چند قدم به کارگاه بعد

مولتی‌تردینگ و همزمانی

برای جلوگیری از غیرقابل‌پاسخ شدن رابط هنگام اجرای کارهای بلندمدت، باید بار پردازشی را به Threadهای فرعی منتقل کرد. PyQt6 کلاس `QThread` را برای این منظور ارائه می‌دهد Python Tutorials – Real Python.

نکات کلیدی

- هر Thread جدید از `QThread` مشتق شود و منطق طولانی در متد `run()` اجرا گردد Stack Overflow.
- ارتباط با Thread اصلی از طریق سیگنال و اسلات انجام می‌شود تا از دسترسی همزمان به منابع مشترک جلوگیری شود Python Tutorials – Real Python.
- از `moveToThread()` برای انتقال یک `QObject` به Thread دیگر استفاده کنید و سپس سیگنال‌ها را متصل نمایید sihabsahariar.medium.com.

QRunnable و QThreadPool

برای مدیریت مجموعه‌ای از کارهای کوتاه و قابل تکرار، استفاده از `QThreadPool` به‌صرفه‌تر است Python GUIs .

```
from PyQt6.QtCore import QRunnable, QThreadPool

class Task(QRunnable):
    def run(self):
        # long-running operation
        ...

pool = QThreadPool.globalInstance()
pool.start(Task())
```

- `QThreadPool` به طور خودکار تعداد Threadها را مدیریت می‌کند و از سربار ایجاد و نابودی Threadهای جدید جلوگیری می‌کند Python GUIs.

استایل و Qt Style Sheets (QSS)

معرفی Qt Style Sheets

خب Qt Style Sheets مشابه CSS وب هستند و به شما امکان می‌دهند ظاهر ویجت‌ها را بدون تغییر کد پایتون تنظیم کنید Qt Documentation. فایل‌های `.qss` را می‌توان در ابتدای برنامه بارگذاری کرد:

```
with open('style.qss', 'r') as f:
    app.setStyleSheet(f.read())
```

نکات کاربردی

- می‌توانید states ویجت (hover, pressed) را هدف قرار دهید:

```
QPushButton:hover { background-color: lightgray; }
QPushButton:pressed { background-color: gray; }
 ``` :contentReference[oaicite:7]{index=7}.
```

- برای سفارشی‌سازی ویجت‌های اختصاصی، از `qproperty-&lt;name&gt;` در QSS استفاده کنید Python GUIs Forum.

## بین‌المللی‌سازی (i18n & l10n)

### ابزارهای Qt برای ترجمه

خب Qt از ابزارهای `lupdate`، `linguist` و `lrelease` برای استخراج متن‌های قابل ترجمه و تولید فایل‌های `.qm` پشتیبانی می‌کند BCCNSoft Documentation.

#### گام‌های اصلی

۱. در کد از `self.tr("…")` برای متن‌های رابط استفاده کنید GitHub.

۲. با `pylupdate5 project.pro` فایل `.ts` بسازید.
```

۳. در Qt Linguist فایل `.ts` را ترجمه و `lrelease` کنید تا فایل `.qm` تولید شود.

۴. در برنامه با `translator.load('fa.qm')` ترجمه را فعال کنید.

نکته‌ی عملی

برای تغییر زبان در زمان اجرا، باید دوباره به `QCoreApplication.installTranslator()` فراخوانی کنید و رابط را ری‌بار کنید Stack Overflow.

بهینه‌سازی عملکرد

بارگذاری تنبل (Lazy Loading)

تنظیم ویجت‌ها یا داده‌های سنگین تنها در هنگام نیاز، مصرف حافظه و زمان شروع را کاهش می‌دهد Python Central.

کاهش Flicker

- از بافر ثانویه (`QPixmap` buffer) برای رسم مکرر استفاده کنید تا flicker کاهش یابد (همانند مثال Paint App) cuteprogramming.
- در صورت امکان از OpenGL via `QOpenGLWidget` برای رندر سریع‌تر بهره ببرید.

پروفایلینگ

- از ماژول‌های استاندارد `cProfile` و ابزارهای بصری‌ساز مانند SnakeViz برای شناسایی گلوگاه‌ها استفاده کنید.

بسته‌بندی و انتشار (Packaging & Distribution)

روی ویندوز PyInstaller

برای ساخت EXE:

```
pip install pyinstaller
pyinstaller --noconsole --onefile main.py
```

- از گزینه `--add-data` برای درج فایل‌های UI و منابع استفاده کنید Python GUIs.

بسته‌بندی در لینوکس

می‌توانید با PyInstaller بسته‌ی اجرایی بسازید و با ابزارهایی مانند `fpm` یا `dpkg` فایل `.deb` تولید کنید Python GUIs.

macOS

با `py2app` یا PyInstaller می‌توانید برنامه‌ی macOS Bundler بسازید.

نکات نهایی

- فایل‌های `.ui` را پیش از انتشار به کد پایتون تبدیل و حذف کنید تا وابستگی کاهش یابد Stack Overflow.
- از ابزارهای ساخت خودکار (Makefile, CI scripts) برای تولید بسته در پلتفرم‌های مختلف بهره ببرید.