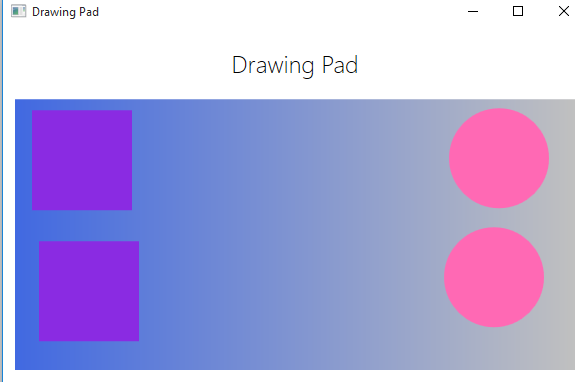
Student: Brian Johnston

Class: 2362

Assignment: 3-3

Printscreen:



Code:

DrawingPad.Xaml.CS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace Drawing

{

public partial class DrawingPadWindow : Window

{

public DrawingPadWindow()

{

InitializeComponent();

}

private void drawingCanvas\_MouseLeftButtonDown(object sender, MouseButtonEventArgs e)

{

Point mouseLocation = e.GetPosition(this.drawingCanvas);

Square mySquare = new Square(100);

if (mySquare is IDraw)

{

IDraw drawSquare = mySquare;

drawSquare.SetLocation((int)mouseLocation.X, (int)mouseLocation.Y);

drawSquare.Draw(drawingCanvas);

}

if (mySquare is IColor)

{

IColor colorSquare = mySquare;

colorSquare.SetColor(Colors.BlueViolet);

}

}

private void drawingCanvas\_MouseRightButtonDown(object sender, MouseButtonEventArgs e)

{

Point mouseLocation = e.GetPosition(this.drawingCanvas);

Circle myCircle = new Circle(100);

if (myCircle is IDraw)

{

IDraw drawCircle = myCircle;

drawCircle.SetLocation((int)mouseLocation.X, (int)mouseLocation.Y);

drawCircle.Draw(drawingCanvas);

}

if (myCircle is IColor)

{

IColor colorCircle = myCircle;

colorCircle.SetColor(Colors.HotPink);

}

}

}

}

Color.cs:

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Media;

using System.Windows.Shapes;

using System.Windows.Controls;

namespace Drawing

{

class Circle : IDraw, IColor

{

private int diameter;

private int locX = 0, locY = 0;

private Ellipse circle = null;

public Circle(int diameter)

{

this.diameter = diameter;

}

void IDraw.SetLocation(int xCoord, int yCoord)

{

this.locX = xCoord;

this.locY = yCoord;

}

void IDraw.Draw(Canvas canvas)

{

if (this.circle != null)

{

canvas.Children.Remove(this.circle);

}

else

{

this.circle = new Ellipse();

}

this.circle.Height = this.diameter;

this.circle.Width = this.diameter;

Canvas.SetTop(this.circle, this.locY);

Canvas.SetLeft(this.circle, this.locX);

canvas.Children.Add(this.circle);

}

void IColor.SetColor(Color color)

{

if (this.circle != null)

{

SolidColorBrush brush = new SolidColorBrush(color);

this.circle.Fill = brush;

}

}

}

}

Iconor.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Media;

namespace Drawing

{

interface IColor

{

void SetColor(Color color);

}

}

Idraw.CS

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Controls;

namespace Drawing

{

interface IDraw

{

void SetLocation(int xCoord, int yCoord);

void Draw(Canvas canvas);

}

}

Square.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Media;

using System.Windows.Shapes;

using System.Windows.Controls;

namespace Drawing

{

class Square : IDraw, IColor

{

private int sideLength;

private int locX = 0, locY = 0;

private Rectangle rect = null;

public Square(int sideLength)

{

this.sideLength = sideLength;

}

void IDraw.SetLocation(int xCoord, int yCoord)

{

this.locX = xCoord;

this.locY = yCoord;

}

void IDraw.Draw(Canvas canvas)

{

if (this.rect != null)

{

canvas.Children.Remove(this.rect);

}

else

{

this.rect = new Rectangle();

}

this.rect.Height = this.sideLength;

this.rect.Width = this.sideLength;

Canvas.SetTop(this.rect, this.locY);

Canvas.SetLeft(this.rect, this.locX);

canvas.Children.Add(this.rect);

}

void IColor.SetColor(Color color)

{

if (this.rect != null)

{

SolidColorBrush brush = new SolidColorBrush(color);

this.rect.Fill = brush;

}

}

}

}