

The cover features a minimalist design with three overlapping blue circles of varying sizes. Two circles are positioned in the upper right quadrant, and a larger one is partially visible at the bottom right. Two thin, light blue lines intersect to form a large 'V' shape that frames the central text area.

Artur Żarski

Expression Blend

Przewodnik

Expression Blend: Przewodnik

© 2009 Artur Żarski

Projekt graficzny okładki: Tomasz Maroński

APN PROMISE Sp. z o. o. Warszawa 2009

APN PROMISE Sp. z o. o., biuro: ul. Kryniczna 2,03-934 Warszawa
tel. (022) 355-16-00; fax (022) 355-16-99
e-mail: mspress@promise.pl

Wszystkie prawa zastrzeżone. Żadna część niniejszej książki nie może być powielana ani rozpowszechniana w jakiegokolwiek formie i w jakikolwiek sposób (elektroniczny, mechaniczny), włącznie z fotokopiowaniem, nagrywaniem na taśmy lub przy użyciu innych systemów bez pisemnej zgody wydawcy.

Microsoft, ActiveX, Internet Explorer, Visual Basic, Visual C#, Visual Studio, Windows, Windows Media, Windows Server oraz Windows Vista są zarejestrowanymi znakami towarowymi Microsoft Corporation.

Wszystkie inne nazwy handlowe i towarowe występujące w niniejszej publikacji mogą być znakami towarowymi zastrzeżonymi lub nazwami zastrzeżonymi odpowiednich firm odnośnych właścicieli.

APN PROMISE Sp. z o. o. dołożyła wszelkich starań, aby zapewnić najwyższą jakość tej publikacji. Jednakże nikomu nie udziela się rękojmi ani gwarancji.
APN PROMISE Sp. z o. o. nie jest w żadnym wypadku odpowiedzialna za jakiegokolwiek szkody będące następstwem korzystania z informacji zawartych w niniejszej publikacji, nawet jeśli APN PROMISE została powiadomiona o możliwości wystąpienia szkód.

ISBN: 978-83-7541-048-8

Redakcja: Ewa Swędrowska, Marek Włodarz
Korekta: Anna Wojdanowicz
Skład i łamanie: MAWart Marek Włodarz

Spis treści

1. Co to jest Expression Blend	5
2. Instalacja produktu	9
3. Pierwsze kroki z Expression Blend	11
Menu File	13
Menu Edit	14
Menu View	15
Menu Object	15
Menu Project	17
Menu Tools	18
Opcje środowiska pracy	18
Toolbox	19
Biblioteka zasobów (<i>Asset Library</i>)	21
Interaction Panel	22
Właściwości (<i>Property Panel</i>)	23
Zasoby – resources	29
4. Układ dokumentu	30
Typy dokumentów	30
5. Rysowanie obiektów i kształtów	37
Rozciąganie kształtów	41
Praca ze ścieżkami	44
6. Pędzel i jego zastosowanie	50
SolidColorBrush	50
LinearGradientBrush.....	51
RadialGradientBrush.....	52
Oś gradientu	52
ImageBrush	56
DrawingBrush	57
VisualBrush.....	58
Obiekty geometryczne	61
7. Grid	66
Rozmieszczenie elementów.....	68
Grid Splitter	71
8. Animacje	72
System animacji w WPF	73
Typy animacji.....	73

Klasa timeline	76
Właściwość BeginTime	76
Obsługa zdarzeń	78
Animacje łączone	78
Właściwości czasu: Duration	79
Właściwości czasu: AutoReverse	79
Właściwości czasu: RepeatBehavior	79
Storyboard	80
Animacje transformacji	81
Rozpoczęcie animacji	83
Animacja stylów	85
Animacje szablonów kontroltek	85
Klatki animacji	87
Interpolacja klatek	90
Klasy animacji klatkowych	91
Typy klatek animacji	92
Metody interpolacji	93
Struktura elementu KeyTime	96
Przykład animacji krok po kroku	98
9. Zasoby	100
Słownik zasobów	105
10. Style i szablony	109
Style generyczne	112
Określanie stylów kontroltek	113
Style i wyzwalacze	118
Proste kontrolki (<i>Simple Controls</i>)	124
11. Praca z tekstem	142
Label	142
TextBox	143
RichTextBox	144
PasswordBox	145
Typografia	145
12. Media Element	150
Tryb Independent	151
Tryb Clock	152
Klasa MediaElement	152
Zakończenie	155

1. Co to jest Expression Blend

Microsoft Expression Blend jest nowym, profesjonalnym narzędziem projektowym do tworzenia aplikacji oraz stron internetowych o bogatym interfejsie użytkownika. Produkt umożliwia użytkowanie pełnego spektrum różnych typów medialnych, takich jak video, tekst, obiekty 3D oraz ich animacje oraz grafika wektorowa. Przy użyciu Expression Blend możliwe jest dostarczanie aplikacji wyglądających i zachowujących się zgodnie z oczekiwaniami klienta. Expression Blend współpracuje z Microsoft Visual Studio, co umożliwia oddzielenie pracy projektanta interfejsu od programisty, a mimo to pozwala zagwarantować, by obie grupy pracowały na tych samych, zawsze aktualnych danych.

