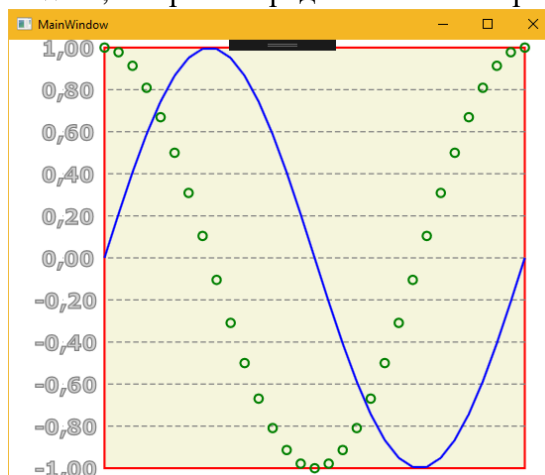


Практична робота 5

Розробка розвинуеного інтерфейсу користувача

№	Тема	К-ть балів
1.	<i>Захист принаймні одного завдання з роботи</i>	1
2.	Завдання 1	0,8*
3.	Завдання 2	0,6*
4.	Завдання 3	0,4*
5.	Завдання 4	0,4*
6.	Завдання 5	0,4*
7.	Завдання 6	1
8.	<i>Здача звіту</i>	0,4
	Всього	5

1. (Побудова графіків) Пропонується побудувати графік функції за допомогою елементів 2D-графіки WPF, на зразок представленого на рисунку додатку.



Побудова таких графіків складатиметься з поступового нашарування прошарків, зокрема

- 1) Фону (червоної квадратної рамки)
- 2) Горизонтальної пунктирної сітки
- 3) Графіку синусу (лінія)
- 4) Графіку косинусу (точки)
- 5) Підписів

Алгоритм роботи з прошарками приблизно однаковий:

- 1) Створити об'єкт чи колекцію для опису геометричних фігур
- 2) Описати та зберегти геометрію фігур(и)
- 3) Налаштувати перо
- 4) Налаштувати пензель
- 5) Додати готовий прошарок в контейнер для відображення

Приклад методу, що відрисує фон, записано нижче

```
// Контейнер прошарків рисунків
```

```

DrawingGroup drawingGroup = new DrawingGroup();

private void BackgroundFun() {
    // (1)
    GeometryDrawing geometryDrawing = new GeometryDrawing();

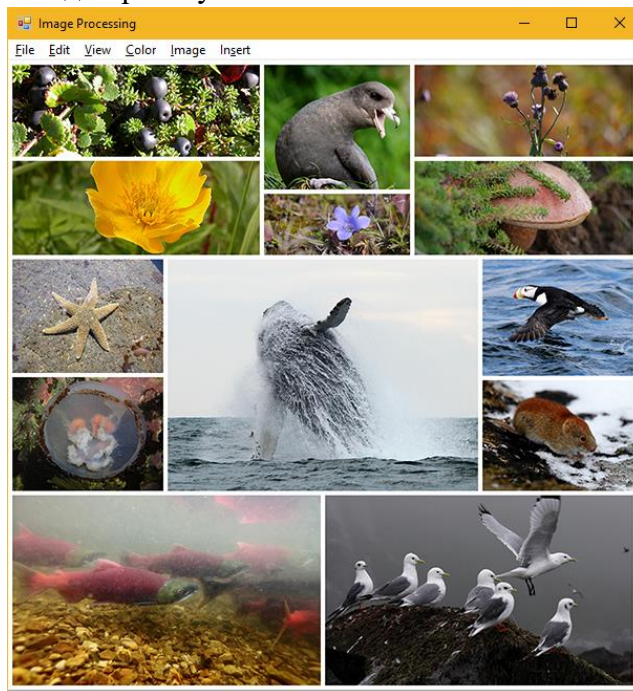
    // (2)
    RectangleGeometry rectGeometry = new RectangleGeometry();
    rectGeometry.Rect = new Rect(0, 0, 1, 1);
    geometryDrawing.Geometry = rectGeometry;

    // (3) i (4)
    geometryDrawing.Pen = new Pen(Brushes.Red, 0.005); // Перо рамки
    geometryDrawing.Brush = Brushes.Beige; // Пензель зафарбовування

    // (5)
    drawingGroup.Children.Add(geometryDrawing);
}

```

2. (Груповий міні-проект) Дослідити код [WinForms-проекту](#) та переписати його на базі технології WPF. Не реалізовувати пункти меню File, Edit, View. Зображення мають підключатись до проекту.

