

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor SOČ: 12. Tvorba učebních pomůcek, didaktická technologie

Binar

Vypracoval: Marek Havránek

Třída: IT3/A

Škola: Střední škola spojů a informatiky, Tábor,
Bydlinského2474

Kraj: Jihočeský

Konzultant: Ing. Dana Almášiová

Čestné prohlášení

Prohlašuji, že jsem svou práci SOČ vypracoval samostatně a použil (a) jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v seznamu vloženém v práci SOČ. Prohlašuji, že tištěná verze a elektronická verze soutěžní práce SOČ jsou shodné.

Nemám závažný důvod proti zpřístupnění této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) v platném znění.

V.....dne.....

Podpis.....

Obsah

Úvod	5
Funkcionalita programu	6
Procvičování	6
Test	8
Visual Studio	10
Popis aplikace	11
Formuláře	11
Přidávání řádků.....	13
Náhodný generátor čísel	15
Zadávání hodnot a jejich řešení.....	16
Závěr	18
Zdroje	19

Anotace

Cílem této práce bylo vytvořit program ve Visual Studiu, který pomůže studentům pochopit princip dvojkové soustavy. Naučit se jí ovládat a pomocí procvičování se zlepšit. Po dokončení cvičení mají studenti možnost pomocí grafu, který se jim ukáže, porovnávat své výsledky.

Klíčová slova

Dvojková soustava, Procvičování, Porovnávání, Počítání, Rychlost

Úvod

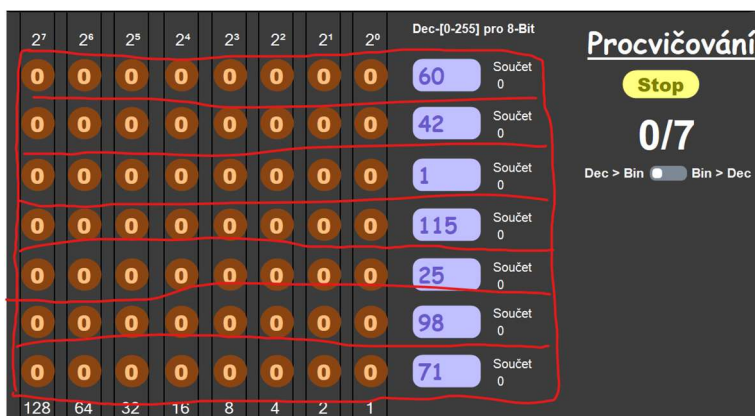
Mým cílem bylo vytvořit program, který hravou formou usnadní převody z dvojkové soustavy do desítkové a naopak. Program je jednoduchý na ovládání. Obsahuje jednoduchý návod k použití. V aplikaci je možnost si zvolit buď procvičování nebo test. Při procvičování je možnost nastavit si směr převodu. Při testu je druh převodu určen.

Funkcionalita programu

Procvičování

Po zmáčknutí tlačítka Start se vygeneruje 7 řádků.

V desítkovém módu je v každém řádku 8 aktivních tlačítek, které se můžou měnit na 1 nebo 0. Hodnoty se mohou měnit pouze pomocí myši. Uživatel má za úkol zadat správnou kombinaci čísel tak, aby odpovídala náhodně vygenerovanému zadání, které je v rozmezí od 0 až 255.



Ve dvojkovém módu je to naopak. Po jeho spuštění se zobrazí 7 řádek s již vyplněnými 8 tlačítky. Tyto hodnoty jsou stálé a nelze je měnit. Vedle těchto tlačítek se nachází oranžová pole. Do těchto polí musí uživatel vepsat správnou hodnotu vygenerovaných kombinací čísel.