Wraz z rozwojem oprogramowania zaczęto kłaść coraz większy nacisk na sposoby budowy interfejsu użytkownika. Zaczął pojawiać się termin User Experience (UX), czyli wrażenia użytkownika z pracy z aplikacją.

Jest bardzo wiele definicji tego pojęcia i większość z nich opisuje je w dość dokładny sposób. UX może być zatem „czytelnością”, gdzie interfejs użytkownika będzie intuicyjny, a wszystkie informacje podane w sposób bardzo przystępny i prosty do znalezienia. Może być również „wiarygodny”, gdy będziemy w stanie przewidzieć, co się stanie po wybraniu takiej, a nie innej opcji, będzie bezpieczny dla użytkownika i skalowany. Takie przykłady można mnożyć. Warto natomiast zwrócić uwagę na to, że w istocie pojęcie to składa się z dwóch głównych cech – pierwsza z nich to łatwość użycia, dzięki któremu program będzie łatwiejszy w użytkowaniu czy w nauce, bezpieczniejszy, optymalny i zmieniający się w zależności od kontekstu pracy. Z drugiej strony oczekujemy pewnego rodzaju bogactwa treści, czyli elementów składających się na ten interfejs. Do tych elementów wizualnych zaliczyć można między innymi grafikę i media, wizualizację danych, wysoką dokładność informacji i integrację z posiadanym sprzętem. Jeśli teraz poskładamy oba te elementy, dostaniemy to, czego oczekuje użytkownik: sukces, produktywność, zdolność pamiętania, porównywalność, satysfakcja oraz powtarzalność użycia.

W praktyce jednak okazywało się bardzo często, że budowane aplikacje, pomimo wielu doskonałych funkcjonalności, miały jedną zasadniczą wadę – były trudne do ogarnięcia przez użytkowników. Ich interfejs był mało intuicyjny, a korzystanie z aplikacji stanowiło nierzadko problem i spowalniało pracę. Główną przyczyną takiego stanu rzeczy było to, że interfejs ten był budowany przez programistów. Bardzo często zdarza się, że programista nie buduje aplikacji dla klienta, tylko na swój własny użytek, a co za tym idzie, nie bierze pod uwagę ergonomii i możliwości pracy z programem przez osoby niezwiązane z danym rozwiązaniem. W rezultacie powstają programy wprawdzie funkcjonalne, ale niewygodne i trudne w użyciu.

Wspomniałem, że dużą część winy za ten stan rzeczy ponoszą programiści, ale czy tylko i wyłącznie oni? Może to również wynikać z ograniczeń oprogramowania, z którego korzystają w celu stworzenia produktu. Warto zwrócić uwagę na proces wytwarzania oprogramowania. Biorą w nim jeszcze udział projektanci aplikacji i to oni rozmawiają z klientem zamawiającym dany produkt. To oni zbierają wymagania i robią wstępne szkice. Rysunki te w dowolnej formie (papierowej czy elektronicznej) dostarczane są programiście. Osoba ta po otrzymaniu takiego projektu zwykle sprawdza, czy dana kontrolka jest dostępna w jego narzędziach, czy też jej brak. W większości przypadków okazuje się, że nie ma czegoś takiego, co narysował projektant, i w tym miejscu programista staje przed dylematem – napisać taką kontrolkę od zera (co wiąże się z ryzykiem niedotrzymania terminów oddania aplikacji) czy też zmieścić się w czasie i wykorzystać to, co jest. Efekt jest łatwy do przewidzenia, ponieważ najbardziej popularnym wyborem jest opcja druga. Czy zatem „świat” programisty i projektanta to zupełnie odrębne byty? Czy nie ma możliwości, aby obie strony mogły porozumieć się przy pomocy tego samego języka? Okazuje się, że jest taka możliwość. Wraz z pojawieniem się pierwszej wersji WinFX, czyli nowego .NET Framework 3.0, pokazał się nowy silnik graficzny WPF (wcześniej znany pod nazwą kodową Avalon). WPF to zestaw bibliotek, dzięki którym możliwa jest integracja interfejsu użytkownika, grafiki 2D i 3D oraz m.in. multimediów. WPF został wykorzystany w Windows Vista do zarządzania interfejsem użytkownika. Główna idea to oddzielenie definicji interfejsu użytkownika w aplikacjach Windows od kodu. Interfejs użytkownika jest teraz definiowany za pomocą odmiany XML-a, który nazywa się XAML.

