Gra w kości w MAUI (styl INF.04)

Czas: 2 × 45 min | praca indywidualna | Wykład 2.24

Cel: Zbudujesz mobilną grę w kości według specyfikacji w stylu arkusza INF.04: obrazy kości, rzut, blokowanie dotknięciem z przezroczystością, wynik losowania i gry — z klasą Kosc z roku 1.

Po zajęciach potrafisz:

- wyświetla i zmienia obrazy z Resources/Images
- obsługuje gest dotknięcia z parametrem
- realizuje wygląd zgodnie ze specyfikacją (kolory, marginesy, czcionki)
- wykorzystuje istniejącą klasę logiki w nowej platformie

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 Androida** — emulator wymaga wirtualizacji sprzętowej; awaryjnie uruchamiaj jako Windows Machine learn.microsoft.com/dotnet/maui/get-started/installation

Przygotowanie przed zajęciami

- Klasa `Kosc` z roku 1 (zadanie 15) i obrazy `kosc0.png`–`kosc6.png` z folderu `pliki` przy tym zadaniu.

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 Specyfikacja

Wymagania wyglądu (traktuj jak arkusz egzaminacyjny).

1. Tło strony #F0F9FF, marginesy wewnętrzne strony 20, odstęp między elementami 16.
2. Nagłówek „Gra w kości”: 28, pogrubiony, kolor #0E7490, wyśrodkowany.
3. Pięć obrazów kości 64 × 64 z marginesem 4, w FlexLayout zawijanym i wyśrodkowanym.
4. Przycisk „RZUĆ KOŚĆMI”: tło #0E7490, biały pogrubiony tekst, wysokość 50, zaokrąglenie 10.
5. Etykiety: „Rzut N z 3”, „Wynik tego losowania: X” (20), „Wynik gry: Y” (24, pogrubiony).

2 Zasoby i układ

Projekt MAUI `GraWKosci`.

1. Obrazy do Resources/Images (Akcja kompilacji MauiImage); klasa Kosc do projektu bez zmian.
2. Każdy obraz z `TapGestureRecognizer Tapped="OnKoscTapped" CommandParameter="0"` (..."4").
3. Zbuduj stronę zgodnie ze specyfikacją.

3 Logika

Code-behind strony.

```
private void OnKoscTapped(object? sender, TappedEventArgs e)
{
    if (_rzut == 0 || e.Parameter is not string tekst) return;
    int nr = int.Parse(tekst);
    _kosci[nr].Zablokowana = !_kosci[nr].Zablokowana;
    Odswiez();
}
```

1. Rzut: wszystkie niezablokowane kości losują; licznik rzutów; po 3 rzutach przycisk nieaktywny.
2. Dotknięcie kości (po pierwszym rzucie) blokuje/odblokowuje; zablokowana ma `Opacity = 0.4`.
3. Wynik losowania: `Kosc.Punkty(...)`; Zapisz turę dodaje go do wyniku gry i odblokowuje kości.
4. Nowa gra z potwierdzeniem `DisplayAlertAsync` (gdy wynik > 0).

4 Dokumentacja

Jak na egzaminie.

1. Komentarze w kodzie strony (nazwa, opis, parametry).
2. Zrzuty z emulatora: start, po rzucie z 2 zablokowanymi kośćmi, po 3 turach.
3. Lista kontrolna zgodności ze specyfikacją (każdy punkt: tak/nie).
4. Zatwierdź pracę w Git.

W sprawozdaniu umieść

- ☐ kod XAML i C# strony
- ☐ lista kontrolna zgodności ze specyfikacją
- ☐ trzy zrzuty z emulatora
- ☐ opis wykorzystania klasy Kosc

Pytania kontrolne

1. Jak przekazać numer kości do metody obsługi dotknięcia?
2. Dlaczego w XAML piszemy kosc1.png?
3. Co zmieniłbyś, aby gra miała 6 kości?
4. Które elementy specyfikacji najłatwiej przeoczyć na egzaminie?

Kryteria oceny

Kryterium	Punkty
Zgodność ze specyfikacją wyglądu	6
Rzut i obrazy	4
Blokowanie z przezroczystością	4
Tury i wynik gry	3
Dokumentacja i zrzuty	3
Razem	**20**

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