Python i Pygame

Programiranje arkadnih igrica u Pythonu koristeći Pygame

Kontroleri i grafika

Kreiranje objekta

- Najprije ćemo kreirati "čovječuljka" - objekt kojeg ćemo pomicati po ekranu
  Kod ćemo napisati u obliku funkcije koja ima tri ulazna parametra: ekran, x i y koordinate čovječuljka

```
def nacrtaj_covjeculjka(screen, x, y)
    # Glava
    pygame.draw.ellipse(screen, BLACK, [x-4,y-17,10,10], 0)
    # Noge
    pygame.draw.line(screen, BLACK, [x,y], [x+5,y+10], 2)
    pygame.draw.line(screen, BLACK, [x,y], [x-5,y+10], 2)
    # Tijelo
    pygame.draw.line(screen, RED, [x,y], [x,y-10], 2)
    # Ruke
    pygame.draw.line(screen, RED, [x,y-10], [x+4,y], 2)
    pygame.draw.line(screen, RED, [x,y-10], [x-4,y], 2)
```

Određivanje pozicije mišem

- Pozicija miša će odrediti koordinate na kojima ćemo iscrtati objekt

```
# Procitaj koordinate misa
    pos = pygame.mouse.get_pos()
    kx = pos[0]
    ky = pos[1]
```

- Nakon određivanja koordinata pozivamo funkciju za crtanje čovječuljka

```
nacrtaj_covjeculjka(screen, kx, ky)
```

- Prije glavne petlje programa ćemo isključiti prikazivanje strelice miša

```
# Sakrij kursor misa
pygame.mouse.set_visible(False))
```

Određivanje pozicije tipkovnicom

- Uvodimo varijable za promjenu x i y smjera

```
x_pomak = 0
y_pomak = 0
```

- For petlja za hvatanje akcije igrača

```
for event in pygame.event.get():         # Hvatanje akcije igrača
       if event.type == pygame.QUIT: # Igrač je pritisnuo close window
           done = True # postavljamo varijablu za kraj programa
       # Igrač je pritisnuo tipku
       elif event.type == pygame.KEYDOWN:
       # Ako je pritisnuta strelica podesi promjenu koordinata
          if event.key == pygame.K_LEFT:
              x_pomak = -3
          elif event.key == pygame.K_RIGHT:
              x_pomak = 3
          elif event.key == pygame.K_UP:
              y_pomak = -3
          elif event.key == pygame.K_DOWN:
              y_pomak = 3
       # Igrač je otpustio tipku
       elif event.type == pygame.KEYUP:
           # Ako je otpuštena strelica ponisti promjenu koordinata
           if event.key == pygame.K_LEFT or event.key == pygame.K_RIGHT:
               x_pomak = 0
           elif event.key == pygame.K_UP or event.key == pygame.K_DOWN:
               y_pomak = 0
```

Određivanje pozicije tipkovnicom, nastavak

- Promjena koordinata i crtanje objekta

```
kx += x_pomak
ky += y_pomak

nacrtaj_covjeculjka(screen, kx, ky)
```

Kviz

- Slijedi link ispod:

  http://programarcadegames.com/quiz/quiz.php?file=controllers&lang=en

Kviz odgovori

- P1: What code will draw a circle at the specified x and y locations?
  - def draw_circle(screen, x, y):
    pygame.draw.ellipse(screen, WHITE, [0, 0, 25 + x, 25 + y])
  - def draw_circle(screen,x,y):
    pygame.draw.ellipse(screen, WHITE, [x, y, 25 + x, 25 + y])
  - def draw_circle(screen, x, y):
    pygame.draw.ellipse(screen, WHITE, [x, y, 25, 25])
- P2: The following code draws an "X". What would the code look like if it was moved from the main program loop to a function, with the ability to specify the coordinates of X appears?

```
pygame.draw.line(screen, RED, [80, 80], [100, 100], 2)
pygame.draw.line(screen, RED, [80, 100], [100, 80], 2)
```

  - def draw_x(screen, x, y):
    pygame.draw.line(screen, RED, [80+x, 80+y], [100, 100], 2)
    pygame.draw.line(screen, RED, [80+x, 100+y], [100, 80], 2)
  - def draw_x(screen, x, y):
    pygame.draw.line(screen, RED, [80, 80], [100, 100], 2)
    pygame.draw.line(screen, RED, [80, 100], [100, 80], 2)
  - def draw_x(screen, x, y):
    pygame.draw.line(screen, RED, [x, y], [20+x, 20+y], 2)
    pygame.draw.line(screen, RED, [x, 20+y], [20+x, y], 2)

Kviz, nastavak

- P3: What code will get the x and y position of the mouse?
  - pos = pygame.mouse.get_pos()
    x = pos[x]
    y = pos[y]
  - x = pygame.mouse.get_pos(x)
    y = pygame.mouse.get_pos(y)
  - pos = pygame.mouse.get_pos()
    x = pos[0]
    y = pos[1]
- P4: In the keyboard example from chapter 10, if x_speed and y_speed were both set to 3, then:
  - The object would move up and to the right at 3 pixels per second.
  - The object would move down and to the right at 3 pixels per frame.
  - The object would move up and to the left 3 pixels per frame.