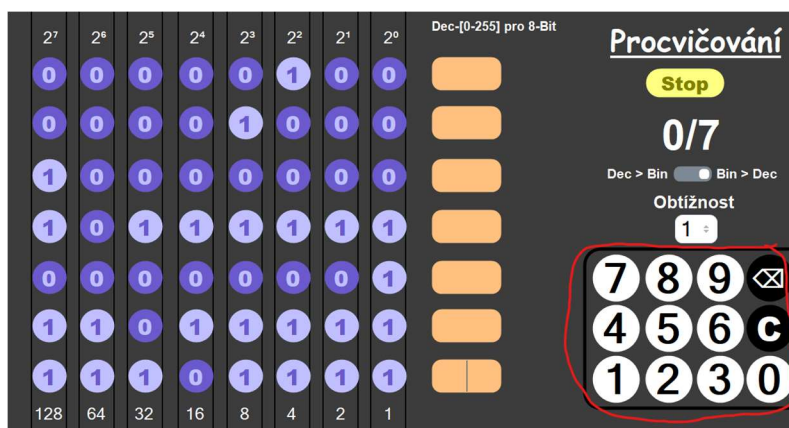


Při správném zadání hodnoty se celá řádka zabarví zeleně. Přibližně po 2 sekundách zmizí. V pravém horním rohu se zobrazí stav již vyřešených řádků ku celku.

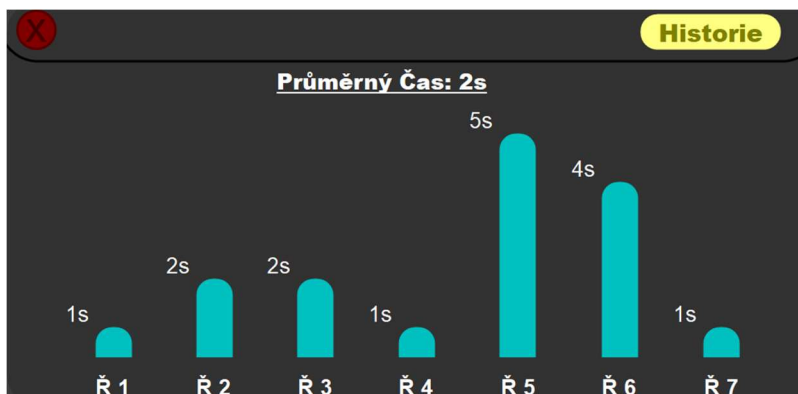


Pokud se hodnota zadá chybně žádná změna se neuskuteční a program čeká na zadání správné hodnoty.

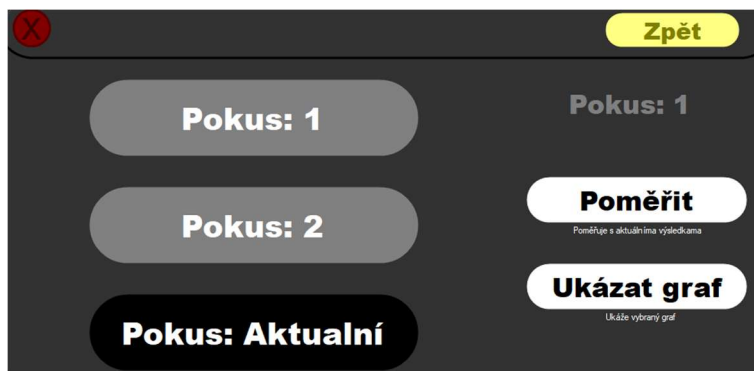
Zadávání číselných hodnot je možné i pomocí myši na numerické klávesnici v pravém dolním rohu.



