

A stylized pink lightbulb icon with radiating lines, positioned to the left of the word 'ragazze'.

ragazze DIGITALI

IDEE PER UN FUTURO SMART

Cosa faremo oggi



Cosa faremo oggi

Nel programma che faremo oggi il giocatore usera' le frecce della tastiera per muovere un quadrato nero (il nostro pacman) a destra, sinistra, in alto e in basso.

Ci saranno degli altri quadrati verdi (il cibo) che appariranno in posizioni casuali e che verranno "mangiati" dal quadrato nero quando questo li toccherà'.

Con il tasto ESC potremo uscire dal gioco e con il tasto X potremo teletrasportare il quadrato nero in una posizione random

Outline

- 1 Disegniamo la finestra di gioco e gli elementi al suo interno
- 2 Gestione degli eventi
- 3 Input da tastiera
- 4 Input da mouse
- 5 Muovere il giocatore
- 6 Disegnare il giocatore
- 7 Collision detection
- 8 Clock e `display.update()`

La finestra di gioco

```
WINDOWWIDTH = 400
WINDOWHEIGHT = 400
windowSurface =
    pygame.display.set_mode((WINDOWWIDTH,
                              WINDOWHEIGHT), 0, 32)
pygame.display.set_caption('Input')
```

set_mode(..)

- La funzione `set_mode(resolution, flags, depth)` inizializza una nuova finestra
- I suoi parametri:
 - resolution
 - flags
 - depth

set_mode(..)

- La funzione `set_mode(resolution, flags, depth)` inizializza una nuova finestra
- I suoi parametri:
 - resolution: una coppia di numeri nella forma (0, 0) che rappresenta rispettivamente la larghezza e l'altezza della finestra
 - flags: opzioni aggiuntive, alcuni esempi:
`pygame.FULLSCREEN` create a fullscreen display
`pygame.RESIZABLE` display window should be sizeable
 - depth: profondita' di colori. Possiamo non preoccuparcene, se settato a 0 pygame decide il valore migliore per noi.

pygame.Rect(...)

disegna un rettangolo e lo memorizza nella variabile player.
Quando dovremo cambiare il rettangolo che rappresenta il giocatore andremo a modificare questa variabile

pygame.Rect(...)

`pygame.Rect(X_POSITION, Y_POSITION, WIDTH, HEIGHT)`

- **X_POSITION**: Posizione della coordinata x del rettangolo sulla finestra
- **Y_POSITION**: Posizione della coordinata y del rettangolo sulla finestra
- **WIDTH**: Larghezza del rettangolo
- **HEIGHT**: Altezza del rettangolo

```
X_POSITION = 300
Y_POSITION = 100
WIDTH = 50
HEIGHT = 50
player = pygame.Rect(X_POSITION, Y_POSITION,
                      WIDTH, HEIGHT)
```


Outline

- 1 Disegniamo la finestra di gioco e gli elementi al suo interno
- 2 Gestione degli eventi**
- 3 Input da tastiera
- 4 Input da mouse
- 5 Muovere il giocatore
- 6 Disegnare il giocatore
- 7 Collision detection
- 8 Clock e `display.update()`

Pygame puo' generare eventi in seguito a input da tastiera da parte dell'utente.

Eventi

- **QUIT:** Evento generato quando il giocatore chiude la finestra di gioco.
- **KEYDOWN:** Evento generato quando il giocatore tiene premuto un tasto qualsiasi.
- **KEYUP:** Evento generato quando il giocatore rilascia un tasto qualsiasi che precedentemente stava tenendo premuto.
- **MOUSEMOTION:** Evento generato quando il mouse si muove all'interno della nostra finestra.
- **MOUSEBUTTONDOWN:** Evento generato quando viene premuto un tasto del mouse all'interno della nostra finestra
- **MOUSEBUTTONUP:** Evento generato quando viene premuto un tasto del mouse all'interno della nostra finestra

Outline

- 1 Disegniamo la finestra di gioco e gli elementi al suo interno
- 2 Gestione degli eventi
- 3 Input da tastiera**
- 4 Input da mouse
- 5 Muovere il giocatore
- 6 Disegnare il giocatore
- 7 Collision detection
- 8 Clock e `display.update()`

Input da tastiera

Settiamo 4 variabili, che rappresentano le diverse posizioni, con valori booleani inizialmente False.

