

# Portal spre **Jumea muzicii**

Aplicație Visual Studio C#

Profesor coordonator:  
Muntean Mihaela

Elev:  
Gligor Sorana-Maria  
XII MI1  
Liceul Teoretic  
„Nicolae Bălcescu”  
Cluj-Napoca

2023

## Cuprins

<b>1. Motivul alegerii temei .....</b>	<b>3</b>
<b>2. Structura lucrării.....</b>	<b>4</b>
<b>3. Descrierea lucrării .....</b>	<b>5</b>
<b>4. Resurse hardware și software necesare .....</b>	<b>15</b>
<b>5. Posibilități de dezvoltare și concluzii.....</b>	<b>16</b>
<b>6. Bibliografie .....</b>	<b>17</b>

## 1. Motivul alegerii temei

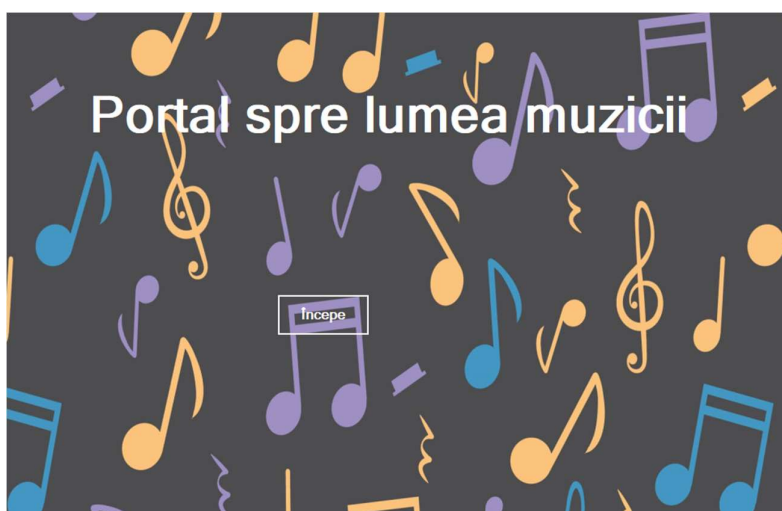
Consider că muzica face parte într-o anumită măsură din viața fiecăruia dintre noi, fie că este vorba de cântat cu vocea, la un instrument, ori ascultat într-un cadru organizat sau pur și simplu când avem o anumită stare. Fiind de mică pasionată de acest domeniu frumos și vast, am ales să fac o aplicație care să introducă utilizatorul în lumea muzicii, învățând câteva concepte de bază și după având ocazia să le aplice pe un pian virtual.

Am decis să fac această aplicație în Visual Studio, întrucât mi s-a părut o oportunitate potrivită pentru a învăța mai multe despre limbajul C#, atât pe cont propriu, cât și prin intermediul orelor de informatică de la școală.

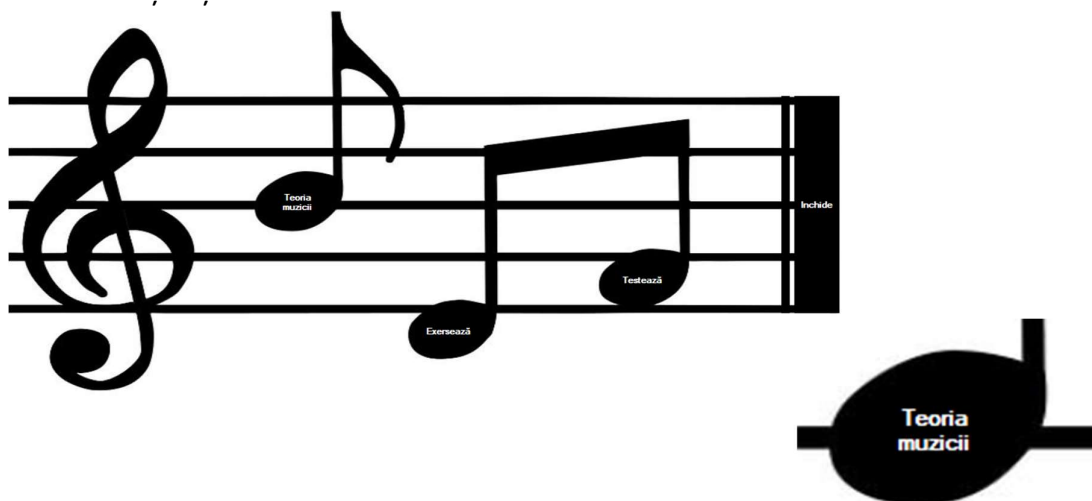
## 2. Structura lucrării

Programul a fost realizat în aplicația Visual Studio, în limbajul de programare C# și cuprinde mai multe componente, care conferă un aspect unitar și plăcut.

