

Wyświetlacz mozaikowy

Projekt komponentu

Wykonawcy komponentu:

Patryk Wychowaniec

Radosław Wlekliński

Krzysztof Walczak

1. Opis ogólny

1.1. Wstęp

1.1.1. Cel

Niniejszy dokument precyzuje sposób realizowanych prac. Określa założenia projektu, standardy, narzędzia i komponenty wchodzące w skład implementacji, oraz opis realizacji tych komponentów.

1.1.2. Zakres

Założeniem projektu Wyświetlacza mozaikowego jest stworzenie okienkowej aplikacji Wyświetlacz. Aplikacja ma za zadanie umożliwiać użytkownikowi wyświetlenie tekst jako mozaiki oraz przesunięcie go z góry do dołu i na odwrót.

System tworzony jest w technologii C#, a do jego obsługi wystarczy komputer wykorzystujący tę technologię.

1.1.3. Omówienie

Dokument ten powstał na podstawie specyfikacji wymagań systemowych. Zawiera on definicje standardów, strategii i konwencji które będą przestrzegane podczas realizacji projektu.

2. Standardy projektu, konwencje, procedury

2.1. Standardy projektowe

Projekt aplikacji powstał w oparciu o paradygmat programowania obiektowego. Dziedzina problemu została przeanalizowana i zaprojektowana zgodnie z jego regułami. Dzięki temu oprogramowanie będzie łatwe w utrzymaniu i rozwoju, przez wzgląd na dostępność wielu narzędzi i wykwalifikowanych programistów.

2.2. Standardy dokumentacyjne

Dokumentacja projektu w sposób jednoznaczny określa jego strukturę logiczną i fizyczną. Wszystkie użyte pojęcia i skróty są wyjaśnione w odpowiednich miejscach. Specyfikacja wymagań jest zgodna ze standardem IEEE 830-1998. Diagramy zawarte w dokumentacji zostały sporządzone wg standardu UML 2.0. Wszystkie dokumenty dotyczące projektu są sporządzone na ustalonym szablonie firmowym. Kody źródłowe zawarte w dokumentacji są pisane czcionką o stałej szerokości.

2.3. Standardy programistyczne

Aplikacja zostanie wykonana i będzie działać na platformie C#. Pozwoli to stworzyć przejrzysty, łatwy do utrzymania kod, z rozdzieloną warstwą logiki biznesowej i prezentacji.

2.4. Narzędzia rozwijania oprogramowania

Środowiskiem RAD w którym powstanie aplikacja jest Microsoft Visual Studio 2013.

Ponadto wykorzystane zostaną:

- serwer GitHub
- narzędzie do projektowania – aplikacje online ze stron creately.com, glify.com
- edytor tekstu – LibreOffice Writer 4.4.0.3

3. Opis komponentu

3.1. Pola właściwości

Nazwa pola	Opis
DotEnabledColor	Kolor pikseli aktywnych
DotDisabledColor	Kolor pikseli nieaktywnych
DotRadius	Średnica piksela
MatrixText	Tekst do wyświetlenia na wyświetlaczu
ScrollDirection_	Kierunek przesuwania tekstu
MaxChars	Maksymalna liczba znaków wyświetlana na wyświetlaczu

3.2. Rysowanie wyświetlacza

3.2.1. Kierunek rysowania

Wyświetlacz jest rysowany wpierw po wierszach, po tym po kolumnach.

3.2.2. Metoda rysowania

```
for(int i=0;i<dotsOnWidth;++i) {  
    byte charIndex = charToIndex(MatrixText[i/7]);  
    for (int j = 0; j < dotsOnHeight; ++j) {  
        int x = i * (DotRadius * 2) + DotRadius * 2;  
        int y = j * (DotRadius * 2) + DotRadius * 2;  
        byte _char = 2;  
        if (charIndex < 128) {  
            _char = Letters.SLetters[charIndex, Math.Abs((j + Pivot) % 7), i % 7];  
            if (_char == 0) brush.Color = DotDisabledColor;  
            else if (_char == 1) brush.Color = DotEnabledColor;  
        }  
        e.Graphics.FillEllipse(brush, x, y, DotRadius * 2, DotRadius * 2);  
    }  
}
```

