

QGIS を用いたプラグインツールの作成手法について

エンジニアリング本部 国土基盤情報部 建設情報グループ

内田 照久

1. はじめに

総務省では、災害発生時にも行政情報の業務を継続するため、効率的で災害に強い電子自治体の実現に向けて取り組んでいる。自治体における地理情報システム(以降 GIS と記す)の整備状況は、平成 27 年 4 月 1 日現在、都道府県では 46 団体(97.9%)、市区町村では 1402 団体(80.5%)で整備済みである⁽¹⁾。都道府県では 1 団体を除いて全国で整備済であるが、市区町村では未だ 20%が未整備である。その主な原因は「財政状況」である。

一般的な有償型 GIS を導入するには、1ライセンスあたり、ソフトのライセンス費用が数十万円、ソフト年間保守料が数万円必要となる。さらに専用システム開発となると仕様によっては、数百万は必要であり、財政規模の小さな自治体には相当な負担になる。

一方、オープンソース GIS は有償型 GIS と比べると高度な解析機能では劣るが、一般的な操作するには十分な機能を備えている。また簡単なプラグインを作成することで複雑な処理を少ない操作にて実現することも可能である。最近では、過去に作成した GIS のリプレースにあたり、少しでも費用を抑えるため、オープンソース GIS にてシステム構築したいという要望も増えてきている。

上記の状況を踏まえて、本稿ではオープンソー

ス GIS を利用した簡単なプラグインツール開発の手法について紹介することとした。

2. 開発環境の整備

2.1 使用ソフトウェアについて

GIS はオープンソース GIS としては最も有名な QGIS を使用する。

開発言語は広く普及していて、記述や理解、維持、配布が簡単である Python にて行う。

統合開発環境は Eclipse を使用する。

2.2 インストール

(1) QGIS のインストール

QGIS のインストールは OSGeo4W サイト (https://trac.osgeo.org/osgeo4w/wiki/OSGeo4W_jp)よりダウンロードしてインストールを行う。

通常「C:¥OSGeo4W¥」フォルダにインストールされ、「C:¥OSGeo4W¥bin¥qgis.bat」を実行すると QGIS が起動する。

(2) Eclipse のインストール

Eclipse は pleiades 版(<http://mergedoc.osdn.jp/>)を使用する。また、Python にて開発を行うため、プラグインとして Pydev のインストールも行う。

3. プラグインツールの概要

3.1 ツールの基本機能

本稿では以下の2つのプラグインツールを作成した。その作成方法について紹介する。

(1) 情報表示機能

1つ目として専用ツールを作成するにあたり、よく見受けられる専用画面を使用した情報表示機能(以降:情報表示機能と記す)を作成した。

これは、図形を選択すると、関連する情報が専用画面に表示される機能である。



図1 情報表示機能画面イメージ

(2) 図形拡張機能

2つ目としては現時点では QGIS の基本機能にない図形編集の機能で、選択した図形の縦または横のどちらか一方方向のみにドラッグすると図形が合わせて拡張する機能(以降:図形拡張機能と記す)を作成した。

これは、選択したポリゴンの外接矩形の辺上の表示される点(□)を縦または横のどちらか一方方向のみにドラッグすると図形が合わせて拡張される機能である。

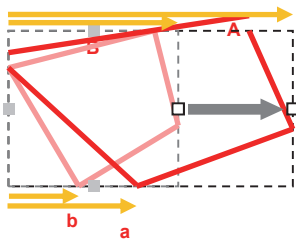


図2 図形拡張機能画面イメージ

4. プラグインツールの開発手法

4.1 プラグインモジュールの開発準備について

QGIS のプラグインは保存場所が以下の 2 カ所に決まっており、そのフォルダにソースを置くことでプラグインとして認識する。

- 共通プラグイン保存先
C:\OSGeo4W\apps\qgis\python\plugins
- ユーザープラグイン保存先
C:\Users\ユーザー名\qgis2\python\plugins

