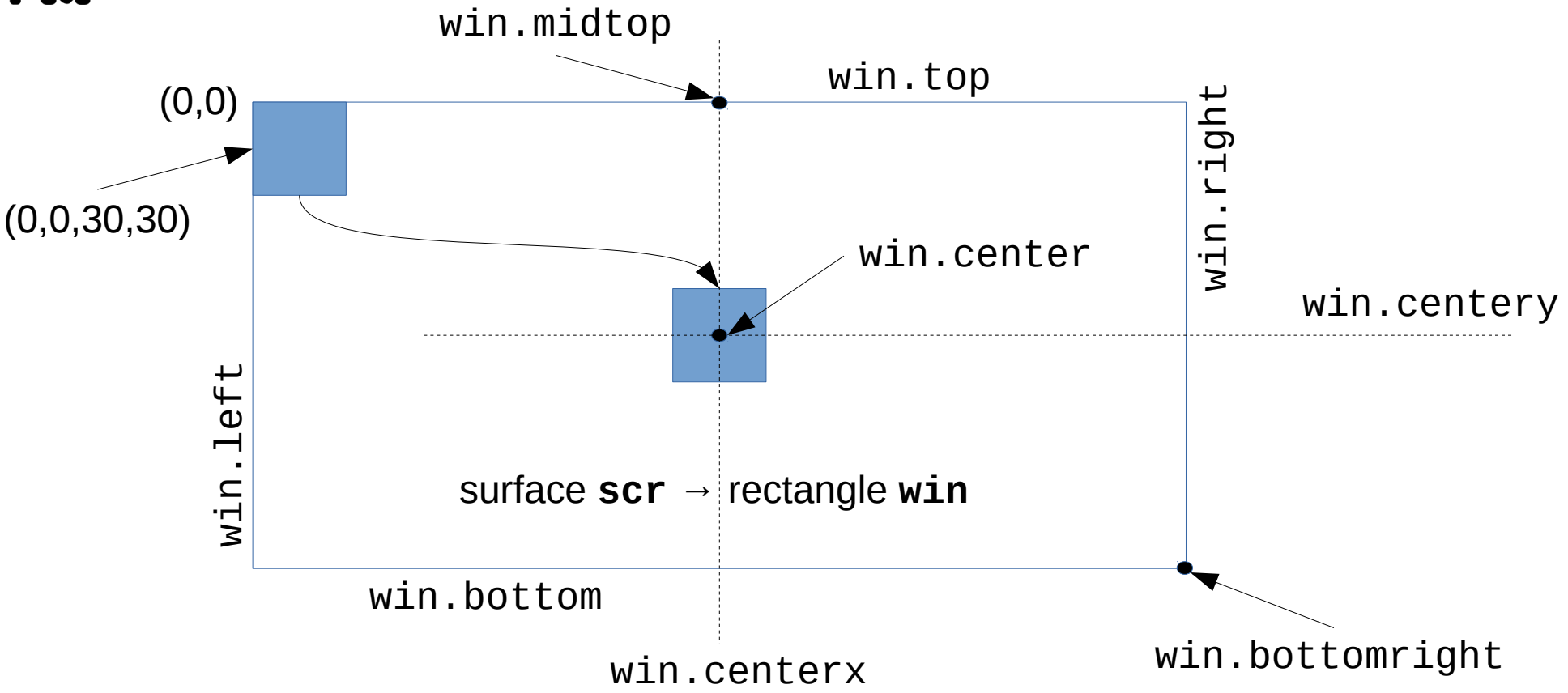
- Modify attributes
  - http://www.pygame.org/docs/ref/rect.html

```
r.x = 10
r.center = (100,100)
```

- Apply actions
  - move, copy, inflate, ...

```
r = r.move(10,10)
r = r.inflate(2,-2)
```

Rectangle positioning




```
scr = pygame.display.set_mode((360, 240))
win = scr.get_rect()

box = pygame.Rect( 0, 0, 30, 30)
box.center = win.center
```

Handling keys

- Event type → KEYDOWN, KEYUP
- Attribute:
  - `key` → http://www.pygame.org/docs/ref/key.html
  - `mod` → modifier (shift, alt, ctrl)
- Repetition
  - `pygame.key.set_repeat()` → in miliseconds

```
pygame.key.set_repeat(1000,250)
```