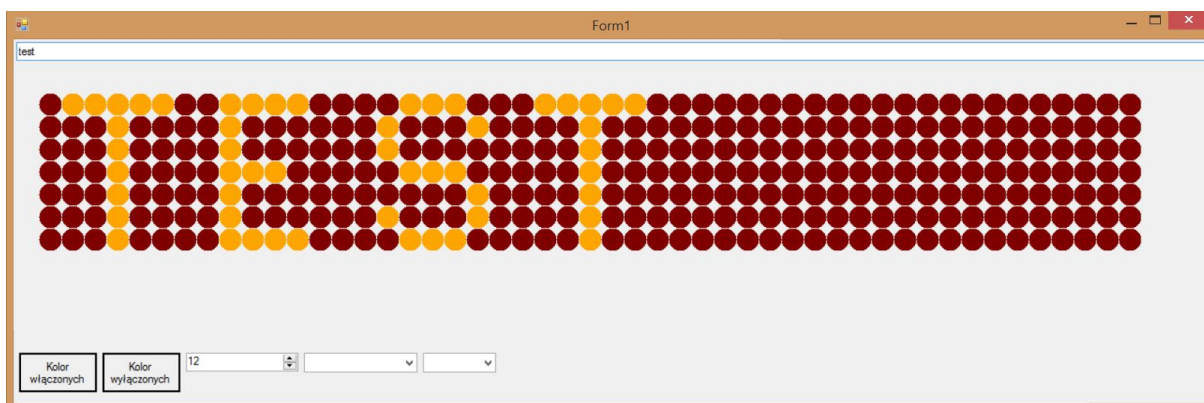
Fragment kodu źródłowego – ustalanie pozycji (x,y), koloru oraz wyrysowanie poszczególnych pikseli wyświetlacza.

3.3.3. Zmienna *Pivot* – ustalenie kierunku rysowania

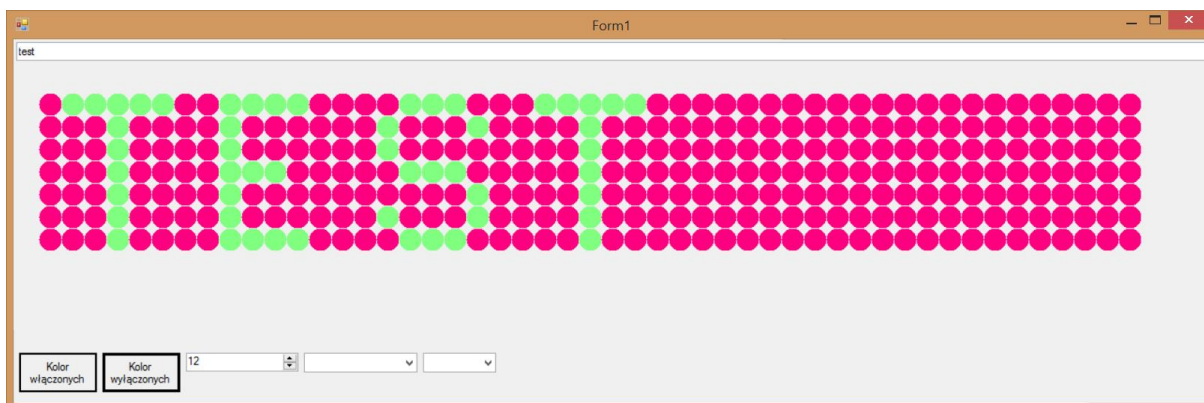
```
switch (Direction)
{
    case ScrollDirection.DOWN: Pivot--;
        if (Pivot < -7) Pivot = 0;
        Invalidate();
        break;
    case ScrollDirection.UP: Pivot++;
        if (Pivot > 7) Pivot = 0;
        Invalidate();
        break;
}
```

Zmienna *Pivot* jest wykorzystywana podczas ustalania pozycji piksela (→ 3.3.2)

4. Zrzuty ekranu aplikacji



Ilustracja 1: Widok aplikacji testującej - standardowe ustawienia wyświetlacza



Ilustracja 2: Widok aplikacji testującej - zmieniony kolor wyłączonych pikseli

5. Podsumowanie

Komponent spełnia nasze oczekiwania pod względem funkcjonalnym. Uważamy, że jest on ukończony i może być wykorzystywany w innych projektach.