UNIVERZITA KARLOVA

Přírodovědecká fakulta

Katedra aplikované geoinformatiky a kartografie

Studijní program: Geoinformatika, kartografie a dálkový průzkum Země

Studijní obor: N-GKDPZ

David Šklíba

Automatizovaná tvorba atlasu v prostředí PyQGIS

Open Source GIS

Jablonec nad Nisou, 2022

Cíle semestrální práce

1) Seznámení se s objektově orientovaným programováním.
2) Seznámení se s PyQGIS.
3) Navržení a implementace automatizované tvorby tematického atlasu.
4) Porovnání využití Pythonu mezi SW QGIS a ArcGIS for Desktop

Využití Pythonu v SW QGIS Desktop

Existuje několik způsobů, jakými lze využít Python v SW QGIS. Nejjednodušší variantou je přímé vydávání příkazů do konzole otevřeného programu. Dále je možné vytvářet a používat pluginy, při zapnutí programu automaticky spouštět kód nebo vytvářet processingové funkce. Tato práce bude těžit z možnosti vytváření vlastních aplikací založených na QGIS API.

QGIS poskytuje integrovanou Python konzoli pro psaní skriptů. V menu ji nalezneme pod položkou „Zásuvné modely“. Výhodou SW QGIS je jeho vlastní editor zdrojového kódu, a tak pro psaní skriptů není třeba žádné druhé aplikace. Kdyby uživatel přece jenom trval na použití jeho oblíbeného editoru zdrojového kódu, neobejde se bez pár nezbytných řádků kódu navíc:

```
from qgis.core import *
```

Načtení QGIS knihovny.

```
QgsApplication.setPrefixPath("/path/to/qgis/installation", True)
```

Zde je místo prvního argumentu třeba nastavit cestu do místa instalace SW QGIS.

```
qgs = QgsApplication([], True)
```

Vytvoření odkazu na QGIS aplikaci a povolení využívání grafického uživatelského rozhraní.

```
qgs.initQgis()
```

Tímto příkazem započíná kód vlastní aplikace.

```
qgs.exitQgis()
```

Poslední příkaz skriptu vymaže registry použitých vrstev z paměti.

Vstupní údaje aplikace automatického vytvoření atlasu hustoty zalidnění

Nezbytným vstupem pro vytvoření jednotlivých map jsou názvy shapefilů mapových prvků a cesta, kde jsou uložené. Bude se jednat o vrstvy okresů, obcí polygonů, obcí bodů, silnic a vodních toků. Poté ještě program potřebuje znát názvy některých atributů, a to atribut s názvem kraje, atribut bodové vrstvy s názvem obce a atribut polygonové vrstvy s počtem obyvatel obce.

Použité funkce knihovny QGIS

Manipulace s vrstvami a jejich editace

```
layer = QgsVectorLayer(shapefile, "layer_name", "ogr")
```

Vytvoření objektu vrstvy. Parametry: shapefile, název vrstvy, „ogr“ – knihovna vektorových formátů zahrnující shapefile.

```
QgsProject.instance().addMapLayer(layer)
```

Přidání vrstvy do QGIS projektu.

```
layer_group = QgsProject.instance()
layer_group.removeMapLayer(layer_group.mapLayersByName("layer_name")[0].id())
```

Odstranění vrstvy z QGIS projektu podle jejího jména.

```
layer_provider = layer.dataProvider()
layer_provider.addAttributes([QgsField("new_field_name", QVariant.Double)])
layer.updateFields()
```

Přidání nového atributu datového typu *double* do existující objektu vrstvy.

```
features = layer.getFeatures()
```

Vytvoření iterovatelného objektu přes jednotlivé řádky atributové tabulky vrstvy.

```
for f in features:
    area = f.geometry().area()/10**6
    new_value = f["population_attibute_name"]/area
    with edit(layer):
        f["new_field_name"] = new_value
        obce_lyr.updateFeature(f)
```