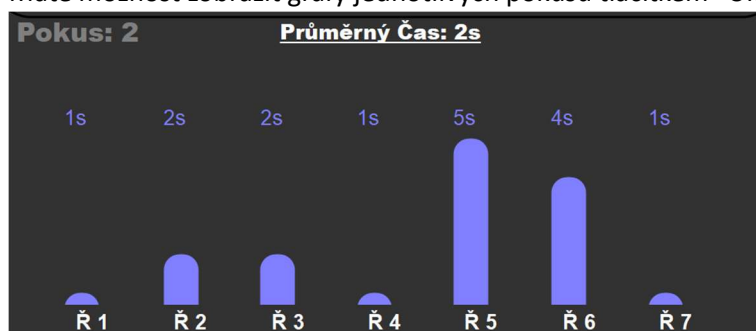
Na pozadí se přičítá čas pro řádku, která se zrovna řeší. Pokud je řádka vyřešená, tak se čas pro danou řádku uloží. Po vyřešení kompletní úlohy se zobrazí graf. Tento graf nám ukáže, jaký čas byl potřeba k vyřešení jednotlivých řádek.



Po stisknutí tlačítka "Historie" se zobrazí seznam předchozích pokusů, které jste již úspěšně dokončili.



Máte možnost zobrazit grafy jednotlivých pokusů tlačítkem “Ukázat graf”.



Nebo je porovnat s aktuálním pokusem pomocí tlačítka “Poměřit”.



Test

Test pracuje na podobném principu jako procvičování, avšak s časovým limitem a omezeným počtem kliknutí na 8 aktivních tlačítek v řádku. Možnosti převodu nelze předem určit.

Po stisknutí tlačítka “Start” test vygeneruje 7 řádků, v nichž jsou obě možnosti převodu vyváženě zastoupeny.

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	Dec-[0-255] pro 8-Bit
1	0	0	0	1	1	0	1	<input type="text"/>
0	0	0	1	0	1	1	1	<input type="text"/>
1	0	0	0	1	1	0	1	<input type="text"/>
0	0	0	1	1	1	0	1	<input type="text"/>
0	0	0	0	0	0	0	0	120
0	0	0	0	0	0	0	0	143
0	0	0	0	0	0	0	0	144
128	64	32	16	8	4	2	1	

Test na 1

Stop

0/7

Zbývá: 297 s
Limit kliknutí: 40

Jakmile je test dokončen, zobrazí se výsledné okno, kde je napsáno, jestli jste v testu uspěl či nikoliv. Důsledkem neúspěchu je buď vyprcháání časového limitu nebo vyplývání počet kliknutí.

Test na 4

Uspěl

Test na 4

Neuspěl

Visual Studio

Visual Studio je vývojové prostředí od společnosti Microsoft, které umožňuje vývojářům softwaru vytvářet a spravovat aplikace. Jedná se o komplexní nástroj, který zahrnuje editor kódu, debugger, kompilátor, ladicí nástroje a mnoho dalších funkcí.

Používá se k vývoji široké škály aplikací, včetně: Konzolové aplikace, Aplikace s grafickým uživatelským rozhraním (GUI), Webové stránky a webové aplikace, Mobilní aplikace, Hry

Uspadňuje vývojářům psaní, ladění, testování a nasazení aplikací. Poskytuje také řadu funkcí, které pomáhají vývojářům být produktivnější, včetně:

IntelliSense: Tato funkce automaticky doplňuje kód a poskytuje návrhy na základě kontextu.

Ladění: Visual Studio obsahuje výkonný debugger, který umožňuje vývojářům krok za krokem procházet kód a identifikovat chyby.

Testování: Visual Studio obsahuje řadu nástrojů pro testování, které vývojářům pomáhají najít a opravit chyby v jejich kódu.

Nasazení: Visual Studio umožňuje vývojářům snadno nasadit své aplikace do produkčního prostředí.

Visual Studio podporuje širokou škálu programovacích jazyků, včetně: C++, C#, Visual Basic .NET, F#, JavaScript, Python, Java, PHP, Go

Visual Studio také podporuje vývoj aplikací v platformách .NET a Windows.

Popis aplikace

Program jsem dělal na platformě Visual Studio pomocí kódu C#.

