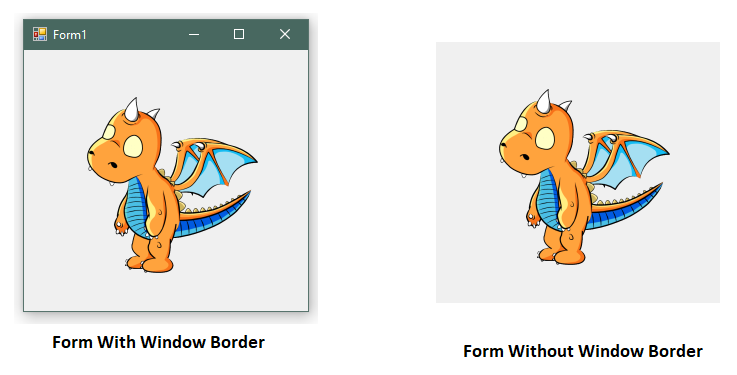
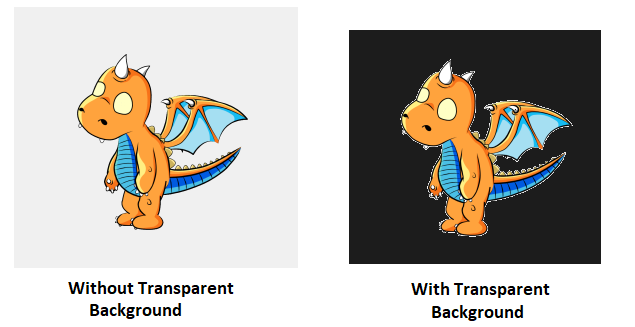
Background Transparency

In Windows everything on screen is necessarily just a Windows Form based application. That is, it is a program with its own window called form. C# is developed by windows to facilitate features like this. Our project requires that the background of the current frame being displayed is transparent. That too needs to be done quite efficiently. Windows Form application provides many helpful tool for the same. We can quite easily remove border of the current window just by changing few flags and variables. 

Also as the dragon is always going to stay at the topmost level, so the isTopMost flag is set true.

Now to get background transparency first we tried to use very simple method provided by windows form application. That is to set the background transparency key. Which basically means that whenever the given RGB values are detected in the current frame they we will be set to transparent.

But the issue with this method is of aliasing. Which is more prominently visible against dark background as can be seen in the diagram. So better solution need to searched for. For this we did research. After doing study we come to know better alternative which would solve this problem for us. The idea is to use special style of Windows Form called Layered Windows Style. Layered Windows Style allows us to set the specific part of form to transparent. 

For this following modules are required to be imported to program:

* UpdateLayeredWindow
* GetDC
* ReleaseDC
* CreateCompatibleDC
* DeleteDC
* SelectObject
* DeleteObject

These modules are provided in the DLLs user32.dll and gdi32.dll.

After doing this new windows form class is required to be made which will be used as base class. Following is code for the same:

using System;

using System.Drawing;

using System.Drawing.Imaging;

using System.Windows.Forms;

using System.Runtime.InteropServices;

namespace TransParentModule

{

class Win32

{

public enum Bool

{

False = 0,

True

};

[StructLayout(LayoutKind.Sequential)]

public struct Point

{

public Int32 x;

public Int32 y;

public Point(Int32 x, Int32 y)

{

this.x = x;

this.y = y;

}

}

[StructLayout(LayoutKind.Sequential)]

public struct Size

{

public Int32 cx;

public Int32 cy;

public Size(Int32 cx, Int32 cy)

{

this.cx = cx;

this.cy = cy;

}

}

[StructLayout(LayoutKind.Sequential, Pack = 1)]

struct ARGB

{

public byte blue;

public byte green;

public byte red;

public byte alpha;

}

[StructLayout(LayoutKind.Sequential, Pack = 1)]

public struct BLENDFUNCTION

{

public byte blendOp;

public byte blendFlags;

public byte sourceConstantAlpha;

public byte alphaFormat;