Výpočet hustoty zalidnění (new_value) pro každou obec (objekt f) a uložení této hodnoty do nového atributu.

```
layer.select(index)
```

Výběr prvku vrstvy podle jeho pořadí v atributové tabulce.

```
QgsVectorFileWriter.writeAsVectorFormat(layer, output_path, "UTF-8", layer.crs(), "ESRI
Shapefile", onlySelected = True)
```

Uložení vybrané části vrstvy jako shapefile. Parametry: vrstva, úložiště, kódování, souřadnicový systém, vektorový driver.

```
layer.removeSelection()
```

Odstranění aktuálního výběru.

Processingové funkce

```
processing.run("native:clip", {'INPUT' : input_layer, 'OUTPUT' : output_path, 'OVERLAY' : overlay_layer})
```

Oříznutí vrsty (input_layer) vrstvou (overaly_layer) a uložení do souboru (output_path).

Symbologie

```
min_value = layer.minimumValue(layer.fields().indexFromName("field_name"))
```

Výpočet minimální hodnoty atributu.

```
breaks = QgsGraduatedSymbolRenderer().calcEqualIntervalBreaks(min_value, max_value, 5, False, 0, False)
```

Vytvoření seznamu horních hranic intervalů podle minimální (min_value) a maximální hodnoty (max_value) atributu a počtu tříd legendy (5) metodou stejně širokých intervalů.

```
symbol = QgsSymbol.defaultSymbol(layer.geometryType())
```

Vytvoření objektu pro určitý typ geometrie.

```
symbol.setColor(QtGui.QColor("color_name"))
symbol.setOpacity(1)
```

Nastavení barvy např. pomocí HTML anotace („color name“) a průhlednosti symbolu.

```
range = QgsRendererRange(minVal, maxVal, symbol, lab)
```

Vytvoření objektu intervalu pomocí jeho hraničních hodnot (minVal, maxVal), objektu symbol a anotace intervalu v legendě.

```
groupRenderer = QgsGraduatedSymbolRenderer('', range_list)
```

Vytvoření objektu pomocí seznamu objektů jednotlivých intervalů.

```
groupRenderer.setClassAttribute("field_name")
layer.setRenderer(groupRenderer)
```

Aplikace odstupňovaného symbolu přidáním reference atributu.

```
symbol = QgsLineSymbol.createSimple({'color': 'black', "width": 0.38})
layer.renderer().setSymbol(symbol)
```

Aplikace jednoduchého liniového symbolu nastavením nové barvy a šířky linie.

Vytvoření popisu

```
settings = QgsPalLayerSettings()
```

Vytvoření objektu pro nastavení popisu v mapě.

```
text_format = QgsTextFormat()
```

Vytvoření objektu pro nastavení formátu textu.

```
text_format.setFont(QFont("Arial", 9))
text_format.setSize(9)
```

Nastavení velikosti a fontu písma textového objektu.

```
buffer_settings = QgsTextBufferSettings()
```

Vytvoření objektu pro nastavení bufferu kolem popisků.

```
buffer_settings.setEnabled(True)
buffer_settings.setSize(0.7)
buffer_settings.setColor(QColor("white"))
text_format.setBuffer(buffer_settings)
```

Zapnutí bufferu, nastavení jeho velikosti a barvy a přidání do objektu textového formátu.

```
settings.setFormat(text_format)
settings.fieldName = city_name_field
settings.placement = 2
settings.enabled = True
layer_settings = QgsVectorLayerSimpleLabeling(settings)
layer.setLabelsEnabled(True)
layer.setLabeling(layer_settings)
```

Sled příkazů, jimiž dojde k vytvoření již nastaveného popisu.