Najważniejsze właściwości WPF:

- Deklaratywne budowanie interfejsów przy pomocy języka XAML.
- Dwa modele tworzenia aplikacji – uruchamiana w przeglądarce oraz jako gruby klient.
- Wykorzystanie sprzętowej akceleracji grafiki.
- Wsparcie dla dwóch rodzajów dokumentów: Fixed i Flow.
- Konfigurowalność i rozszerzalność.

Warto zwrócić uwagę na to, że wszystkie elementy wizualne są obsługiwane przez WPF – wektory, bitmapy, obiekty 3D, elementy audio, video, obiekty tekstowe oraz różnego rodzaju efekty. Najbardziej rewolucyjnym elementem jest Composition Engine. Silnik ten pozwala na umieszczenie dowolnej zawartości w dowolnym elemencie. Oznacza to tyle, że można na przykład osadzić animowany obiekt 3D wewnątrz kontrolki reprezentującej klawisz lub odtwarzać multimedia na powierzchni obiektów 3D. Taka drzewiasta struktura dostępna jest dla każdej kontrolki istniejącej w WPF. WPF jest nie tylko zestawem kontrolki tworzących interfejs użytkownika, ale również umożliwiających przechowywanie różnego rodzaju zawartości lub połączeń do bazy danych.

Ważnym aspektem są również definicje rozmiarów elementów w WPF, które posługują się jednostkami metrycznymi. W związku z tym można definiować fizyczne rozmiary elementów, czyli np. określić, że klawisz mierzy 2x5 cm i będzie miał taki rozmiar zarówno na monitorze, jak i po wydrukowaniu. Można nadal używać pikseli, ale w tym przypadku piksele te są niezależne od rozdzielczości. Jeden piksel w WPF ma rozmiar 1/96 cala (a 96 dpi to standardowa rozdzielczość w Windows).

Istotnym elementem jest rozdzielenie definicji interfejsu użytkownika od kodu, który odpowiada za jego wykonanie. Definicja interfejsu opisywana jest językiem XAML (Extensible Application Markup Language), który jest oparty na XML-u. Obsługa zdarzeń może być napisana w jednym z dwóch języków: C# lub VB.NET.

Zdefiniowanie przycisku w naszej aplikacji może wyglądać następująco:

```
<Window x:Class="obiekt"
<Button Name ="button1" Click="bc">OK.
    <Button.Background>
        LightBlue
    </Button.Background>
</Button>
```

Kod ten spowoduje stworzenie prostego przycisku, na którym będzie umieszczony napis „OK”, a jego tło będzie koloru jasnoniebieskiego.

Oprócz oczywiście samego języka XAML potrzebne są również narzędzia przeznaczone właśnie dla osób zajmujących się projektowaniem aplikacji lub też projektowanie różnego rodzaju interakcji w aplikacjach. Dla takich osób Microsoft stworzył zestaw produktów o nazwie Expression Studio. W pakiecie tym możemy znaleźć następujące produkty:

- **Expression Web** – narzędzie do budowy stron internetowych, które wywodzi się z produktu FrontPage, ale całkowicie go zastępuje, dodając wiele nowych funkcjonalności oraz możliwości programowania.
- **Expression Blend** – czyli narzędzie do tworzenia interaktywnej zawartości dla aplikacji Web.
- **Expression Design** – narzędzie do budowy elementów graficznych do aplikacji oraz stron interaktywnych.
- **Expression Media** – produkt, który pozwala na zarządzanie multimediami, katalogowanie ich oraz organizowanie.
- **Expression Media Encoder** – program do obróbki plików multimedialnych, oferujący możliwość skrócenia filmu, dodania trailera oraz innych elementów

Podstawowe funkcjonalności Blenda to między innymi:

- Praca z obiektami 3D – w tym import i eksport z i do formatów popularnych narzędzi, manipulowanie obiektami 3D przy użyciu kamery, definiowanie materiałów, animacje w przestrzeni „Z”.
- Łatwy w użyciu wizualny interfejs wraz z bogatym kontekstowym menu.
- Animacje przy użyciu linii czasu.
- Integracja z mediami, takimi jak np. wideo o różnej jakości oraz wielkości (w tym wideo w formacie HD – High Definition).
- Integracja z danymi zewnętrznymi.
- Projektowanie na zasadzie „przeciągnij i upuść” w trybie WYSWIG, co pozwala na bieżąco kontrolować stan projektu.
- Wykorzystanie XAML wraz z jednoczesną możliwością jego edycji.