Quando il giocatore userà il tasto sinistra cambieremo il valore corrispondente in True (e faremo lo stesso per gli altri tre tasti).

Quando il giocatore lascerà il tasto premuto setteremo di nuovo il valore corrispondente a False

```
moveLeft = False  
moveRight = False  
moveUp = False  
moveDown = False
```

Gestire l'evento KEYDOWN

Se l'evento scatenato e' KEYDOWN allora l'evento stesso ha un attributo chiamato "key" che indica quale tasto e' stato premuto.

```
if event.type == KEYDOWN:
    if event.key == K_LEFT:
        moveRight = False
        moveLeft = True
    if event.key == K_RIGHT:
        moveLeft = False
        moveRight = True
    if event.key == K_UP:
        moveDown = False
        moveUp = True
    if event.key == K_DOWN:
        moveUp = False
        moveDown = True
```

Table 19-2: Constant Variables for Keyboard Keys

pygame constant variable	Keyboard key	pygame constant variable	Keyboard key
K_LEFT	Left arrow	K_HOME	HOME
K_RIGHT	Right arrow	K_END	END
K_UP	Up arrow	K_PAGEUP	PGUP
K_DOWN	Down arrow	K_PAGEDOWN	PGDN
K_ESCAPE	ESC	K_F1	F1
K_BACKSPACE	Backspace	K_F2	F2
K_TAB	TAB	K_F3	F3
K_RETURN	RETURN or ENTER	K_F4	F4
K_SPACE	Spacebar	K_F5	F5
K_DELETE	DEL	K_F6	F6
K_LSHIFT	Left SHIFT	K_F7	F7
K_RSHIFT	Right SHIFT	K_F8	F8
K_CTRL	Left CTRL	K_F9	F9
K_RCTRL	Right CTRL	K_F10	F10
K_LALT	Left ALT	K_F11	F11
K_RALT	Right ALT	K_F12	F12

Gestire l'evento KEYUP

Se l'evento scatenato e' KEYUP il nostro quadrato si dovra' fermare. Settiamo quindi la direzione del tasto rilasciato a False.

```
if event.type == KEYUP:
    if event.key == K_ESCAPE:
        pygame.quit()
        sys.exit()
    if event.key == K_LEFT:
        moveLeft = False
    if event.key == K_RIGHT:
        moveRight = False
    if event.key == K_UP:
        moveUp = False
    if event.key == K_DOWN:
        moveDown = False
```

Outline

- 1 Disegniamo la finestra di gioco e gli elementi al suo interno
- 2 Gestione degli eventi
- 3 Input da tastiera
- 4 Input da mouse**
- 5 Muovere il giocatore
- 6 Disegnare il giocatore
- 7 Collision detection
- 8 Clock e `display.update()`

Gestire eventi del mouse

- **MOUSEMOTION**: Evento generato quando il mouse si muove all'interno della nostra finestra
Ha degli attributi. In particolare ci sarà utile **pos**: Indica la posizione del mouse all'interno della finestra. La posizione è rappresentata come tupla (x, y) dove x e y sono le coordinate del mouse.
- **MOUSEBUTTONDOWN**: Evento generato quando viene premuto un tasto del mouse all'interno della nostra finestra
Ha degli attributi. In particolare ci sarà utile **button**: Indica quale tasto del mouse è stato premuto.
1 rappresenta il tasto sinistro, 2 il tasto centrale (se presente), 3 il tasto destro, 4 se la rotella è stata mossa verso l'alto o 5 se mossa verso il basso.
- **MOUSEBUTTONUP**: Ha gli stessi attributi di **MOUSEBUTTONDOWN**.

