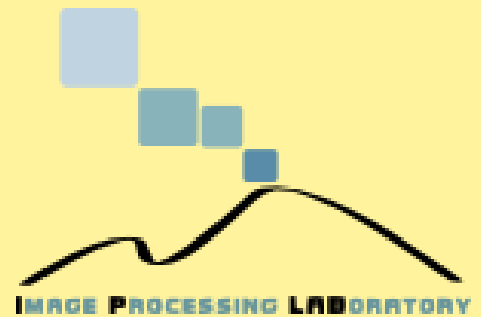


Pygame



LORENZO DI SILVESTRO

**www.dmi.unict.it/~disilvestro
disilvestro@dmi.unict.it**



Cos'è Pygame



2

- Pygame è un insieme di moduli progettati per scrivere videogiochi.
- Il progetto è stato portato avanti da Pete Shinnars nel 2000.
- Basato su SDL (Simple Directmedia Layer)
- Funziona su Windows, Linux, OSX, ...
- Documentazione <http://www.pygame.org/docs/>

Moduli



3

- **cdrom** - gestisce le unità cdrom e la riproduzione audio
- **cursors** - gestisce le immagini dei cursori
- **display** - controlla la finestra di visualizzazione sullo schermo
- **draw** - disegna semplici forme
- **event** - gestisce gli eventi e le code di eventi
- **font** - crea e visualizza caratteri TrueType
- **image** - carica e salva immagini
- **key** - gestisce la tastiera
- **mouse** - gestisce il mouse
- **movie** - visualizza e gestisce filmati mpeg
- **sndarray** - manipola suoni tramite librerie per il calcolo numerico
- **surfarray** - manipola immagini con librerie per il calcolo numerico
- **time** - controlla le operazioni di “timing”
- **transform** - ridimensiona, ruota e capovolge le immagini

Struttura base



4

1. Inizializzazione

1. Import moduli
2. Creazione Surface dello schermo
3. Assegnamento variabili

2. Impostazione sfondo Surface e visualizzazione

3. Inizio ciclo principale

1. Gestione eventi utente (mouse, tastiera ...)
2. Disegno qualcosa sullo schermo
3. Mostro lo schermo

4. Ripeto il passo 3 finché non esco

Surface



5

- Un'immagine in pygame è rappresentata come un oggetto Surface.
- È caratterizzato da:
 - Risoluzione
 - Formato dei pixel
- Anche la finestra principale di ogni programma pygame è un oggetto Surface

Impostare la Screen Surface



6

- `set_mode` per impostare il display e `set_caption` per impostare il titolo

```
# Creiamo un oggetto Surface per lo schermo  
srfcScreen = pygame.display.set_mode((800, 600))
```

```
# Impostiamo il titolo della finestra  
pygame.display.set_caption('Il mio titolo')
```

Impostare lo sfondo



7

```
# creiamo una nuova surface con la stessa dimensione  
della finestra  
srfcBackground = pygame.Surface(srfcScreen.get_size())  
  
# Impostiamo il formato dei pixel in base all  
eimpostazioni dello schermo  
srfcBackground = srfcBackground.convert()  
  
# Coloriamo lo sfondo di nero  
srfcBackground.fill((0, 0, 0))
```

oppure

```
# Utilizziamo un'immagine  
background = pygame.image.load(filename).convert()
```

Visualizzare immagini



8

- Usiamo la funzione `blit()` per disegnare un'immagine (surface) su un'altra surface

```
# Disegniamo la surface dello sfondo sulla surface della finestra
srfcScreen.blit(srfcBackground, (0, 0))
```


Modulo DRAW



9

- Il modulo DRAW disegna semplici figure su una surface
 - cerchi
 - rettangoli
 - linee

```
# ESEMPIO Disegniamo un rettangolo rosso con il primo  
vertice in posizione (200,50) con altezza 250px e  
lunghezza 125px
```

```
pygame.draw.rect(srfcBackground, (255, 0, 0),  
(200, 50, 250, 125), 0)
```

Gestire gli Eventi



10

- Acquisiamo gli eventi, li analizziamo uno ad uno, e agiamo solo in risposta a quelli che ci interessano

```
# ESEMPIO Eseguiamo il ciclo di gioco finchè non si
chiude la finestra o si preme ESC

run=True
while run:
    for event in pygame.event.get():
        if event.type == QUIT or (event.type == KEYDOWN
and event.key == K_ESCAPE):
            run=False
```

