Kapitel 12.2: Tastatursteuerung – Glossar

Tastatursteuerung

Drückt der Benutzer eine Taste auf der Tastatur, dann merkt `pygame` sich das und wir können es bei unserem nächsten Schleifendurchlauf abfragen und darauf reagieren.

Zur Erinnerung: Wir kennen so etwas schon: Wenn der Benutzer mit der Maus auf das X im Fenster drückt, dann schließt sich das Fenster, weil wir das `pygame.QUIT`-Event in jedem Schleifendurchlauf abfangen. Drückt der Benutzer eine Taste, dann wird allerdings kein Event ausgelöst, sondern man kann einfach zu jedem beliebigen Zeitpunkt abfragen, welche Taste(n) der Benutzer gerade im Moment gedrückt hält.

pygame.key.get_pressed()

Die Liste der gerade vom Benutzer gedrückten Tasten erhält man mit `pygame.key.get_pressed()`

Die Tasten

Auf die einzelnen Tasten kann man über in `pygame` vordefinierte Variablen zugreifen.

- `pygame.K_LEFT` → Pfeiltaste nach links
- `pygame.K_RIGHT` → Pfeiltaste nach rechts
- `pygame.K_UP` → Pfeiltaste nach oben
- `pygame.K_DOWN` → Pfeiltaste nach unten
- `pygame.K_SPACE` → Leertaste
- `pygame.K_RETURN` → Return-Taste

Abfrage

Mit bedingten Anweisungen (*if statements*) kann man jetzt abfragen, ob eine bestimmte Taste gedrückt wurde.

Beispiel: `keys[pygame.K_LEFT]` ist `True`, wenn die Taste mit dem Pfeil nach unten gedrückt wird. Im Grunde kann man alle Tasten, die ein Benutzer drücken kann, auf diese Weise abfragen. Die Tasten-Variablen fangen alle mit `pygame.K_` an.

None

Wenn es ein Objekt manchmal nicht gibt, dann kann man es auf den Wert `None` setzen. Häufig ist es z. B. so, dass Objekte beim ersten Durchlauf einer for-Schleife erzeugt werden. Wenn sie dann im nächsten Durchlauf existieren, kann man etwas mit ihnen tun. Sobald eine bestimmte Bedingung erfüllt ist, werden sie wieder gelöscht.

Greift man auf Methoden (oder Attribute) eines Objekts zu, das noch nicht erzeugt wurde, erhält man eine Fehlermeldung. Setzt man das Objekt zuerst auf None, dann hat man die Möglichkeit abzufragen, ob das Objekt bereits erzeugt wurde oder nicht.

is-Operator

Um abzufragen, ob ein Objekt bereits erzeugt wurde oder nicht, braucht man den is-Operator.

```
if obj is None:
# Objekt erzeugen

if obj is not None:
# mit dem Objekt kann man arbeiten, weil es existiert

if obj:
# das geht auch und ist das gleiche wie `if obj is not None`
```