また、プラグインの雛形は「Plugin Builder」を使用して生成することができる。QGIS を起動後、プラグインメニューより「Plugin Builder」をクリックすると設定画面が表示される。メニューバーに「Plugin Builder」が表示されない場合は、「プラグインの管理とインストール」より新規に追加する。

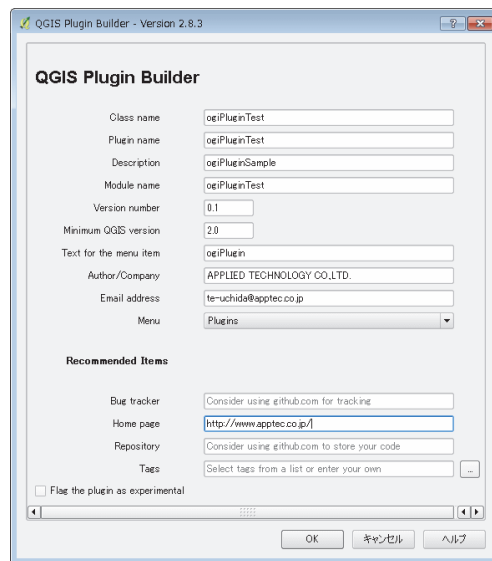
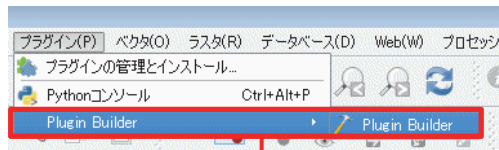


図3 Plugin Builder 設定画面

Plugin Builder 設定画面にて必要な項目を入力 (プラグイン名: ogiPluginTest と設定) すると、プラグイン保存場所にフォルダ (C:\OSGeo4W\apps\python\plugins\ogiPluginTest) とモジュール一式が生成される

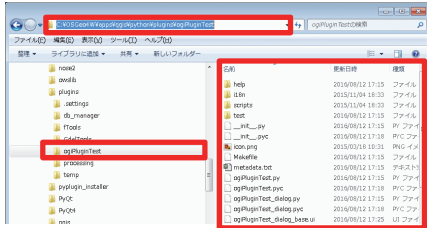


図 4 Plugin フォルダ

次にフォルダ内にある「resource.qrc」より「resource.py」を生成する。「C:\OSGeo4W\OSGeo4W.bat」を実行後、「ogiPluginTest」フォルダに移動し、「Make」と実行すると「resource.py」が作成される。

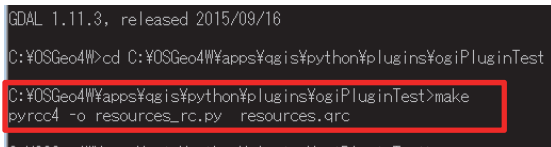


図 5 コマンドプロンプト画面

以上で開発を行う準備が整った。

4. 2 情報表示機能の作成

(1) 表示画面 (GUI) の作成

はじめに画面の作成を行う。Python では QT Designer というツールが用意されていて、それを使用して画面を作成する。「C:\OSGeo4W\bin\designer.exe」を実行すると QT Designer が起動する。先ほど作成したフォルダより、「ogiPluginTest_dialog_base.ui」を開く。「ogiPluginTest_dialog_base.ui」(以降:メニュー画面と記す)は2つの機能呼び出すためのボタンを配置する。続いて情報表示

を行うための新規 UI (ogiPlugin_InfoView.ui と命名、以降:情報表示画面と記す)を追加し、写真を表示するための Label コントロールと、属性情報を表示する TableWidget コントロールを配置する。

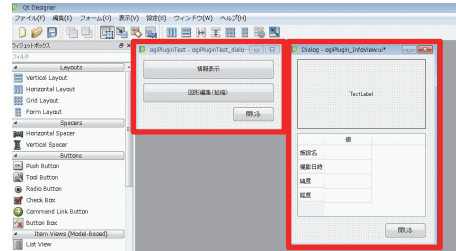


図 6 QT Designer

新規に作成した情報表示画面は py 形式に変換する。「C:\OSGeo4W\OSGeo4W.bat」を実行後、「ogiPluginTest」フォルダに移動し、「pyuic4 ogiPlugin_InfoView.ui -o ogiPlugin_InfoView.py」と実行すると、「ogiPlugin_InfoView.py」ファイルが作成される。

