2016

**План создания программы**

**“Dark Jedi’s JA Unlimited Table”**

(и черновик)

от Dark Jedi

Содержание

1. Описание программы и ее представление ее дизайна.
2. План проектирования.
   1. Создание виджета окна.
      1. Установка цвета фона.
      2. Добавление Layot’ов.
   2. Создание виджета Group Box.
   3. Создание виджета таблицы.
   4. Создание виджета ScrollBar.
   5. Создание виджета ScrollArea.
      1. Создание виджета JWidget в помощь к виджету ScrollArea.
   6. Создание Label’a - переключателя цвета команды.
   7. Создание истории копирования скинов в буфер обмена.
   8. Прикрепления всего к виджету окна.
   9. Настройка стилей виджетов.
   10. Связывание всех виджетов друг с другом.
3. Журнал.

Описание программы и ее представление дизайна

**Основное описание:** Программа “Dark Jedi’s JA Unlimited Skins Table” создана для упрощения поиска и установки скинов в игре Jedi Academy. Поскольку стандартно в игре установлено ограничение на количество скинов в таблице, эта программа будет загружать скины в бесконечную таблицу из pk3 файлов. Имена скинов будут в большинстве случаев будут соотвествовать именам папок скинов, в которых они расположены внтури pk3 файла, а иконками будут в большинстве картинки с именами icon\_default,

icon\_red, icon\_blue.

Пользователь выбирает иконку и у него автоматически копируется в буфер обмена команда

/model имя\_скина.

**Дизайн:** Ниже представлено два изображения: первое - обычный дизайн, второе - дизайн элементов с событием hover.





. . .

**JWidget**

1. Установка фона.

2. Установка ширины границы.

2.1. Перенос всех дочерних виджетов на другие координаты в соотвествии с шириной границы и calculateContentsRect().

2.1.1. В childEvent()

2.1.2. При изменении ширины границы: Сдвигаем все дочерние виджеты на новую точку.

А - точка contentsRect() B - точка дочернего виджета

B old + A new - A old = B new.

3. Установка цвета границы, при этом перекрывая дочерние виджеты.

3.1. Создание нового виджета - класса границы.

3.2. Размещение класса - границы в JWidget.

3.3. Создание функции запрещающей/разрешающей поднятие класс - границу над другими дочерними виджетами.

3.4. Создание функций управления цветом границы.

3.5. Зарисовка границы цветом.

**Ошибка #1**: Класс - граница не появлялся.

**Решение**: У него не было размеров. В JWidget::resizeEvent() установил размеры класса - границы по размерам JWidget’a.

3.6. Поднятие класса - границы над дочерними в соответствии со значением запрещающей/разрешающей поднятие класс - границу функцией.

**Ошибка #2:** Приложение, сразу же при влючении, падает.

**Решение:** Поставил к условию “разрешено ли поднятие” условие “является ли ребенком сам класс - граница”. Все очень бредово и ненормально, но работает.

**Итог:** Все готово.

**Возможные нюансы в будущем:** Кажется, размеры дочернего виджета расчитываются не правильно.

**Важные особенности:**

1. При наследовании, при переопределении resizeEvent(), нужно вызывать метод JWidget::resizeEvent() для нормального функционирования границы.

**JScrollArea**

1. Создание основы класса.

**Предупреждение в выводе приложения #1:** “*setGeometryDp: Unable to set geometry 100x30+480+190 on QWidgetWindow/'JWidgetClassWindow'. Resulting geometry: 116x30+480+190 (frame: 8, 30, 8, 8, custom margin: 0, 0, 0, 0, minimum size: 0x0, maximum size: 16777215x16777215)*”.

**Предположение #1:** Возможно это из-за того что в JWidget нет sizeHint().

**Решение:** Добавить в класс метод sizeHint() с установленными в нем значениями и установить setSizePolicy() в конструкторе.

2. Автоматическая установка JScrollBar’ов.

2.1. Установка скроллбаров в виджете.

2.2. Настройка нормальных размеров.

2.3. Перемещение скроллбаров на свои места.

2.3.1. Создание функций настроек положения скроллбаров.

2.5. Добавление функций настроек scrollBarPolicy.

2.6. Применение функций настроек scrollBarPolicy.

2.7. Настройка нормальных размеров слайдеров.

2.7.1. Установка minimum значений.

2.7.2. Установка maximum значений.

**Ошибка #3:** В событии showEvent размеры слайдеров вычисляются неправильно, но при событии resizeEvent() все нормально.

