

Windows Presentation Foundation a kytara

bakalářská práce

Jan Krajčovič

Vedoucí bakalářské práce: Ing. Václav Novák, CSc.

Jihočeská univerzita v Českých Budějovicích

Pedagogická fakulta

Katedra informatiky

2010

Anotace

Tato práce se obecně zabývá technologií Windows Presentation Foundation, jež je zaměřena zejména na vytváření aplikací s uživatelsky přívětivým rozhraním, určeným pro operační systémy Windows Vista (a novější). Pomocí technologie WPF vytvořil autor této práce výukový program pro hraní na kytaru, jehož testování na malém vzorku žáků a vyhodnocení dosažených výsledků je hlavní cíl této práce.

Abstract

This work deals with general overview about technology of Windows Presentation Foundation which is specialized for creating applications with user friendly interface designated for operating systems Windows Vista (or later). By using technology of WPF author of this thesis created tutorial for playing guitar. Few of students will test the program. Main goal of this thesis is evaluation of testing student's achieved results.

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách.

V Českých Budějovicích dne 15. 4. 2010

.....
Jan Krajčovič

Poděkování

Rád bych na tomto místě poděkoval svému vedoucímu práce
Ing. Václavu Novákovi, CSc. za cenné rady a čas, který mi věnoval při
osobních konzultacích.

Obsah

MOTIVACE	14
1 ÚVOD	15
1.1 VÝVOJ GRAFICKÝCH UŽIVATELSKÝCH ROZHRAŇÍ PŘED WPF	16
1.2 SEZNÁMENÍ S WPF	18
1.2.1 <i>Architektura WPF</i>	20
1.2.2 <i>DirectX</i>	21
1.2.3 <i>Grafické funkce</i>	22
1.2.4 <i>Aplikační rozhraní vyšší úrovně</i>	23
1.2.5 <i>XAML</i>	27
2 SOUČASNÝ STAV.....	30
2.1.1 <i>Podle zpracování</i>	30
2.1.2 <i>Podle využití</i>	33
2.1.3 <i>Podle dalšího využití</i>	35
2.1.4 <i>Způsoby výuky</i>	36
2.1.5 <i>Další zaměření</i>	38
3 KONCEPCE PROGRAMU	39
3.1 CÍLOVÁ SKUPINA UŽIVATELŮ	39
3.2 STRUKTURA A ZAMĚŘENÍ VÝUKY	41
3.2.1 <i>Teoretický úvod</i>	42
3.2.2 <i>Výuka akordů</i>	44
3.2.3 <i>Výukové lekce</i>	45
4 METODIKA.....	47
4.1 PŘED VYTVOŘENÍM PROGRAMU	47

4.2	TVORBA PROGRAMU.....	49
4.3	PO VYTVOŘENÍ PROGRAMU.....	52
4.3.1	<i>Hodnocení</i>	52
4.3.2	<i>Testování</i>	53
4.3.3	<i>Výsledky testování</i>	55
5	ZÁVĚR	58
	LITERATURA	60
A	INSTALACE A SPUŠTĚNÍ PROGRAMU	61
B	OBSAH PŘILOŽENÉHO CD	62
C	DOTAZNÍKY	63

Seznam obrázků

<i>Obrázek 1: Architektura WPF.</i>	21
<i>Obrázek 2: Automatické generování kódu XAML.</i>	28
<i>Obrázek 3: Virtuální hmatník.</i>	36
<i>Obrázek 4: Video lekce.</i>	38
<i>Obrázek 5: Srovnání: řazení strun v tabulaturách vs. špatná představa začátečníka.</i>	42
<i>Obrázek 6: Vizualizace poloh jednotlivých prstů.</i>	43
<i>Obrázek 7: Class diagram (program Kytarace).</i>	51

Seznam grafů

<i>Graf 1: Výsledky dotazníku č. 1.</i>	49
<i>Graf 2: Schopnosti uživatelů před absolvováním výuky.....</i>	54
<i>Graf 3: Schopnosti uživatelů po absolvování výuky.....</i>	56
<i>Graf 4: Výsledky dotazníku č. 5.</i>	57

Motivace

Pokud se člověk rozhodne, že se naučí hrát na kytaru, není hned nutné začít navštěvovat lidovou školu umění a učit se vše od úplného základu. Někdy je, vzhledem k věku, již poněkud pozdě a většinou ani není třeba začít takový druh školy navštěvovat. Aby byl člověk schopný, zahrát si jednoduchou píseň ze zpěvníku, stačí umět pár základních akordů, které jsou ve většině takových publikací jednoduše popsány. Pokud se ovšem začínající hráč na kytaru naučí nějaké základní akordy a má zájem své schopnosti dále rozvíjet, začne přemýšlet, jak nejlépe se naučit hrát také různé stupnice apod. Obecně by se každý jistě rád vyvaroval zbytečných chyb, kterých se mnozí začátečníci dopouští. Způsobů, jak získat pevné základy, díky kterým pak může každý své schopnosti dále rozvíjet i bez cizí pomoci, je několik. Lze začít navštěvovat soukromé lekce, vyzkoušet výukový videokurz, nebo se pokusit vzdělat pomocí výukového programu.

Jelikož jsem sám aktivním kytaristou, který se naučil hrát na kytaru v podstatě pouze s dopomocí programu, který je určený pro zápis, tvorbu a přehrávání hudby. Rozhodl jsem se, že se pokusím vytvořit, na základě vlastních zkušeností, výukový program pro začínající kytaristy. Protože se většině lidí z mého okolí líbí moderní vzezření nových operačních systémů od firmy Microsoft (tj. Windows Vista a Windows 7) a mě zaujala nová technologie programování Windows Presentation Foundation, určena právě pro tyto operační systémy, rozhodl jsem se, že tento program vytvořím právě pomocí WPF.

