Elenco dei comandi di pygame

|  |  |
| --- | --- |
| screen = pygame.display.set\_mode(size) | Restituisce la Surface dell’intera finestra |
| pygame.display.set\_caption(caption) | Setta la scritta sulla barra superiore della finestra |
| clock = pygame.time.Clock() | Costruttore della classe Clock |
| clock.tick(fps) | Aspetta che passi il prossimo tick cioè intervallo di tempo che misura 1/fps secondi |
| pygame.event.get() | Restituisce la lista degli eventi avvenuti |
| event.type | Restituisce il tipo di evento (se la variabile event è un evento) |
| QUIT, MOUSEBUTTONDOWN, MOUSEBUTTONUP | Codici dei tipi di evento importati da pygame.locals |
| pygame.init() | Inizializza la libreria pygame |
| pygame.quit() | Disattiva la libreria pygame, da eseguire prima di chiudere l’applicazione |
| sys.exit() | Termina l’applicazione, serve importare sys |
| pygame.mouse.get\_pressed() | 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 |
| pygame.mouse.get\_pos() | Restituisce la tupla (x, y) delle coordinate del cursore del mouse |
| pygame.key.get\_pressed() | 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 pygame.locals. Ad esempio: K\_w è la posizione per il tasto ‘w’, K\_UP è la freccia in alto… |
| pygame.display.flip() | funzione che aggiorna lo schermo |
| pygame.display.update() | Funzione analoga a flip ma che con l’utilizzo di parametri aggiuntivi permette di aggiornare porzioni di schermo. Senza parametri è identica a flip |
| pygame.Surface((larghezza, altezza)) | Costruttore della classe Surface che di fatto è un immagine |
| pygame.Rect(posx, posy, larghezza, altezza) | Costruttore della classe Rect, che indica posizione e ingombro di qualcosa, in pratica una hitbox |
| rect = image.get\_rect() | Funzione che a partire da una Surface restituisce un Rect con le stesse dimensioni. È necessario poi assegnare la posizione al Rect |
| rect.bottom, .bottomleft, .bottomright, .center, .centerx, .centery, .top, .topright… | 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 |
| rect.colliderect(altrorect) | Funzione della classe Rect che controlla se il rect collide con un altro rect ricevuto come parametro |
| rect.collidepoint(coordinate) | 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. |