

SPRAWOZDANIE

Zajęcia: Informatyczne systemy medyczne
Prowadzący: prof. dr hab. Vasyl Martsenyuk

Laboratorium 4

04.11.2020

Temat: Użycie standardu medycznych obrazów DICOM.

Wariant 2.

Radosław Siwiec
Informatyka II stopień,
stacjonarne,
2 semestr,
Gr.1B 1.

Zadanie. Opracować Player zkompresowanych plików DICOM według wariantu

1. Wykorzystana technologia

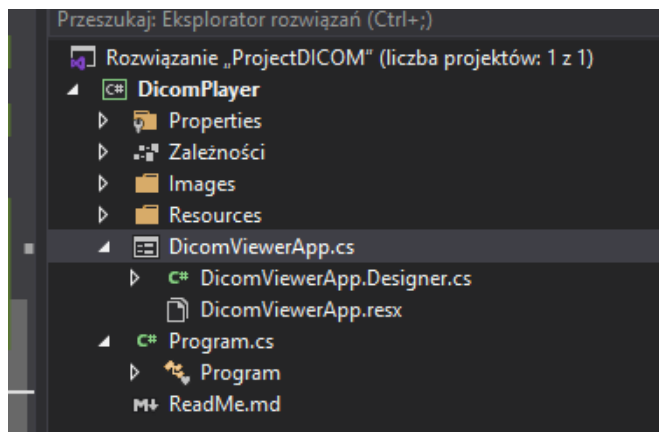
NET Core 3.1, aplikacja systemy Windows.

2. Wykorzystane biblioteki:

fo-dicom –prosty w implementacji zestaw funkcjonalności do manipulowania plikami DICOM.

3. Opis implementacji:

- Struktura projektu:



- Kod:

Importowanie plików DICOM:

```
private void BtnOpenFile_Click(object sender, EventArgs e)
{
    OpenFileDialog openFileDialog = new OpenFileDialog
    {
        InitialDirectory = ".",
        Filter = "DiCom Image (.dcm;.dic)|*.dcm;*.dic",
        RestoreDirectory = true,
        Multiselect = false
    };

    if (openFileDialog.ShowDialog() == DialogResult.OK)
    {
        if (openFileDialog.FileNames != null)
        {
            fileName = openFileDialog.FileName;
            pathDicomFiles = Directory.GetFiles(Path.GetDirectoryName(fileName));

            index = pathDicomFiles.ToList().IndexOf(fileName);
            ShowDicomImage(pathDicomFiles[index]);
        }
    }
}
```

Po wciśnięciu przycisku „Open DICOM Image”, otwiera się okno wyboru pliku typu: „.dcm”, „.dic”.

User wybiera 1 plik ale program przeszukuje cały folder i daje możliwość przeglądania wszystkich plików w katalogu z którego otwarto plik.

Wyświetlanie plików DICOM:

```
private void ShowDicomImage(string pathFile)
{
    dicomFile = DicomFile.Open(pathFile);
    dicomImage = new DicomImage(dicomFile.Dataset);
    ImageManager.SetImplementation(WinFormsImageManager.Instance);
    imageDisplay.Image = dicomImage.RenderImage().AsClonedBitmap();
    imageDisplay.SizeMode = PictureBoxSizeMode.Zoom;
    imageDisplay.Refresh();
}
```

Ta funkcja służy do odczytu plików DICOM – biblioteka fo-dicom.

Przewijanie plików:

```
private void NextFile()
{
    if (++index >= pathDicomFiles.Length)
        index = 0;

    ShowDicomImage(pathDicomFiles[index]);
}

private void PrevFile()
{
    if (--index < 0)
        index = pathDicomFiles.Length - 1;

    ShowDicomImage(pathDicomFiles[index]);
}
```

```
public DicomViewerApp()
{
    BackgroundImage = Properties.Resources.img;
    BackgroundImageLayout = ImageLayout.Stretch;

    InitializeComponent();

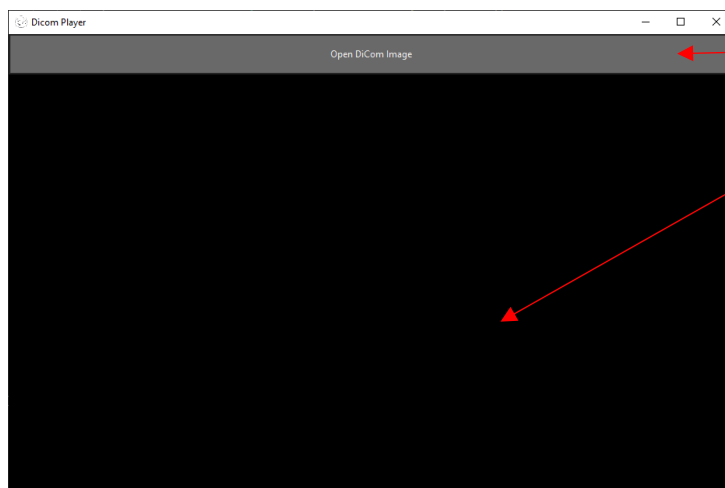
    imageDisplay.MouseWheel +=
        new MouseEventHandler(this.imageDisplay_MouseWheel);
}

private void imageDisplay_MouseWheel(object sender, MouseEventArgs e)
{
    if (pathDicomFiles == null)
        return;

    if (e.Delta < 0)
        NextFile();
    else
        PrevFile();
}
```