Aplicația este formată dintr-o pagină principală, care are doar un buton de start întrucât nu încurajează utilizatorul să mai iasă din lumea muzicii.



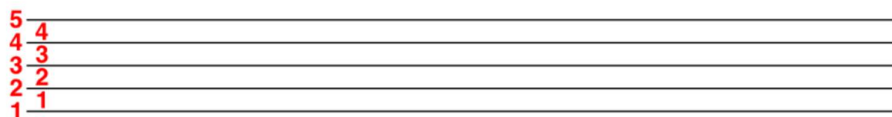
Mai apoi, din meniu se poate observa că programul conține 3 părți principale: tutorialul, un pian unde poate exersa, iar în cele din urmă o testare a cunoștințelor dobândite.



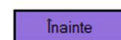
### 3. Descrierea lucrării

Programul se deschide odată cu apariția primului form, iar de aici singura opțiune a utilizatorului este de a intra în lumea muzicii. După aceea, apare cel de-al doilea form, unde se află meniul care îi prezintă variantele, în ordinea ideală a parcurgerii lor.

- 1) *Primul pas ar fi apăsarea butonului „Teoria muzicii” care deschide un alt form cu un tutorial legat de scrierea notelor pe portativ.*



Portativul este format din 5 linii și 4 spații, numerotate de jos în sus.



De aici, utilizatorul poate comuta între imaginile însoțite de text sugestiv pentru a învăța ideea principală despre cum se cântă la pian.

Pentru a realiza acest lucru, am pus toate imaginile necesare într-un folder și le-am numerotat crescător, ca mai apoi cu ajutorul unui pictureBox să le pot afișa pe ecran. Butonul „Înainte” incrementează numărul imaginii, astfel încât să se afișeze următoarea, împreună cu textul aferent, în timp ce butonul „Înapoi” face opusul.



Fiecărei note îi corespunde o literă de la A la G, iar notația începe de pe La. Liniile ajutătoare extind portativul atunci când nu mai este spațiu pentru note.

Înapoi

Înainte

```
private void LoadNextImage()
{
    slidePic.ImageLocation = string.Format(@"poze\{0}.png", imageNumber);
}
```

1 reference

```
private void button1_Click(object sender, EventArgs e)
{
    if (imageNumber < 20)
    {
        imageNumber++;
        label1.Text = Texte[imageNumber];
    }
    LoadNextImage();
}
```

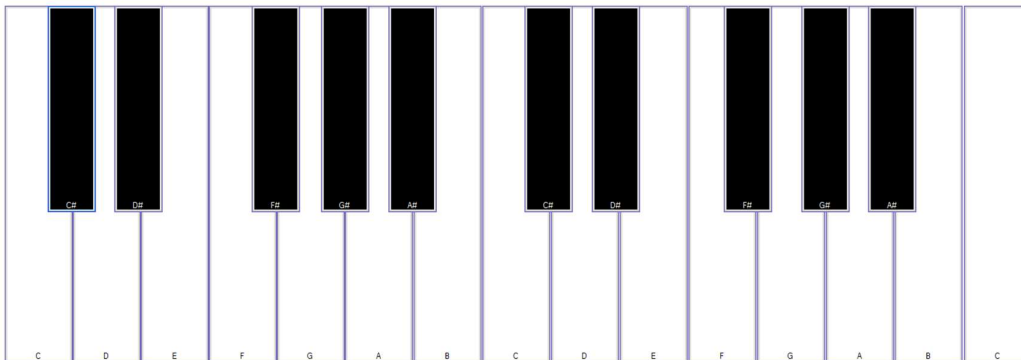
1 reference

```
private void button2_Click(object sender, EventArgs e)
{
    if (imageNumber > 1)
    {
        imageNumber--;
        label1.Text = Texte[imageNumber];
    }
    LoadNextImage();
}
```