1 Úvod

V současné době jsou lidé po celém světě zvyklí na počítačové aplikace s plnohodnotným grafickým uživatelským rozhraním. Vše začíná u operačních systémů, které většinou každý uživatel chápe, jako softwarový základ počítače a tudíž také právem očekává zejména příjemné grafické a snadno ovladatelné uživatelské rozhraní. Co se týče České republiky, tak je zde stále nejpoužívanější a nejoblíbenější operační systém Windows od firmy Microsoft.

Ke tvorbě aplikací speciálně pro předposlední vývojovou verzi Windows, tj. Windows Vista, byl vyvinut nový framework s názvem Windows Presentation Foundation (dále jen WPF). Tento framework nabízí nové možnosti zejména v oblasti grafického zpracování aplikace. Předchůdcem WPF byla technologie s názvem WinForms, na niž WPF volně navazuje, avšak ji přímo nenahrazuje. Obecné seznámení s WPF je také dílčím úkolem této práce.

Program, jehož testování (na malém množství studentů) a vyhodnocení, je hlavním cílem této práce, byl dle zadání vyvinut právě ve WPF.

Jedná se o výukový program, s jehož dopomocí by se jeho uživatelé (začínající hráči na kytaru) měli naučit základům hraní na kytaru. Takový typ programu je ideální vytvořit ve WPF, protože právě znázorňování akordů, či přehrávání výukových lekcí, je ve většině výukových kurzů řešeno pomocí videí. Úkolem je pokusit se právě tento klasický postup úplně eliminovat a vše vyřešit pouze pomocí přehledných a zároveň jednoduchých grafických prvků, jež WPF poskytuje.

1.1 Vývoj grafických uživatelských rozhraní před WPF

U klasických zobrazovacích technologií je takřka nemožné oddělit grafický obsah od programového kódu. U technologie WinForms je hlavním problémem, že každý formulář (okno programu), který vytvoříte, je definován výhradně kódem v jazyku C#. Jakmile je nějaký prvek přesunut na pracovní plátno a tam dále upravován, Visual Studio veškerou činnost automaticky převede na kód, obsažený v odpovídající třídě. Grafičtí designéři tak nemají možnost žádným, jim přirozeným nástrojem, grafický vzhled aplikace upravit. Neexistuje totiž takový nástroj, který by byl určen pro tvorbu grafiky a zároveň uměl pracovat s kódem, psaném v jazyku C#. Designéři jsou tak nuceni své návrhy převést na bitmapové obrázky. Až po té je možno tyto obrázky použít k úpravě vzhledu formulářů, tlačítek, či jiných prvků. Při jednoduchých, či částečných úpravách grafického vzhledu aplikace, tento postup není nijak zdlouhavý, ani komplikovaný. Nicméně v některých situacích se takový postup stává značně omezujícím.

Problémy, ze kterých zmíněná omezení vycházejí

- Každý grafický prvek (pozadí, tlačítko atd.) je nutné zvlášť vytvořit a převést na bitmapový obrázek. To omezuje možnost kombinovat různé bitmapové obrázky a používat dynamické efekty jako je vyhlazování hran, průhlednost, nebo stínování.
- Pro některé funkce uživatelských rozhraní je nutné, aby byly začleněny přímo v programovém kódu. To platí pro vlastnosti, jako jsou velikosti tlačítek, pozicování, tzv. mouse over efekty, nebo animace. Grafici pak tyto funkce de facto nemohou ovlivnit.

- Velikost bitmapových obrázků nelze měnit, aniž by tato změna neměla vliv na jejich kvalitu. Uživatelské rozhraní, založené na bitmapovém obrázku je tím pádem závislé na rozlišení. V praxi to znamená, že výsledné zobrazení aplikace není odpovídající na širokoúhlých monitorech, či displejích s vysokým rozlišením. To je hlavní rozdíl oproti základním principům tvorby designu WPF technologií.

Tvorba vlastního grafického vzhledu WinForms aplikací tedy pro vývojáře znamená přínos spousty komplikací. Pokud je k dispozici nejdříve grafický návrh, tak musí programátor veškeré úpravy převést do C# kódu.

WPF tyto problémy řeší pomocí XAMLu. Upravování grafického designu WPF aplikací ve Visual Studiu, neznamena, že se, jako tomu je u WinForms, veškeré změny překládají do programového jazyka. Namísto toho se vše zaznamenává pomocí XAML tagů (značek) do souboru s příponou *.xaml*. Při spuštění aplikace jsou tagy použity pro generování objektů, které dohromady tvoří uživatelské rozhraní. [4]

1.2 Seznámení s WPF

Grafický framework WPF (dříve známý pod názvem Avalon) je součástí .NET Framework (od verze 3.0) a je určen zejména pro tvorbu grafických uživatelských rozhraní aplikací, běžících na systému Windows od firmy Microsoft. WPF je součástí Windows 7, Windows Vista a Windows Server 2008. Je možné ji dodatečně nainstalovat i do Windows Server 2003 a Windows XP. Je totiž součástí aktualizčního balíčku s názvem Service Pack 2 (a novějších verzí). WPF aplikace není možné spouštět na jiných operačních systémech, než je právě Windows.