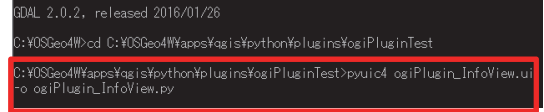


図 7 コマンドプロンプト画面

(2) コードの作成

Eclipse を起動し、「ogiPluginTest」フォルダ内の「ogiPluginTest_dialog.py」を表示し、必要なモジュールをインポートする。

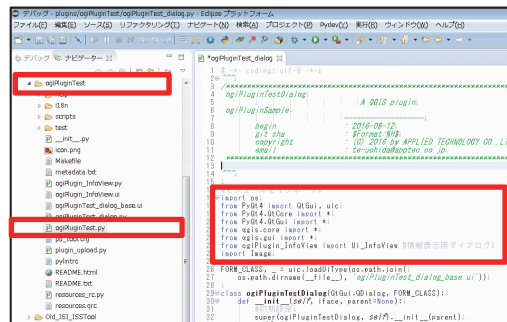


図 7 Eclipse

メイン画面に配置した2つのボタン押下時の関数を生成し、呼び出すよう設定する。

```
● 情報表示ボタンクリック時の関数
def ButtonInfoView_Set(self):
● 図形編集ボタンクリック時の関数
def ZukeiKakuShuku_Set (self):
● ボタン押下時に呼び出す方法
self.ButtonMenu1.clicked.connect(self.ButtonInfoView_Set)
```

次に地図上の図形情報を取得するため、「QgsMapTool」クラスを継承したクラスを作成する⁽²⁾。作成したクラス内に選択した図形情報を取得するための「canvasPressEvent（マウスクリック）」関数を生成し、選択した図形情報を取得するコードを記述する。

< canvasPressEvent 関数 >

- ① クリックした座標(ピクセル値)を取得し、地理座標に変換
- ② 座標と重なる図形(空間検索)を取得

```
● 図形選択用クラスとマウスクリック関数
class ogiSelectTool(QgsMapTool):
    def canvasPressEvent(self, event):
● クリック座標を地理座標に変換する方法
self.canvas.getCoordinateTransform().
toMapCoordinates(event.pos())
● 空間検索の方法
QgsFeatureRequest().setFilterRect(検索範囲)
```

続いてダイアログ表示用のクラスを生成し、選択した図形情報を表示するコードを生成した。

< ダイアログ表示用のクラス >

- ① 選択図形より属性情報を取得
- ② 取得した属性情報を tableWidget に表示
- ③ Label に画像を設定

最後に「canvasPressEvent」関数内にダイアログ

```
● ダイアログ表示用のクラス
class InfoViewDialog(QDialog, Ui_InfoView):
● 属性情報の取得方法
feature.attributes()[フィールド番号]
● tableWidget への表示方法
self.ui.tableWidget.setItem(行番号,列番号
QTableWidgetItem(表示する文字列))
● Label に画像を設定する方法
self.pixmap = QPixmap(ファイルパス)
self.ui.label.setPixmap(self.pixmap)
```

表示用クラスを呼び出すよう設定することで、選択した図形の情報が表示される。以上で情報表示機能のコード生成が完了する。今回は情報を画面上に表示するだけであったが、Excel などの帳票に出力する機能も作成可能である。

