

STŘEDOŠKOLSKÁ ODBRORNÁ ČINNOST

OBOR SOČ: 18. INFORMATIKA

Vědomostní hra v jazyce C#

AUTOR: Pavel Vojta

**ŠKOLA: Střední škola spojů a informatiky
Bydlinkého 2474, Tábor, 390 11**

KRAJ: Jihočeský kraj

Tábor 2016

Čestné prohlášení

Prohlašuji, že jsem svou práci vypracoval samostatně. Použil jsem pouze podklady uvedené v příloženém seznamu a postup při zpracování a dalším nakládání s prací je v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně.

V Táboře dne 10.dubna 2017

.....
Pavel Vojta

Anotace

Cílem práce bylo vytvořit počítačovou vědomostní hru napsanou v programovacím jazyce C# s uživatelsky přívětivým prostředím. Program je přehledný, může se používat ve školách pro práci s interaktivními tabulemi. Výsledná aplikace umožňuje střídání dvou hráčů. Tito hráči odpovídají po celou dobu hry na náhodně přidělené neopakující se otázky. Hráči mají zobrazeno, kolik času je na zodpovězení otázky a také zbývajících čas celé hry. Dále je zobrazeno jméno současného hráče, jeho průběžný počet bodů a kolikáté kolo je rozehráno. Soutěžící mohou hru ukončit či jen pozastavit a poté se vrátit opět do rozehrané hry. Aplikace obsahuje jednoduchý editor otázek a odpovědí.

Program je psaný v programovacím jazyce C#. V aplikaci jsou použity časovače, dialogová okna, ověření podmínek. Dále se pracuje s externím textovým souborem, kde je možné upravovat, přidávat a odebírat otázky a správné odpovědi. Součástí práce je nejen zdrojový kód hry, ale i spustitelná verze.

Obsah

Čestné prohlášení.....	2
Anotace.....	3
Cíl.....	5
O jazyku C#.....	5
Integrované vývojové prostředí.....	6
Přehled vybraných vývojových prostředí pro jazyk C#.....	6
MonoDevelop.....	6
JetBrains Rider.....	6
Visual Studio.....	6
Instalace Visual Studio.....	7
Vytvoření nového projektu.....	7
Vzhled programu Vědomostní hra v jazyce C#.....	8
Hlavní okno hry.....	8
Editor odpovědí.....	9
Vysvětlení pojmů.....	9
Proměnná.....	9
Seznam.....	10
Analýza kódu.....	11
Form1.cs.....	11
Form2.cs.....	14
Soubor otázek.....	15
Ikona aplikace.....	16
Závěr.....	17
Zdroje.....	18
Seznam příloh.....	19

Cíl

Můj záměr je v této práci vytvořit počítačovou vědomostní hru napsanou v programovacím jazyce C# s uživatelsky přívětivým prostředím. Pro původní inspiraci mi posloužila soutěžní vědomostní hra AZ-Kvíz vysílaná Českou televizí, z důvodu náročnosti vytvořit grafiku programu, která by byla stejná jako v tomto pořadu, rozhodl jsem se od tohoto záměru ustoupit. Aplikace ale stále ověřuje znalosti hráčů.

Protože předpokládám, že by se program mohl používat ve školách pro práci s interaktivními tabulemi, rozhodl jsem se, že tento program bude přizpůsobovat velikosti obrazovek, na kterých se bude zobrazovat. Dále jsem se rozhodl v návrhu designu programu napodobit moderní vzhled programů, které jsou vytvořeny pro prostředí Metro/Modern UI, i když tento program bude jen běžná aplikace. Program by tak měl být uživatelsky přívětivý, přehledný a líbivý. Vzhledově by tak měl zapadat do posledních verzí systému Windows.

Výsledná aplikace by měla umožnit střídání dvou hráčů. Tito hráči by měli po celou dobu hry odpovídat na náhodně přidělené otázky. Ty by se neměli po sobě opakovat. Na zodpovězení zadání bude mít každý hráč dostatečný čas, aby si mohl přečíst otázku a zapsat řešení. Jestliže hráč nenapíše správnou odpověď, měl by se dozvědět její správné znění. Maximální délku hry si zvolí soutěžící sami. Hráči taktéž budou vidět, kolik času mají na zodpovězení otázky a také zbývající čas celé hry. Dále se bude zobrazovat jméno současného hráče, jeho průběžný počet bodů a kolikáté kolo je rozehráno. V průběhu hry budou mít soutěžící možnost hru ukončit či jen pozastavit a posléze se vrátit opět do rozehrané hry. Ke konci soutěže si budou hráči moci přečíst její výsledek.

Hra bude načítat otázky z externího textového souboru, měla by být schopna si tento soubor v případě nutnosti i vytvořit. Aby se aplikace mohla rozvíjet i bez mé podpory, bude obsahovat i jednoduchý editor otázek a odpovědí, ve kterém bude možné nejenom upravovat všechna současná zadání a jejich řešení, ale i vytvářet vlastní.

