

ELENCO DEI COMANDI DI PYGAME

<code>screen = pygame.display.set_mode(size)</code>	Restituisce la Surface dell'intera finestra
<code>pygame.display.set_caption(caption)</code>	Setta la scritta sulla barra superiore della finestra
<code>clock = pygame.time.Clock()</code>	Costruttore della classe Clock
<code>clock.tick(fps)</code>	Aspetta che passi il prossimo tick cioè intervallo di tempo che misura 1/fps secondi
<code>pygame.event.get()</code>	Restituisce la lista degli eventi avvenuti
<code>event.type</code>	Restituisce il tipo di evento (se la variabile event è un evento)
<code>QUIT, MOUSEBUTTONDOWN, MOUSEBUTTONUP</code>	Codici dei tipi di evento importati da <code>pygame.locals</code>
<code>pygame.init()</code>	Inizializza la libreria pygame
<code>pygame.quit()</code>	Disattiva la libreria pygame, da eseguire prima di chiudere l'applicazione
<code>sys.exit()</code>	Termina l'applicazione, serve importare sys
<code>pygame.mouse.get_pressed()</code>	Restituisce la tupla contenente una lista di valori true o false associata ad ogni tasto del mouse. Posizione 0 tasto sinistro, 1 tasto centrale, 2 tasto destro
<code>pygame.mouse.get_pos()</code>	Restituisce la tupla (x, y) delle coordinate del cursore del mouse
<code>pygame.key.get_pressed()</code>	Restituisce un oggetto che puoi usare come una tupla che in ogni posizione contiene un valore true o false che indica se un determinato tasto è premuto o no. Per le posizioni si usano i codici importati da <code>pygame.locals</code> . Ad esempio: <code>K_w</code> è la posizione per il tasto 'w', <code>K_UP</code> è la freccia in alto...
<code>pygame.display.flip()</code>	funzione che aggiorna lo schermo
<code>pygame.display.update()</code>	Funzione analoga a flip ma che con l'utilizzo di parametri aggiuntivi permette di aggiornare porzioni di schermo. Senza parametri è identica a flip
<code>pygame.Surface((larghezza, altezza))</code>	Costruttore della classe Surface che di fatto è un'immagine
<code>pygame.Rect(posx, posy, larghezza, altezza)</code>	Costruttore della classe Rect, che indica posizione e ingombro di qualcosa, in pratica una hitbox
<code>rect = image.get_rect()</code>	Funzione che a partire da una Surface restituisce un Rect con le stesse dimensioni. È necessario poi assegnare la posizione al Rect
<code>rect.bottom, .bottomleft, .bottomright, .center, .centerx, .centery, .top, .topright...</code>	Modi per accedere (leggere o modificare) alla posizione del rect. Modificata una si modificano tutte. Può riferirsi ad una coppia x, y (come center o topleft) o solo x o y (come top o centerx). la posizione indicata nel costruttore equivale a topleft
<code>rect.colliderect(alaltrorect)</code>	Funzione della classe Rect che controlla se il rect collide con un altro rect ricevuto come parametro
<code>rect.collidepoint(coordinate)</code>	Funzione della classe Rect che controlla se un punto ricevuto come parametro collide col rect

colore = (255,255,255)	I colori si indicano col formato rgb
image.fill(colore)	Riempie una Surface o immagine con un colore
pygame.draw.rect(image, colore, (x, y, larg, alt), spessore) oppure pygame.draw.rect(image, colore, rect, spessore)	Funzione per disegnare un rettangolo, riceve come parametri una Surface o immagine su cui disegnare il rettangolo, un colore, una tupla con i dati del rettangolo o un Rect da cui estrae i valori e opzionalmente uno spessore. Se lo spessore non è indicato disegna un rettangolo pieno se no disegna il bordo del rettangolo con quello spessore.
pygame.draw.circle(image, colore, centro, raggio, spessore)	Analogo a draw.rect ma riceve due parametri centro e raggio invece che la tupla del rettangolo
screen.blit(image, rect) oppure screen.blit(image, pos)	Disegna su screen (una Surface) l'immagine 'image' nella posizione indicata dal rect. Posso indicare la posizione con una tupla (x, y) invece che con un Rect
image = pygame.image.load("cartella/nomefile.estensione)	Funzione che carica un'immagine con un certo path (composto da percorso cartella più nome) e restituisce una Surface. ./ cartella corrente (inutile scriverlo) ../ cartella superiore immagini/ entra nella cartella immagini ../immagini esce dalla cartella corrente ed entra in immagini
image = pygame.transform.scale(image, size)	Funzione che modifica la dimensione di un'immagine (Surface). Riceve una Surface e una size (tupla largh altezza) e restituisce la nuova Surface ridimensionata.
image = oygame.transform.rotate(image, gradi)	Funzione che riceve un immagine e restituisce l'immagine ruotata di tot gradi
font = pygame.font.Font(None, 50) oppure font = pygame.font.Font("arial", 50)	Costruttore dell'oggetto Font, serve per poter poi costruire un'immagine con una scritta. Il primo parametro è il nome del font, se indico None prende il font di default. Il secondo parametro è l'altezza del font in pixel.
imgtext = font.render(testo, True, colore)	Crea un'immagine (Surface) che contiene la scritta indicata dal parametro "testo" del colore indicato. È necessario passare il secondo parametro che indica se l'antialiasing deve essere attivo o no. Posso scrivere True o 1 oppure False o 0.