Outline

- 1 Disegniamo la finestra di gioco e gli elementi al suo interno
- 2 Gestione degli eventi
- 3 Input da tastiera
- 4 Input da mouse
- 5 Muovere il giocatore**
- 6 Disegnare il giocatore
- 7 Collision detection
- 8 Clock e `display.update()`

Muovere il giocatore

Muovere il giocatore

Se una delle 4 variabili, `moveDown`, `moveUp`, `moveLeft` o `moveRight` ha valore `True` dobbiamo muovere il nostro giocatore (memorizzato nella variabile `player`).

In particolare possiamo muovere il giocatore verso l'alto se questo non e' gia' arrivato all'estremo superiore della finestra, in basso se non e' arrivato a quello inferiore e cosi' via..

Per rappresentare il movimento cambieremo la posizione verso la quale ci vogliamo muovere aggiungendo o sottraendo un valore `MOVESPEED` (che rappresenta il quanto ci stiamo muovendo)

```
if moveDown and player.bottom < WINDOWHEIGHT:
    player.top += MOVESPEED
if moveUp and player.top > 0:
    player.top -= MOVESPEED
```

Outline

- 1 Disegniamo la finestra di gioco e gli elementi al suo interno
- 2 Gestione degli eventi
- 3 Input da tastiera
- 4 Input da mouse
- 5 Muovere il giocatore
- 6 Disegnare il giocatore**
- 7 Collision detection
- 8 Clock e `display.update()`

Disegnare il giocatore

Per adesso abbiamo solo detto, definendo player, quali sono le caratteristiche del nostro giocatore.

Non lo abbiamo ancora disegnato e mostrato a video

```
pygame.draw.rect(windowSurface, BLACK, player)
```

Outline

- 1 Disegniamo la finestra di gioco e gli elementi al suo interno
- 2 Gestione degli eventi
- 3 Input da tastiera
- 4 Input da mouse
- 5 Muovere il giocatore
- 6 Disegnare il giocatore
- 7 Collision detection**
- 8 Clock e `display.update()`

Collision detection

player.colliderec()

Andiamo a rilevare le collisioni tra il giocatore (player) e i cibi (foods). Utilizziamo la funzione colliderect che verifica se due rettangoli si toccano.

```
new_foods = foods
for food in foods:
    if player.colliderect(food):
        foods.remove(food)
new_foods = []
```

Outline

- 1 Disegniamo la finestra di gioco e gli elementi al suo interno
- 2 Gestione degli eventi
- 3 Input da tastiera
- 4 Input da mouse
- 5 Muovere il giocatore
- 6 Disegnare il giocatore
- 7 Collision detection
- 8 Clock e `display.update()`

pygame.display.update()

pygame.display.update()

Le proprietà del nostro giocatore sono cambiate, come anche quelle dei cibi. Il giocatore è stato spostato, alcuni cibi rimossi e alcuni aggiunti. È necessario aggiornare la finestra di gioco con le nuove caratteristiche di giocatore e cibi.

```
pygame.display.update()
```

Clock

Inizializziamo, all'inizio del nostro programma, una variabile globale chiamata `mainClock` che ci aiuterà (in generale) a tenere traccia del passare del tempo.

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

pygame.time.tick(..)

pygame.time.tick(..)

come ultima riga del while utilizziamo la funzione tick(..) che, lascia scorrere del tempo tra una chiamata e l'altra della funzione stessa. Quindi, secondo l'esempio, dopo il primo passeranno 40 millisecondi prima che venga eseguito di nuovo il tick.

```
mainClock.tick(40)
```

Perche' questo?

Proviamo a togliere la riga mainClock.tick(40)

E' il vostro turno

E' il vostro turno!