2. Instalacja produktu

Podobnie jak dla większości produktów firmy Microsoft, także i w tym przypadku proces instalacji jest intuicyjny i nie powinien sprawić kłopotu nawet początkującym użytkownikom. Po umieszczeniu płyty CD w napędzie wybieramy system operacyjny, na którym będzie zainstalowany Expression Blend (tak jak na rysunku 1).

Skąd ten wybór? Wynika to z faktu, że system Windows XP SP2 standardowo nie zawiera .NET Framework 3.0, który jest wymagany do prawidłowej pracy Expression (tym wymaganiem elementem jest komponent o nazwie Windows Presentation Foundation). W systemie Vista .NET Framework 3.0 jest zainstalowany standardowo, a i sam system Windows Vista korzysta z Windows Presentation Foundation. Po zainstalowaniu .NET Framework 3.0, jeśli wystąpiła taka potrzeba, możemy rozpocząć instalację. Po tym kroku pojawi się standardowy ekran powitalny (rysunek 2). Wciskamy klawisz „Next” i przechodzimy do następnego kroku, którym jest akceptacja lub odrzucenie warunków licencji (rysunek 3). Aby przejść dalej i rozpocząć instalację, należy zaakceptować wymienione warunki.



Rysunek 1. Wybór systemu operacyjnego

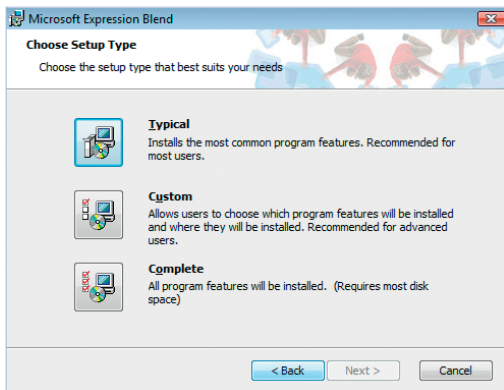


Rysunek 2. Rozpoczęcie procesu instalacji

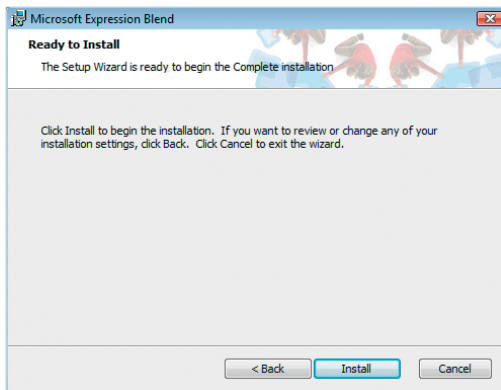


Rysunek 3. Akceptacja warunków licencji

Kolejne kroki są również standardowe dla produktów Microsoft. Wybieramy rodzaj instalacji: minimalna, pełna lub własna (Rysunek 4), a po dokonaniu wyboru możemy rozpocząć proces instalacji (Rysunek 5).

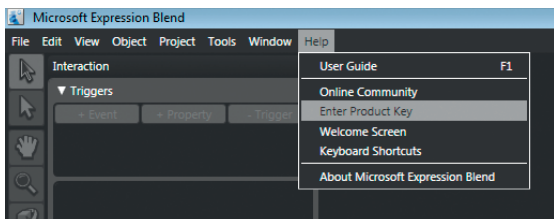


Rysunek 4. Wybór rodzaju instalacji



Rysunek 5. Rozpoczęcie instalacji

Klawisz „**Install**” rozpoczyna instalację. Warto zwrócić uwagę na to, że podczas instalacji system nie prosi o podanie klucza aktywacyjnego. Aby produkt aktywować, musimy teraz uruchomić Expression Blend i wprowadzić klucz produktu. W tym celu wybieramy opcję w menu *Help* → *Enter Product Key* (rysunek 6).



Rysunek 6. Wprowadzenie klucza produktu

3. Pierwsze kroki z Expression Blend

Produkt został zainstalowany i jest gotowy do użycia. Przyjrzyjmy się zatem, jak wygląda interfejs użytkownika oraz zapoznajmy się z podstawowymi opcjami menu.