}

public const Int32 ULW\_COLORKEY = 0x00000001;

public const Int32 ULW\_ALPHA = 0x00000002;

public const Int32 ULW\_OPAQUE = 0x00000003;

public const Int32 AC\_SRC\_OVER = 0x00;

public const Int32 AC\_SRC\_ALPHA = 0x01;

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Design", "CA1060:MovePInvokesToNativeMethodsClass")]

[DllImport("user32.dll", ExactSpelling = true, SetLastError = true)]

public static extern Bool UpdateLayeredWindow(IntPtr hwnd, IntPtr hdcDst, ref Point pptDst, ref Size psize, IntPtr hdcSrc, ref Point pprSrc, Int32 crKey, ref BLENDFUNCTION pblend, Int32 dwFlags);

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Design", "CA1060:MovePInvokesToNativeMethodsClass")]

[DllImport("user32.dll", ExactSpelling = true, SetLastError = true)]

public static extern IntPtr GetDC(IntPtr hWnd);

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Design", "CA1060:MovePInvokesToNativeMethodsClass")]

[DllImport("user32.dll", ExactSpelling = true)]

public static extern int ReleaseDC(IntPtr hWnd, IntPtr hDC);

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Design", "CA1060:MovePInvokesToNativeMethodsClass")]

[DllImport("gdi32.dll", ExactSpelling = true, SetLastError = true)]

public static extern IntPtr CreateCompatibleDC(IntPtr hDC);

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Design", "CA1060:MovePInvokesToNativeMethodsClass")]

[DllImport("gdi32.dll", ExactSpelling = true, SetLastError = true)]

public static extern Bool DeleteDC(IntPtr hdc);

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Design", "CA1060:MovePInvokesToNativeMethodsClass")]

[DllImport("gdi32.dll", ExactSpelling = true)]

public static extern IntPtr SelectObject(IntPtr hDC, IntPtr hObject);

[System.Diagnostics.CodeAnalysis.SuppressMessage("Microsoft.Design", "CA1060:MovePInvokesToNativeMethodsClass")]

[DllImport("gdi32.dll", ExactSpelling = true, SetLastError = true)]

public static extern Bool DeleteObject(IntPtr hObject);

}

public class TransparentForm : Form

{

public TransparentForm()

{

FormBorderStyle = FormBorderStyle.None;

TopMost = true;

}

public void SetBitmap(Bitmap bitmap)

{

SetBitmap(bitmap, 255);

}

public void SetBitmap(Bitmap bitmap, byte opacity)

{

if (bitmap.PixelFormat != PixelFormat.Format32bppArgb)

throw new ApplicationException("The bitmap must be 32ppp with alpha-channel.");

IntPtr screenDc = Win32.GetDC(IntPtr.Zero);

IntPtr memDc = Win32.CreateCompatibleDC(screenDc);

IntPtr hBitmap = IntPtr.Zero;

IntPtr oldBitmap = IntPtr.Zero;

try

{

hBitmap = bitmap.GetHbitmap(Color.FromArgb(0));

oldBitmap = Win32.SelectObject(memDc, hBitmap);

Win32.Size size = new Win32.Size(bitmap.Width, bitmap.Height);

Win32.Point pointSource = new Win32.Point(0, 0);

Win32.Point topPos = new Win32.Point(Left, Top);

Win32.BLENDFUNCTION blend = new Win32.BLENDFUNCTION()

{

blendOp = Win32.AC\_SRC\_OVER,

blendFlags = 0,

sourceConstantAlpha = opacity,

alphaFormat = Win32.AC\_SRC\_ALPHA

};

Win32.UpdateLayeredWindow(Handle, screenDc, ref topPos, ref size, memDc, ref pointSource, 0, ref blend, Win32.ULW\_ALPHA);

}

finally

{

Win32.ReleaseDC(IntPtr.Zero, screenDc);

if (hBitmap != IntPtr.Zero)

{

Win32.SelectObject(memDc, oldBitmap);

Win32.DeleteObject(hBitmap);

}

Win32.DeleteDC(memDc);

}

}

protected override CreateParams CreateParams

{

get

{

CreateParams cp = base.CreateParams;

cp.ExStyle |= 0x00080000 | 0x20; // To Set Windows Layered And Transparent + Click Through

return cp;

}

}

}

}