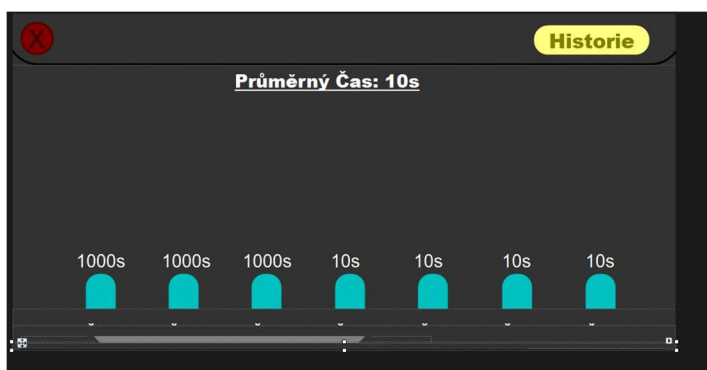
Vytvořená aplikace je stavěna na přednastaveném buildu, který se jmenuje: **Windows Forms App (.NET Framework)**

Formuláře

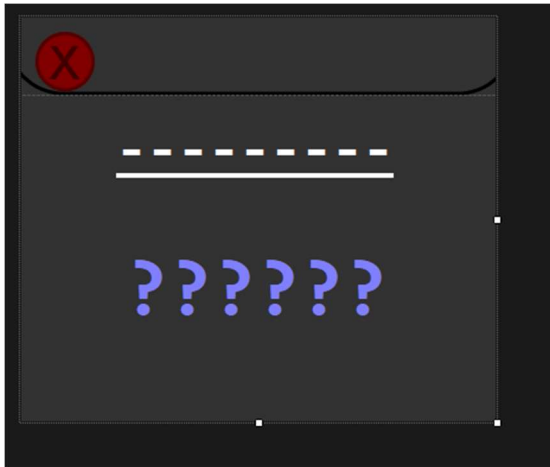
Program obsahuje 3 Formuláře, které jsou navzájem propojeny.



Hlavní formulář obsluhuje celou aplikaci. Posílá veškeré informace a signály zbylým formulářům. Tento formulář v sobě 2 módy Test, Procvičování. Tyto módy používají hlavní panel, kde se pro jednotlivý mód vygenerují cvičení a pak následně vyhodnocuje správnost řešení. Pokud je cvičení správně vyřešené, hlavní panel začne volat zbylé dva formuláře.



Druhý formulář se spustí při úspěšném dokončení cvičení jménem Procvičování. Tento formulář v sobě obsahuje graf. Ten se vytvoří pomocí uložených časů na jednotlivých řádcích okamžitě potom, co je formulář vytvořen hlavním formulářem. Jakmile je v programu zaznamenáno více pokusů, tak formulář vytvoří graf pro každý udělaný pokus.



Druhý formulář se spustí při dokončení cvičení jménem Test. Tento formulář pouze vyhodnocuje, zda byl test dokončen úspěšně či nikoliv.

Přidávání řádků

Hlavní přísadou tohoto programu je kód, který má za úkol přidávat řádky. Spouští ji sepnutí tlačítka Start.

```
Label24.Text = "Součet"; Label26.Text =  
if (stop == false)  
{ cooldown = true;  
  BinarStart.Text = "...";  
  stop = true;  
  ProklikniPanel();  
  Info.Visible = false;  
  TimerLines.Enabled = true;  
  await Task.Delay(3500);  
  BinarStart.Text = "Stop";  
  cooldown = false;  
}  
else
```

Tento úryvek zavolá funkci, která začne generovat řádky. Funkce se dělají proto, aby zjednodušili strukturu celého kódu.

```
}  
if (!DecBin.Checked)  
{  
  //Dec To Bin  
  if (start)
```

Toto rozhoduje, jaký druh převodu je vybrán. A na to následuje kód, který začíná generovat řádky. Kód, který generuje řádky funguje tak, že najde hlavní panel. Hlavní panel má v sobě Tag hodnotu, která říká, na jaké řádce se zrovna nachází.