Blend po standardowej instalacji znajduje się w menu **Start → Microsoft Expression → Microsoft Expression Blend**. Po jego uruchomieniu pojawia ekran powitalny podzielony na trzy zakładki, tak jak na rysunku 7:

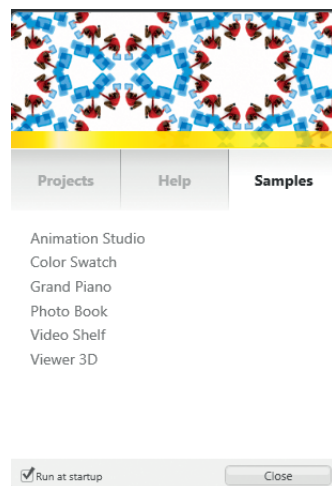
Projects – lista ostatnich projektów.

Help – pomoc dla produktu.

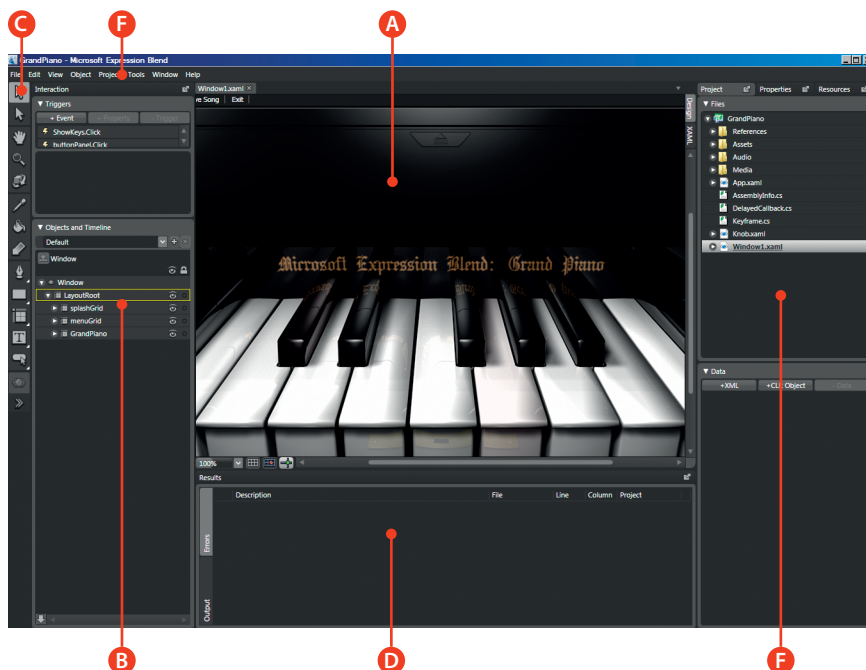
Samples – przykłady gotowych aplikacji.

Opcja **Run at startup** pozwala na wybranie, czy chcemy, aby ten ekran pojawiał się przy starcie programu, czy też nie.

Aby zobaczyć, jak wyglądają poszczególne elementy ekranu, wybierzmy z przykładów gotowy projekt p.t. „Grand Piano”. Po uruchomieniu zobaczymy obraz, taki jak na rysunku 8.



Rysunek 7.
Ekran powitalny
Expression Blend



Rysunek 8. Wygląd pulpitu Expression Blend

Na rysunku zaznaczono kilka podstawowych elementów. Zanim jednak zaczniemy je szczegółowo omawiać – kilka słów wstępu. Expression Blend składa się z dwóch głównych środowisk pracy – *Design View* oraz *XAML View*. *Design View* jest widokiem standardowym i służy do projektowania wizualnego naszego rozwiązania. *XAML View* jest trybem edycji kodu aplikacji, ukazującym kod XAML definiujący konkretny obraz i jego elementy.

Całe środowisko pracy nosi nazwę *Workspace* i składa się z następujących elementów (wg liter na rysunku):

- A Artboard**, czyli przestrzeń do graficznego modelowania aplikacji. To w tym miejscu mamy do dyspozycji ekran projektowy, gdzie wstawiamy poszczególne elementy i na którym tworzymy widok ogólny.
- B Interaction Panel** – jest to element, który zawiera wszystkie komponenty związane z interaktywnością, czyli przede wszystkim listę obiektów i powiązań między nimi (np. jak wygląda hierarchia elementów), listę wyzwalaczy (ang. *triggers*), czyli elementów decydujących, co ma się wydarzyć, jeśli nastąpi jakieś zdarzenie, oraz to, co najważniejsze – element o nazwie *Timeline*, czyli linię czasu. Linia czasu zawiera informacje o tym, jak w określonym przedziale czasu ma się zachowywać wybrany obiekt.
- C Toolbox** – przybornik narzędzi.
- D Result Panel** – czyli informacje systemu na temat projektu (np. błędy kompilacji).
- E** Lista elementów związanych z projektem (zasoby, pliki źródłowe).
- F Menu** główne programu.