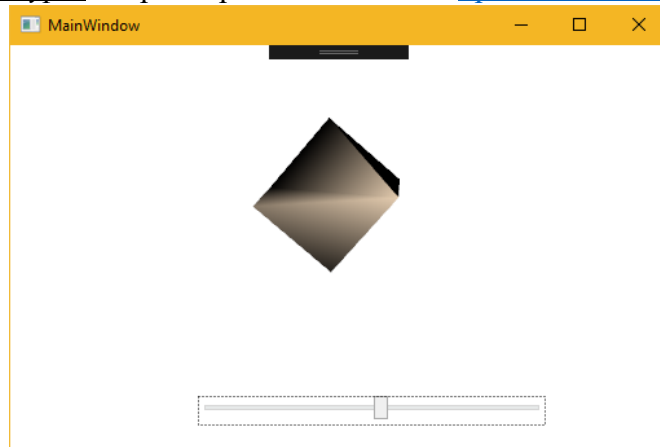


Відповідно до варіанту

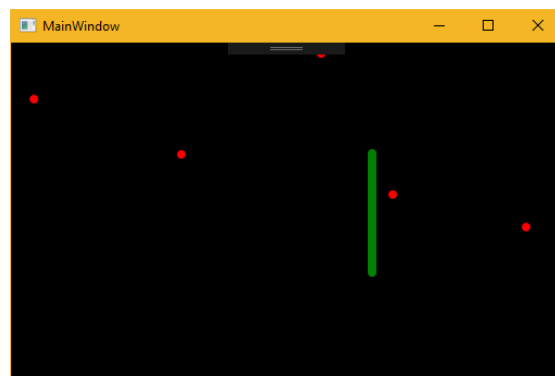
- 1) Кольоровий фільтр, гамма-корекція
- 2) Яскравість та контрастність
- 3) Відтінки сірого та інвертація (ефект негативу)
- 4) Зміна розміру зображення
- 5) Повороти, дзеркальні відображення
- 6) Обрізка зображення
- 7) Нанесення на зображення тексту
- 8) Нанесення на зображення інших зображень чи фігур

Підгрупа відповідає за інтеграцію проекту: створення меню, вибір із доступних у проекті зображень, підключення в проект методів, написаних усіма варіантами.

3. Опрацювати глави 8 (Vector Graphics) та 9 (User Controls and Custom Controls) книги «Blend for Visual Studio 2012 By Example Beginner's Guide» за авторством Abhishek Shukla і створити власний елемент управління.
4. Анімована 3D-фігура. Зберіть проект відповідно [представленого коду](#).



5. Опрацювати главу 6 (Understanding Animation and Storyboards) книги «Blend for Visual Studio 2012 By Example Beginner's Guide» за авторством Abhishek Shukla та анімуйте заданий об'єкт.
6. Гра «Змійка». Зазвичай гравець керує змієподібною істотою, яка повзає по площині та збирає їжу, уникаючи зіткнень зі своїм хвостом та краями ігрового поля. Кожного разу, коли змія з'їдає шматок їжі, вона стає довшою, що поступово ускладнює гру. Гравець керує напрямком руху голови змії, але зупинити рух не може.



З точки зору реалізації змійка та набір їжі визначаються як набір точок (клас Point). Швидкість руху змійки визначається часовими проміжками, наприклад,

```
private TimeSpan FAST = new TimeSpan(1);
private TimeSpan MODERATE = new TimeSpan(10000);
private TimeSpan SLOW = new TimeSpan(50000);
private TimeSpan DAMNSLOW = new TimeSpan(500000);
```

Динамічний рух створюється через запуск таймеру, який зсуває точки тіла змійки

```
DispatcherTimer timer = new DispatcherTimer();
timer.Tick += new EventHandler(timer_Tick);

timer.Interval = MODERATE;
timer.Start();
```

Метод `timer_Tick(object sender, KeyEventArgs e)` у даному випадку визначатиме нові координати голови у процесі руху.

Управління рухом змійки відбувається з клавіатури, тому потрібно створювати подію «натиснення на клавішу», яка визначатиме напрям руху (у даній реалізації – метод `OnButtonKeyDown(object sender, KeyEventArgs e)`):

```
this.KeyDown += new KeyEventHandler(OnButtonKeyDown);  
paintSnake(startingPoint);  
currentPosition = startingPoint;
```

```
// генерації їжі  
for (int n = 0; n < 10; n++) paintBonus(n);
```

За відрисовку змійки та їжі тут відповідають методи `paintSnake()` та `paintBonus()` відповідно, які пропонується реалізувати самостійно.