**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC AN GIANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**ĐỀ TÀI**

**FLASH FORM**

**THÀNH VIÊN NHÓM**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **HỌ VÀ TÊN** | **MSSV** | **LỚP** |
| 1 | Phạm Thanh Lộc | DTH216004 | DH22TH3 |
| 2 | Võ Phước Lộc | DTH216006 | DH22TH3 |
| 3 | Nguyễn Đặng Gia Tường | DTH216237 | DH22TH3 |
| 4 | Trần Quang Minh | DTH216023 | DH22TH3 |

**AN GIANG, THÁNG 1 NĂM 2024**

.

1. Giới thiệu

Flash form là một thành phần quan trong các giao diện thường của một chương trình quản lý. Khi chạy một chương trình thì hệ thống sẽ chờ load chương trình, cấu hình hệ thống vì vậy ta cần có form thể hiện khoản thời gian chờ đó để tăng trải nghiệm người dùng, tránh trường hợp khi chương trình đang xử lý, mà người dùng không biết lại tiếp tục thao tác các chức năng khác, có thể làm cho ứng dụng bị treo.

1. Timer

Timer là một lớp trong thư viện System.Windows.Forms được sử dụng để tạo bộ hẹn giờ.

Timer rất hữu ích cho các tác vụ như:

Cập nhật giao diện người dùng theo thời gian thực

Thực hiện các tác vụ lặp lại

Đo thời gian thực hiện một công việc

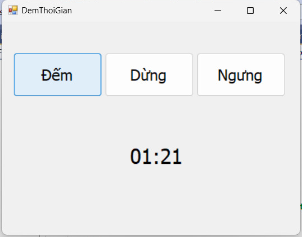
Một số thuộc tính của Timer:

| **Thuộc tính** | **Mô tả** |
| --- | --- |
| Interval | Thiết lập giá trị là một số nguyên. Giá trị nguyên này là thời lượng của một chu kì tính |
| Enable | Thiết lập giá trị True hoặc False. Nếu là giá trị True thì điều khiển Timer hoạt động, nếu là giá trị False thì điều khiển Timer không hoạt động |

Một số phương thức thường dùng của Timer:

| **Phương thức** | **Mô tả** |
| --- | --- |
| Start() | Start() kích hoạt điều khiển Timer hoạt động. Phương thức này tương ứng với việc thiết lập giá trị thuộc tính Enable là True |
| Stop() | Dừng hoạt động của điều khiển Timer. Phương thức này tương ứng với việc thiết lập giá trị thuộc tính Enable là False |

**Ví dụ:**



*Ứng dụng đơn giản về timer*

* **Code**

|  |
| --- |
| ﻿using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace FlashForm  {  public partial class DemThoiGian : Form  {  public DemThoiGian()  {  InitializeComponent();  }  int dem;  private void btnDem\_Click(object sender, EventArgs e)  {  timer1.Start();  }  private void timer1\_Tick(object sender, EventArgs e)  {  int mi, se;  dem++;  mi = dem/60;  se = dem % 60;  label1.Text = (mi < 9 ? "0" + mi : mi + "" )+ ":"+ (se < 9 ? "0" + se : se + " ");  }  private void button1\_Click(object sender, EventArgs e)  {  timer1.Stop();  }  private void button2\_Click(object sender, EventArgs e)  {  dem = 0;  label1.Text = "00:00";  timer1.Stop();  }  private void DemThoiGian\_Load(object sender, EventArgs e)  {  }  }  } |

1. Progress bar

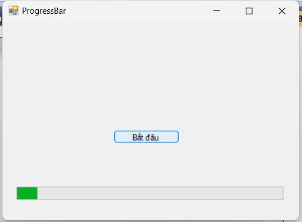
Khi lập trình ứng dụng Winform, nếu chúng ta xử lý một khối lượng lớn dữ liệu, thời gian chờ đợi quá trình xử lý kết thúc tương đối lâu ví dụ như tải file, tìm kiếm dữ liệu,v.v… thì chúng ta có thể sử dụng điều khiển ProgressBar để hiển thị quá trình theo phần trăm để cho biết chương trình xử lý đã được bao nhiêu phần trăm, sắp xong hay chưa.

| **Thuộc tính** | **Mô tả** |
| --- | --- |
| Maximum | Giá trị tối đa của ProgressBar. Khi ProgressBar được lấp đầy nghĩa là ProgressBar đã đạt giá trị Maximum |
| Minimum | Giá trị nhỏ nhất của ProgressBar. Khi ProgressBar trông rỗng nghĩa là ProgressBar đang có giá trị Minimum |
| Value | Giữ giá trị hiện tại của ProgressBar, giá trị này nằm trong đoạn Minimum và Maximum |
| Style | Kiểu hiển thị của ProgressBar |
| Step | Lượng giá trị thêm vào Value khi phương thức PerformStep() được gọi |