O jazyku C#

Jazyk C# je vyšší programovací jazyk, objektově orientovaný, podporující tzv. type-safe programování, dále je case-sensitive (rozlišuje malá a velká písmena) a multiplatformní (kromě Windows může běžet i na Linuxu). Je založen na akcích (kliknutí na tlačítko, stisknutí klávesy apod.)

Ačkoli se programovací jazyk C# poprvé objevil v roce 2000, teprve začátkem roku 2002 se verze 1.0 spolu s vydáním Microsoft Visual Studio .NET 2002 dostal blíže k veřejnosti. Microsoft tak reagoval na oblibu jazyku Java, avšak C# má spíše blíže C++. Další verze 1.1 a 1.2 se objevily s Visual Studio .NET 2003, poté 2.0 s Visual Studio 2005, verze 3.0 byla ve Visual Studio 2008 a 2010, obsahovala i C# 4.0. Pro Visual Studio 2012 a Visual Studio 2013 se připravila verze 4.0. Visual Studio 2015 má verzi 6.0. Zatím poslední Visual Studio 2017 podporuje verzi C# 7.0. Jednotlivé verze jsou zpětně kompatibilní, což znamená, že novější verze podporuje funkce verze starší.

Zdrojové kódy jazyka C# mívají příponu .cs a jednotlivé příkazy končí středníkem.

Integrované vývojové prostředí

I když se programy dají tvořit pouze pomocí textového editoru a příkazového řádku, daleko častěji se vytváří pomocí vývojového prostředí (IDE - Integrated Development Environment), které celý proces usnadňuje – například může našeptávat jednotlivé příkazy (IntelliSense), upozorňovat na případné chyby nebo vytvářet instalační soubory.

Přehled vybraných vývojových prostředí pro jazyk C#

MonoDevelop

Podporované platformy: Linux, Windows a Mac OS X

Podporované jazyky: C/C++, C#, F#, Vala, Visual Basic

Odkaz ke stažení: <http://www.monodevelop.com/download/>

JetBrains Rider

Podporované jazyky: C#, CSS, HTML, JavaScript, JSON, SQL, TypeScript, VB.NET, XAML, XML

Odkaz ke stažení: <https://www.jetbrains.com/rider/download/>

Visual Studio

Podporované jazyky: C++, C#, F#, JavaScript, Python, TypeScript, Visual Basic

Odkaz ke stažení: <https://www.visualstudio.com/cs/downloads/>

Při tvorbě této práce se používalo Visual Studio Community 2015.

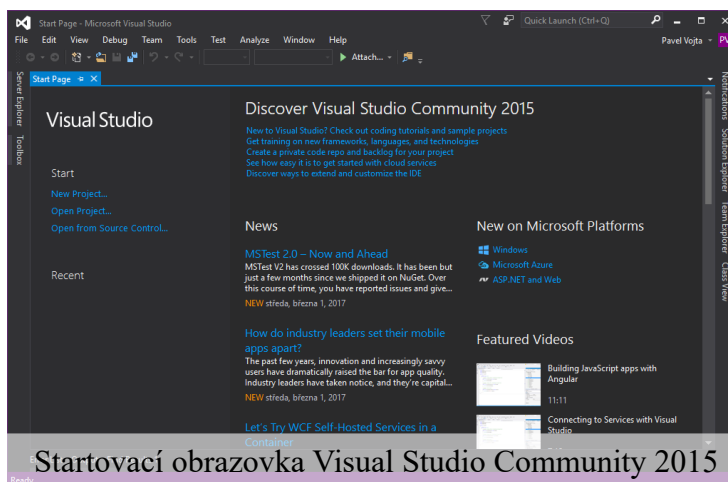
Instalace Visual Studio

Instalační soubor posledního vydání Visual Studio lze získat z <https://www.visualstudio.com/post-download-vs/>, při použití toho odkazu bude však potřeba připojení k Internetu. Pro offline instalaci Visual Studio použijte instalační soubor z https://download.microsoft.com/download/b/e/d/bedddfc4-55f4-4748-90a8-ffe38a40e89f/vs2015.3.com_enu.iso (7,09 GB). V průběhu instalace se stačí řídit pokyny průvodce. Po nainstalování bude Visual Studio v angličtině.

