Real example

- *"Talk is cheap. Show me the code."*
  Linux Torvalds, Fri, 25 Aug 2000 11:09:12 -0700 (PDT)

Space invaders in Python

```
#!/usr/bin/env python3

import pygame
import sys
from pygame.locals import KEYDOWN, KEYUP, K_SPACE, K_ESCAPE, \
                          K_RIGHT, K_LEFT

width         = 320
height        = 240
imageWidth    = 32
imageHeight   = 32

goingLeft     = True
invaderHeight = 0
gunLeft       = False
gunRight      = False
gunXpos       = (width/2)-(imageWidth/2)
delay         = 10
```

Space invaders in Python

```
class BoxSprite(pygame.sprite.Sprite):
    image = None

    def __init__(self, initial_position):
        pygame.sprite.Sprite.__init__(self)
        if BoxSprite.image is None:
            BoxSprite.image = pygame.image.load("ball.png")
        self.image = BoxSprite.image

        self.rect = self.image.get_rect()
        self.rect.topleft = initial_position
        self.next_update_time = 0  # as soon as possible
        self.yPos = initial_position[1]
```

Space invaders in Python

```
    def update(self, current_time, left, right):
        global goingLeft, invaderHeight, imageWidth, delay
        # check update
        if self.next_update_time < current_time:
            # If we're at the left or right the screen, switch directions.
            if self.rect.topleft[0] == left:
                goingLeft = False
                invaderHeight += 1
            elif self.rect.topleft[0] == right-imageWidth:
                goingLeft = True
                invaderHeight += 1
            if goingLeft == True:
                self.rect.topleft = [self.rect.topleft[0]-1,
                                     self.rect.topleft[1]]
            else:
                self.rect.topleft = [self.rect.topleft[0]+1,
                                     self.rect.topleft[1]]
            self.rect.topleft = [self.rect.topleft[0],
                                 invaderHeight+self.yPos]
            self.next_update_time = current_time + delay
```

Space invaders in Python

```
class missile(pygame.sprite.Sprite):
    image = None

    def __init__(self, initial_position):
        pygame.sprite.Sprite.__init__(self)
        if missile.image is None:
            missile.image = pygame.image.load("arrow.png")
        self.image = missile.image

        self.rect = self.image.get_rect()
        self.rect.topleft = initial_position
        self.next_update_time = 0 # update() hasn't been called yet.
```

Space invaders in Python

```
    def update(self, current_time):
        global missile
        # check update
        if self.next_update_time < current_time:
            # If we're reached the top then stop
            if self.rect.topleft[1] == 0:
                missiles.remove(self)
                self.kill()
                return
            else:
                self.rect.topleft = [self.rect.topleft[0],
                                     self.rect.topleft[1]-1]
            self.next_update_time = current_time + 4
```

Space invaders in Python

```
class gun(pygame.sprite.Sprite):
    image   = None

    def __init__(self):
        global width, imageHeight, gunXpos
        pygame.sprite.Sprite.__init__(self)
        if gun.image is None:
            gun.image = pygame.image.load("gun.png")
        self.image = gun.image

        self.rect = self.image.get_rect()
        self.rect.topleft = [gunXpos, height-imageHeight]
        self.next_update_time = 0 # update() hasn't been called yet.
```

Space invaders in Python

```
    def update(self, current_time):
        global gunXpos, width, imageWidth

        # check update
        if self.next_update_time < current_time:
            if gunLeft and gunXpos>0:
                gunXpos -= 1
            if gunRight and gunXpos<width-imageWidth:
                gunXpos += 1
            self.rect.topleft = [gunXpos, self.rect.topleft[1]]
            self.next_update_time = current_time + 1
```

Space invaders in Python

```
def checkInput():
    global gunLeft, gunRight, missiles, gunXpos, height
    for event in pygame.event.get():
       if event.type == KEYDOWN:
          if event.key == K_ESCAPE:
              sys.exit(0)
          elif event.key == K_RIGHT:
              gunLeft = False
              gunRight = True
          elif event.key == K_LEFT:
              gunLeft = True
              gunRight = False
          else:
              missiles.append(missile([gunXpos, height]))
       elif event.type == KEYUP and event.key != K_SPACE:
          gunRight = False
          gunLeft = False
```

Space invaders in Python

```
def checkCollisions():
    global missiles, invaders
    if missiles != [] and invaders != []:
        for m in missiles:
            found = False
            for b in pygame.sprite.spritecollide(m, invaders, False):
                invaders.remove(b)
                b.kill()
                found = True
            if found:
                missiles.remove(m)
                m.kill()
```

Main section of space invaders - initialisation

```
pygame.init()
invaders = []
missiles = []

for x in range(0, width, 32):
    for y in range(0, 96, 32):
        invaders.append(BoxSprite([x, y]))

screen = pygame.display.set_mode([320, 240])
gunControl = gun()
```

Main section of space invaders - initialisation

```
while invaders != []:
    screen.fill([0, 0, 0]) # blank the screen.
    time = pygame.time.get_ticks()
    for b in invaders:
        b.update(time, 0, width)
        screen.blit(b.image, b.rect)

    checkInput()
    checkCollisions()
```

Main section of space invaders - initialisation

```
    gunControl.update(time)
    screen.blit(gunControl.image, gunControl.rect)
    for m in missiles:
        m.update(time)
        screen.blit(m.image, m.rect)
    pygame.display.update()
    if pygame.sprite.spritecollide(gunControl, invaders, 0) != []:
        pygame.time.delay(50)
        print "loser"
        sys.exit(0)
    if len(invaders)<10:
        delay = len(invaders)

pygame.time.delay(50)
print "winner"
```