Poté, co nalezne panel, začne kód prohledávat všechny prvky co se v něm nachází což jsou tlačítka, text box a jeden panel. Všechny prvky mají v sobě zabudovanou Tag hodnotu, aby zase kód věděl, na jaké pozici se prvek nachází.

```

}
var LinesNumbers = PanelBinSous.Controls.OfType<Panel>().ToList();
foreach (var LineNumber in LinesNumbers)
{
    if (int.TryParse(Convert.ToString(LineNumber.Tag), out int TAG) && TAG == AddingLine)
    {
        LineNumber.Visible = true;
        var Buttons = LineNumber.Controls.OfType<Guna2CircleButton>().ToList();
        var TextBoxes = LineNumber.Controls.OfType<Guna2TextBox>().ToList();
        var SoucPanels = LineNumber.Controls.OfType<Panel>().ToList();
        int LineTag = 0;
        foreach (var Button in Buttons)
        {
            LineTag++;
            if (int.TryParse(Convert.ToString(Button.Tag), out int TAGE) && TAGE == LineTag)
            {
                await Task.Delay(new Random().Next(1, 51));

                Button.Text = Convert.ToString(0);

                Button.Visible = true;
                Button.Enabled = true;

                if (Button.Text == "1")
                {
                    Button.FillColor = Color.FromArgb(255, 192, 128);
                    Button.ForeColor = Color.SaddleBrown;
                }
                else if (Button.Text == "0")
                {
                    Button.FillColor = Color.SaddleBrown;
                    Button.ForeColor = Color.FromArgb(255, 192, 128);
                }
            }
        }
    }
}

```

Funkce začíná tím, že si vyhledá 7. řádek. Určil jsem, že řádky budou panely, kterým jsem přidal hodnotu Tag a očísloval je od 1 do 7. Dále si kód začne vyhledávat co se jednotlivým řádku (panelu) nachází. Upřesnil jsem, aby vyhledával tlačítka, text box a label. Tlačítkům a text boxům bude nastavený stav, ve kterém budou, jestli se budou moci změnit či nikoliv. Udělí se jim barva, která rozliší jejich stav. Nastaví se náhodným generátorem, jakou budou mít hodnotu a poté se zviditelní. Label, který jsem zmiňova,l bude pouze funkční při jednom módu a jeho úkol bude pomáhat počítat jednotlivá tlačítka, aby ze začátku pomohl nováčkům přijít na princip počítání jednotlivých bitů ve dvojkové soustavě.

```
await Task.Delay(new Random().Next(1, 51));
```

Mezi to jsem přidal náhodný odpočet času, aby to dostalo efekt přidávání. Funkce await Task.Delay() zajistí potřebné zdržení, udává se v milisekundách. Tato funkce je dobrá, aby kód byl pozastaven na určenou dobu a poté pokračoval dále. Nebo při opakující se funkci nastavit aby se opakovala pomaleji.