Vytvoření nového projektu

Nový projekt se vytvoří následujícím způsobem:

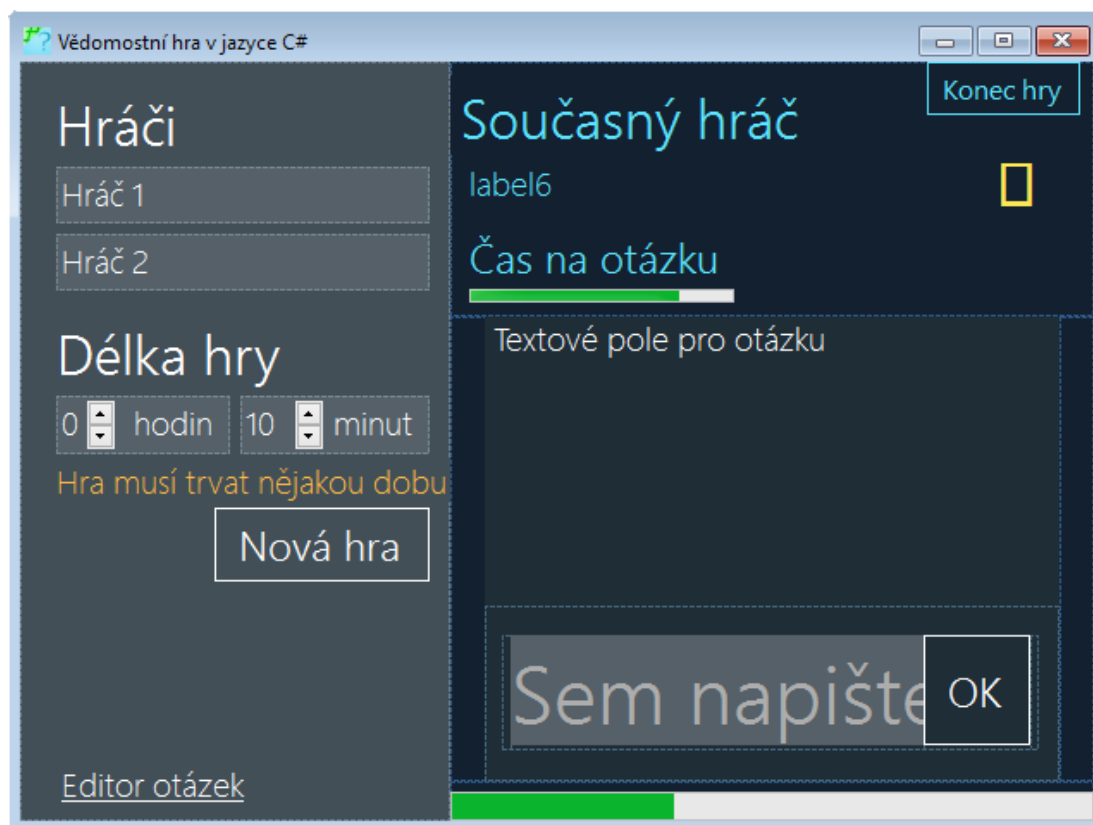
New Project... > Visual C# >
Windows > Classic Desktop >
Windows Forms Application



Vzhled programu Vědomostní hra v jazyce C#

Při práci s vzhledem hry bylo zamýšleno držet se moderního vzhledu programů a použít responzivní design tak, aby se program přizpůsoboval velikosti obrazovky. Projekt se skládá ze dvou formulářů – samotné hry a editoru otázek a odpovědí.

Hlavní okno hry



Tento formulář (*Form1*) je rozdělen na dva hlavní panely, při spuštění je zobrazen pouze levý panel (*panel1*), *panelHra* je skrytý. U prvku *panel1* je vlastnost *Dock* nastavena na *Left*, *panelHra* má vlastnost *Dock* rovnu *Fill* a je viditelný pouze, když je spuštěna hra. Dále byla ve *Form1* upravena vlastnost *Text* na „Vědomostní hra v jazyce C#“ a přidala se vlastní ikona aplikace. Aplikace používá písma rodiny Segoe UI.

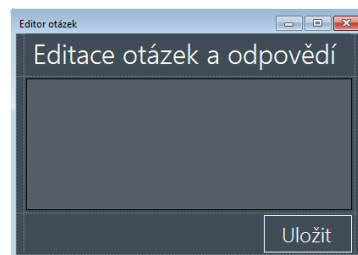
Na *panel1* je umístěno 5 labelů (popisek) – *label1* (Hráči), *label2* (Délka hry), *label3* (hodin), *label4* (minut), *label8*, který je zobrazen pouze při špatně zadané délce hry. Dále se zde nacházejí dva textBoxy (*textBox1*, *textBox2*) sloužící pojmenování hráčů a *numericUpDown1* spolu s *numericUpDown2* určující dobu hry, tyto prvky jsou z čistě estetického důvodu umístěny ve vlastních panelech. Dále tlačítko *buttonHraZacatek* (Nová hra) s akcí po kliknutí *but-*

ton1_Click a *linkLabel1* (Editor otázek) spouštějící po kliknutí *Form2*.