Prendiamo come riferimento il codice al link, potete copiarlo
Download

Completiamo il codice fornito secondo le indicazioni della pagine successive.

E' il vostro turno - Parte 1

E' il vostro turno!

- Crea una variabile che contenga, inizialmente, una lista vuota (chiamandola foods)
- Crea anche una costante chiamata FOODSIZE che rappresenterà sia la larghezza sia l'altezza del cibo (dandole il valore che preferisci, per il giocatore avevamo usato 50)

E' il vostro turno - Parte 1

E' il vostro turno!

- Con un ciclo for (che si ripeta 20 volte, aiuto: `range(20)`) aggiungiamo alla lista `foods` (aiuto: `append(..)`) dei rettangoli che rappresenteranno i "cibi" del nostro gioco.
Usiamo `pygame.Rect(..)` come visto nell'esempio con cui abbiamo creato il rettangolo per il giocatore.
La larghezza e l'altezza del rettangolo sara' `FOODSIZE`.
- Prima abbiamo usato `X_POSITION` e `Y_POSITION` per posizionare il rettangolo del giocatore. In questo caso usa un numero generato casualmente compreso tra 0 e la larghezza della finestra `WINDOWWIDTH - FOODSIZE` (con `random.randint(..)` come abbiamo visto)

E' il vostro turno - Parte 2

E' il vostro turno!

Abbiamo visto come gestire eventi generati dalla pressione e dal rilascio dei tasti su, giu, destra o sinistra.

Dai la possibilita' all'utente di usare **anche** le lettere **W** per muoversi verso l'alto, **S** verso il basso, **A** verso sinistra e **D** verso destra (quindi o le frecce o le lettere).

```
event.key == K_w
# Per la pressione del tasto W
event.key == K_s
# Per la pressione del tasto S
event.key == K_a
# Per la pressione del tasto A
event.key == W_d
# Per la pressione del tasto D
```

E' il vostro turno - Parte 3

E' il vostro turno!

Ora vogliamo dare la possibilita' al giocatore di essere teletrasportato in una posione casuale alla pressione del tasto X

Aggiungi la gestione dell'evento "pressione del tasto X" agli if di keyup.

```
# per teletrasportare il giocatore in una
# posizione random usa, dentro all'if:
player.top = random.randint(0, WINDOWHEIGHT -
    player.height)
player.left = random.randint(0, WINDOWWIDTH -
    player.width)
```


E' il vostro turno - Parte 4

E' il vostro turno!

Aggiungi un "cibo" nella posizione in cui si trovava il mouse al momento della pressione del tasto sinistro (la pressione in questo caso non ci interessa, ci interessa solo il rilascio `MOUSEBUTTONUP`)
Lo si puo' fare prendendo spunto dall'esempio in cui abbiamo aggiunto i cibi in posizione random. In questo caso li aggiungeremo nella posizione del mouse (elementi 0 e 1 della lista `event.pos[0]`).

E' il vostro turno - Parte 5

E' il vostro turno!

Abbiamo visto come muovere il giocatore verso l'alto e verso il basso. Seguendo quell'esempio aggiungi il movimento verso sinistra e verso destra (le posizioni sinistra e destra del giocatore sono rispettivamente `player.left` e `player.right`)

E' il vostro turno - Parte 6

E' il vostro turno!

Analogamente a come abbiamo disegnato e mostrato a video il quadrato del giocatore, mostra a video/disegna i quadrati di tutti i food presenti nella lista foods

E' il vostro turno - Parte 7

E' il vostro turno!

La funzione **pygame.time.get_ticks()** restituisce il tempo (in millisecondi) trascorso da quando e' stato avviato pygame (e quindi, nel nostro caso, il programma).

Come possiamo fare un countdown del tempo? Prova a farlo stampando il tempo con una print

Ad esempio:

- 10 s
- 9 s
- 8 s
- ...

Materiale rilasciato con licenza
Creative Commons - Attributions, Share-alike 4.0