Handling keys and movement

```
import pygame, sys
black = (0, 0, 0); white = (255, 255, 255)
pygame.init()

scr = pygame.display.set_mode((360, 240))
win = scr.get_rect()

box = pygame.Rect( 0, 0, 30, 30)
box.center = win.center

step = 5

while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            sys.exit()
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT:
                box = box.move(-step,0)

    scr.fill(black)
    pygame.draw.rect(scr,white,box)
    pygame.display.flip()
```

ex-1.py

Add the code for box movement in all-directions using cursor keys

Detecting limits & key repetition

```
...

step = 5

pygame.key.set_repeat(50,50)

while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            sys.exit()
        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_LEFT:
                box = box.move(-step,0)

    if box.left < 0:
        box.left = 0

    scr.fill(black)
    pygame.draw.rect(scr,white,box)
    pygame.display.flip()
```

Add the code for movement limits in all-directions using attributes: `win.left`, `win.right`, `win.top`, `win.bottom`

Timekeeping

- Loop timing with delay (simple, not accurate):

```
pygame.time.wait(ms)
pygame.time.delay(ms)
```

- Loop timing with frames per second
  - create Clock object outside the main loop

```
fps = pygame.time.Clock()
```

  - use this object with tick() ones per frame

```
fps.tick(100)
```

    - by calling Clock.tick(**n**) once per frame,
      the program will never run at more than **n**-frames per second.

Animation effect

```
import pygame, sys
black = (0, 0, 0); white = (255, 255, 255)
pygame.init()

scr = pygame.display.set_mode((360, 240))
win = scr.get_rect()
box = pygame.Rect( 0, 0, 30, 30)
box.center = win.center

vec = [2,0]
fps = pygame.time.Clock()

while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            sys.exit()

    box = box.move(vec)
    if box.left < win.left or box.right > win.right:
        vec[0] = -vec[0]

    scr.fill(black)
    pygame.draw.rect(scr,white,box)
    pygame.display.flip()

    fps.tick(60)
```

ex-2.py

Add the code for both horizontal and vertical box animation

Detecting collisions

- Direct comparison of Rectangle attributes
  - applicable for small number of moving objects

    ```
    x,y, top, left, bottom, right
    topleft, bottomleft, topright, bottomright
    midtop, midleft, midbottom, midright
    center, centerx, centery, width, height, w,h
    center, size
    ```

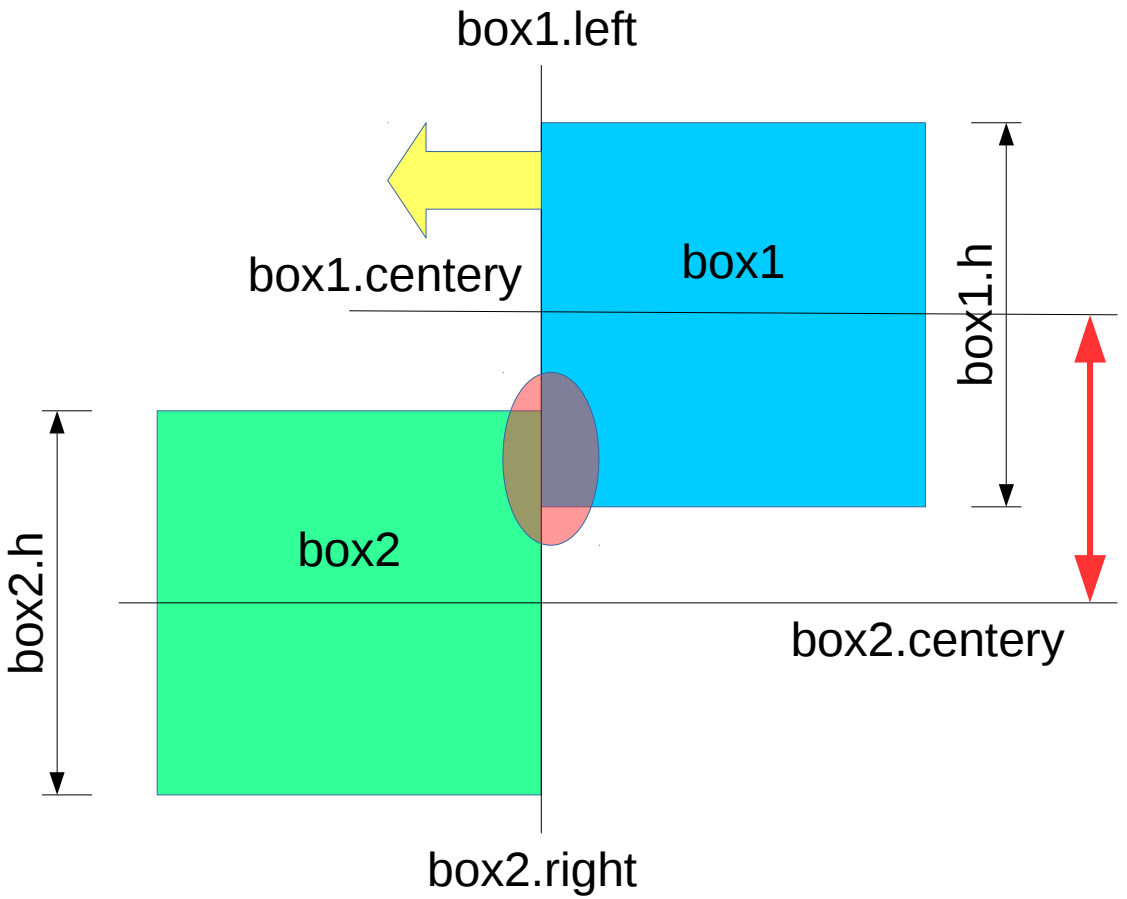
- Using `collide()` methods for Rectangles
  - collisions detection between many objects
    - handling of detected collision involves comparison of Rectangle attributes – as above

