

Woordraadselspel met Windows Forms in C#

5INF

6 juni 2023

1 Inleiding

In deze tutorial zullen we een uitdagend woordraadselspel maken met behulp van Windows Forms in C#. Het doel van het spel is om binnen een bepaalde tijdslimiet een woord te ontrafelen. We zullen gebruikmaken van de functies van Windows Forms om de gebruikersinterface te verbeteren. Laten we beginnen!

2 Projectinstellingen

1. Open je voorkeursontwikkelomgeving (IDE) voor C#-programmering.
2. Maak een nieuw Windows Forms-project.
3. Noem het project "WoordraadselSpel" en klik op "Maken".

3 Vereiste bibliotheken importeren

Voeg de volgende regels toe aan het begin van je codebestand om de benodigde bibliotheken te importeren:

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Windows.Forms;
```

4 Woordraadsellogica maken

In de code van het formulier, declareer een lijst van woorden die in het spel gebruikt zullen worden:

```
List<string> woorden = new List<string> { "appel", "banaan", "kers", "sinaasappel", "aardbe"
```

Gebruik de Random-klasse van .NET om een willekeurig woord uit de lijst te selecteren:

```
Random random = new Random();
string geselecteerdWoord = woorden[random.Next(0, woorden.Count)];
```

Maak een methode genaamd `WoordMixer` om het geselecteerde woord te mixen:

```
static string WoordMixer(string woord)
{
    char[] karakters = woord.ToArray();
    Random random = new Random();

    for (int i = 0; i < karakters.Length; i++)
    {
        int j = random.Next(i, karakters.Length);
        char temp = karakters[i];
        karakters[i] = karakters[j];
        karakters[j] = temp;
    }

    return new string(karakters);
}
```

Roep de `WoordMixer`-methode aan en sla het gemixte woord op in een variabele:

```
string gemixtWoord = WoordMixer(geselecteerdWoord);
```

5 Implementeren van de spellogica

Voeg een `Label` toe aan het formulier en gebruik deze om een welkomstbericht en instructies voor het spel weer te geven:

Voeg een `TextBox` toe waarin de speler zijn gok kan invoeren:

Voeg een `Button` toe waarmee de speler zijn gok kan controleren:

Voeg een `Label` toe waarin de resterende tijd wordt weergegeven Voeg een `Timer` toe aan het formulier om de tijd bij te houden Implementeer de `Timer.Tick`-event om de resterende tijd bij te werken en het tijdslimiet te controleren:

```
private void Timer_Tick(object sender, EventArgs e)
{
    if (resterendeTijd > 0)
    {
        resterendeTijd--;
        tijdLabel.Text = $"Resterende tijd: {resterendeTijd} seconden";
    }
    else
    {
        timer.Stop();
        MessageBox.Show("Tijd is op! Je bent door je tijdslimiet heen.");
    }
}
```

Implementeer de klik voor de `Button` om de gok van de speler te controleren:

6 Het spel testen

Bouw en voer het project uit om het spel te spelen. Voer je gokken in binnen de tijdslimiet om het woord te ontrafelen. Het spel geeft feedback op je gokken en meldt het wanneer je door je tijdslimiet heen bent of het woord succesvol hebt ontrafeld.

Gefeliciteerd! Je hebt een uitdagend Woordraadselspel gemaakt met behulp van Windows Forms in C# en hebt enkele .NET-functies toegepast. Je kunt het spel verder uitbreiden door een score-systeem, een grotere woordendatabase of zelfs een grafische gebruikersinterface (GUI) toe te voegen.

Veel plezier met het spelen en aanpassen van je Woordraadselspel!

7 snake

In dit tutorial zullen we de gegeven code voltooien om een functionerende kop van de slang te creëren in het Snake-spel. De code is geschreven in C# en gebruikt Windows Forms voor de gebruikersinterface.

8 Stap 1: Het Begrijpen van de Code

Laten we eerst naar de gegeven code kijken en de structuur en componenten begrijpen. Open het bestand "Form1.cs" in je ontwikkelomgeving.

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Runtime.ExceptionServices;
using System.Runtime.Remoting.Metadata.W3cXsd2001;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace snake
{
    public partial class Form1 : Form
    {
        snake bob;
        Graphics paper;
        Pen pen = new Pen(Color.Black, 2);
    }
}
```

```

public Form1()
{
    InitializeComponent();
}
public void tekenSnake()
{
    for (int i = 0; i < bob.slang.Count; i++)
    {
        if (i == 0)
        {
            paper.FillRectangle(Brushes.Green,
                                bob.slang[i].x,
                                bob.slang[i].y, 50, 50);
        }
        else
        {
            paper.DrawRectangle(pen,
                                bob.slang[i].x,
                                bob.slang[i].y, 50, 50);
        }
    }
}

// Rest van de code...