Náhodný generátor čísel

```
1 reference
static void KontrolaRadku(List<String> vybranaCisla, List<int> selectedNumbers, Random rnd, int x)
{
    for (int i = 0; i < vybranaCisla.Count; i++)
    {
        for (int j = i + 1; j < vybranaCisla.Count; j++)
        {
            if (vybranaCisla[i] == vybranaCisla[j])
            {
                vybranaCisla.RemoveAt(j);
                selectedNumbers.Clear();
                VybiraniRadku(vybranaCisla, selectedNumbers, rnd, x);
                j--;
            }
        }
    }
}

2 references
static void VybiraniRadku(List<String> vybranaCisla, List<int> selectedNumbers, Random rnd, int x)
{
    while (selectedNumbers.Count < x)
    {
        int randomNumber = rnd.Next(1, 9);

        if (!selectedNumbers.Contains(randomNumber))
        {
            selectedNumbers.Add(randomNumber);
        }
    }

    string cislaOddelenaCarkami = string.Join(" ", selectedNumbers);
    vybranaCisla.Add(cislaOddelenaCarkami);

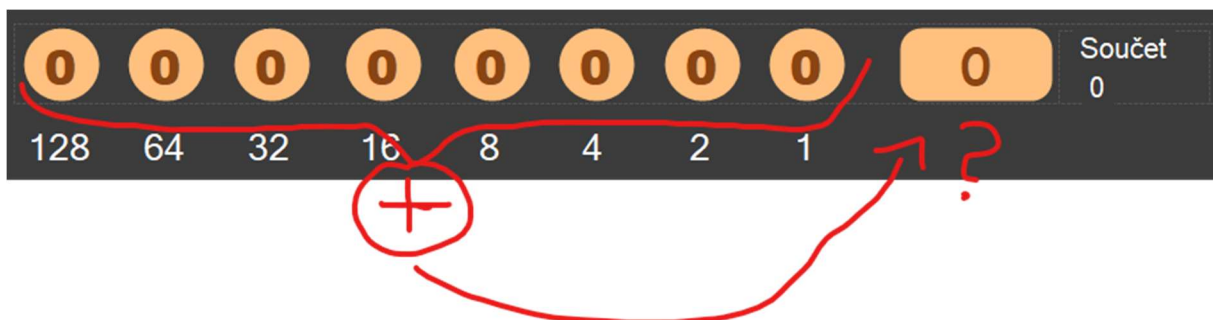
    KontrolaRadku(vybranaCisla, selectedNumbers, rnd, x);
}
```

Tento úryvek obsahuje 2 funkce, které na sebe navazují. Tyto funkce mají zařídít, aby se náhodný výběr čísel nikdy neopakoval. Aby se v cvičení nikdy neukázala stejná kombinace 1 a 0 nebo stejné číslo v rozmezí 0-255. Funkce vybere celou řádku a začne prohledávat kombinaci 1 a 0 ve všech 8 tlačítkách, poté ji porovná ve své paměti, jestli se už tato kombinace někdy neopakovala. Pokud tomu tak není, tak se kombinace do paměti zapíše. Pokud se v paměti taková kombinace už objevila, tak funkce bude muset novou kombinaci čísel znovu vygenerovat a proces se bude následně opakovat. Po dokončení cvičení se paměť s kombinacema číslic promaže.

Zadávání hodnot a jejich řešení

```
12 references
private void CalculatorButton_Click(object sender, EventArgs e)
{
    Guna2CircleButton Button = (Guna2CircleButton)sender;
    if (FinalNumberTextbox != null && FinalNumberTextbox is Guna2TextBox && CalculatorPanel.Visible)
    {
        if (Button.Text != "⌫" & Button.Text != "C")
        {
            FinalNumberTextbox.Text = FinalNumberTextbox.Text + Button.Text;
        }
        else if (Button.Text == "⌫")
        {
            if (!string.IsNullOrEmpty(FinalNumberTextbox.Text))
            {
                FinalNumberTextbox.Text = FinalNumberTextbox.Text.Substring(0, FinalNumberTextbox.Text.Length - 1);
            }
        }
        else if (Button.Text == "C")
        {
            FinalNumberTextbox.Text = "";
        }
    }
}
```

Díky tomuto kódu může uživatel ovládat numerickou klávesnici pomocí myši. Numerická klávesnice se aktivuje pomocí poklepání myši na text box, do kterého chceme psát. Po kliknutí na jedno z čísel na numerické klávesnici se číslo přečte a následně vloží do vybraného textboxu. Pokud klepneme na symbol šipky zpět, číslo se od konce začne mazat. Pokud klepneme na písmeno C, tak se celé číslo v text boxu smaže.