Prvek *panelHra* obsahuje *label5* (Současný hráč), *label6* sloužící k zobrazení jména právě hrajícího hráče, *label7* (Čas na otázku), klikatelný *label9*, jenž dokáže pozastavit hru či v ní dále pokračovat, dále tlačítka *buttonHraKonec* (Konec hry) s akcí *button2_Click* a *buttonAnswerOK* (OK) s akcí *buttonAnswerOK_Click*, *progressBar1* zobrazující zbývající čas k zodpovězení otázky a *progressBar2* ukazující dobu do konce hry. Výčet prvků uzavírají textové pole na zadání odpovědi *textBox3*, *textBox4*, který slouží k zobrazení otázky a který má vlastnost *ReadOnly*, aby uživatel nemohl měnit jeho obsah, dále celkem čtyři *tableLayoutPanely*, které upravují rozvržení prvků. Prvky, které nejsou vidět, jsou zde *timer1*, *timer2*, prázdný *contextMenuStrip1* a *toolTip1*.

Editor odpovědí

Podstatně méně prvků náleží Editoru otázek. Zde lze nalézt *label1* (Editace otázek a odpovědí), tlačítko *button1* (Uložit), *tableLayoutPanel* a *dataGridView1*, ve kterém se nachází dva sloupce (Otázka a Odpověď). Právě tento prvek umožňuje editaci otázek a jejich odpovědí. Vizuální styl editoru respektuje design hlavního okna.



Vysvětlení pojmů

Proměnná

Proměnná má totožnou funkci jak v programování, tak i v matematice. Do proměnných lze ukládat po dobu běhu programu data, pracovat s nimi a měnit je. Aby se daly používat, musí se nejprve vytvořit – deklarovat. Systém si pro ně pak vyhradí místo v paměti. Velikosti a rozsahy vybraných typů ukazuje následující tabulka.

Druh	Typ	Bajty	Rozsah hodnot
celé číslo	int	4	-2 147 483 648 až 2 147 483 647
	uint	4	0 až 4 294 967 295
	int16	2	-32 768 až 32 767
	uint16	2	0 až 65 535
	int64	8	-2^{63} až $2^{63} - 1$
	uint64	8	0 až $2^{64} - 1$
ano/ne	bool	1	true/false
znak	char	1	
text	string		

Příklad deklarace proměnných:

```
16      int delkaHry, hrac1Skore, hrac2Skore, otazka, kolo;
17      string hrac1Jmeno, hrac2Jmeno, hracSoucasny, odpoved;
```

Protože se tyto proměnné deklarují ještě před třídou *Form1()*, nazývají se tyto proměnné jako členské. Znamená to, že se mohou využít v celém formuláři. Zatím však nemají žádnou hodnotu, v programu se jim přiděluje postupně, například zde:

```
177      kolo = 1;
```

Seznam

Seznam se značí v jazyce C# slovem *List*. Ve své podstatě se jedná o rozšíření prvku pole. Pole se využívají především v případě, kdy je třeba vytvořit více proměnných stejného typu sloužící ke stejnému účelu (představa deklarovat proměnné obsahující například průměrnou denní teplotu za posledních sto dní po jedné je děsivá). Při deklaraci pole se uvádí i jeho velikost – počet položek, které se budou vytvářet, toto množství je konečné a nelze ho v budoucnu měnit. Avšak seznam umožňuje měnit počet položek, lze tedy do seznamu položku jak přidat, vložit, přesunout, tak i odebrat. Konkrétně možnost odebrat ze seznamu se v tomto programu využívá k zajištění losování otázek, které by se neměly za sebou ihned opakovat. S každou položkou se pracuje jako s proměnnou.

Každá proměnná má v poli i v seznamu své pořadí, které se nazývá index. Index v obou případech začíná od nuly. Například osmá proměnná v seznamu má index sedm.

Analýza kódu

Form1.cs

V tomto formuláři se nejdříve deklarují členské proměnné číselného typu (*delkaHry*, *hrac1Skore*, *hrac2Skore*, *otazka*, *kolo*), textového typu (*hrac1Jmeno*, *hrac2Jmeno*, *hracSoucasny*, *odpoved*) a seznam *otazkySeznam*, protože se potřebují používat ve více třídách.

Ve třídě *Form1()* se nejdříve vytvoří prvky, které se umístily ve formuláři, *AcceptButton* se mění na *buttonHraZacatek*, což zajišťuje, že se spustí po stisknutí klávesy Enter stejná akce, jako když se klikne na toto tlačítko, a poté se skryje *panelHra*.

Kdyby uživatel poklikal na *panel2* nebo na *panel3*, bude moci psát do *textBox1* respektive do *textBox2*, stejně jako by poklikal na *textBox* náležícímu svému panelu. Stejný postup byl použit i pro *panel4* a *panel5*, kde se změní aktivní prvek na *numericUpDown1* popřípadě *numericUpDown2*.