În ceea ce privește textul, am utilizat un dicționar, unde fiecărui câmp de la 1 la 20 îi corespunde un text pentru o anumită imagine.

```
private Dictionary<int, string> Texte = new Dictionary<int, string>()
{
    [1] = "Portativul este format din 5 linii și 4 spații, numerotate de jos în sus.",
    [2] = "Cheile alocă fiecărei note un loc pe portativ, fie pe o linie, fie pe un spațiu.\nCe",
    [3] = "Cheia Sol atribuie oricărei note plasate pe a doua linie a portativului valoarea Sol",
    [4] = "Astfel, nota de deasupra notei Sol va fi La (A), după Si (B) și așa mai departe.",
    [5] = "Fiecărei note îi corespunde o literă de la A la G, iar notația începe de pe La.\nLin",
    [6] = "Cheile arată locul notei pe portativ. De aceea, cheia Fa marchează locul notei Fa, i",
    [7] = "De obicei, partitura pentru pian conține câte 2 perechi de portative, unde cheia Sol",
    [8] = "La pian, fiecărei note de pe portativ îi corespunde o clapă.",
    [9] = "Un semiton reprezintă distanța dintre două clape alăturate de pe pian.\nSpre exemplu",
    [10] = "Un ton este format din 2 semitonuri, spre exemplu între clapele 1 și 3 este un ton."
```

2) După ce utilizatorul este familiarizat cu noțiunile, poate trece la pasul următor, și anume să încerce să cânte la un pian virtual cu 2 octave.



Am creat pianul din butoane, pe care le-am poziționat să arate cât mai realist.

25 references

```
public void Play(string notePath)
{
    System.Media.SoundPlayer player = new System.Media.SoundPlayer(notePath);
    player.Play();
}
```

1 reference

```
private void c4_Click(object sender, EventArgs e)
{
    string fileName = "c4.wav";
    string path = Path.Combine(Environment.CurrentDirectory, @"note\", fileName);
    Play(path);
}
```

Pentru sunete, am descărcat un folder de pe site-ul freesound.org cu notele de care aveam nevoie și am folosit clasa SoundPlayer.

3) Nu în cele din urmă, utilizatorul are posibilitatea de a-și testa cunoștințele dobândite, apăsând pe butonul „Testează”.

Încearcă și tu!



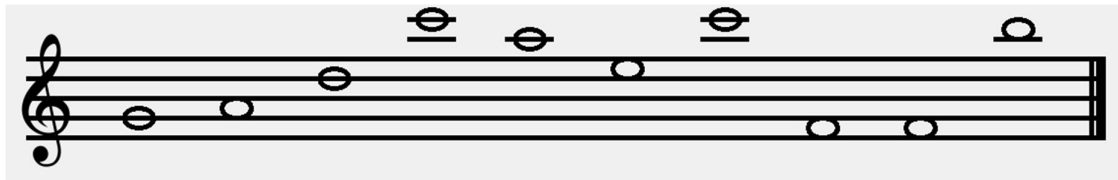
Încearcă să cânti notele generate, iar după te poți verifica.

OK

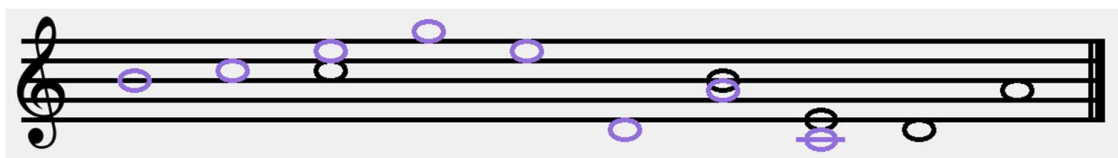
Acesta este întâmpinat de un MessageBox, iar mai apoi se deschide un form unde se află exact același pian, însă însoțit acum de 3 butoane: „Începe”, „Verifică” și „Cântă”.



Odată ce butonul „Începe” este apăsător, se generează aleatoriu 10 note pe un portativ.



De aici, utilizatorul trebuie să încerce să apese exact notele indicate, în ordinea precizată. Pe parcurs sau doar după ce termină, poate apăsa butonul „Verifică” pentru a vedea ce note a cântat, scrise cu violet. Astfel, își poate da seama cât de bine a reușit să învețe poziția notelor pe portativ.