Tipi di Eventi



11

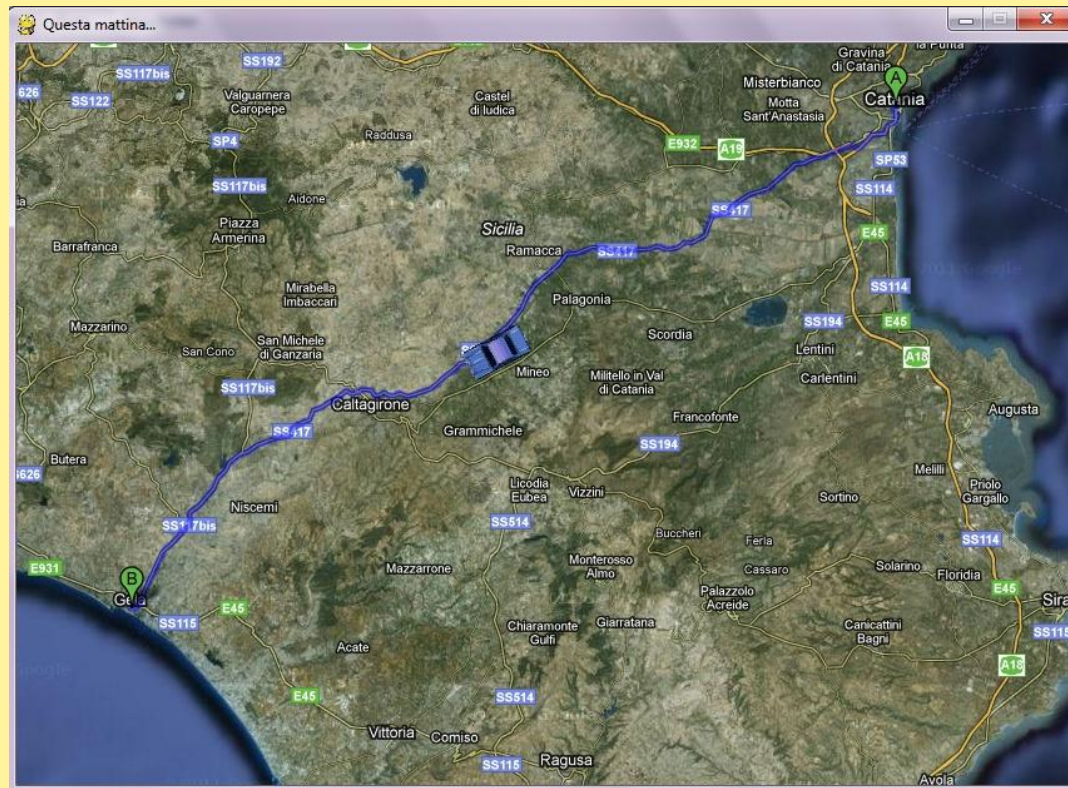
- QUIT none
- KEYDOWN unicode, key, mod
- KEYUP key, mod
- MOUSEMOTION pos, rel, buttons
- MOUSEBUTTONUP pos, button
- MOUSEBUTTONDOWN pos, button
- VIDEORESIZE size, w, h
- USEREVENT code

Pygame all'opera



12

- Gita a Gela
 - Manipolazione immagini
 - Gestione eventi da tastiera

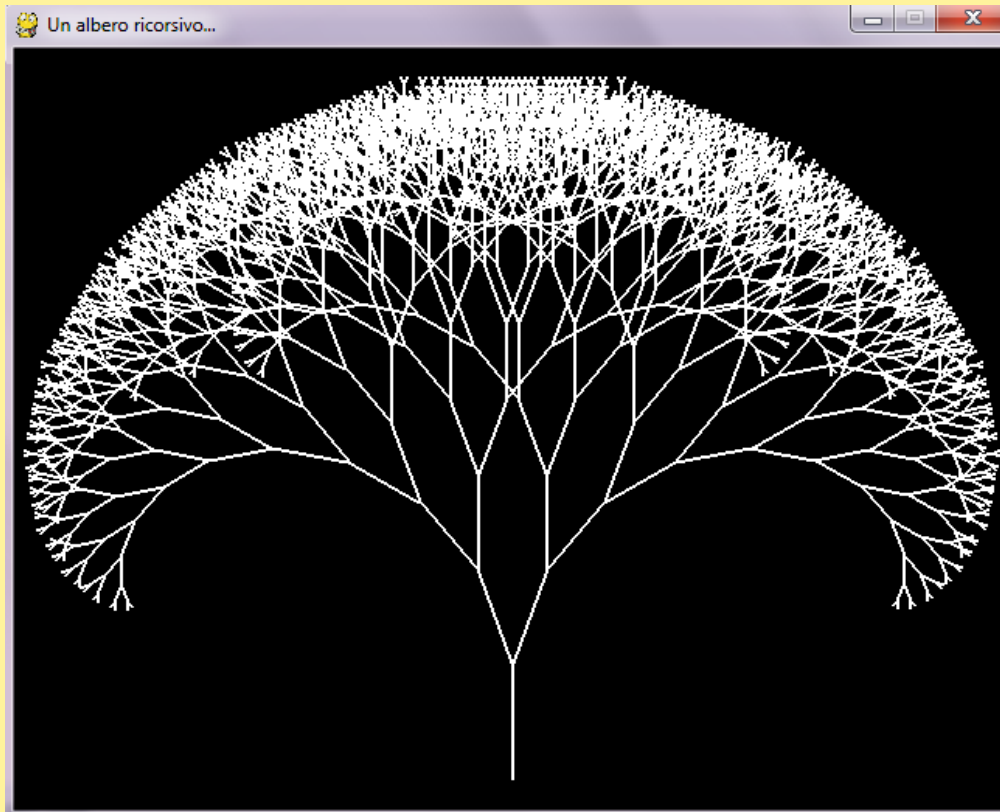


Pygame all'opera (2)



13

- L'albero ricorsivo
 - Disegni parametrici con chiamate ricorsive
 - Gestione eventi da tastiera



Pygame all'opera (3)



14

- Il costruttore di poligoni
 - Disegno interattivo di poligoni
 - Gestione eventi del mouse