Klikne-li uživatel na *textBox1*, aby zadal jméno prvního hráče, nebo na něj přejde pomocí klávesy Tab, změní svou barvu, stejně tak i jeho panel, a pokud tento *textBox* obsahuje text „Hráč 1“, text se vymaže. Odejde-li z něj, barva pozadí se vrátí na původní a ověří se, jaký text *textBox* obsahuje. Pokud v něm žádný není, přidá se „Hráč 1“, pokud je však stejný jako druhý *textBox*, změní se barva textu na odstín červené a schová se tlačítko *buttonHraZacatek* do doby, než se text změní na jiný, než je v druhém *textBoxu* *textBox2*. V ideálním případě se však změní pouze barva textu na bílou a je vidět tlačítko ke spuštění hry. Obdobně toto platí pro *textBox2*, kde je pouze změna z textu „Hráč1“ na „Hráč2“. Tímto se program snaží donutit uživatele, aby měli jako soutěžící jiná jména.

Kromě změny barvy při změně statusu aktivace *textBox3* určeného pro zadání odpovědi, vytváří tento *textBox* i efekt zástupného textu (placeholder), tj. místo žádného textu bude šedou barvou zobrazen text „Sem zadejte svou odpověď“, avšak uživatel tento text nemusí mazat, text se smaže sám.

Pokud se vstoupí do *panel4*, změní se fokus na prvek *numericUpDown1*, jehož hodnota určuje délku hry hodinách. Stejně se chová také *panel5* obsahující *numericUpDown2*, kde se nastavuje počet minut.

Pro lepší rozlišení aktivních prvků se *textBox1*, *panel2*, *textBox2*, *panel3*, *numericUpDown1*, *panel4*, *numericUpDown2*, *panel5* a *textBox3* mění jejich barva pozadí na světlejší nebo na původní, podle toho jestli jsou právě aktivní či nikoli.

Tlačítko *button1* spouštějící novou hru nejprve ověřuje, jestli je zadaná správná doba hry. Poté se ukládá jméno prvního hráče z *textBox1* do proměnné *hrac1Jmeno*, jméno druhého hráče do proměnné *hrac2Jmeno* z *textBox2*. Jejich skóre uložené v proměnných *hrac1Skore* a *hrac2Skore* je nula bodů. Skryje se *label8* spolu s *panel1* a naopak se zobrazí *panelHra*. *AcceptButton* se mění na *buttonAnswerOK*, aby se mohlo po napsání odpovědi, odpovědět také i klávesou Enter. Dále se do proměnné *delkaHry* ukládá délka hry zadané uživatelem do prvků *numericUpDown1* a *numericUpDown2* a začne se zobrazovat tento zbývajíc čas hry v *progressBar2*. Nejdříve se však nastaví maximální hodnota *progressBar2*, aby zde byla zajištěna správná funkce tohoto prvku. Prvek *timer1* začne odpočítávat čas. Do proměnné *hracSoucasny* se uloží hodnota „hrac1“. Hodnota proměnné *kolo* se bude rovnat 1 a spustí se funkce *novaOtazka()*, která se mimojiné stará o zobrazení nového zadání pro soutěžící. Popisek *label6* bude zobrazovat jméno hráče a jeho počet bodů, například „Jakub (0 bodů)“. Panel *tableLayoutPanel4*, ve kterém je umístěn *textBox* pro zadání odpovědi, se přizpůsobí velikosti okna. Popisek *label9* určený pro pozastavování a opětovného pokračování hry bude obsahovat žlutý symbol pro pauzu. Prvek *toolTip1* zobrazuje vysvětlující zprávu po najetí na *label9*. Poslední akce tlačítka *button1* je povolení a případně zobrazení časovačů *timer1* a *timer2*, textových polí *textBox3* a *textBox4*, tlačítka *buttonAnswerOK* a ukazatelů průběhu *progressBar1* a *progressBar2*.

Jestliže bude uživatel chtít ukončit hru ještě před uplynutím časového limitu, klikne na tlačítko *button2*, které spustí funkci *konecHry()*.

Při každém tiknutí prvku *timer1* se zmenší hodnota v *progressBar2* do doby, než se bude rovnat nule. Tímto se měří zbývajíc čas hry, po jehož uplynutí hra skončí a následně se začne vykonávat funkce *konecHry()*.

Podobně jako *timer1* měří zbývajíc čas hry, tak *timer2* odpočítává do konce doby, kdy bude možné zadat řešení otázky. Tento čas je zobrazen za pomoci prvku *progressBar1*, jehož hodnota se postupně zmenšuje, než se následně rovná nule. Po vyčerpání tohoto časového limitu program zkusí ověřit, zda je text v prvku *textBox3* shodný s řešením otázky, poté přichází na řadu funkce *novaOtazka()*.