Pentru a desena portavul cu cheia sol și notele, am folosit mai multe elemente grafice din clasa Graphics, cum ar fi linii sau elipse. Am folosit funcția DrawImage



pentru a adăuga o poză cu cheia sol la începutul portativului, întrucât este un element foarte greu de desenat în Visual Studio.

```
g = this.CreateGraphics();
Pen p = new Pen(Color.Black, 5);
Point p1 = new Point(100, 60);
Point p2 = new Point(100, 80);
Point p3 = new Point(100, 100);
Point p4 = new Point(100, 120);
Point p5 = new Point(100, 140);

g.DrawLine(p, p1, p1 + new Size(1100, 0));
g.DrawLine(p, p2, p2 + new Size(1100, 0));
g.DrawLine(p, p3, p3 + new Size(1100, 0));
g.DrawLine(p, p4, p4 + new Size(1100, 0));
g.DrawLine(p, p5, p5 + new Size(1100, 0));

g.DrawLine(p, new Point(1190, 58), new Point(1190, 143));
g.DrawLine(new Pen(Color.Black, 10), new Point(1200, 58), new Point(1200, 143));

Image cheiaSol = Image.FromFile(@"extra\cheie.png");
g.DrawImage(cheiaSol, 75, 25, 90, 155);
```

Mai apoi, am generat aleatoriu 10 coordonate pe axa absciselor, cuprinse între 10 și 150, pentru a se genera 10 note în diferite locații pe portativ. Fiecare astfel de număr a fost salvat într-o listă *generate*, pentru ca mai apoi să poată fi desenate notele.

```
for (int i = 1; i <= 10; i++)
{
    int nr = random.Next(10, 150);
    while (nr % 10 != 0)
        nr = random.Next(10, 150);
    generate.Add(nr);
}

int nota = 100;
foreach (int elem in generate)
{
    nota += 100;
    if (nota <= 1100)
    {
        Point point = new Point(nota, elem);
        draw_note(sender, e, point, p);
    }
}
```

Funcția *draw\_note* desenează o notă întreagă la coordonatele date, adăugând linii ajutătoare dacă nota nu se plasează pe portativ.

3 references

```
private void draw_note(object sender, EventArgs e, Point point, Pen p)
{
    g.DrawEllipse(p, point.X, point.Y, 30, 20);
    if (point.Y == 150)
        g.DrawLine(p, point - new Size(10, -10), point + new Size(40, 10));
    if (point.Y == 30 || point.Y == 20 || point.Y == 10)
        g.DrawLine(p, new Point(point.X, 30) - new Size(10, -10), new Point(point.X, 30) + new Size(40, 10));
    if (point.Y == 10)
        g.DrawLine(p, point - new Size(10, -10), point + new Size(40, 10));
}
```

De data aceasta, la fiecare apăsare a unei clape albe de pe pian, pe lângă că se aude sunetul, este și adăugat în lista *cantate*.

```
private void c4_Click(object sender, EventArgs e)
{
    string fileName = "c4.wav";
    string path = Path.Combine(Environment.CurrentDirectory, @"note\", fileName);
    Play(path);
    cantate.Add(150);
}
```

Astfel, odată ce este apăsat butonul „Verifică”, se vor desena cu culoarea violet primele 10 sunete cântate de către utilizator la pian.

```
Pen p = new Pen(Color.MediumPurple, 5);
int x = 100;
foreach (int nota in cantate)
{
    x += 100;
    if (x <= 1100)
    {
        Point point = new Point(x, nota);
        draw_note(sender, e, point, p);
    }
}
```

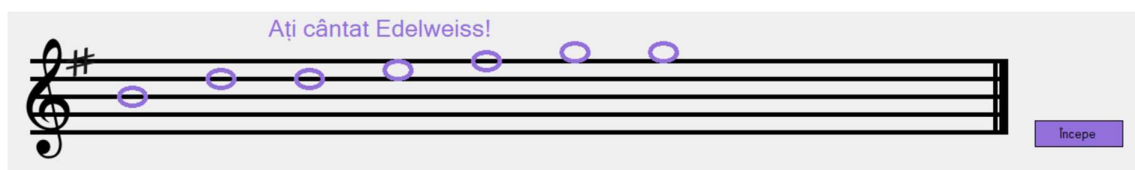
Într-un final, am pregătit mai multe cântece scurte cunoscute într-un folder *Cantece*, de unde programul le citește și le memorează într-o listă definită cu ajutorul unei clase *Cantec*. Pentru fiecare cântec, este salvat titlul, gama în care este scris, numărul de portative și, evident, notele.

```
public class Cantec
{
    public int _nrPortative;
    public List<List<int>> _note = new List<List<int>>();
    public string _titlu;
    public string _gama;
```