4.3 図形拡張機能の作成

(1) コードの作成

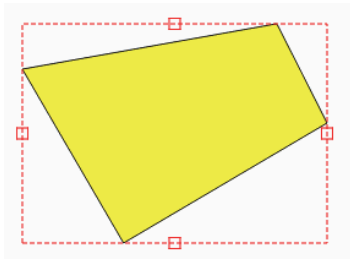
指定した図形を拡張するため新しく「QgsMapTool」クラスを作成する⁽²⁾。作成したクラスは選択した図形に対して拡張を行うため、「情報表示機能」時に作成した図形選択クラス内の「canvasPressEvent」関数より呼び出すようにした。

```
● 図形拡張を行うクラス
class ZukeiKakuShukuSet (QgsMapTool):
● canvasPressEvent 内に以下を記述する
tool = ZukeiKakuShukuSet (self.canvas)
self.iface.mapCanvas().setMapTool(tool)
```

次に選択した図形の外接矩形とその辺上に表示される点(□)を表示するコードを生成した。表示図形は、一時的な図形であるため外接矩形は「QgsRubberBand」クラスを点(□)は「QgsVertexMarker」クラスを使用した⁽²⁾。

- 外接矩形を行う `QgsRubberBand` クラス
`self.rubberBand.reset(Qgis.Polygon)`
`self.rubberBand.setLineStyle(線種を指定)`
`self.rubberBand.setColor(色を指定)`
`self.rubberBand.addPoint(左下座標値, False)`
`self.rubberBand.addPoint(左上座標値, False)`
`self.rubberBand.addPoint(右上座標値, False)`
`self.rubberBand.addPoint(右下座標値, True)`
`self.rubberBand.show()`
- 点 (□) を表示す `QgsVertexMarker` クラス
`vMarker.setIconType(QgsVertexMarker.ICON_BOX)`
`vMarker.setColor(色を指定)`
`vMarker.setCenter(点の座標値)`

以下のような外接矩形と点 (□) が表示される。



続いて図形拡張クラス内に「`canvasPressEvent`」関数を生成し、選択した点 (□) の座標を取得し共通変数に保存しておく。移動距離に合わせて図形を変更することから「`canvasMoveEvent`」関数を生成し、図形を変更するコードを生成した。

< `canvasMoveEvent` 関数 >

- ① 移動座標を取得
- ② 選択した点 (□) との移動距離を算出
- ③ 選択図形の各座標に移動距離を加味する。
 ※横方向に移動した場合は各 X 座標を縦方向に移動した場合は各 Y 座標のみを変更する。
- ④ 外接矩形及び点 (□) も移動距離に合わせて再作成を行う

以上で図形拡張機能のコード生成が完了した。

- 選択座標を取得し、共通変数に保存
`def canvasPressEvent(self, event):`
 選択座標 = `self.toMapCoordinates(event.pos())`
- 移動座標を取得
`def canvasMoveEvent(self, event):`
 移動座標 = `self.toMapCoordinates(event.pos())`
- 図形の変更方法 (X 座標変更の時)
 移動距離 = 移動座標.x - 選択座標.x
 変更座標値 = `QgsPoint(x + 移動距離, y)`
 `pr = Layer.dataProvider()`
 `pr.changeGeometryValues({feature.id(): 変更座標値})`
 `Layer.commitChanges()`
- 外接矩形及び点 (□) の変更
 外接矩形は `QgsRubberBand` クラス、点 (□) は `QgsVertexMarker` クラスを用いて、変更座標値にて再作成を行う

4. 4 インストール及び動作確認

(1) インストール

QGIS を起動後、プラグインメニューより「プラグインの管理とインストール」を選択するとプラグイン設定画面が表示される。「`ogiPluginTest`」を選択すると、プラグインメニュー及びツールバーアイコンに「`ogiPluginTest`」が追加される。