WPF se stala následovníkem WinForms, nejedná se sice o přímou náhradu, ale spíše o novější variantu. V oblasti grafického zpracování aplikace je však ve WinForms řešení některých problémů velice komplikované, nebo dokonce nemožné. Dá se říci, že WPF je nejradikálnější změnou, v oblasti tvorby uživatelského rozhraní aplikací, určených pro operační systémy Windows, od vzniku Windows 95. To znamená, že více než patnáct let vývojáři vyvíjeli stejnými technologiemi všechna uživatelská rozhraní, počínaje aplikacemi pro Windows 95, konče u Windows XP. Standardní Windows aplikace se tedy výhradně opírala o dvě spolehlivé, avšak postupem času pomalu zastarávající části pro vývoj uživatelského rozhraní.

User32 - poskytuje klasický vzhled prvků ve Windows, jako jsou tlačítka, text-boxy a vlastně celé prostředí, ve kterém se každý běžný uživatel pohybuje.

GDI/GDI+ - zajišťuje vykreslování různých obrazců, textu a obrázků, avšak poměrně komplikovaným způsobem a často s ne

příliš přesvědčivým výsledkem (pokud bereme v potaz poměr využití výkonu počítače/celkový dojem).[4]

WPF obsahuje sadu knihoven poskytujících aplikační rozhraní (API), známou jako DirectX [1]. Díky tomu je možné využívat hardwarové akcelerace a jednoduše pracovat s novými funkcemi, jako jsou například zprůhlednění, natáčení, či transformace. WPF aplikace jsou renderovány na grafické kartě a tím se ulehčuje práce procesoru počítače [2].

Technologie WPF je nový způsob programování a je ovlivněna moderními zobrazovacími technologiemi, jako jsou HTML, či Flash [2]. To v praxi znamená, že je rozdělen vývoj aplikace na dvě části, tzn. je oddělena část, kde se vytváří grafické uživatelské rozhraní (XAML), tj. vzhled, od samotného funkčního programového kódu (C#, Visual Basic). Jedná se v podstatě o dvě oddělená, vzájemně spolupracující vývojová prostředí a také dva různé soubory. Pro upřesnění je třeba dodat, že pokud není uvedeno jinak, vývojovým prostředím, určeným pro tvorbu WPF aplikací, je (v této práci) míněno Microsoft Visual Studio 2008 (a novější).

Uživatelské rozhraní je tvořeno v novém, značkovacím jazyce XAML (eXtensible Application Markup Language). Ten je jednoduchý, rychlý a přehledný. XAML umožňuje vytvářet uživatelské prvky a vztahy mezi nimi, stejně tak lze tvořit kontroly, vlastní design a využívat další možnosti, jako jsou např. 2D i 3D kreslení, vektorovou i rastrovou grafiku, data binding, zvuky, videa, či vytvářet animace.

1.2.1 Architektura WPF

Presentation framework – obsahuje velký počet standardních prvků, jako jsou tlačítka, menu, panely, labely a další. S touto vrstvou pracuje valná většina vývojářů WPF aplikací.

PresentationCore.dll – zahrnuje základní subsystemy, jako jsou UIElement a Visual, ze kterých vychází uspořádání vzhledu WPF aplikací (tzv. layout), způsoby vykreslení scén apod.

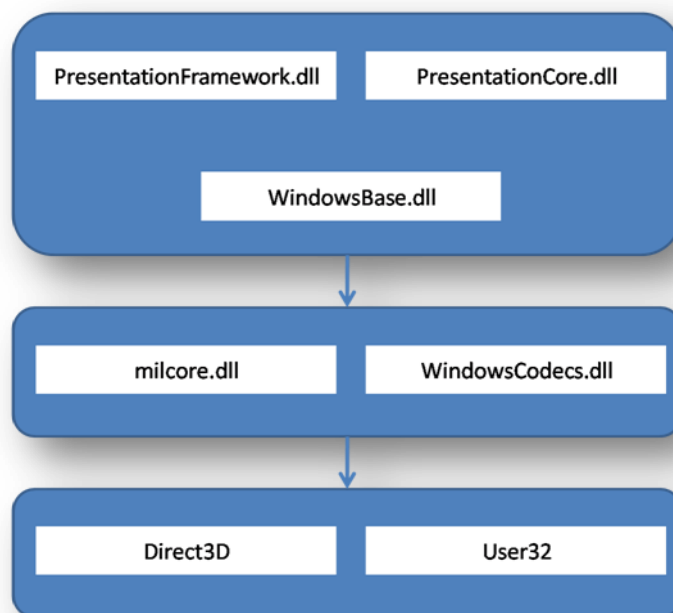
WindowsBase.dll – obsahuje další ze základních nástrojů, jako jsou třídy DispatcherObject a DependencyObject jež se dají využít i mimo WPF.

Milcore.dll – je jádrem systému zobrazení WPF a základ vrstvy MIL (Media Integration Layer). Úkolem milcore.dll je přetvářet veškeré vizuální prvky na trojúhelníky a textury, které vyžaduje Direct3D. Ačkoliv je knihovna milcore.dll považována za součást WPF, tak je i základní systémovou součástí Windows Vista. Využívá ji DWM (Desktop Window Manager) při vykreslování plochy.

WindowsCodecs.dll – je nízkoúrovňové programovací rozhraní (neboli API, tzn. Application Programming Interface), poskytující podporu zobrazování, tj. zpracovávání, zobrazování, či modelování bitmapových, nebo JPEG obrázků.

Direct3D - nízkoúrovňové aplikační rozhraní, skrze které se renderují (vykreslují) veškeré grafické prvky WPF aplikací.