Odată ce utilizatorul apasă butonul „Cântă”, se va deschide ultimul form creat de mine și se va alege aleatoriu un cântec din listă. Butonul „Începe” afișează pe portativ primele note ale cântecului, iar butonul „Mai departe” schimbă notele de pe portativ cu cele care urmează.



De data aceasta, apare direct pe portativ fiecare notă cântată la pian cu violet, pentru ca utilizatorul să observe rapid dacă a făcut greșeli. În caz că acesta nu a recunoscut cântecul, la final misterul se va desluși.



Pentru a realiza acest form, am pus în fișier numărul de cântece, iar după aceea fiecare melodie în parte, formată din titlu, gamă și notele “așezate” pe portative.

<pre> 5 Gaudeamus Igitur Do Major do2 sol sol do2 la la la si do2 re2 si do2 mi2 do2 do2 sol sol do2 la la la si do2 re2 si do2 mi2 do2 si do2 re2 re2 mi2 do2 re2 re2 si do2 re2 re2 mi2 do2 re2 re2 do2 si la fa2 mi2 re2 mi2 re2 do2 do2 si la fa2 mi2 re2 sol2 si do2 . Rapsodia română Do Major do do mi mi sol sol fa re mi fa mi fa sol mi do do mi mi sol sol fa re mi fa mi fa sol mi fa la la la la sol sol sol fa fa fa fa mi mi fa mi re re do re mi fa sol la fa mi fa sol mi . </pre>	<pre> Sunetul muzicii Do Major do re mi do mi do mi mi re mi fa fa mi re fa mi fa sol mi sol mi sol sol fa sol la la sol fa la sol do re mi fa sol la la re mi fa sol la si si mi fa sol la si do2 si sib la fa si sol do2 . Edelweiss Sol Major si re2 la2 sol2 re2 do2 si si si do2 re2 mi2 re2 si re2 la2 sol2 re2 do2 si re2 re2 mi2 fa2 sol2 sol2 la2 re2 re2 fa2 mi2 re2 si re2 sol2 mi2 sol2 la2 sol2 fa2 re2 si re2 la2 sol2 re2 do2 si re2 re2 mi2 fa2 sol2 sol2 . </pre>
---	--

Am folosit un dicționar pentru a face conversia între notele din fișier și poziția lor pe portativ. Astfel, nota do se va situa pe poziția 150, nota re pe 140 și tot așa, pâna la nota do3, pe 10. De asemenea, am decis să notez și dacă nota are alterații, spre exemplu exista *dod* pentru do diez, *dob* pentru do bemol și *don* pentru do natural, dar cu becar în față.

```

private Dictionary<string, int> convertireNote = new Dictionary<string, int>()
{
    ["do"] = 150,
    ["dod"] = 151,
    ["dob"] = 152,
    ["don"] = 153,
    ["re"] = 140,
    ["red"] = 141,
    ["reb"] = 142,
    ["ren"] = 143,
    ["mi"] = 130,
    ["mid"] = 131,
    ["mib"] = 132,
    ["min"] = 133,
    ["fa"] = 120,
    ["fad"] = 121,
    ["fab"] = 122,
    ["fan"] = 123,
    ["sol"] = 110,

```

```

StreamReader sr = new StreamReader(@"Cantece.txt");
int nrCantece = Convert.ToInt32(sr.ReadLine());
for (int i = 0; i < nrCantece; i++)
{
    string nume = sr.ReadLine();
    string gama = sr.ReadLine();
    string valori = sr.ReadLine();
    List<List<int>> note = new List<List<int>>();
    int linie = 0;
    while (valori[0] != '.')
    {
        string[] denumireNote = valori.Split(' ');
        List<int> portativ = new List<int>();
        foreach (var denumire in denumireNote)
            portativ.Add(convertireNote[denumire]);
        note.Add(portativ);
        linie++;
        valori = sr.ReadLine();
    }
    Form6.cantece.Add(new Cantec(nume, gama, linie, note));
}