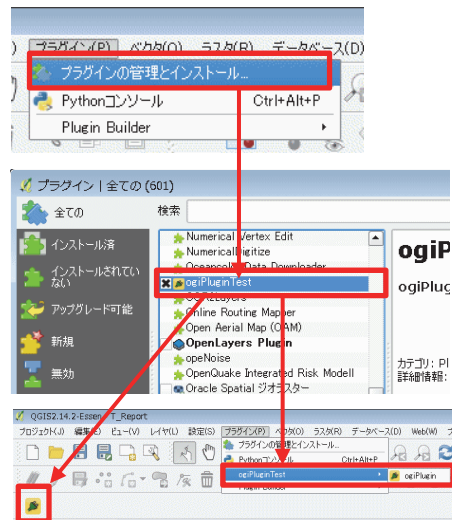
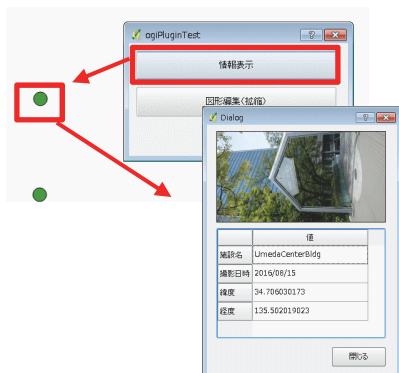


図 8 プラグインメニュー及びプラグイン設定画面

(2) 動作確認

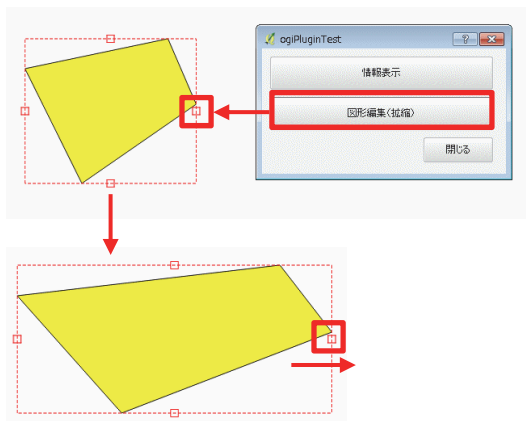
① 情報表示機能の動作確認

メニューまたはアイコンより「ogiPluginTest」を選択すると、メニュー画面が起動する。「情報表示」ボタンを選択後、情報表示したい図形を選択すると、情報表示画面が正しく表示されたことを確認した。



② 図形拡張機能の動作確認

メニュー画面より「図形編集(拡張)」ボタンを選択後、編集したいポリゴン図形を選択すると、外接矩形とその辺上に点(□)が表示される。移動したい点を指定後、拡張したい方向にマウスを移動すると図形が変更されることを確認した。



5. まとめ

最後に有償型 GIS とオープンソース GIS の開発における、それぞれの長所と短所を述べる。

- ① 有償型 GIS の長所
有償ではあるが、開発サポートが充実している。
- ② サンプルソースが充実している。
- ③ 有償型 GIS の短所
初期導入費用がかかる。
開発サポートを受けるためには別途、年間サポート料が発生する。
- ④ オープンソース GIS の長所
初期導入費用、年間サポート料が無償。他のユーザーが作成したツールのソースが公開されているため、参考にできる。
- ⑤ オープンソース GIS の短所
不明箇所の調査に時間を要する。
サンプルソースが体系だってまとめられていないため、検索に時間を要する。

※なお、以前は英語の解説サイトがほとんどで、日本語の解説サイトはあまりなかったが、最近では日本語のサイトもいくつか見受けられるようになり、今回作成した2つの機能は、その日本語の解説サイトを参考⁽²⁾⁽³⁾⁽⁴⁾にすることで、ほとんどがカバーできた。

当社では今後、技術をさらに磨きオープンソース GIS 分野のツール開発にも積極的に取り組んでいく予定である。

<参考文献>

- 1) 「地方自治情報管理概要～電子自治体の推進状況(平成27年4月1日現在)」(総務省)
http://www.soumu.go.jp/denshijiti/060213_02.html
- 2) 「月の杜工房 - QGIS 豆知識」
<http://mf-atelier.sakura.ne.jp/mf-atelier/modules/tips/software/qgis/>
- 3) 「だら\$らいぶらりい~/QGIS はじめてのプラグイン作成」
<http://www.arenanine.gr.jp/~banjo/QGIS/plugin/index.html>
- 4) 「PyQGIS 開発者用 Cookbook」
http://docs.qgis.org/2.2/ja/docs/pyqgis_developer_cookbook/