User32 – používá se pro určení toho, s čím ve skutečnosti program pracuje. Je sice součástí WPF, nemá však žádný vliv na renderování většiny prvků. [5]



Obrázek 1: Architektura WPF. Zdroj: [5]

1.2.2 DirectX

Sada knihoven s názvem DirectX byla vytvořena společností Microsoft, aby bylo možné vyhnout se při vývoji grafických aplikací omezením vycházejícím z knihoven User32 a GDI/GDI+. Zpočátku byly knihovny DirectX poměrně nestabilním nástrojem, sloužícím k vývoji her na platformě Windows. Microsoft začal spolupracovat s výrobcí grafických karet, aby mohly mít knihovny DirectX lepší přístup přímo k hardwarové

akceleraci, potřebné pro zobrazování složitějších textur, nebo speciálních efektů (částečná průhlednost, 3D grafika atd.).

Poprvé se DirectX (s podporou všech moderních grafických karet) stal nedílnou součástí Windows ve verzi Windows 95. Na programovacím rozhraní pro DirectX byl však stále znát vliv původního účelu (nástroj pro vývojáře her). Celková spletnost knihoven DirectX zapříčinila, že se nikdy DirectX nevyužil při vývoji klasických Windows aplikací (např. podnikové softwary).

WPF však své základy technologie grafiky nestaví na GDI/GDI+, ale na DirectX. Není to nástavba na GDI/GDI+, ale přímo jeho náhrada. Je to oddělená vrstva pracující s DirectX. WPF aplikace používají knihovny DirectX nehlédě na to, jaký typ uživatelského rozhraní je vytvářeno. Nezáleží, zda je tvořeno 3D grafikou, nebo tlačítka a prostým textem. Veškeré vykreslování vždy probíhá skrze DirectX. [4]

1.2.3 Grafické funkce

Všechny grafické funkce jsou renderovány prostřednictvím Direct3D. Díky tomu se příliš nezatěžuje procesor, ale naopak se všechny výpočty řeší pomocí grafické karty, respektive grafického procesoru (GPU – Graphic Processing Unit) na této kartě. Díky paralelnímu zpracování pixelů, pracují grafické procesory rychle a umožňují tak i zrychlení obnovování obrazu. Na využití procesoru počítače dojde až v situaci, kdy grafická karta není schopna vše zvládnout sama. Na pohled složitá grafická část aplikace tedy již neznamená snížení výpočetního výkonu počítače, jako tomu bylo dříve u WinForms.

Úkolem WPF je poskytnout programátorovi komplexní nástroj, který mu umožní v jednom okně používat, či vytvářet různé, a to nejen grafické, prvky (tj. 2D a 3D grafiku, animace, média atd.).

WPF poskytuje, oproti WinForms a jeho GDI, pokročilejší grafické funkce. [3]

1.2.4 Aplikační rozhraní vyšší úrovně

WPF nabízí, programátorům aplikací pro Windows, množství moderních prostředků, které slouží nejen pro ulehčení práce, ale nabízí i nové možnosti.

Vybrané změny, které WPF přináší oproti předchozím technologiím

- **Rozvržení (layout) aplikace** – přibyly nové možnosti ke klasickým postupům, kdy se veškeré prvky, zobrazované v okně aplikace, pevně pozicují danými souřadnicemi. WPF klade důraz, aby bylo možné snadno libovolně přizpůsobit rozvržení veškerých prvků v okně.
- **Širší možnosti kreslení** – ve WPF je umožněno pracovat s jednoduchými grafickými prvky (jako jsou například základní tvary), které umožňují rychlejší a snadnější tvorbu vlastních grafických prvků.
- **Textový model** – dosavadní možnosti práce s textem a textovými prvky byly poměrně omezené. WPF umožňuje vytvořit a zobrazit vlastní styly textu a textových prvků. Je možné kombinovat text

se seznamy, obrázky, či jinými prvky, tvořícími uživatelské rozhraní. Například při práci s rozsáhlým textem je možné používat pokročilých možností formátování a zobrazení textu, jako je zalamování, dělení do sloupců, nebo zarovnávání.

- **Animace** – tvorba animací je součástí frameworku WPF. Není tedy pokaždé třeba používat časovač k pravidelnému vyvolávání překreslení celého okna (pro navození dojmu plynulého pohybu). Díky vektorové grafice a implementaci animací ve WPF lze vytvářet efekty, jejichž tvorba byla ve WinForms zcela nemožná. Vytváření animací je velice jednoduché a ve většině případů se tvoří výhradně v jazyce XAML. Ve WPF se s animacemi nepracuje s klasickou časovou osou, ale s vlastnostmi jednotlivých kontrol. Animace je v podstatě generátor hodnot, které jsou postupně přiřazovány některé z vlastností kontroly. Animace lze různě kombinovat ve stylech i šablonách. Ve WPF jsou tři typy animací (základní, frame a path), které se vzájemně liší způsobem generování hodnot. Základní typ animace je v podstatě lineární generátor hodnot, který generuje hodnoty od počáteční po koncovou. Frame animace umožňují zadat více, než dvě hodnoty (framey). Jsou tři typy framů – lineární (hodnoty se generují postupně), diskrétní (nedochází k postupnému přechodu mezi vygenerovanými hodnotami) a spline (hodnoty jsou generovány podle křivky). Path je speciální, zřídka používaný typ animace, který generuje hodnoty na základě zadané trajektorie.

