

Sintesi vocale con Visual C# "PcCheParla"

Progetto Visual C# "PcCheParla" - Classe IVBI ITIS Teramo 2014/2015 - Prof. Mauro De Berardis



Il progetto PcCheParla illustrato in queste pagine e adatto agli studenti con competenze di base del linguaggio Visual C#, realizza un TTS - Text to Speech converter, ovvero un convertitore da testo a parlato. Il testo può essere letto oppure registrato in un file wav che può essere successivamente ascoltato con un qualsiasi player audio.

Come funziona l'applicazione?

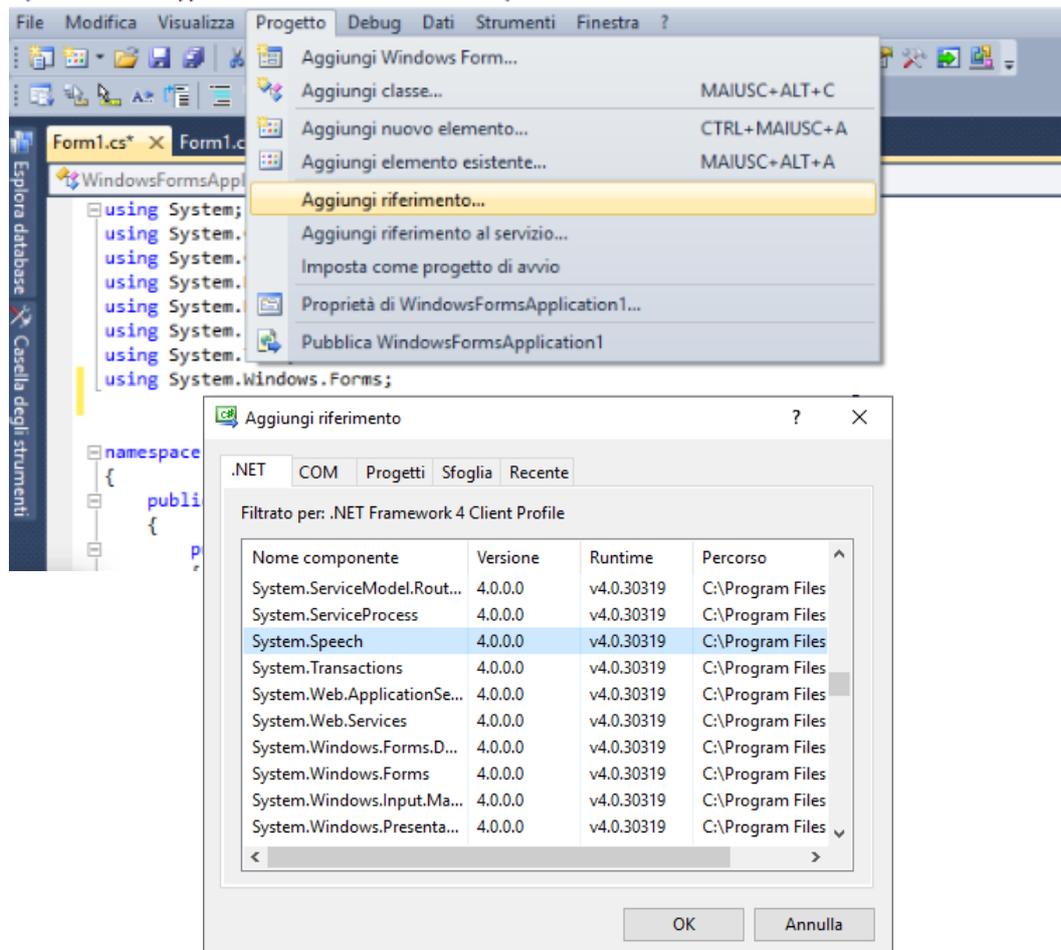
1. Il testo inserito (o copiato con Ctrl-C e incollato con Ctrl-V) viene letto con la voce standard di Windows cliccando sul bottone **Leggi**. Quando è attiva la lettura, non può essere effettuata la registrazione
2. I bottoni **Pausa** e **Riparti** consentono di interrompere la lettura e riprenderla successivamente
3. Il bottone **Stop** consente di resettare la lettura e di abilitare la registrazione
4. Il bottone **Clear** pulisce il testo inserito
5. Il bottone **PathName** consente di selezionare la cartella e il nome del file wav che si vuole salvare
6. Il bottone **Registra** consente di registrare il parlato relativo al testo inserito, nella cartella e con il nome selezionati con il bottone PathName
7. Le due comboBox permettono di "regolare" la velocità di lettura e il volume prima di leggere o di registrare il testo. La qualità del parlato è accettabile: le soluzioni professionali di TTS sono molto più complesse e utilizzano altre informazioni relative ad esempio alle pause, all'intonazione e alle pronuncia di parole particolari.