Ačkoli je prvek *label9* popisek, chová se po kliknutí stejně jako tlačítko. Jeho účel je pozastavovat hru nebo umožnit hráčům v jejím pokračování. Po kliknutí na tento prvek si aplikace pomocí zobrazeného symbolu v *label9* zjistí, zda je hra pozastavena či nikoli. Jestliže je, prvky *timer1*, *timer2*, *textBox3*, *textBox4*, *buttonAnswerOK*, *progressBar1* a *progressBar2* se zakáží a skryjí se, aby hráč nemohl číst otázku ani na ni odpovídat, ale také aby dále neubíhal čas ve hře. Změní se symbol pro pokračování hry v *label9* a jeho barva bude v odstínu zelené. Upraví se vysvětlující popisek *toolTip1* prvku *label9*. Po stisknutí klávesy Enter by se neměla vykonávat akce žádného tlačítka. Pokud je hra pozastavena, povolí se stejné prvky, které byly při pozastavení zakázány, *label9* bude zobrazovat symbol pro pauzu v odstínu žluté barvy. Poupřaví se vysvětlující popisek *toolTip1* prvku *label9*. Pro snadnější ovládání bude aktivní *textBox3* pro zadání odpovědi a po stisku klávesy Enter se vykoná akce tlačítka *buttonAnswerOK*.

Tlačítko *buttonAnswerOK* je určeno k potvrzení napsání odpovědi. Nejprve si uloží odpověď hráče, která se nachází v textBoxu *textBox3*, do proměnné *odpovedOverit*. Text v této proměnné se předvede na malá písmena, což zajišťuje, že hráči nemusejí rozlišovat malá a velká písmena. Do funkce *vyhodnoceniOtazky()* se předá obsah proměnné *odpovedOverit*, aby se mohla ověřit správnost odpovědi hráče. Jako poslední akce v tomto tlačítku se spustí funkce *novaOtazka()*.

Odkazovací popisek *linkLabel1* otevírá editor otázek a odpovědí. Proto se však musí deklarovat, jaké okno formuláře se chce spustit, zde je to *Form2*. Tento formulář se spouští jako dialogové okno. Tento způsob spouštění znemožňuje dále pracovat s hlavním oknem hry, jestliže je spuštěn editor otázek.

Protože není žádoucí, aby hráč vybíral text otázek v *textBox4*, aplikace při každém kliknutí do tohoto prvku změni výběr aktivního prvku na jiný. Zde byl zvolen popisek *label5*.

Funkce *novaOtazka()* si nejprve ověří, jestli seznam otázek *otazkySeznam* obsahuje nějaké, které by se mohly hráči zobrazit. Jestliže v tomto seznamu už žádné nezbyývají, načtou se znovu ze souboru *questions.txt*. V případě, že tento soubor neexistuje, program jej do adresáře, ze kterého se spouští, zkopíruje. Třída *nahodneCisloOtazky* poslouží k vygenerování nového náhodného čísla, které se uloží do proměnné *otazka*. Zde se losuje celý rozsah seznamu *otazkySeznam*. Následně se do proměnné *otazkaZadani* uloží ze stejného seznamu text otázky nacházející se na daném indexu. Poté dojde k rozdělení textu z proměnné *otazkaZadani* na dvě části do dvou proměnných *otazkaZadane*, kde se do znaku svislé čáry („|“) tento text bude dále považovat za otázku a po tomto znaku se nachází odpověď. Protože vylosovaný text ze seznamu

otazkySeznam není potřebný, odebere se, čímž se docílí neopakování stejných otázek za sebou. Aby se uživatel nemusel zabývat psaním velkých či malých písmen, převede se celý text odpovědi na malá písmena. Spolu s číslem kola bude v *textBox4* zobrazena na dalším řádku i otázka. Hodnota prvku *progressBar1* se nastaví podle délky textu v *textBox4*, ke které se přičte číslo 30, maximum tohoto prvku bude stejné a *timer2* začne opět odpočítávat čas. Hráči se ulehčí možnost, že bude mít již zvolený *textBox3*, kam bude moci zadat svoji odpověď, kterou následně zadá klávesou Enter, pokud bude chtít.

Zda je odpověď, kterou zadal hráč, správná, vyhodnocuje funkce *vyhodnoceniOtazky()*. Začíná tak, že si nejprve převezme do proměnné *odpovedHrace* odpověď, kterou této funkci předala už jiná část programu, a zastaví odpočítávání prvku *timer2*. Poté se text v proměnné *odpovedHrace* srovná s členskou proměnnou *odpoved*, která obsahuje správné řešení otázky. Shodují-li se, zobrazí se vyskakovací okno informující o správném zodpovězení otázky, současně mu hráči se přičte jeden bod, změní se současný hráč na jeho soupeře a v prvku *label6* se zobrazí soupeřovo jméno s jeho počtem bodů. V opačném případě se zobrazí vyskakovací okno, které kromě sdělení, že otázka je zodpovězena špatně, se zobrazí i správné řešení, změní se současně hrající uživatel na jeho soupeře a v prvku *label6* bude opět vidět soupeřovo jméno společně s jeho počtem bodů. Pokud právě dokončil svou otázku druhý hráč, zvýší se i číslo kola o jedno. Poté se vždy vymaže *textBox3*, aby nový hráč nemusel tento prvek vymazávat.

