

Wir können aus unserem Projekt eine ausführbare Datei erstellen und diese dann weitergeben. Auf dem Zielsystem muss kein Python installiert sein.

Zu PyInstaller gibt es eine ausführliche Anleitung, die du auf jeden Fall einmal anschauen solltest.
<https://pyinstaller.org/en/stable/>

PyInstaller installieren

Pyinstaller kann direkt in Pycharm installiert werden. Dazu rufst du Python Packages auf und suchst nach 'pyinstaller'. Momentan ist Version 6.3.0 aktuell. Diese oder eine neuere Version kannst du installieren.

Projekt vorbereiten

Es sollte immer die neuste Version der Pygame-Tools verwendet werden. Daher sollte das Lib - Verzeichnis immer mit der neusten Version auf Github aktualisiert werden.

<https://github.com/hobbyelektroniker/SpasMitPygame>

Wir arbeiten mit einer Kopie des lauffähigen Projekt. Deshalb sollte zuerst überprüft werden, ob das Projekt noch funktioniert.

Alle Installationsprojekte werden innerhalb eines **Install** - Verzeichnisses erstellt. Falls es noch nicht existiert, solltest du es erstellen.

Als Beispiel wandeln wir hier das Projekt **Aliens** in ein ausführbares Programm um.

Das ursprüngliche Projektverzeichnis **Aliens** wird in das **Install** - Verzeichnis kopiert.
Das Verzeichnis **Install/Aliens** dient jetzt als Projektverzeichnis für das Installationsprojekt.

Jetzt kopieren wir noch das **Lib** - Verzeichnis in das neue Projektverzeichnis.

Wir haben jetzt diese Struktur:

```
Install
  Aliens
    Lib
    ...
    media
    ...
    mygame.py
    others.py
    player.py
```

Das Verzeichnis **data** wurde nicht verwendet, daher wird es hier nicht benötigt.

Die Kommandozeile

Der erste Build - Prozess muss mit einer Kommandozeile gestartet werden. Deshalb müssen wir ein Terminal öffnen. In Pycharm gehen wir mit der rechten Maustaste auf das **Aliens** - Verzeichnis und wählen **Open In Terminal**.

Mac

```
pyinstaller mygame.py -w --onefile --name "aliens" --add-data "media/*:media" --add-data "Lib/*:."
```

Windows

Unter Windows sind oft noch weitere Vorbereitungen notwendig. Da hier aus Scripts eine ausführbare Datei zusammengebastelt wird, stuft Windows den Vorgang als sehr gefährlich ein. So wird der Windows Defender die meisten Dateien im Verzeichnis löschen. Dasselbe gilt auch für viele andere Schutzprogramme. Deshalb klappt der Vorgang nur, wenn das Install - Verzeichnis von der Überwachung ausgeschlossen wird.

```
pyinstaller mygame.py -w --onefile --name "aliens" --add-data "media/*;*media" --add-data "Lib/*:."
```

Die beiden Befehle sind weitgehend identisch. Unter Windows wird lediglich ; anstelle von : verwendet.

Durch diesen Befehl wird ein erster Build durchgeführt und die Konfigurationsdatei **aliens.spec** erzeugt. Weiter Builds können dann direkt mit `pyinstaller aliens.spec` durchgeführt werden. Dabei kann die Konfiguration direkt in **aliens.spec** angepasst werden.

Das build - Verzeichnis

Das ist ein temporäres Verzeichnis, das nur während des Build - Prozesses benötigt wird. Es kann jederzeit gelöscht werden.

Das dist - Verzeichnis

Mac

Hier findet man die beiden ausführbaren Dateien **aliens** und **aliens.app**.

Beide laufen eigenständig und können direkt weitergegeben werden.

aliens ist eine Kommandozeilen - Applikation, die normalerweise nicht gebraucht wird.

aliens.app ist eine normale Desktop - Applikation. Wie im Mac üblich, handelt es sich dabei um ein Applikationpackage. Deshalb wird es in Pycharm als Verzeichnis angezeigt.

Windows

aliens.exe ist eine normale ausführbare Datei, die direkt weitergegeben werden kann.