- **Podpora médií** – u předchozích technologií byly nástroje, co se týče práce s médii, pro tvorbu uživatelských rozhraní poměrně omezené. WPF obsahuje podporu pro přehrávání veškerých zvukových, nebo video souborů, které podporuje Windows Media Player. Možné je také přehrávat více médií najednou, nebo třeba umístit video na plochu nějakého 3D objektu.
- **Styl** – styl je skupina vlastností jako například barva pozadí, velikost a styl písma, nebo šířka a barva rámečku, kterou lze aplikovat na vybrané kontroly. Používání stylů dělá kód flexibilní a zároveň přehlednější, eliminuje se totiž neustálé opisování stejného kódu, například u vlastností stejných kontrol. Používání stylů ve WPF funguje obdobně jako CSS. U stylů lze využít vlastnosti zvané dědičnost. Dědičnost umožňuje založit styl nový a to na vlastnostech již vytvořeného stylu. K jeho vlastnostem lze pak další přidat, nebo stávající upravit.
- **Šablony** – pomocí šablon lze kompletně změnit vzhled dané kontroly a nahradit tak standardní vzhled aplikace. Například u tlačítka lze i bez šablon, tj. pouze úpravou vlastností, změnit velikost písma, nebo upravit barvu pozadí. Přepsáním šablony tlačítka je možné upravit i jeho tvar, nebo přímo nadefinovat prvky, ze kterých se bude skládat. Šablony lze používat i přímo ve stylech, takže se dá upravit třeba vzhled všech tlačítek v aplikaci.
- **Spouštěče** – umožňují reagovat na události kontrol (*event triggers*), hodnoty jejich vlastností (*property triggers*), nebo datové hodnoty (*data triggers*). *Property trigger* lze nastavit

hodnoty vlastností, nebo pracovat s animacemi. Pokud podmínka spuštění přestane platit, vrátí se hodnoty zpět do původního stavu. Pokud je třeba u *property trigger* spustit animaci při nastání více podmínek najednou, používá se tzv. *multi trigger*. *Event trigger* pracuje pouze s animacemi. *Data trigger* se používá pouze na vlastnosti kontrol. Umožňuje reagovat na hodnotu uživatelských proměnných a využívají se zejména v tzv. data bindingu.

- **Data binding** – umožňuje v XAMLu pracovat s daty. Je to jakési vzájemné provázání vlastností. Změna hodnoty zdrojové vlastnosti okamžitě vyvolá změnu v provázaných vlastnostech. Stejně tak se změna v provázaných vlastnostech ihned projeví i ve zdrojové vlastnosti. Data binding tedy představuje obousměrnou synchronizaci hodnot.

Existují tři způsoby synchronizace hodnot. Způsob *TwoWay* synchronizuje hodnoty obousměrně. Změna hodnoty na zdroji se projeví na hodnotě provázané (nabindované) vlastnosti a naopak. *OneWay* synchronizuje hodnoty pouze ze zdroje do nabindované vlastnosti. Změna hodnoty nabindované vlastnosti je zdrojem přehlížena a neprojeví se. Při použití módu *OneTime* jsou hodnoty synchronizované pouze při první změně a po té jsou ignorovány.

Změnou hodnoty atributu *UpdateSourceTrigger* lze nastavit, kdy se projeví změna na zdrojové hodnotě. Hodnota *Explicit* má za následek, že k synchronizaci dojde pouze pokud se volá metoda *Update Source*. Při hodnotě *LostFocus* se zdrojová hodnota aktualizuje až po té, co přestane být aktivní nabindovaná kontrola. *PropertyChanged* zajistí aktualizaci zdrojové hodnoty při změně nabindované hodnoty.

Data Binding není pro tvorbu aplikací ve WPF zcela nezbytný. Případy, kdy je vhodné data binding využít, jdou vyřešit i pomocí událostí volaných metod. Takto však nelze postupovat v XAMLu, kde se právě data binding nejlépe uplatní. [2] [4] [6] [7] [8] [9]

1.2.5 XAML

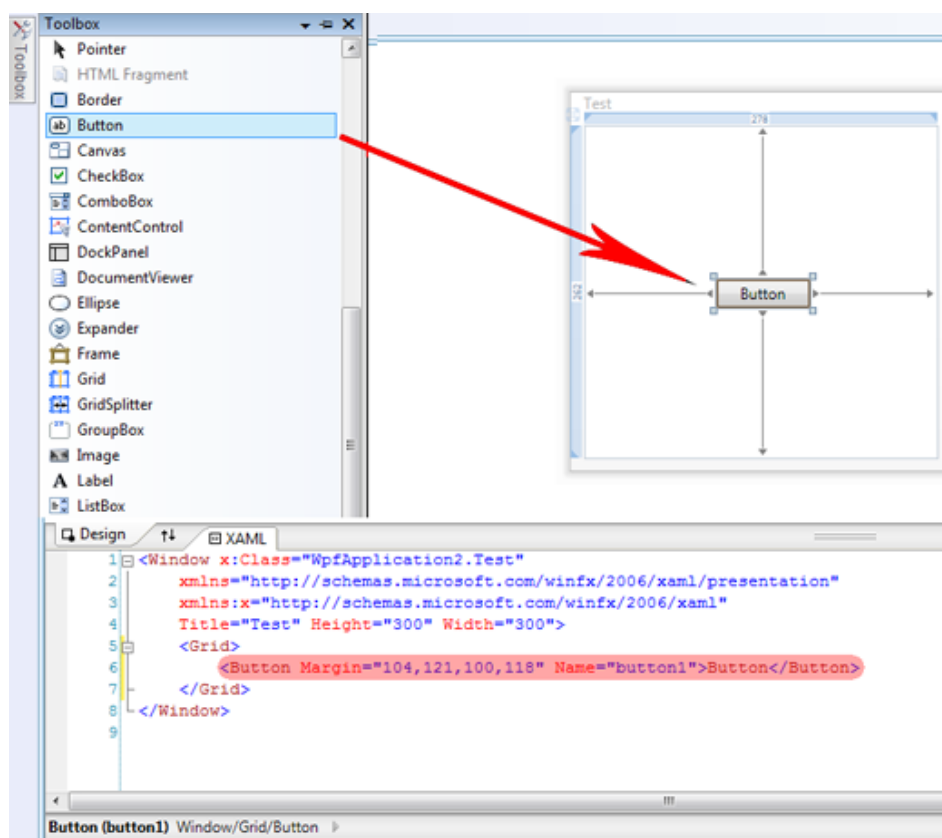
Jak již bylo zmíněno, tvorba aplikace ve WPF je rozdělena na dvě části a každá část má svoje vývojové prostředí a jazyk. Oddělení základního programového kódu, od kódu, určeného k vývoji grafické části aplikace, má jasný účel. Obě části mohou být zpracovávány odděleně, a tudíž mají svůj vlastní kód, potažmo prostředí i samotní grafici.