Một số phương thức của ProgressBar:

| **Phương thức** | **Mô tả** |
| --- | --- |
| PerformStep() | Phương thức giúp tăng ProgressBar. Giá trị tăng là giá trị được thiết lập trong thuộc tính Step |
| Increment(<giá trị>) | Phương thức giúp tang ProgressBar. Giá trị tăng là tham số đầu vào <giá trị> của phương thức |

**Ví dụ:**



*Ứng dụng đơn giản về progress bar*

* **Code**

|  |
| --- |
| **﻿**using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace FlashForm  {  public partial class ProgressBar : Form  {  public ProgressBar()  {  InitializeComponent();  }    private void timer1\_Tick(object sender, EventArgs e)  {  progressBar1.Increment(1);  if(progressBar1.Value == 100)  {  timer1.Stop();  }  }    List<int> ints = new List<int>();  private void button1\_Click(object sender, EventArgs e)  {  timer1.Start();    }  }  } |

1. Flash Form

Flash Form là một màn hình hiển thị tạm thời khi khởi động ứng dụng C#. Nó thường hiển thị logo, tên ứng dụng, thông tin bản quyền hoặc thanh tiến trình để thông báo cho người dùng rằng ứng dụng đang được khởi chạy.

* Cách thức hoạt động

Flash Form thường được triển khai bằng cách sử dụng một hình ảnh hoặc một giao diện người dùng (UI) đơn giản được hiển thị trước khi giao diện chính của ứng dụng được tải.

* Lợi ích của Flash Form

1. Tạo ấn tượng đầu tiên tốt: Flash Form giúp tạo ấn tượng đầu tiên tốt cho người dùng bằng cách hiển thị logo và tên ứng dụng một cách chuyên nghiệp.

2. Che giấu thời gian khởi động: Flash Form có thể che giấu thời gian khởi động ứng dụng, đặc biệt hữu ích nếu ứng dụng của bạn cần tải nhiều dữ liệu hoặc thực hiện các thao tác khởi tạo phức tạp.

3. Hiển thị thông tin hữu ích: Flash Form có thể hiển thị thông tin hữu ích cho người dùng, chẳng hạn như thông tin bản quyền, phiên bản ứng dụng hoặc trạng thái kết nối.

4. Tăng tính chuyên nghiệp: Sử dụng Flash Form giúp tăng tính chuyên nghiệp cho ứng dụng của bạn, khiến nó trông giống như một sản phẩm được đầu tư kỹ lưỡng.

5. Giảm thiểu sự nhàm chán: Flash Form có thể giúp giảm thiểu sự nhàm chán cho người dùng trong khi chờ ứng dụng khởi động.

* **Lưu ý:**

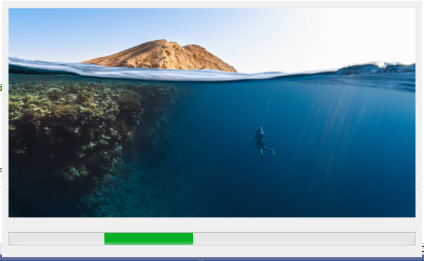
Không nên để Splash Screen hiển thị quá lâu, vì nó có thể khiến người dùng cảm thấy khó chịu.

Nên đảm bảo rằng Splash Screen được thiết kế đẹp mắt và phù hợp với giao diện người dùng của ứng dụng.

Nên sử dụng Splash Screen một cách tiết kiệm, chỉ khi thực sự cần thiết.

**Ví dụ**

|  |
| --- |
| namespace FlashForm  {  public partial class MainForm : Form  {  public MainForm()  {  Thread t = new Thread(new ThreadStart(StartForm));  t.Start();*chạy luồng Flash*  Thread.Sleep(5000); *mô phỏng thời gian chờ*  InitializeComponent();  t.Abort();  }  public void StartForm()  {  Application.Run(new FlashForm());  }  private void MainForm\_Load(object sender, EventArgs e)  {  }  }  } |



*Flash form*

1. Kết luận

Qua trình bày trên chúng ta đã biết cách tạo một Flash Form cũng như một số thành phần thú vị khác của c#. Sử dụng Splash Screen có thể mang lại nhiều lợi ích cho ứng dụng của bạn. Tuy nhiên, bạn nên sử dụng nó một cách hợp lý để tránh gây khó chịu cho người dùng.

1. Nguồn tham khảo

<https://freetuts.net>

<https://www.w3schools.com>

https://www.youtube.com/@foxlearn

https://www.youtube.com/@KTeam