

Sammlung von Aktionen, die zusammengefügt werden. Die `update(dt)` Funktion hat den Namen `update` und sie nimmt, das Argument `dt`. Sie wird sehr oft von PyGame Zero aufgerufen: Dort gibts du den Code ein, der den Status deines Spiels ändert. Momentan macht unser Spiel "nichts". Da in Python eine Funktion nicht leer gelassen werden kann, schreiben wir einfach den leeren Befehl `pass`.

```
7 def draw():
8     screen.fill((128, 0, 0))
```

Die `draw` Funktion wird direkt nach der `update(dt)` Funktion ausgeführt. In diesem ersten Programm ruft es einfach die `fill()` Funktion (die sich in dem PyGame Zero Modul namens `screen` befindet), um den Hintergrund des Fensters auf ein dunkles Rot zu setzen: Das Argument, das wir an die `screen.fill` Funktion übergeben, ist ein Tupel, das aus dem roten, grünen und blauen Komponente der Farbe besteht.



Ale Rimoldi, Coderdojo Zürich

Sammlung von Aktionen, die zusammengefügt werden. Die `update(dt)` Funktion hat den Namen `update` und sie nimmt, das Argument `dt`. Sie wird sehr oft von PyGame Zero aufgerufen: Dort gibts du den Code ein, der den Status deines Spiels ändert. Momentan macht unser Spiel "nichts". Da in Python eine Funktion nicht leer gelassen werden kann, schreiben wir einfach den leeren Befehl `pass`.

```
7 def draw():
8     screen.fill((128, 0, 0))
```

Die `draw` Funktion wird direkt nach der `update(dt)` Funktion ausgeführt. In diesem ersten Programm ruft es einfach die `fill()` Funktion (die sich in dem PyGame Zero Modul namens `screen` befindet), um den Hintergrund des Fensters auf ein dunkles Rot zu setzen: Das Argument, das wir an die `screen.fill` Funktion übergeben, ist ein Tupel, das aus dem roten, grünen und blauen Komponente der Farbe besteht.

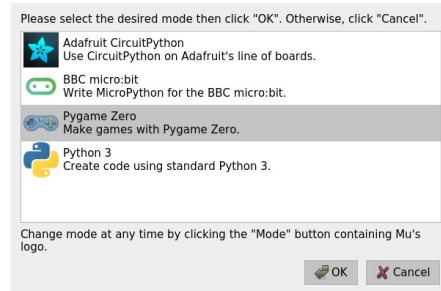


Ale Rimoldi, Coderdojo Zürich

Dein erstes Programm



Wenn du den Mu-Editor startest, kannst du ein Modus auswählen: wähle "PyGame Zero"



Falls du bereits einen anderen Modus ausgewählt hast, kannst du es ändern, indem du auf "Mode" in der Werkzeugleiste klickst. Jetzt kannst du "PyGame Zero" Modus auswählen.

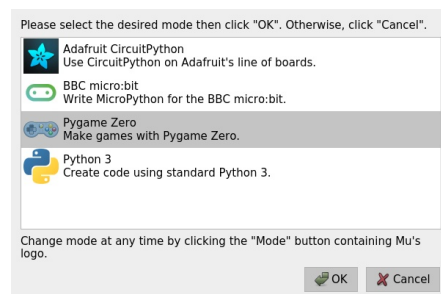
In dem Editorfenster kannst du den Platzhalter entfernen:

```
1 # Write code here :-)
```

Dein erstes Programm



Wenn du den Mu-Editor startest, kannst du ein Modus auswählen: wähle "PyGame Zero"



Falls du bereits einen anderen Modus ausgewählt hast, kannst du es ändern, indem du auf "Mode" in der Werkzeugleiste klickst. Jetzt kannst du "PyGame Zero" Modus auswählen.

In dem Editorfenster kannst du den Platzhalter entfernen:

```
1 # Write code here :-)
```

und dein erstes Code eintippen:

```
1 WIDTH = 640
2 HEIGHT = 480
3
4 def update(dt):
5     pass
6
7 def draw():
8     screen.fill((128, 0, 0))
```



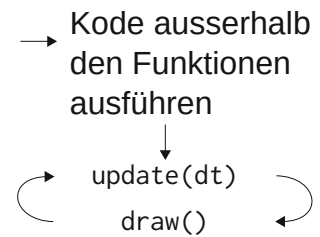
Klicke auf "Play": speichere die Datei als `main.py` in einem neuen Verzeichnis, das nach deinem Projekt benannt ist. Sobald du auf "Save" geklickt hast, sollte ein dunkelrotes Fenster auftauchen: dein Programm läuft!



Du kannst es schließen, indem du in der Symbolleiste des Mu-Editors auf die Schaltfläche "Stop" klickst.

Was macht der Code, den du gerade geschrieben hast?

Python Script laden



```
1 WIDTH = 640
2 HEIGHT = 480
```

Die beiden Variablen `WIDTH` und `HEIGHT` werden ausserhalb aller Funktionen definiert: Sie sind globale Variablen, auf die man von überall aus zugreifen kann.

Die zwei Variablen enthalten die Breite und Höhe des Fensters und werden "auf magische" Art und Weise von PyGame verwendet, um die Größe des Spiels festzulegen.

```
4 def update(dt):
5     pass
```

Das Schlüsselwort `def` startet die Definition einer Funktion. Eine Funktion ist eine An-

und dein erstes Code eintippen:

```
1 WIDTH = 640
2 HEIGHT = 480
3
4 def update(dt):
5     pass
6
7 def draw():
8     screen.fill((128, 0, 0))
```



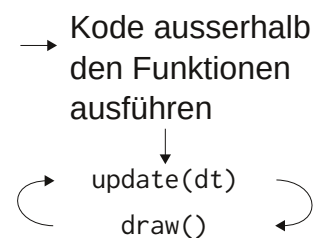
Klicke auf "Play": speichere die Datei als `main.py` in einem neuen Verzeichnis, das nach deinem Projekt benannt ist. Sobald du auf "Save" geklickt hast, sollte ein dunkelrotes Fenster auftauchen: dein Programm läuft!



Du kannst es schließen, indem du in der Symbolleiste des Mu-Editors auf die Schaltfläche "Stop" klickst.

Was macht der Code, den du gerade geschrieben hast?

Python Script laden



```
1 WIDTH = 640
2 HEIGHT = 480
```

Die beiden Variablen `WIDTH` und `HEIGHT` werden ausserhalb aller Funktionen definiert: Sie sind globale Variablen, auf die man von überall aus zugreifen kann.

Die zwei Variablen enthalten die Breite und Höhe des Fensters und werden "auf magische" Art und Weise von PyGame verwendet, um die Größe des Spiels festzulegen.

```
4 def update(dt):
5     pass
```

Das Schlüsselwort `def` startet die Definition einer Funktion. Eine Funktion ist eine An-