XAML je jazyk, jež je výhradně určený pro grafický oddíl vývoje. Je tedy určen zejména pro tvorbu uživatelského rozhraní, tzn. rozvrh a umístění různých panelů, tlačítek, či kontrol, dohromady tvořících jednotlivá okna WPF aplikace. Ačkoliv je XAML technologie, která může být využita k řešení problémů v mnoha rozdílných sférách, hlavní jeho úlohou je právě tvorba uživatelských rozhraní WPF aplikací.

Na rozdíl od klasického programového kódu se XAML téměř vůbec nepíše ručně, až na výjimky jako je tvorba šablon, stylů, animací apod. Kód se tedy obvykle generuje automaticky na základě změn, provedených v editoru designu zvaném Cider. Díky nástrojům (jako je drag and drop), jež usnadňují vývoj grafického rozhraní, může vzniknout mylný dojem, že není třeba syntaxi XAMLu rozumět. Do jisté míry je možné se bez takových znalostí obejít, ne však v případech jako je již zmíněná tvorba šablon atp. [4]

1.2.5.1 Cider

Cider je WPF designer, který je součástí Visual Studia (od verze 2008). Je to takzvaný WYSIWYG (What You See Is What You Get) editor pro tvorbu, či úpravu vzhledu uživatelského rozhraní WPF aplikací. Ve WYSIWYG editorech všeobecně platí pravidlo, že vše, co vývojář změní a vidí v editoru, se naprosto totožně projeví a tudíž i zobrazí ve spuštěné aplikaci. Toto pravidlo samozřejmě platí i pro Cider. Stejně jako tomu bylo u WinForms, i zde se pracuje technikou zvanou *drag and drop* (táhni a pusť). Ta umožňuje umístění libovolného objektu (např. tlačítko, text box apod.) přímo na pracovní plátno, přičemž se tato aktivita okamžitě projevuje i v generovaném XAML kódu.



Obrázek 2: Automatické generování kódu XAML.

1.2.5.2 XAML – varianty

XAML může být používán i v jiných platformách, kde slouží k znázornění jiných objektů, než klasických objektů, jako jsou např. tlačítka, známá z .NET frameworku.

Některé verze XAMLu

- **WPF XAML** obsahuje prvky, sloužící ke tvorbě veškerých vizuálních složek (jako jsou různé ovladače, vektorová grafika apod.), tvořících vzhled WPF aplikace. V současnosti je využívání XAMLu nejrozšířenější právě pro tyto účely ve WPF.
- **XPS XAML** je část WPF XAML, která definuje zastoupení XML pro elektronické formáty dokumentů. XPS XAML byl vytvořen jako samostatný standard XPS (XML Paper Specification).
- **Silverlight XAML** je podmnožinou WPF XAML, určenou pro Silverlight aplikace. Silverlight je multiplatformní plugin pro prohlížeče poskytující možnost vytvoření webu s bohatým rozhraním obsahujícím 2D grafiku, animace, audio a video.
- **WF XAML** zahrnuje prvky, tvořící obsah vzhledu aplikací Windows Workflow Foundation.

2 Současný stav

Publikací ve formě textu, ať už knih, nebo webových stránek, existuje mnoho. Ale stejně jako je někdy obtížně porozumět, co se snaží autor takové publikace popsat, či vysvětlit, tak je i komplikované předat zkušenosti začínajícím hráčům skrze výukový program. Existuje spousta programů - výukových kurzů, které byly vytvořeny, aby bylo možné naučit se s jejich dopomocí hrát na nějaký hudební nástroj. Zřídka se vyskytne výukový program, určený výhradně pro výuku jednoho konkrétního nástroje. Co se týče kytary, tak je tento nástroj po celém světě natolik oblíbený, že není problém narazit na výukový program, určený výhradně začínajícím hráčům na kytaru (a to jak klasickou, tak elektrickou). Naopak se lze setkat téměř s přemírou těchto typů aplikací. Tudíž se nejen kvalita zpracování, ale i metoda samotných výukových postupů (a z nich vycházejících nabitých zkušeností, jež získají začínající hudebníci využíváním programu), mnohdy velmi liší. Proto je dobré tyto programy rozlišovat podle různých kritérií.

2.1.1 Podle zpracování

Výukový program pro hraní na kytaru může vytvořit, stejně jako jakýkoliv jiný program, v podstatě kdokoliv. Zapojit se tak do řady tisíců výukových programů může se svým výtvořem každý, kdo má základní znalosti z jakéhokoliv programovacího jazyka a má pocit, že by se o své zkušenosti, ohledně hry na kytaru, měl podělit s ostatními. S ohledem na lepší orientaci mezi programy je tedy lepší, rozdělovat je dle způsobu zpracování do následujících tří kategorií.

