

Soluzioni degli esercizi

Queste soluzioni sono proposte soprattutto per favorire un'acquisizione progressiva delle conoscenze. Bisogna partire dall'assunto che esse *non* siano le uniche o le migliori. Prima di studiarle, ognuno deve cercare in autonomia le *proprie*, che potranno anche essere molto diverse da quelle proposte. Alcune delle soluzioni seguenti potrebbero essere incomplete e presentare solo alcune idee per risolvere gli aspetti più critici del problema.

Esercizi capitolo 5 - Animazioni

Cerchi al click

```
def tick():
    if g2d.mouse_clicked():
        pos = g2d.mouse_pos()
        if dist(pos, center) <= R and g2d.confirm("Center! Exit?"):
            g2d.close_canvas()
        else:
            col = (randrange(256), randrange(256), randrange(256))
            g2d.set_color(col)
            g2d.draw_circle(pos, R)
```

 https://fondinfo.github.io/play/?exs/c05_circles.py

Eliminando la chiamata a `clear_canvas` i cerchi rimangono permanenti.

Veicolo in orizzontale

```
x, y, dx = 50, 50, 5
ARENA_W, ARENA_H = 480, 360
XMIN, XMAX, LENGTH = -100, ARENA_W + 100, ARENA_W + 200

def tick():
    global x, dx
    if g2d.mouse_clicked():
        dx = -dx
    if x + dx < XMIN:
        x += LENGTH
    if x + dx > XMAX:
```

```
x -= LENGTH
g2d.clear_canvas()
g2d.draw_image("ball.png", (x, y))
x += dx
```

▶ https://fondinfo.github.io/play/?exs/c05_vehicle.py

I veicoli e i tronchi di uno dei primi e più famosi videogame (*il gioco Frogger*) possono essere gestiti in maniera simile ¹.

Alieno

Il movimento orizzontale e quello verticale sono mutuamente esclusivi. Per garantire questo requisito, separiamo il movimento orizzontale e quello verticale nei due rami di una selezione `if-else`. Anche questo movimento ricorda uno dei primi e più famosi videogame: *Space invaders*.

```
x, y, dx, dy = 50, 50, 5, 5

def tick():
    global x, y, dx
    g2d.clear_canvas()
    g2d.draw_image("ball.png", (x, y))
    if not 0 <= x + dx <= ARENA_W - BALL_W:
        dx = -dx
        y += dy
    else:
        x += dx
```

▶ https://fondinfo.github.io/play/?exs/c05_alien.py

Rimbalzi con gravità

```
x, y, dx, dy, g = 50, 50, 5, 0, 0.5

def tick():
    global x, y, dx, dy
    g2d.clear_canvas()
    g2d.draw_image("ball.png", (x, y))
    if not 0 <= x + dx <= ARENA_W - BALL_W:
        dx = -dx
    if not 0 <= y + dy <= ARENA_H - BALL_H:
        dy = -dy
    else:
        dy += g
    x += dx
    y += dy
```

¹<https://www.dizionariovideogiochi.it/doku.php?id=frogger>

▶ https://fondinfo.github.io/play/?exs/c05_gravity.py

Basta aggiungere una piccola costante a dy , a ogni frame. Se al momento del rimbalzo si cambia solo segno a dy , il risultato è più preciso.

Movimento per 5 fotogrammi

```
import g2d

x, y, dx, count = 40, 40, -4, 0
ARENA_W, ARENA_H = 480, 360

def tick():
    global x, dx, count
    g2d.clear_canvas()
    g2d.draw_image("ball.png", (x, y))
    if g2d.mouse_click() and count == 0:
        count = 5
        dx = -dx
    if count > 0:
        count -= 1
        x += dx

def main():
    g2d.init_canvas((ARENA_W, ARENA_H))
    g2d.main_loop(tick, 5) # call tick 5 times/second

main()
```

▶ https://fondinfo.github.io/play/?exs/c05_nframes.py

Se c'è un click e la pallina è ferma, allora si avvia il conteggio. Se il conteggio è in corso, la pallina avanza.

La rana del gioco Frogger e altri personaggi hanno un comportamento simile: si muovono per pochi frame e poi si fermano.

Poligono in rotazione

```
from p21_polygon import draw_polygon
i, n = 0, 0
ARENA_W, ARENA_H, R = 600, 600, 250

def tick():
    global i
    g2d.clear_canvas()
    draw_polygon(n, (ARENA_W/2, ARENA_H/2), R, i)
    i += 1
```

▶ https://fondinfo.github.io/play/?exs/c05_rotate.py

Free climbing

 https://fondinfo.github.io/play/?exs/c05_???.py