```

Dacă melodia nu este în Do Major, se adaugă alterațiile necesare, iar același lucru se întâmplă și pentru notele alterate.

```

g.DrawImage(cheiaSol, 75, 25, 90, 155);
if (cantece[nrCantec]._gama == "Sol Major")
    g.DrawImage(diez, 140, 45, 33, 33);
if (cantece[nrCantec]._gama == "Fa Major")
    g.DrawImage(bemol, 143, 78, 33, 33);

foreach (int element in cantec[nrPortativ])
{
    nota += 100;
    Point point = new Point(nota, element / 10 * 10);
    if (element % 10 == 1)
        g.DrawImage(diez, nota - 38, element - 8, 35, 35);
    if (element % 10 == 2)
        g.DrawImage(bemol, nota - 40, element - 15, 35, 35);
    if (element % 10 == 3)
        g.DrawImage(becar, nota - 35, element - 8, 25, 35);
    draw_note(sender, e, point, p);
}

```

Pentru fiecare notă apăsată pe pian, se apelează această funcție care scrie instant nota pe portativ, însă doar dacă se suprapune peste o notă deja scrisă cu negru.

```
void apasare_nota(object sender, EventArgs e, int nota)
{
    locatie += 100;
    if (nrPortativ <= nrPortative && locatie <= (cantec[nrPortativ-1].Count()+1) * 100)
    {
        Point point = new Point(locatie, nota);
        draw_note(sender, e, point, pen);
    }
}
```

La ultimul portativ, se pune linie dublă la final și se elimină butonul „Mai departe”.

```
if (nrPortativ == nrPortative)
{
    g.DrawLine(p, new Point(1190, 58), new Point(1190, 143));
    g.DrawLine(new Pen(Color.Black, 10), new Point(1200, 58), new Point(1200, 143));
    button3.Visible = false;
    label1.Visible = true;
}
```

## 4. Resurse hardware și software necesare

Pentru a putea accesa această aplicație, orice utilizator ar avea nevoie de un calculator dotat cu:

- ♪ sistem de operare Windows
- ♪ tastatură
- ♪ mouse
- ♪ placă de bază cu un procesor de minim 1 Ghz
- ♪ minim 8 GB RAM



## 5. Posibilități de dezvoltare și concluzii

Cu cât am lucrat mai mult, cu atât am observat mai multe lucruri care ar putea fi îmbunătățite sau dezvoltate. Spre exemplu, tutorialul ar putea fi extins astfel încât să cuprindă mai multă teorie muzicală, de pildă noțiuni ritmice, mai multe game, intervale sau acorduri. De asemenea, pianul ar putea fi extins sau conectat la tastatură pentru a se putea cânta mai multe note simultan. Totodată, la notele generate ar putea fi adăugate durate și ar putea fi găsită o soluție pentru apăsarea clapelor negre, întrucât acestea pot simboliza note diferite pe portativ în funcție de context. Nu în cele din urmă, proiectul ar putea fi îmbunătățit prin adăugarea unei baze de date, unde să se poată salva utilizatorii pentru a-și menține progresul.

După ore de încercări, remedieri și erori corectate, sunt sigură că mi-am lărgit aria de cunoștințe. Am avut de la început intenția să fac o aplicație legată de muzică, însă nu am știut exact ce să creez. Abia pe parcurs am realizat cum aș putea să-mi modelez proiectul astfel încât să fie folositor și plăcut atât de utilizat, cât și de lucrat la el. Într-un final, am ajuns să am un produs al meu, care simbolizează toate probleme pe care am reușit să le rezolv, toate cazurile la care m-am gândit și pe care le-am tratat, dar mai ales pasiunea mea pentru muzică.



## 6. Bibliografie

- ♪ <https://visualstudio.microsoft.com/>
- ♪ <https://www.musictheory.net/>
- ♪ <https://stackoverflow.com/>
- ♪ <https://freesound.org/people/pinkyfinger/packs/4409/>
- ♪ <https://rashicode.com/image-slider-in-win-form-app-c-sharp/>