2.1.1.1 Profesionální zpracování

U profesionálně zpracovaných programů, se dá všeobecně předpokládat, že jsou vyvíjeny několika týmy pracovníků (programátoři, grafici, pedagogové, profesionální hudebníci atd.), pracují bezchybně, grafické zpracování je líbivé a funkční, ale také cena za takový produkt je těmto aspektům obvykle přímo úměrná.

Některé profesionálně zpracované programy však, obvykle ve snaze obsáhnout více problematik (např. výuka hraní na více nástrojů, nebo více možných využití programu), mohou být poměrně přehlcené a na uživatele tak působí dojmem celkové nepřehlednosti. Nepřehlednost vede, ve většině případů, k tomu, že se uživatel přestane o další funkce zajímat, nebo si jich nevšimne a tudíž dochází k neúplnému využití. Uživatel také nemusí vůbec zjistit, jak některé funkce pracují. Tvůrci takovýchto programů jsou si zpravidla vědomi, že je jejich produkt může působit těžkopádně. Za účelem lepšího pochopení programu a jeho funkcí se tedy vytváří propracovaná nápověda. Někdy jsou však tyto manuály tak rozsáhlé, že jsou vlastně pro uživatele pouze další zbytečnou součástí programu. Problém vysvětlení práce s programem však nemusí vyřešit ani přehledná interaktivní příručka. V takovém případě však již není chyba na straně tvůrců programu, ale na straně uživatelů, kteří obecně nejsou zvyklí (a tudíž ani neumějí) pracovat s nápovědou.

2.1.1.2 Poloprofesionální

Tato kategorie zpracování programů je poměrně těžko definovatelná. Hranice mezi poloprofesionálním a amatérským zpracováním programu může být totiž velice sporná. Je tedy lepší, rozdělit ji na dvě podkategorie.

Nekomerční - Můžeme říci, že se jedná o projekty, které jsou původně spíše amatérským dílem. Dosahují však kvalit profesionálních programů a zároveň jsou k dostání zdarma.

Komerční - Stejně tak ale můžeme hovořit o programech, které jsou tvořeny s komerčním úmyslem, ale cena není přímo úměrná zpracování a ani výsledkům, ke kterým vede používání těchto programů. V některých případech tedy časem dojde k situaci, kdy začne být takový, co se prodaje týče, neúspěšný program nabízen zdarma. Bývá pak zvykem, že součástí programů je jakási reklamní plocha zajišťující tvůrcům programu alespoň nějaký výdělek.

2.1.1.3 Amatérské

Do kategorie amatérsky zpracovaných programů spadají ty, které vytvořil, obvykle jeden, programátor (vývojář). Tyto programy zpravidla vytváří sami hudebníci a to za účelem zjednodušení práce, potažmo ušetření času, při cvičení hry na kytaru. Většinu takových produktů je pak možno dohledat na internetu a zdarma stáhnout i používat.

Amatérsky zpracované programy mohou být vytvořeny různými způsoby. Od webových kurzů, kde je výuka realizována pomocí fotografií, či videí, přes jednodušší (např. java, či WinForms) aplikace, až po poměrně propracované projekty. V této kategorii tedy nutně nemusí být každý projekt automaticky označen za špatný, neúplný, nebo dokonce bezvýznamný. Nicméně je samozřejmě vyšší pravděpodobnost, že většina programů nebude koncipována jako profesionální výukový kurz.

2.1.2 Podle využití

Využití výukového programu, určeného pro začínající kytaristy, se zdá být zcela jasné. Nicméně způsobů zpracování programu je několik a podle toho bývají také možnosti výuky některých programů omezeny. Výukový program, určený začínajícím hráčům na kytaru, může být zaměřen pouze na jednu problematiku, stejně jako může obsáhnout téměř vše, co by kytarista měl umět. Variant různých výukových metod, které je možno v programu obsáhnout, může být několik a nejlépe jsou na tom zpravidla ty, které obsahují kombinace všech (nebo většiny) z níže uvedených možností výuky.

2.1.2.1 Výuka akordů

Pokud je program zaměřen jen a pouze na výuku akordů, jedná se s největší pravděpodobností spíše o amatérské, či poloprofesionální zpracování. Výuka akordů obvykle probíhá pomocí grafického zobrazování daných akordů. Jednoduše se vyobrazí poloha prstů na hmatníku a v lepším případě jsou označeny i čísla jednotlivých prstů,

které tento akord mají hrát. Někdy je grafické vyobrazení nahrazení, nebo doprovázeno autentickými fotografiemi hraných akordů. Pokud jsou fotografie správně nafocené, mohou poskytnout uživateli lepší představu, než pouhé grafické znázornění. Grafické znázornění akordu bývá doplněno i zvukovou interpretací.

2.1.2.2 Výuka not

Existují programy, určené výhradně pro výuku not. Jejich počet je ale výrazně nižší, například oproti těm, co slouží k výuce akordů. Začínající kytaristé totiž chtějí spíše rovnou hrát, než se učit hudební teorii. Proto je obvykle výuka not obsažena v některých z rozsáhlejších projektů. Ani zde se mnohdy nejedná o výuku not a je spíše na každém jednotlivci, zda si vše časem dokáže odvodit, podle různých zápisů hudby.