O ukončení hry se stará funkce *konecHry()*, která nejprve zastaví *timer2* a *timer1*, které následně zakáže. Zkontroluje počet bodů obou hráčů a vypíše tento stav ke konci hry do vyskakovacího okna a hráči se tak dozví, zda hra skončila remízou, či vyhrál jeden soutěžící. Ve *Form1* se opět povolí a zobrazí *panell* a *panelHra* se naopak zakáže a skryje. Tlačítko, jehož akce se spustí po stisknutí klávesy Enter, bude od této chvíle opět *buttonHraZacatek*.

Form2.cs

Tento formulář po svém načtení spustí funkci *nacistOtazky()*, která nejprve vytvoří textový seznam *otazkySeznam*. Do tohoto seznamu se postupně načtou všechny řádky ze souboru *questions.txt*. Pokud tento soubor se nepodaří načíst, program se pokusí o jeho zkopírování s výchozí sadou otázek ze souborů projektu. Vytvoří se nová tabulka dat (prvek *DataTable*) *tabulka*, která bude široká dva sloupce, z nichž bude jeden zobrazovat otázku a druhý odpověď. Pomocí cyklu *for*, který zajistí, že tabulka bude obsahovat všechna potřebná data, se nejprve uloží daný index do proměnné *otazkaZadani*, která se později rozdělí v proměnné *otazkaZa-*

dane na dvě části – první část bude otázka, druhá řešení této otázky. Rozdělovací znak je v tomto případě svislá čára („|“). Následně se do tabulky vloží řádek, ve kterém bude nejdříve otázka a správná odpověď. Toto celé se opakuje do skončení tohoto cyklu `for`. Do prvku *dataGridView1* se načtou data z tabulky *tabulka*.

K ukládání změn provedených uživatelem slouží tlačítko *button1*. Po kliknutí na toto tlačítko, se nejdříve vytvoří textový seznam *otazkySeznam* a proměnná *otazka*. První cyklus `for` se opakuje do doby, dokud nejsou zapsané všechny řádky. Aby každý řádek obsahoval dva sloupce, je v tomto cyklu vnořen ještě jeden další cyklus `for`, který ukládá počet sloupců. Aby se zachovala správná struktura dokumentu s otázkami, první sloupec, který vidí uživatel, obsahuje otázku, jenž se uloží do proměnné *otazka*, která poté bude na svém konci obsahovat ještě rozdělovací znak. Při dalším opakování vnořeného cyklu `for` dojde ke zpracování z pohledu uživatele druhého sloupce s odpovědí, jehož hodnota v tomto řádku se přidá do seznamu *otazkySeznam* za hodnotu proměnné *otazka*. Po zpracování všech řádků v tabulce *tabulka* se program pokusí o otevření souboru *questions.txt* k zápisu. Do tohoto souboru se postupně uloží hodnoty ze seznamu *otazkySeznam* tak, aby byla každá hodnota na novém řádku. Soubor s otázkami se následně zavře, aby s tímto souborem mohly pracovat i jiné programy. Že se zápis do souboru podařil, bude značit barva ukládajícího tlačítka *button1*, která se změní na odstín zelené. Jestliže se zápis nezdaří, tlačítko *button1* změní svou barvu na odstín červené.

Pro přehlednější práci s tímto editorem otázek se po změně hodnoty jakékoli buňky v prvku *dataGridView1* změní barva tlačítka *button1* na odstín žluté, do doby než dojde k pokusu uložení otázek a odpovědí do souboru.

Soubor otázek

Název souboru otázek a odpovědí byl zvolen jednoduše – *questions.txt*. Při každém spuštění aplikace se nejdříve zkontroluje verze tohoto souboru nacházející se v adresáři programu. V případě, že je ve složce starší verze, než jaká je uložena v souborech tohoto celého projektu, soubor *questions.txt* do adresáře tohoto programu zkopíruje.

Struktura tohoto dokumentu je tvořena ze zadání otázky, oddělovacího znaku a správné odpovědi. Jako rozdělovací znak byla zvolena svislá čára („|“), protože se v běžném textu téměř vůbec nevyskytuje.

S tímto souborem pracují oba formuláře tohoto programu. Hlavní okno hry z tohoto souboru

pouze čte zadání a řešení, zatímco editor otázek do tohoto souboru může i zapisovat.