Vytvoření layout view, přidání jednotlivých prvků a export

```
project = QgsProject.instance()
manager = project.layoutManager()
layout = QgsPrintLayout(project)
layout.initializeDefaults()
layoutName = "layout_name"
layout.setName(layoutName)
```

Sled příkazů, který vytvoří layout view.

```
pc = layout.pageCollection()
pc.pages()[0].setPageSize('A4', QgsLayoutItemPage.Orientation.Portrait)
manager.addLayout(layout)
```

Nastavení velikost listu na A4 a orientaci na výšku.

```
ms = QgsMapSettings()
ms.setLayers(list_of_layers)
rect = QgsRectangle(ms.fullExtent())
rect.scale(1)
ms.setExtent(rect)
map.setExtent(rect)
map.setBackgroundColor(QColor(255,255,255,0))
layout.addLayoutItem(map)
```

Sled příkazů, který přidá mapu do layout view a nastaví jí bílé pozadí.

```
layout_feature.attemptMove(QgsLayoutPoint(10, 10, QgsUnitTypes.LayoutMillimeters))
```

Umístění layout prvku pomocí milimetrové vzdálenosti od levého horního rohu stránky.

```
layout_feature.attemptResize(QgsLayoutSize(190, 250, QgsUnitTypes.LayoutMillimeters))
```

Určení velikosti prvku v layout view pomocí jeho rozměrů v milimetrech.

```
legend = QgsLayoutItemLegend(layout)
legend.setTitle("legend_title")
layerTree = QgsLayerTree()
layerTree.addLayer(layer)
legend.model().setRootGroup(layerTree)
layout.addLayoutItem(legend)
```

Sled příkazů, který přidá legendu vrstvy do layout view.

```
scalebar = QgsLayoutItemScaleBar(layout)
scalebar.setStyle("Line Ticks Up")
scalebar.setUnits(QgsUnitTypes.DistanceKilometers)
scalebar.setNumberOfSegments(2)
scalebar.setNumberOfSegmentsLeft(0)
scalebar.setUnitsPerSegment(5)
scalebar.setLinkedMap(map)
scalebar.setUnitLabel("km")
scalebar.setFont(QFont("Arial", 9))
scalebar.update()
```

```
layout.addLayoutItem(scalebar)
```

Sled příkazů, který přidá grafické měřítko do layout view.

```
title = QgsLayoutItemLabel(layout)
title.setText(title_text)
title.setFont(QFont("Arial", 24))
title.adjustSizeToText()
layout.addLayoutItem(title)
```

Sled příkazů, který přidá nadpis do layout view.

```
exporter = QgsLayoutExporter(layout)
exporter.exportToPdf(output_path, QgsLayoutExporter.PdfExportSettings())
```

Export pdf souboru do output_path.

Zjednodušený pseudokód automatické tvorby atlasu

Načti polygonovou vrstvu obcí.

Přidej nový atribut pro výpočet hustoty zalidnění polygonové vrstvě obcí.

Pro každou obec vypočítej hustotu zalidnění a aktualizuj atribut.

Načti všechny zbývající vrstvy: vrstvu okresů, silnic, vodních toků a bodovou vrstvu obcí.

Opakuj tolikrát, kolik je počet okresů:

- Vyber nový okres.
- Ořízni podle něho polygonovou vrstvu obcí.
- Získej minimální a maximální hodnotu hustoty zalidnění z oříznutých obcí.
- Vypočítej hraniční hodnoty pro 5 intervalů metodou Equal Interval Breaks
- Přiřaď každému intervalu barvu symbolu a jeho popis v legendě.
- Aplikuj odstupňovanou symbologii na vrstvu obcí.
- Ořízni i zbylé vrstvy: vrstvu silnic, vodních toků a bodovou vrstvu obcí.
- Nastav vrstvám vhodnou jednoduchou symbologii.
- U bodové vrstvy obcí přidej popis včetně bílého bufferu.
- Vytvoř layout view s orientací na výšku o velikosti formátu A4.
- Přidej mapový rám se všemi vrstvami, jimž byla upravena symbologie.
- Přidej legendu všech zobrazených mapových vrstev.
- Přidej grafické měřítko.
- Přidej nadpis mapy, který se shoduje s názvem okresu.
- Proveď export do pdf s unikátním názvem.
- Smaž všechny aktivní vrstvy.