      ```
      collidepoint, colliderect, collidelist,
      collidelistall, collidedict, collidedictall
      contains
      ```

example

Comparison of attributes

```
if box1.left < box2.right and \
   abs(box1.centery-box2.centery) < (box1.h+box2.h)/2:
```

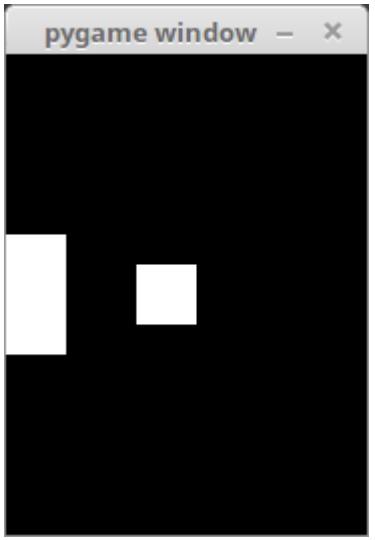


Collision of 2 objects

```
...
box1 = pygame.Rect( 0, 0, 30, 30)
box1.center = win.center
box2 = pygame.Rect( 0, 0, 30, 60)
box2.midleft = win.midleft

vec = [1,0]
pygame.key.set_repeat(50,50)
fps = pygame.time.Clock()

while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            sys.exit()
    ...

    box1 = box1.move(vec)
    if box1.left < win.left or box1.right > win.right:
        vec[0] = -vec[0]
    if box1.left < box2.right and \
        abs(box1.centery-box2.centery) < (box1.h+box2.h)/2:
        vec[0] = -vec[0]

    scr.fill(black)
    pygame.draw.rect(scr,white,box1)
    pygame.draw.rect(scr,white,box2)
    pygame.display.flip()
    fps.tick(260)
```

ex-3.py

Add the code for cursor keys and limits



Rectangle `collide()` methods

```
...
box1 = box1.move(vec)

if box1.left < win.left or box1.right > win.right:
    vec[0] = -vec[0]

if box1.left < box2.right and \
   abs(box1.centery-box2.centery) < (box1.h+box2.h)/2:
    vec[0] = -vec[0]
...
```

```
...
box1 = box1.move(vec)

if not win.contains(box1):
    vec[0] = -vec[0]

if box1.colliderect(box2):
    vec[0] = -vec[0]
...
```

After collision detection
it is necessary
to compare the attributes
to find out the region
of collision

Displaying text

- Set up active font object

```
myfont = pygame.font.Font('fontname.ttf',size)
```

- Render text to a new surface

```
text = myfont.render("text", True, color)
```

- Obtain text rectangle from text surface

```
text_box = text.get_rect()
```

  - position your text_box setting attributes

- Draw text onto main screen surface (respecting transparency)

```
screen.blit(text, text_box)
```

Displaying text

```
import pygame, sys

black = (0, 0, 0); red = (255, 0, 0)

pygame.init()
scr = pygame.display.set_mode((360, 240))
win = scr.get_rect()

myfont = pygame.font.Font('freesansbold.ttf', 48)
msg = myfont.render("The Game !!!", True, red)

msg_box = msg.get_rect()
msg_box.center = win.center

while True:
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            sys.exit()

    scr.fill(black)
    scr.blit(msg, msg_box)

    pygame.display.flip()
```

ex-4.py