|  |  |  |
| --- | --- | --- |
| C:\Users\krzysiek\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\731AB5F4.tmp | Akademia Górniczo-Hutnicza  W Krakowie  Wydział EAIiIB | Autor:  Wasik Natalia |
| WDM | | |

**CZĘŚĆ I**

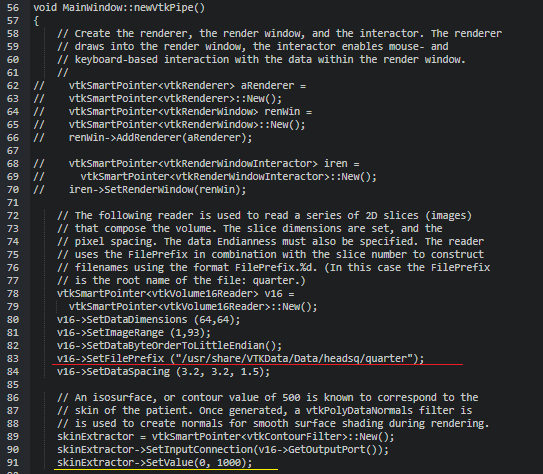
**1.Cel projektu**

Zadaniem projektu było zwizualizowanie modelu głowy w VTK przy pomocy Qt oraz dodanie możliwości wizualizacji różnych poziomów danych.  
Projekt powstał w Qt Creator 4.5.2., Qt 5.9.5 i zakłada użycie biblioteki VTK w wersji 7.1.

**2. Przebieg projektu**

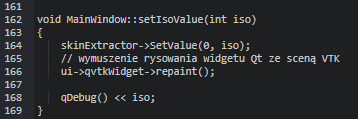
Do wykonania projektu wykorzystano gotowy kod. Suwak przedstawiony w okienku został wykorzystany do zmiany wizualizowanych danych.

Kod przedstawiający dane głowy został dodany w miejsce przedstawiający obiekt w wyskakującym oknie. Zostało to zaimplementowane na poniższej grafice w linjce podkreślonej na czerwono. Była to ścieżka do pliku gdzie dane były podzielone na podfoldery. Dodatkowo również zadany rozmiar zmiany wartości suwaka został dodany eksperymentalnie przez kilka prób, które pokazały w jakim zakresie przedstawiane bedą dane. Kod podkreślony na żółto.



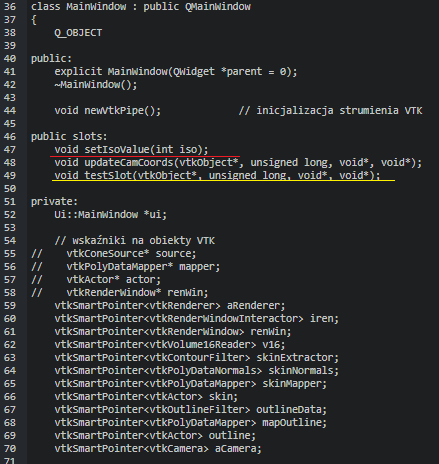
Rys.1. Kod źródłowy.

Następnie została edytowana funkcja umożliwiająca przejście przez wszystkie poziomy danych modelu, tak jak w Rys.3 poniżej.



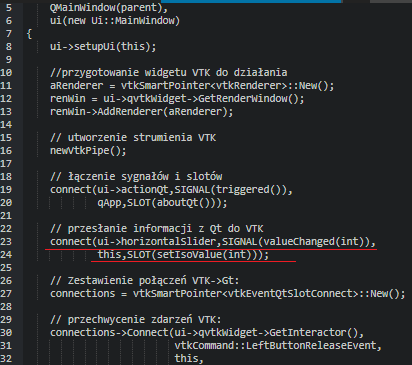
Rys.2. Kod źródłowy.

Funkcja została zadeklarowana globalnie i zaimplementowana w kodzie modelu. Przedstawiono w poniższej grafice w wersie podkreślonym na czerwono.



Rys.4. Kod źródłowy.

Ostatnim krokiem było połączenie akcji suwaka ze zmiana wyświetlanych danych. Do miejsca deklarującego akcję zmiany suwaka wstawiono funkcję zmiany przedstawianych danych.



Rys.5. Kod źródłowy.

**CZĘŚĆ II**

1. **Cel projektu**  
   Celem projektu było otworzenie w programie QT wcześniej zrobionego zadania, a następnie zmodyfikowanie przygotowanego kodu tak aby na końcu można było na interfejsie wybrać opcję, dzięki której użytkownik może zmienić kolor tła oraz czaszki (z danej palety kolorów).
2. **Przebieg projektu**Po załadowaniu pliku w interfejsie dodano opcje „Select Color”. Dodatkowo ustawiliśmy QAction na triggered() signal, dzięki któremy po przyciśnieciu przycisku pojawi nam się okno, którego pożądamy(paleta kolorów).   
   W ‘’mainwindow’’ wypisano trzylinijkowy kod, w którym używano między innymi nowej funkcji QColorDialog. Na załączonym kodzie jest również widoczne, iż do tego stworzono 3 elementową tablice(druga linijka) w której są załączone 3 barwy : czerwona, zielona oraz niebieska, które jak wiemy tworzą model przestrzeni barw „RGB”  
   Poglądowy kod:

**QColor color = QColorDialog::getColor();**

**if (color.isValid())**

**aRenderer->SetBackground(color.redF(), color.greenF(), color.blueF());**

Bardzo podobne czynności wykonaliśmy dla zmienienia koloru czaszki. W tym jednak etapie musieliśmy na początku sami dodać do kodu metode, która będzie zmieniać jej kolor i skasować pewnie niepotrzebne formuły.  
Finalnie część kodu wygladał tak:

**QColor color = QColorDialog::getColor();**

**if (color.isValid()) ;**

**skin->GetProperty()->SetColor(color.redF(), color.greenF(), color.blueF());**  
  
Zdjecie 1. Zmiana koloru

