

# Pygame - Zusammenfassung

## Aufbau eines Programms:

```
import pygame
pygame.init()

FPS = 60
screen_width = 800
screen_height = 600
clock = pygame.time.Clock()

screen = pygame.display.set_mode((screen_width, screen_height))
pygame.display.set_caption("Mein Pygame Spiel")

#----

while True:
    screen.fill((0, 0, 0)) # Fenster schwarz füllen

    #----

    clock.tick(FPS)
    pygame.display.flip()
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            pygame.quit()
            exit()
```

Pygame importieren und initialisieren

Variablen für das Spiel festlegen (FPS, Punktestände...), Bilder, Sounds, Schriftarten und weiteres laden.

Fenster für das Spiel erstellen.

Funktionen für das Spiel definieren.

Game Loop - die Funktionen des Spiels ausführen:

- Fenster leeren (-> einfarbig füllen)
- Input vom Benutzer (z.B. über Tastatur) abfragen
- Spiellogik: Funktionen, Grafiken zeichnen
- Prüfen, ob das Fenster geschlossen werden soll.

## Funktionen und wie sie Funktionieren

Aufbau einer Funktion - Beispiel: `blit(surface, Position)`

„blit“ ist hier der Name der Funktion.

Innerhalb der Klammern stehen Argumente (=Daten), die man der Funktion geben muss, dass sie funktioniert.

Surface muss ein `pygame.Surface` - also ein Bild oder Fenster sein.

Position ist eine Liste aus zwei Zahlen: eine x- und eine y-Koordinate. z.B. `[400,200]`

## Funktionen von Pygame

(Hinter den Argumenten steht jeweils in grün, was für Daten die Funktion hier erwartet)

`pygame.init()` - initialisiert pygame

`pygame.display.set_mode(size - Liste aus zwei Zahlen)`

-> Erstellt ein Fenster mit einer bestimmten Größe. In Size sind Höhe und Breite des Fensters gespeichert. Ist size bspw. `[800, 600]` ist das Fenster 800 x 600 Pixel groß.

`Surface.fill(RGB Farbe)` Surface = ein Bild oder Fenster

-> Füllt das Bild/Fenster mit einer RGB-Farbe (Liste aus drei Zahlen zwischen 0 und 255)

`Surface.blit(Bild, Position - Liste aus zwei Zahlen)`

-> Zeigt ein Bild an einer Position.

`pygame.image.load(Dateipfad)`

Lädt Bild aus einem Dateipfad (= „Ordner/Ordner/Dateiname.Endung“, z.B. „Bilder/test.png“)

`pygame.display.flip()` - updatet das Fenster

`pygame.quit()` - beendet pygame

`pygame.key.get_pressed()[pygame-taste]`

-> Gibt True (Ja) oder False (Nein) zurück, je nachdem ob die abgefragte Taste gedrückt ist. Pygame-tasten sind z.B. `pygame.K_A` für die Taste A, `pygame.K_Space` für Leertaste etc.

`pygame.font.Font(Dateipfad, Größe - Zahl)`

-> Lädt eine Schriftart aus einem Dateipfad mit einer bestimmten Größe (z.B. 24, 32...)

`Font.render(Text, True, RGB Farbe)` Font = eine mit `pygame.font.Font` geladene Schriftart

-> Zeichnet einen Text in einer bestimmten Farbe und Schriftart. Dieser kann dann z.B. mit `.blit` auf dem Bildschirm angezeigt werden.