Ikona aplikace

Ikonu aplikace určuje vlastnost formuláře Icon, její viditelnost v záhlaví okna se nastavuje ve vlastnosti ShowIcon. Ikona této aplikace byla nejdříve vytvořena v grafickém editoru Paint.NET ve formátu běžného obrázku (.png) a následně byla převedena na soubor ikony (.ico) pomocí webové stránky ConvertICO.com. Pro hlavní okno byla použita ikona *vhCSharp.ico*, v záhlaví editoru otázek se žádná ikona nezobrazuje.

Závěr

V této práci jsem se pokusil vytvořit vědomostní hru za použití programovacího jazyka C#. Domnívám se, že je výsledná hra připravena k běžnému používání.

Designově aplikace zapadá do moderního prostředí systému Windows. Rozložení prvků hry se přizpůsobuje velikosti okna, tedy je responzivní. Naneštěstí se nezvětšuje i velikost písma, což by se ale mohlo objevit v dalších verzích programu.

Hra umožňuje pojmenovat oba dva proti sobě hrající soutěžící. V průběhu hry vidí právě hrající hráč své jméno, již získané body, zbývající čas na zapsání odpovědi a čas do skončení soutěže. Doba na zapsání odpovědi se přiděluje podle délky otázky, celkový čas hry si hráči volí sami. Hráčům je umožněno hru pozastavit i ukončit. Na konci hry se zobrazí výherce.

Nejobtížnější v této práci bylo vymyslet postup, který by umožnil generování náhodných otázek tak, aby se neopakovaly. Výsledného řešení tohoto problému bylo dosaženo pomocí seznamů, avšak můj algoritmus umožňuje opakování dvou stejných otázek za sebou, jestliže po vylosování poslední otázky ze seznamu se vygeneruje další nová, která se losuje znovu z celého původního rozsahu. Jedná se však o chybu, ke které při větším počtu otázek bude docházet zřídka.

Uživatelům tento program umožní i editaci otázek a odpovědí. Budou moci otázky a jejich řešení upravovat, mazat nebo je přidávat. Uživatelé by však měli mít na paměti, že program umožňuje zadání pouze jedné správné odpovědi.

Jako hlavní místo využití této aplikace spatřuji hlavně ve školách, zvláště v hodinách kdy se pracuje s interaktivní tabulí.

Díky této práci jsem se naučil lépe pracovat s funkcemi, seznamy a prvkem dataGridView umožňujícím zobrazení tabulky. Odevzdávám tuto práci s nadějí, že si hra najde své uživatele, kterým umožní trénink svých znalostí.

Zdroje

Pro vytvoření práce byly použity tyto zdroje:

C Sharp (programming language) - *Wikipedia* [online]. [cit. 2017-03-01]. Dostupné z: [https://en.wikipedia.org/wiki/C_Sharp_\(programming_language\)](https://en.wikipedia.org/wiki/C_Sharp_(programming_language))

MonoDevelop [online]. [cit. 2017-03-01]. Dostupné z: <http://www.monodevelop.com/>

Rider Features [online]. [cit. 2017-03-01]. Dostupné z: <https://www.jetbrains.com/rider/features/>

Data Type Ranges [online]. [cit. 2017-03-02]. Dostupné z: [https://msdn.microsoft.com/en-us/library/wew5ytx4\(v=vs.90\).aspx](https://msdn.microsoft.com/en-us/library/wew5ytx4(v=vs.90).aspx)

Segoe MDL2 icon guidelines - UWP app developer | Microsoft Docs [online]. [cit. 2017-03-18]. Dostupné z: <https://docs.microsoft.com/cs-cz/windows/uwp/style/segoe-ui-symbol-font>

C# - Set Segoe UI Symbol font programmatically - Stack Overflow [online]. [cit. 2017-03-18]. Dostupné z: <http://stackoverflow.com/a/17285149>

Lists - Learn C# - Free Interactive C# Tutorial [online]. [cit. 2017-02-16]. Dostupné z: <http://www.learncs.org/en/Lists>

How to Remove a Line From a Text File at Specific Location in C# [online]. [cit. 2016-12-27]. Dostupné z: <http://www.c-sharpcorner.com/blogs/how-to-remove-a-line-from-a-text-file-at-specific-location-in-c-sharp1>)

C# - How to Bind dataGridView with Datatable In C# - C#, JAVA,PHP, Programming ,Source Code [online]. [cit. 2017-02-27]. Dostupné z: <http://1bestcsharp.blogspot.cz/2014/12/c-how-to-bind-datagridview-with.html>)

C# -How To Export dataGridView Data to txt file - C#, JAVA,PHP, Programming ,Source Code [online]. [cit. 2017-02-27]. Dostupné z: <http://1bestcsharp.blogspot.cz/2014/12/c-how-to-export-datagridview-data-to.html>)

Seznam příloh

CD: Dokument práce ve formátu PDF

CD: Projekt Microsoft Visual Studio Vědomostní hra v jazyce C#