```

De code bevat een klasse genaamd **Form1**, die het hoofdformulier van de toepassing vertegenwoordigt. De **snake** structuur bevat informatie over de slang, zoals het lichaam van de slang en het aantal levens. De **blokje** structuur vertegenwoordigt een blok of segment van het lichaam van de slang. Daarnaast hebben we de **Richting** enum, die de mogelijke richtingen van de slang definieert.

```

enum Richting
{
    Boven,
    Omlaag,
    Links,
    Rechts
}

```

9 Stap 2: Het Toevoegen van de Slangkop

Om een functionerende kop van de slang te maken, moeten we een nieuw blok toevoegen aan de voorkant van het lichaam van de slang.

```

public void UpdateSnakeHead()
{
    int headX = bob.slang[0].x;
    int headY = bob.slang[0].y;

    // Bepaal de nieuwe positie van de slangkop op
    // basis van de huidige richting

    switch (bob.Richting)
    {
        case Richting.Boven:
            headY -= 50;
            break;
        case Richting.Omlaag:
            headY += 50;
            break;
        case Richting.Linkskant:
            headX -= 50;
            break;
        case Richting.Rechts:
            headX += 50;
            break;
    }

    blokje newHead = new blokje(headX, headY);

    bob.slang.Insert(0, newHead);
}

```

De `UpdateSnakeHead` methode berekent de nieuwe positie van de slangkop op basis van de huidige richting en voegt een nieuw blok toe aan het begin van het lichaam van de slang.

```

public void opschuiven()
{
    UpdateSnakeHead();

    for (int i = 0; i < bob.slang.Count - 1; i++)
    {
        bob.slang[i + 1] = bob.slang[i];
    }
}

```

De `opschuiven` methode roept nu de `UpdateSnakeHead` methode aan voordat de rest van het lichaam van de slang wordt bijgewerkt.

```

public void tekenSnake()

```

```

{
    for (int i = 0; i < bob.slang.Count; i++)
    {
        if (i == 0)
        {
            paper.FillRectangle(Brushes.Green,
                                bob.slang[i].x, bob.slang[i].y, 50,
                                50);
        }
        else
        {
            paper.DrawRectangle(pen, bob.slang[i].x,
                                bob.slang[i].y, 50, 50);
        }
    }
}

```

De `tekenSnake` methode is ook bijgewerkt om de kop van de slang weer te geven als een gevuld groen rechthoek en de rest van het lichaam van de slang als randen.

10 Stap 3: Omgaan met Toetsenbordinvoer voor Richtingsverandering

Om de speler in staat te stellen de richting van de slang te veranderen, moeten we toetsenbordinvoer afhandelen.

```

private void Form1_KeyDown(object sender,
    KeyEventArgs e)
{
    switch (e.KeyCode)
    {
        case Keys.Up:
            bob.Richting = Richting.Boven;
            break;
        case Keys.Down:
            bob.Richting = Richting.Omlaag;
            break;
        case Keys.Left:
            bob.Richting = Richting.Linken;
            break;
        case Keys.Right:
            bob.Richting = Richting.Rechts;
            break;
    }
}

```

```
}
```

De `Form1.KeyDown` gebeurtenishandler wordt geactiveerd wanneer een toets wordt ingedrukt. Deze handler verandert de richting van de slang op basis van de ingedrukte pijltoets.

11 Stap 4: Het Spel Uitvoeren

Nu kunnen we het spel uitvoeren en de slangkop laten bewegen!

1. Bouw het project en voer het uit.
2. Klik op de "Begin" knop om het spel te starten.
3. Gebruik de pijltoetsen om de richting van de slangkop te veranderen.
4. Klik op de "Opschuiven" knop om de slangkop te verplaatsen en het lichaam van de slang te updaten.

Gefeliciteerd! Je hebt succesvol de slangkop in het Snake-spel geïmplementeerd. Nu kun je verder bouwen aan het spel door het toevoegen van obstakels, voedsel en collision detection om de slang te laten groeien.