Przyjrzyjmy się teraz szczegółowo poszczególnym elementom, zaczynając od Menu. Składa się ono z następujących elementów:

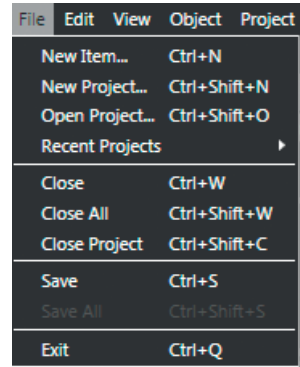
- File** – standardowy element, który pozwala na podstawową pracę z plikami.
- Edit** – standardowe elementy edycyjne.
- View** – funkcje związane z oglądaniem i przeglądaniem projektu.
- Object** – funkcje związane z pracą z obiektami rysunkowymi.
- Project** – menu pozwalające na pracę z projektem i zarządzanie nim.
- Tools** – zarządzanie dodatkowymi narzędziami,
- Window** – praca z poszczególnymi oknami oraz widokami środowiska pracy.
- Help** – pomoc.

Zobaczmy zatem, do czego służą poszczególne elementy menu i w jaki sposób można z nich korzystać.

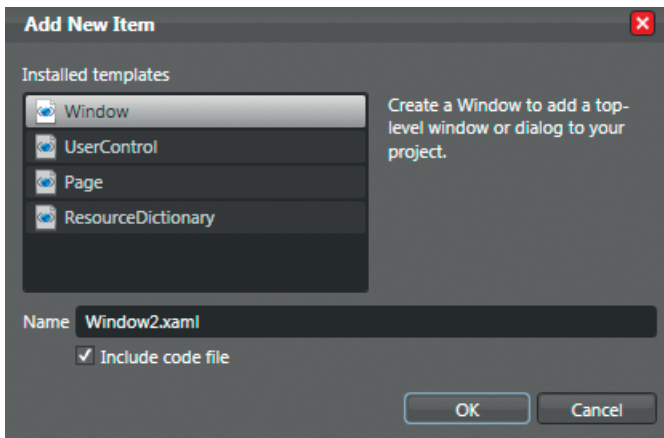
Menu File

Menu File jest dostępne w każdym z produktów. Jego elementy przedstawiono na rysunku 9. Poszczególne elementy menu opisane są poniżej.

New Item ... (CTRL+N) – funkcjonalność ta pozwala na utworzenie nowego elementu w ramach projektu. Umożliwia stworzenie nowego okna, nowej kontrolki użytkownika, strony lub słownika zasobów, który pozwala na przechowywanie i udostępnienie różnych elementów w ramach projektu. Opcja ta pokazana jest na rysunku 10.



Rysunek 9.
Menu File

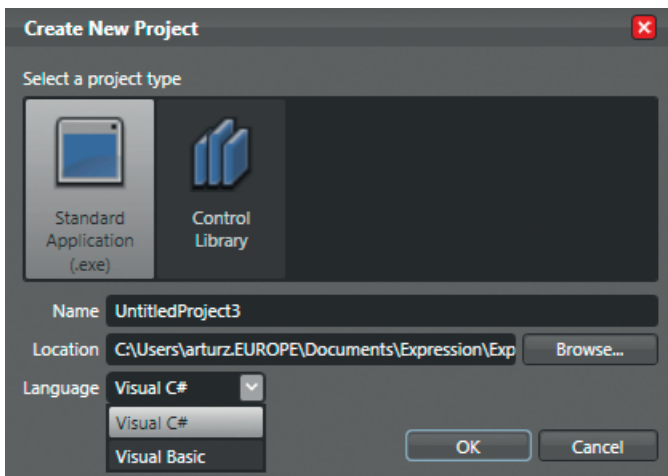


Rysunek 10. *Dodanie nowego elementu*

Warto w tym miejscu zwrócić uwagę na opcję **Include code file**, której wybranie powoduje dodanie do elementu pliku zawierającego tzw. *code-behind*, czyli umieszczenie kodu źródłowego dla elementu poza definicją tworzonego obiektu. W zależności od tego, jaki język wybraliśmy przy tworzeniu projektu, pliki źródłowe będą miały rozszerzenie *.xaml.cs (dla języka C#) oraz *.xaml.vb (dla języka Visual Basic).

