QT DEMO EXPANSION

Widget Town (WT) 1.0

White Paper

Published: March, 2012

Team Exception

University of Utah, CS3505 Project 4

By John Wells and Joel Hough

Abstract:

Explanation of the capabilities and limitations of the QT embedded dialogs demo.

Contents

Illustration Index

Introduction: What is WT?

Widget Town (referred to as WT in the rest of the document) is a simple application created to enhance QT's embedded dialog feature set. It consists of a town created using a tile engine, several NPC characters, and a playable character that can move around the world. WT also has several embedded widgets that run inside of the main window, including a simple QLabel, a QCalendarWidget, and an additional instance of WT.

Feature set

Original features of Embedded Dialogs

EmbeddedDialogs (ED) is a simple demo to outline the functionality of QT's ability to embed widgets in a GraphicsScene. Originally, this consisted of a simple window with a custom form (EmbeddedDialogs.ui) that became larger upon mouse over. The dialog itself contained a few simple options to demonstrate the flexibility of the embedding (Font style, window size, etc.)

Originally Planned Extensions

The original implementation involved adding several QT Advanced Widgets and X11Embed frames (eg, Video Player, Emacs, Web Browser) into the window, and place them in such a way that it appeared to be a desk. When an object was focused on, the window would enlarge and be a fully functional sub program. Unfortunately this implementation was scrubbed, mainly due to lack of compatibility on the CADE machines. There were several QT packages (most notably, Phonon) missing from the version we were using. This prevented us from using a few key advanced widgets that we had planned on using.

Figure 1: WT Calendar Widget

Accomplishments

WT incorporates several QT built in functionalities, many of which focus on the flexibility of the GraphicsScene and GraphicsView system.

- **Tile Engine:** This is the background 'world' that the characters and widgets exist in. The engine utilizes QT's Xml tools, the QgraphicsScene system, and QGraphicsItems to render nearly 11,000 tiles in real time.
- **Characters:** Movable characters in the game. Demonstrates timed tilesheet animation, mask based collision, and Z-ordering.
- **Player Characters:** A keyboard (WASD) controllable character in the game, demonstrating input event propagation in the GraphicsScene.
- **NPCs:** Simple AI that randomly walk around WT, demonstrating QGraphicsItems in class hierarchies.
- **QLabel:** A simple label on top of the banana tile that reads: "I'm a banana!", demonstrating simple widget embedding.
- **QCalendar:** A fully interactive calendar on the wall inside the house, demonstrating advanced widget embedding.
- **Additional TileScenes:** Upon clicking, this changes focus to the sub window inside WT, and places a new controllable character into a new town. This demonstrates multiple levels of advanced, custom widget embedding (The Inception effect, pictured below).

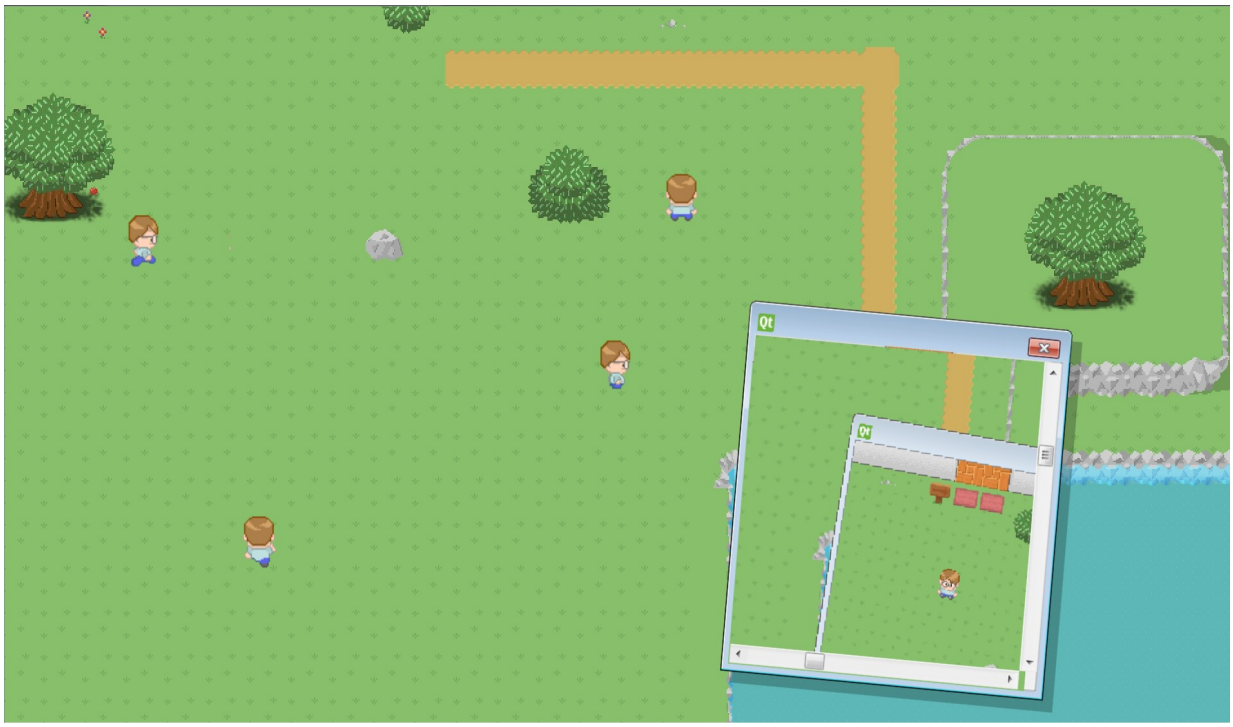

Figure 2: Widget Town "Inception Effect"

Time Usage Report

Time usage was excellent. Most of the time was spent learning Qt will playing with code. The implementation was quick and effective. The only real time sinks were due to CADE issues.

Original Implementation:

Approximately two hours were spent learning the functionality of the EmbeddedDialogs project and another eight were spent trying to find widgets that would load on the CADE lab QT configurations and could be embedded.

Tile Engine

The Tile Engine was built in two major iterations. The initial iteration took nine hours, most of which was spent reading documentation and experimenting with the less documented features to determine how best to structure the engine. The result was a rough engine with minimal features:

- Background tiles (no Z ordering)
- Movable character (no animation, no collisions, hardcoded single player input focus).

The second iteration was a major refactor that restructured the code to improve extensibility, performance, and general sanity. The iteration took six hours. The engine included:

- Tiles (Z ordering, collision layer)
- Player object (animated, collisions, scene focus based input)
- Embedded widgets (label and calendar)

Final Implementation and Polishing

A period of about five hours was dedicated to throwing features in. The engine was very extendable from the earlier refactor and as such gained most of its advanced featureset during this polishing period:

- Character class hierarchy, allowing Player and NPC to share animation and collision code
- Embedded GraphicsView and TileScene widgets, for Inception fans

Summary

Widget Town:

Despite the limited functionality of the CADE QT configurations, we were able to implement a unique modification to the QT demo. These extensions give the user a brief overview of what else is possible using QT's powerful GUIs. Definitely deserves an A.