La progettazione visuale del form è abbastanza semplice e il nome e il tipo dei controlli sono facilmente riconoscibili. In particolare:

- rTesto è la richTextBox in cui inserire il testo da convertire
- bLeggi, bPausa, bRiparti, bStop, bClear, bPathName, bRegistra sono i bottoni
- gBox1, gBox2, gBox3 sono le groupBox che contengono rispettivamente i parametri di velocità e volume, i bottoni per la lettura, i controlli per la registrazione
- comboVelocità e comboVolume sono le comboBox che permettono di impostare la velocità di lettura da -10 a 10 (valore di default 0) e il volume da 0 a 100 (valore di default 50)
- saveFileDialog1 è la finestra di dialogo che deve essere inserita nel form
- gli altri controlli sono label: lPathName è la label che contiene il nome del percorso scelto per memorizzare il file wav

Codice Visual C#

Una volta creato un nuovo progetto Visual C# e scelta un'applicazione windows Form, per poter utilizzare la classe .NET SpeechSynthesizer, occorre includere nel progetto il riferimento a System.Speech



A questo punto, si costruisce il form visuale con i controlli indicati nella prima figura e il codice c# iniziale proposto dall'IDE viene completato "scatenando" i vari eventi e copiando, o rielaborando, il codice utilizzato per questo progetto.

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Speech;
using System.Speech.Synthesis;

using System.IO;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            SpeechSynthesizer Lettore = new SpeechSynthesizer();
            MemoryStream streamAudio = new MemoryStream();

            private void bLeggi_Click(object sender, EventArgs e)
            {
                if (rTesto.Text == "")
                {
                    MessageBox.Show("Inserire un testo!");
                    return;
                }
                comboVolume.Enabled = false;
                comboVelocità.Enabled = false;
                bRegistra.Enabled = false;
                bPathName.Enabled = false;
                Lettore.Dispose();
                Lettore = new SpeechSynthesizer();
                Lettore.Rate = Convert.ToInt32(comboVelocità.Text);
                Lettore.Volume = Convert.ToInt32(comboVolume.Text);
                Lettore.SpeakAsync(rTesto.Text);
            }

            private void bPausa_Click(object sender, EventArgs e)
            {
                if (Lettore.State == SynthesizerState.Speaking) Lettore.Pause();
            }

            private void bRiparti_Click(object sender, EventArgs e)
            {
                if (Lettore != null)
                {
                    Lettore.Rate = Convert.ToInt32(comboVelocità.Text);
                    Lettore.Volume = Convert.ToInt32(comboVolume.Text);
                    if (Lettore.State == SynthesizerState.Paused) Lettore.Resume();
                }
            }

            private void bStop_Click(object sender, EventArgs e)
            {
                Lettore.Dispose();
                comboVolume.Enabled = true;
                comboVelocità.Enabled = true;
                bRegistra.Enabled = true;
                bPathName.Enabled = true;
            }
        }
    }
}

```

```

private void bRegistra_Click(object sender, EventArgs e)
{
    if (rTesto.Text == "")
    {
        MessageBox.Show("Inserire un testo! ");
        return;
    }
    if (lPathName.Text == string.Empty)
    {
        MessageBox.Show("Selezionare il PathName del file wav da salvare",this.Text);

        return;
    }
    Lettore.Dispose();
    Lettore = new SpeechSynthesizer();
    Lettore.Rate = Convert.ToInt32(comboVelocità.Text);
    Lettore.Volume = Convert.ToInt32(comboVolume.Text);
    Lettore.SetOutputToWaveFile(lPathName.Text);
    Lettore.Speak (rTesto.Text);
    Lettore.SetOutputToDefaultAudioDevice();
    MessageBox.Show("File wav salvato con successo",this.Text);
}
private void comboVelocità_DropDown(object sender, EventArgs e)
{
    comboVelocità.Items.Clear();
    for(int i=-10;i<=10;i++)
    {
        comboVelocità.Items.Add(i);
    }
}
private void comboVolume_DropDown(object sender, EventArgs e)
{
    comboVolume.Items.Clear();
    for(int i=0;i<=100;i++)
    {
        comboVolume.Items.Add(i);
    }
}
private void bPathName_Click(object sender, EventArgs e)
{
    saveFileDialog1.Filter = "File Wav|*.wav";
    if (saveFileDialog1.ShowDialog() == DialogResult.OK)
    {
        lPathName.Text = saveFileDialog1.FileName;
    }
}
private void bClear_Click(object sender, EventArgs e)
{
    rTesto.Text = "";
}
private void bReset_Click(object sender, EventArgs e)
{
    comboVolume.Enabled = true;
    comboVelocità.Enabled = true;
    bRegistra.Enabled = true;
    bPathName.Enabled = true;
}
}
}

```