

Qt Scrypt Demo

5.9.2024



QtScript

Introduction

- **What is it**
 - Research project trying to create **dynamic bindings** for Python
- **Use case**
 - Make an application scriptable (think Visual Basic for Excel)
 - **Existing example:** Python scripting for Autodesk Maya using Qt for Python's binding generator
 - **Drawback:** Using the bindings generator is a complicated process. Can this be simplified (given also that in Qt 5, there was *Qt Script* providing JavaScript bindings)?

QtScript

Introduction

- **Design goals**

- Create dynamic bindings of QObjects and other classes for Python
- Extendability, a way to convert arbitrary types to Python
- Should not require a complicated generation/build process

- **History**

- Started out as a Hackathon project by Cristián to provide an equivalent of Qt 5's *Qt Script*

Current State

- Dynamic types in Python achieved by registering custom types with special implementations of *getattr/setattr* slots
- Use Qt metatype system for the conversion from/to *PyObject*
- Signal support
- Currently in internal repo, pending final naming discussion

Client code

- `QScriptClassType`, `QScriptNamespaceType` encapsulating `PyObject`
- Creating `QObject` type automatically adds properties and invocable functions from `QObject`
- Additional functions and constructors can be registered
- Register free classes like `QObject`; all functions must be registered
- Bind instance to name in module `__main__`
- Register free functions

Demo Script

Future plans

- Use it in Qt Creator as a way of scripting things for testing
- Open questions, for example:
 - To what extent is functionality from shiboken needed (reference manipulation of function arguments)?
 - To what extent expose Python APIs?
 - Things might change with reflection in C++ 26