Zde je vidět, jak funguje sčítání jednotlivých tlačítek (bitů). Pokud nějaké tlačítko obsahuje číslo 1, pak se jeho hodnota, která je určena pod ním, sečte s hodnotami ostatních tlačítek, která také obsahují číslo 1. Pokud tlačítka obsahují číslo 0, nepočítají se. Celkový součet tlačítek pak musí odpovídat hodnotě v text boxu. Hodnoty text boxu jsou určeny od 0 do 255.


```

if (clampedValue == ans)
{
    nehotovo--;
    Remain.Text = Convert.ToString(7 - nehotovo) + "/7";
    if (!testee)
    {
        switch (panel.Tag)
        {
        }
        foreach (var groupButton in buttons)
        {
            groupButton.DisabledState.FillColor = Color.FromArgb(192, 255, 192);
            groupButton.DisabledState.ForeColor = Color.DarkGreen;
        }
        finalNumber.FillColor = Color.FromArgb(192, 255, 192);
        finalNumber.ForeColor = Color.DarkGreen;

        await Task.Delay(1000);
        panel.Visible = false;
        finalNumber.Visible = false;
        foreach (var groupButton in buttons)
        {
            groupButton.Visible = false;
            finalNumber.Text = string.Empty;
        }

        if (nehotovo <= 0)
        {
            if (!testee)
            {
                KonecGraf();
            }
            else
            {
                uspesnosttest = true;
                TestKonec();
            }
        }
    }
}

```

Toto je úryvek kódu, který vyhodnocuje, jestli se hodnoty rovnají a pokud ano, celá řádka se označí zeleně a poté zmizí. Na konci se ptá, jestli už jsou všechny řádky splněny a pokud ano, tak se vyvolá funkce, která ukáže pro vybrané cvičení vytvořený graf.

Závěr

Díky tomuto programu, co jsem vytvořil, jsem se naučil tak lépe orientovat ve Visual Studiu. Základy programovacího jazyka C# jsem se naučil ve škole o hodinách programování. Tento jazyk jsem poznal více, když jsem si hrál s tímto programem. Veškeré funkce jsem se naučil na různých webech. Nejvíce jsem používal fórum od společnosti Microsoft, protože to mají hodně spojené s Visual Studiem a podle mě, to mají celkem dobře popsané a vysvětlené. Čas od času, když jsem si nemohl něco dohledat, tak jsem použil chatbota Microsoft Copilot, což je nová umělá inteligence od Microsoftu. Tato umělá inteligence dokáže lépe vysvětlit funkcionalitu nějakého kódu než samotný chat GPT od společnosti Open AI.

Nápad k vytvoření toho programu jsem dostal, když jsem na hodině počítačových sítí ve škole viděl jednu nevyvinutou webovou hru v placeném kurzu od společnosti CISCO. Hra byla strašně demotivující a chtěl jsem ji po pár minutách vypnout. Řekl jsem si, že zkusím vytvořit hru s podobnou myšlenkou ale s opačným pocitem. Program jsem několikrát odzkoušel na ostatních a postupně ho formoval podle zpětných vazeb, co jsem dostával. Jakmile jsem skončil se svoji první publikovatelnou verzí (což je tato), předal jsem program učitelům na naší škole, aby ho vyzkoušeli dát studentům. Studentům se líbil a stal se tak součástí učebních pomůcek. Testovací část je využívána vyučujícími pro závěrečné hodnocení znalostí.



Zdroje

Fórum kódování Microsoft:

<https://learn.microsoft.com/en-us/dotnet/csharp/>

Efekty tutoriál s rozšířením GunaUI:

https://www.youtube.com/watch?v=VO12HzfEO9o&t=193s&ab_channel=GunaUI

Oficiální stránky GunaUI:

<https://gunai.com>

Oficiální stránky Microsoft Visual Studio:

[Visual Studio: Integrované vývojové prostředí \(IDE\) a editor kódu pro vývojáře softwaru a týmy \(microsoft.com\)](https://visualstudio.microsoft.com/)

Pomoc s C# Fórum Detvou:

<https://dev.to/t/csharp>

Oficiální stránky CSharp Fórum:

<https://csharpforums.net>