New Project ... (CTRL+SHIFT+N) – funkcjonalność ta pozwala utworzyć nowy projekt. Opcję tę przedstawia rysunek 11.

Jak widać, mamy do dyspozycji dwa możliwe typy projektów. Pierwszy z nich to *Standard Application* (standardowa aplikacja), a drugi to *Control Library* (biblioteka kontrolki użytkownika). Dodatkowo możemy wybrać język programowania C# lub VB.NET (język programowania jest analogiczny do ActionScript w przypadku Flash). Podajemy nazwę nowego projektu oraz lokalizację, gdzie będzie się znajdował projekt wraz ze wszystkimi powiązаныmi plikami.



Rysunek 11. *Utworzenie nowego projektu*

Open Project ... (CTRL+SHIFT+O) – funkcja ta pozwala na otwarcie gotowego już projektu. Możliwe jest uruchomienie pliku o rozszerzeniu *.csproj (dla projektów w C#), *.vbproj (dla projektów w Visual Basic) oraz *.sln (plik rozwiązania dla Visual Studio).

Recent Projects ... – funkcja ta pozwala na dostęp do ostatnio używanych projektów.

Close (CTRL+W) – opcja ta zamyka aktualnie otwarte okno.

Close All (CTRL+SHIFT+W) – zamyka wszystkie otwarte okna, ale nie zamyka projektu.

Close Project (CTRL+SHIFT+C) – opcja ta zamyka cały projekt.

Save (CTRL+S) – zapisanie zmian w aktywnym oknie.

Save All (CTRL+SHIFT+S) – zapisanie wszystkich zmian we wszystkich oknach.

Menu Edit

To menu pozwala na standardowe prace edycyjne w projekcie z poszczególnymi elementami. Widok menu pokazano na rysunku 12. Opcje te przydane są szczególnie w trybie pracy z kodem XAML.

Undo (CTRL+Z) – cofnięcie ostatniej operacji.

Redo (CTRL+Y) – przywrócenie ostatniej operacji.

Cut, Copy, Paste, Delete – standardowe operacje wycięcia i skopiowania elementu do schowka, skopiowania do schowka, wklejenia ze schowka bądź skasowania elementu.

Find, Find Next, Replace – typowe funkcje wyszukiwania informacji oraz zamiany jednej informacji na drugą.

Go To (CTRL+G) – przejście do odpowiedniej linii kodu.

Edit	View	Object	Project
Undo Rectangle			Ctrl+Z
Redo			Ctrl+Y
Cut			Ctrl+X
Copy			Ctrl+C
Paste			Ctrl+V
Delete			Del
Find...			Ctrl+F
Find Next			F3
Replace...			Ctrl+H
Go To...			Ctrl+G
Select All			Ctrl+A
Select None			Ctrl+D

Rysunek 12.
Menu Edit

Select All (CTRL+A) – zaznaczenie całego kodu źródłowego w trybie widoku XAML lub wszystkich obiektów w trybie pracy Design.

Select None (CTRL+D) – odznaczenie poprzednio wybranych obiektów.

Menu View

Menu służące do zarządzania widokami całego środowiska pracy. Widok tej opcji przedstawiono na rysunku 13.

Design (F11) – widok w trybie projektowania.

XAML (F12) – widok pracy w trybie kodu XAML.

Zoom In (CTRL+=) – powiększenie przestrzeni rysunkowej.

Zoom Out (CTRL+-) – pomniejszenie przestrzeni rysunkowej.

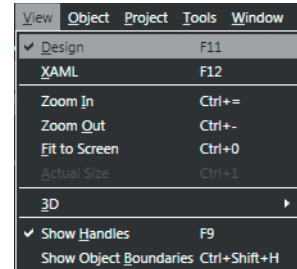
Fit to Screen (CTRL+0) – dopasowanie przestrzeni rysunkowej do wielkości okna.

Actual Size (CTRL+1) – powrót do aktualnej (początkowej) wielkości okna.

3D – zawiera opcję *Toggle Lights*, pozwalającą na przełączanie źródeł światła dla obiektów 3D.

Show Handles (F9) – pokazuje lub ukrywa uchwyty dla elementów na rysunku. Uchwytem jest obramowanie pozwalające na łatwiejsze chwywanie elementów.

Show Object Boundaries (CTRL+SHIFT+H) – chowanie lub pokazywanie granicy obiektu.