Přidej všechna vytvořená pdf do jednoho a vytvoř tak atlas.

Porovnání PyQGIS a ArcPy

Na první pohled je znatelným rozdílem způsob procedurálního programování, jímž jsou obě knihovny sestaveny. ArcPy je vytvořena strukturovaně, zatímco PyQGIS objektově orientovaně. Nejpodstatnějším rozdílem je zcela určitě rozsah možností, které obě knihovny nabízejí. ArcPy je oproti PyQGIS značně omezená, neumožňuje funkce uživatelského rozhraní viz např. tvorbu symbologie. S PyQGIS má uživatel v podstatě neomezené možnosti, s tím se pojí rozsáhlá dokumentace s často ale chybějícím či neúplným vysvětlením jednotlivých metod. Pro nezkušeného uživatele je proto prvotní seznámení velmi obtížné. Na druhé straně ArcPy má velmi přehlednou dokumentaci. Implementaci v Pythonu vždy přechází vysvětlení funkce v desktopové aplikaci, a tak si začátečník může oba způsoby jednoduše propojit. Dokumentace vždy uspokojivě vysvětluje vstupní a výstupní parametry, a tak při jejím prohlížení nevznikají žádné nejasnosti. Na závěr dokumentace poskytuje několik příkladů kódu k danému problému, což velmi usnadňuje pochopení předchozích získaných informací. Zdokonalení dokumentace PyQGIS inspirováním se od ArcPy by jistě zlepšilo její uživatelskou přívětivost.

Vzhledem k možnostem obou knihoven se způsoby automatizované tvorby atlasu značně liší. V případě využití ArcPy musí celému procesu předcházet vytvoření mapy v desktopové aplikaci, která poté slouží jako šablona. Pro každé další území se v editoru zdrojového kódu pouze reklasifikuje symbologie. Tvorba atlasu se tedy neobejde bez manuální práce. Díky neomezeným možnostem PyQGIS se jakékoliv práci v SW vyhýbáme. Symbologii i jednotlivé součásti mapového listu lze vytvořením jejich objektů nastavit podle vlastního přání.

Možná vylepšení

Drobnou nedokonalostí navrženého postupu je vytváření samostatných pdf souborů pro každý okres, které jsou poté pomocí knihovny *fitz* sloučeny do jednoho. Takový způsob sice není výpočetně náročný, pravděpodobně ale existuje možnost, jak celý atlas sestavit pouze za pomoci PyQGIS. Dokumentace hovoří o objektu QgsLayoutAtlas, který by měl pro daný účel sloužit. Nicméně vzhledem k již zmíněné špatné uživatelské přívětivosti se nepodařilo možnosti tohoto objektu dopodrobna prozkoumat.

Dále by po lepším porozumění a hlubším proniknutí do PyQGIS dokumentace šlo zapracovat na kartografických nedokonalostech. Nepodařilo se nastavit optimální měřítko mapy v závislosti na velikosti mapového rámu, a tak se do něj některé okresy nevešly celé. Hranice kategorií kartogramu by bylo lepší zvolit v závislosti na histogramu tak, aby do jedné kategorie nespadala většina hodnot. Žádoucí by bylo též přidání číselného měřítka.

Seznam zdrojů:

1. HAFEN, K. (2019): PyQGIS: Create and Print a Map Layout with Python. https://opensourceoptions.com/blog/pyqgis-create-and-print-a-map-layout-with-python/ (cit. 26. 1. 2022).
2. QGIS Documentation (2022): Documentation for QGIS 3.22. https://docs.qgis.org/3.22/en/docs/index.html (cit. 26. 1. 2022).

Seznam příloh:

1. Skripty: https://github.com/sklibad/PyQGIS_Atlas