**Причина:** Вызов функций setScrollBarPolicy() после вызова show().

**Решение:** Установка setMinimum() & setMaximum() в методах setScrollBarPolicy().

**Нюанс:** Общая функция updateSettings() почему-то неработает.

3. Установка метода setWidget().

**Ошибка #4:** При установке виджета в setWidget либо граница устаналивается поверх виджета, но и поверх скроллбаров, либо скроллбары устанавливаются поверх границы и границы под виджетом.

void JScrollArea::setWidget(QWidget \*wgt) {

**widget\_** = wgt;

allowRaisingBorder(true);

wgt->setParent(this);

allowRaisingBorder(false); // Строка #1.

}

В строке #1 запрет на поднятие границы каким-то образом действует на предыдущее значение, и, не сотря на разрешение поднять границу, указанное перед setParent(), виджет остается под границей.

**Предположение #1:** Возможно, это из-за самой функции childEvent() в JWidget.

**Решение #1:** Использование метода stackUnder():

void JScrollArea::setWidget(QWidget \*wgt) {

**widget\_** = wgt;

wgt->setParent(this);

wgt->stackUnder(**borderClass\_**);

}

**Решение #2:** Установить в классе JWidget, при создании класса-границы применить к классу-границе метод *setAttribute(Qt::WA\_TransparentForMouseEvents).*

**Ошибка устранена решением #2.**

**Ошибка #5:** При появлении одного из скроллбара по надобности, вначале на одном пикселе, другой скроллбар не менял своих размеров, как нужно.

**Решение:** Добавить в функции переключения видимости скроллбаров, после ихних переключений, обновление размеров противоположных скроллбаров.

**Ошибка #6:** Неправильный расчет появления скроллбаров по надобности.

**Причина:** Расчет происходил за счет нижних точек виджета, который скроллится, и внутренней облати ScrollArea (Что является неправильным).

**Решение:** Изменение расчета, за счет размеров виджета, который скроллится, и внутренней области ScrollArea.

4. Установка cornerWidget.

4.1. Автоматическая установка.

4.1.1. Установка переключателя показом.

4.1.1. Установка размеров.

4.2.2. Установка положения.

4.2. Пользовательская установка.

5. Связывание изменений положения слайдеров с изменениями положениями виджета, который скроллится.

**Итог:** Все готово.

**Предупреждение #1:** *overriding recipe for target 'debug/moc\_JHScrollBar.cpp*

**Предупреждение #2:** *old**recipe for target 'debug/moc\_JHScrollBar.cpp*

**Причина предупреждений #1 & #2:** В файле .pro прописано два раза одни и те же файлы.

**Решение:** Убрать повторяющиеся файлы из списка.

**Автоматическое перемещение итемов на предыдущее место**

1. Добавление колонки при добавлении необходимой ширины.

2. Удаление колонки при удалении необходимой ширины.

**Ошибка 7:** При удалении колонки в модели, JTableView еще несколько пикселей неправильно расчитывает ширину таблицы.

**Итог:** Как-нибудь потом сделаю.

**Вытаскивание иконок и имен. Заполнение таблицы**

1. Вытаскивание иконок из pk3.

1.1. Попытаться сначала достать иконку из rar.

**Связывание всех виджетов друг с другом**

1. Добавить в JScrollBar отступы для слайдера.

2. Исправить ошибку конструктора копирования для горизонтального ScrollBar.

**Причина функциональных изменений:** В методах JScrollArea не было обновления связи QObject::connect между скроллбаром и скроллареа.

**Причина стилей:** В конструктор копирования скроллбар передавался как указатель, а нужно как ссылка.

**Ошибки решены.**

3. Исправить ошибку изменения значения слайдера изнутри JTableView.

*Причина:* JTableView, при изменении значения слайдера, связывается со своим слайдером (поскольку JTableView наследован от QScrollArea), а не к добавленному как его родитель JScrollArea.

**Ошибка решена** путем перенаследования JTableView от JScrollArea.

4. Исправить ошибку с размещением слайдера, когда скроллбар должен быть выключен как ScrollBarAsNeeded, но является включен как ScrollBarAlwaysOn (Когда размер скроллбара больше размера виджета, который скроллится). Слайдер в таком случае перемещается и остается в самом низу.

**Решение:** Устанавливать максимум скролла на 0, когда скроллбар должен быть выключен как ScrollBarAsNeeded.

5. Попробовать заставить, при увеличения достаточного места для вмещения итема таблицы, перемещать итемы на предыдущее место (как в проводнике Windows).