Rysunek 13.
Menu View

Menu Object

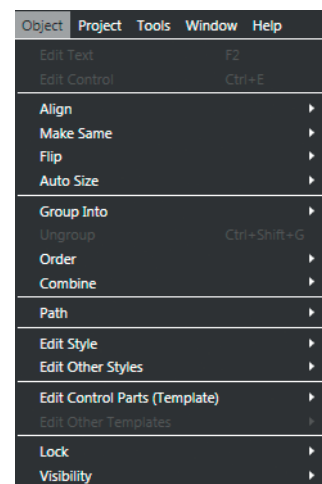
Jest to podstawowe menu, dzięki któremu możliwa jest praca z obiektami oraz ich modyfikacja. Menu to pokazano na rysunku 14.

Edit Text (F2) – opcja pozwala na edycję tekstu, jeśli kontrolka przechowuje jakieś wartości tekstowe. Przykładem może być tekst w polu typu TextBox lub tekst na klawiszu.

Edit Control (CTRL+E) – funkcja ta umożliwi edycję kontrolki użytkownika, o ile mamy dostęp do jej danych źródłowych.

Align – umożliwi wyrównanie ułożenia kontrolki względem siebie.

Make Same – pozwala na edycję zaznaczonych kontrolki, aby miały taką samą wielkość (wysokość i szerokość).



Rysunek 14.
Menu Object

Flip – odbicia względem osi poziomej i pionowej zaznaczonej kontrolki.

Auto Size – pozwala na automatyczny dobór rozmiaru przez kontrolkę. Zawiera opcje *Width*, *Height*, *Both*, *Fill*. Opcja *Width* automatycznie dopasowuje szerokość kontrolki do zawartości – np. jeśli w środku znajduje się tekst. *Height* dopasowuje wysokość, opcja *Both* zawiera w sobie obie wymienione opcje, a *Fill* zmienia rozmiar kontrolki, dopasowując ją do całego rozmiaru kontenera, w którym kontrolka jest umieszczona.

Group Into – możliwość zgrupowania kilku kontrolki do jednego kontenera, który pozwala na łatwiejsze zarządzanie poszczególnymi elementami rysunku. Dostępne kontenery to: *Grid*, *StackPanel*, *DockPanel*, *Canvas*, *ScrollView*, *Border*, *WrapPanel*, *UniformGrid* oraz *ViewBox*.

Ungroup (CTRL+SHIFT+G) – operacja odwrotna do grupowania.


Order – umiejscowienie wybranej kontrolki lub grupy względem innych, na przykład czy ma ona być widoczna na górze wszystkich obiektów, czy też powinna być ukryta pod jakąś inną. Dostępne opcje to *Bring to Front*, *Bring Forward*, *Send To Back*, *Send Backward*.


Combine – opcja pozwalająca na łączenie ze sobą różnych kontrolki na zasadzie działania na zbiorach. Dostępne funkcjonalności pozwalają uzyskać na przykład część wspólną zaznaczonych kontrolki lub też uzyskiwać obszar będący połączeniem obwodów wybranych kontrolki. Dzięki temu możemy uzyskiwać niestandardowe kształty. Dostępne opcje to: *Unite*, *Divide*, *Intersect*, *Subtract* oraz *Exclude Overlap*.

Path – jedna z ciekawszych opcji pozwalająca na zamianę wybranej kontrolki na zestaw krzywych reprezentujących obiekt.

Edit Style oraz **Edit Other Styles** – są to bardzo podobne opcje służące do edycji stylu. Style opisują wygląd poszczególnych elementów i mogą być używane do łatwego tworzenia globalnych wizualizacji elementów w projekcie.

Edit Control Parts (Template) oraz **Edit Other Templates** – opcja pozwala na utworzenie szablonów kontrolki oraz zdefiniowanie zachowania się tych kontrolki.

Lock – umożliwia blokowanie przed edycją poszczególnych elementów (*Lock Selection*) lub zablokowanie wszystkich elementów na oknie i samego okna (*Lock All*). Operacją odwrotną jest dostępna funkcja (*Unlock All* – CTRL+SHIFT+L). Warto zwrócić uwagę na to, że w przypadku wybraniu którejś z tych opcji w oknie **Objects and timeline** pojawia się kłódka  reprezentująca blokadę danego obiektu.

Visibility – jest to opcja podobna do *Lock*, z tą różnicą, że określa, które elementy mają być widoczne lub niewidoczne na ekranie. W panelu **Objects and timeline** widoczna jest jako „oko” , jeśli obiekt jest widoczny.