Motywy i różne ekrany

Czas: 2 × 45 min | praca indywidualna | Wykład 2.29

Cel: Przygotujesz aplikację MAUI działającą w motywie jasnym i ciemnym (systemowym lub wybranym) oraz zmieniającą układ przy obrocie ekranu.

Po zajęciach potrafisz:

- definiuje kolory zależne od motywu AppThemeBinding
- przełącza motyw aplikacji UserAppTheme i zapamiętuje wybór
- zmienia układ zależnie od orientacji (VisualStateManager, OrientationStateTrigger)
- testuje wygląd na różnych platformach i ekranach

Sprzęt i materiały

- komputer z Windows 11 i dostępem do internetu
- pendrive lub konto w repozytorium do zapisania projektu

Oprogramowanie

- **Visual Studio Community 2026 (obciążenie „Programowanie aplikacji klasycznych platformy .NET")** — darmowe dla uczniów i nauczycieli; w pracowni zainstalowane visualstudio.microsoft.com/pl/downloads
- **.NET 10 SDK (LTS)** — instalowany razem z Visual Studio; sprawdzenie: `dotnet --version` dotnet.microsoft.com/pl-pl/download/dotnet/10.0
- **Obciążenie VS „Programowanie aplikacji wieloplatformowych w .NET" (.NET MAUI) z emulatorem Android**a — emulator wymaga wirtualizacji sprzętowej; awaryjnie uruchamiaj jako Windows Machine learn.microsoft.com/dotnet/maui/get-started/installation

Bezpieczeństwo

- Zapisuj projekt w swoim folderze (np. `C:\Users\<login>\source\repos`), a na koniec zajęć skopiuj go na pendrive lub wyślij do repozytorium.
- Nie uruchamiaj programów i pakietów z nieznanych źródeł; pakiety NuGet instaluj tylko te, które wskazuje zadanie.

Przebieg zadania

1 Kolory i style

Słownik `Resources/Styles/Motywy.xaml` dołączony w App.xaml.

```
<Setter Property="BackgroundColor"
        Value="{AppThemeBinding Light={StaticResource TloJasne},
                                Dark={StaticResource TloCiemne}}" />
```

1. Pary kolorów dla jasnego i ciemnego motywu: tło, karta, tekst, akcent.
2. Style z `AppThemeBinding`: ContentPage (tło), Label (tekst), Naglowek, Karta (Border), Button.
3. Sprawdź kontrast tekstu w obu motywach (min. 4,5:1).

2 Wybór motywu

`MotywService`.

1. Picker: Systemowy / Jasny / Ciemny; zapis w Preferences.
2. `Application.Current.UserAppTheme` = Unspecified / Light / Dark; przywrócenie w konstruktorze App.
3. Etykieta z motywem systemu (`PlatformAppTheme`) i aplikacji (`RequestedTheme`), odświeżana zdarzeniem RequestedThemeChanged.

3 Orientacja

Układ strony.

1. Grid z VisualStateManager: pionowo jedna kolumna, poziomo dwie (OrientationStateTrigger).
2. Druga karta przenosi się pod pierwszą lub obok niej; nagłówek w poziomie zajmuje 2 kolumny.
3. Informacja o ekranie: rozdzielczość, gęstość, orientacja (`DeviceDisplay.Current.MainDisplayInfo`).

4 Testy

Na dwóch platformach.

1. Windows: zmiana trybu aplikacji w ustawieniach systemu przy opcji „Systemowy”.
2. Emulator: ciemny motyw w szybkich ustawieniach, obrót ekranu.
3. Tabela 6 zrzutów (2 platformy × motywy/orientacje). Zatwierdź pracę w Git.

W sprawozdaniu umieść

- ☐ Motywy.xaml
- ☐ kod MotywService
- ☐ tabela zrzutów
- ☐ pomiary kontrastu

Pytania kontrolne

1. Czym różni się PlatformAppTheme od RequestedTheme?
2. Jak aplikacja zapamiętuje wybrany motyw?
3. Dlaczego jednostki MAUI nie są pikselami ekranu?
4. Co musi się zmienić w układzie przy orientacji poziomej i dlaczego?

Kryteria oceny

Kryterium	Punkty
Kolory i style z AppThemeBinding	5
Wybór i zapamiętanie motywu	5
Układ zależny od orientacji	5
Testy na platformach	3
Pytania	2
Razem	**20**

Sprawozdanie przygotuj na wzorze dołączonym do zadania (plik .docx o tym samym numerze).