Funkcje NextFile i PrevFile służą do ładowania kolejnych obrazów z folderu. Przeglądanie się zapętla tj. jeśli zostanie osiągnięty ostatni plik to kolejnym plikiem będzie plik pierwszy i analogicznie w przeciwnym kierunku. Przewijanie podłączone jest pod zdarzenie MouseWheel czyli kółko myszy (scrol).

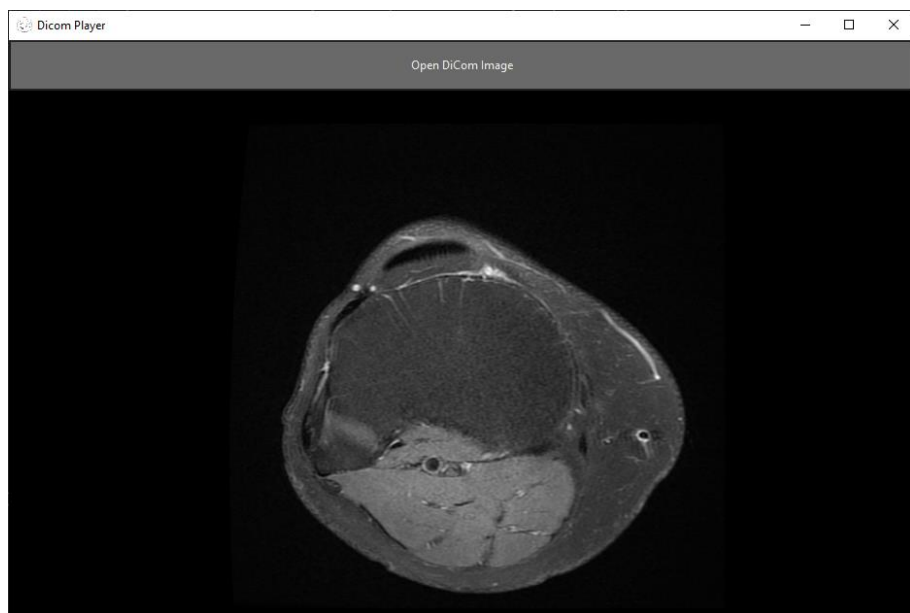
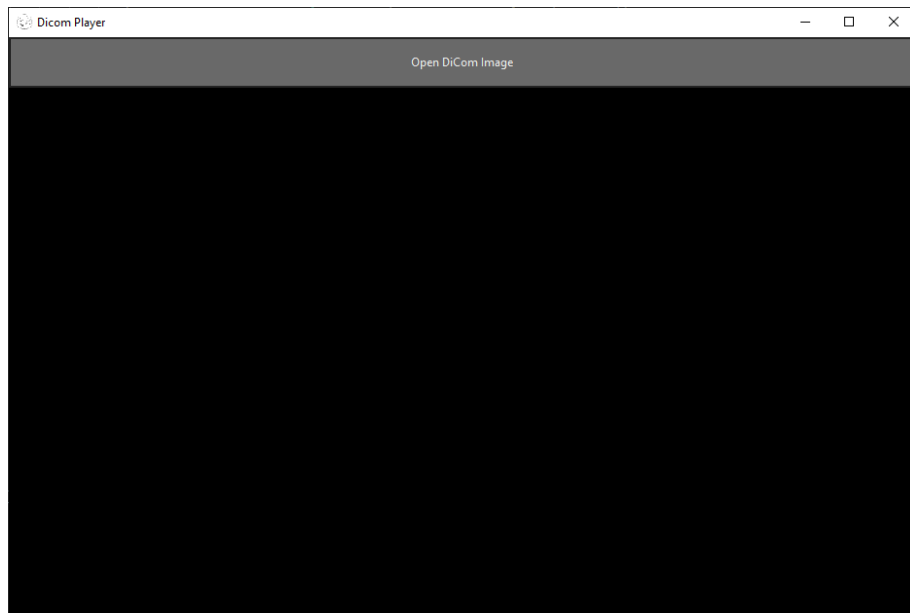
• Wygląd:



Przycisk otwierania plików DICOM

Miejsce do wyświetlania zawartości pliku DICOM

4. Zrzuty ekranu działania programu:



Wnioski końcowe:

Na podstawie wiedzy o technologii DICOM i wiedzy praktycznej zdobytej przy budowie programu, można zauważyć że stworzenie własnego programu viewera plików DICOM nie jest takie trudne. Pliki DICOM tworzą efektywną serię obrazów, co sprawia wrażenie filmu.

Link do zdalnego repozytorium

<https://github.com/SiwiecSoft/ATH/tree/master/ISM>