2.1.2.3 Výuka stupnic, či postupů

Jako tomu je u výuky not, tak i programy, které se zaměřují pouze na výuku stupnic, či různých postupů, jsou spíše výjimkou. Tato problematika bývá součástí rozsáhlejších výukových programů. Výuka stupnic (či postupů) je zpravidla řešena podobně, jako výuka akordů. To znamená, že jsou jednotlivé tóny postupně graficky zobrazovány na virtuálním hmatníku, případně jsou doplněné o čísla prstů, jež tyto tóny mají hrát. Někdy jsou zobrazené i názvy příslušných tónů. Tento postup výuky je někdy nahrazen fotografiemi, nebo videem. Video může uživatele odradit, protože nemusí být příliš přehledné a také chybí regulace rychlosti.

2.1.2.4 Teoretická výuka

Teoretická výuka (týká se zejména držení kytary, poloh rukou apod.) je řešena obvykle textovým, či audio (namluveným) manuálem, s doprovodnými fotografiemi, videem, či grafickým vyobrazením. Teoretickou výukou se opět zabývají (a většinou spíše okrajově) pouze rozsáhlejší výukové programy.

2.1.3 Podle dalšího využití

V současnosti je mnohem snadnější narazit na programy, které se nezaměřují pouze na jeden nástroj, stejně jako nejsou určeny výhradně pro účel výuky. Proto je třeba se zaměřit i na problematiku dalšího využití. Není nutné programy podle dalšího využití rozdělovat do různých kategorií. Je však užitečné o těchto možnostech vědět.

Existují programy, které nemusí být primárně určeny právě pro výuku hry na kytaru. Mnoho programů je, v původním záměru, vytvářeno zejména pro zápis hudby pomocí not, nebo tabulatur. Nabízí ale i možnost následného přehrávání (a to i pomocí autentických zvuků jednotlivých nástrojů), či exportování do zvukových souborů. Škála využití bývá, u kvalitních profesionálních aplikací, opravdu široká. Tyto aplikace jsou povětšinou jakýmsi notovým papírem současných amatérských, ale i profesionálních moderních muzikantů, potažmo skladatelů. Pokud jde o výukové lekce, ty jsou obvykle k programu přidány jako doplňkové, či ukázkové soubory. Lekce tvoří také samotní uživatelé a ukládají je na k tomu určené weby, kde jsou k dispozici i ostatním.

2.1.4 Způsoby výuky

Výuka hry na kytaru může v programech probíhat různými způsoby. Je na každém vývojáři, nebo vývojovém týmu, jaký postup zvolí. Zda se budou držet osvědčených metod, nebo se pokusí vymyslet něco nového. Některé postupy se mohou jedné skupině uživatelů zdát nepřehledné, nejasné, nebo úplně nefunkční a jiným zase ideální pro jejich potřeby.

Pravděpodobně nejoblíbenější realizací výuky je čistě grafické zobrazování akordů, nebo jednotlivých tónů na virtuálním hmatníku. Obvykle bývá právě vyobrazený akord, či tón, synchronizovaně doprovázen zvukovým výstupem.



Obrázek 3: Virtuální hmatník. Zdroj: [program Guitar Pro 5]

Podobně mohou být akordy zobrazovány pomocí fotografií. Zde ale převažují spíše nevýhody a to jak ze strany uživatelů, tak vývojářů. V současné době je jednodušší vytvořit grafický virtuální hmatník, než pořizovat desítky fotografií, přičemž názornost některých fotografií nemusí být zrovna ideální. Využívání fotografií je pro své nevýhody poměrně ojedinělé a používá se spíše v amatérských programech, nebo jako názorné doplnění při zobrazování pomocí virtuálního hmatníku.

Existují výukové programy, jejichž výuka je založena na textově (nebo i hlasem) realizovaných instrukcích. Dalo by se říci, že jsou to spíše knihy převedené do počítačové formy, sloužící jako interaktivní příručka, než přímo jako výukový kurz. Pokud není text, potažmo audio záznam

doprovázen odpovídajícími fotografiemi, jsou uživatelé těchto programů odkázáni čistě na schopnost autora instrukcí, popsat danou problematiku pomocí mluveného slova, či textu.

Další možností, jak někteří vývojáři výukových programů spíše obcházejí způsob vytváření (tj. tvorbu grafických prvků, programování apod.) výukových lekcí sami, je nahrazení čistě grafického zobrazování výukovými videi. Videi se stejně, jako tomu bylo u fotografií, dají použít jako vhodný doplněk grafického zobrazení akordů, či tónů. Video lekce se v programech aplikují zřídka. Spíše se vytváří jako samostatná videa (viz. Obrázek 4), tvořící komplexní výukový soubor lekcí. Názornost sice bývá (samozřejmě dle výkonů kameramanů a figurujících kytaristů) oproti fotografiím výrazně lepší. Oproti čistě graficky zobrazované výuce je sice pro uživatele výhodou, že mohou sledovat například i celkové držení těla, nebo polohy rukou vůči kytáře. Nevýhoda je však v chybějící možnosti regulace rychlosti, což bývá u lekcí, řešených programovým zobrazováním grafických prvků, samozřejmostí.

Welcome to Part 10 of your free Course: "Rock 'n' Roll/Blues Style ". Don't forget to check out the [archive](#) for a huge choice of songs to learn! (All lessons are covered by our 90 Day - 100% satisfaction guarantee)



Obrázek 4: Video lekce.

Zdroj: [http://www.vguitarlessons.com/e_course/part10_467.htm]

2.1.5 Další zaměření

Výukové programy pro kytaristy by samozřejmě mohly být rozděleny do několika dalších kategorií. Zejména rozdělení dle cílových skupin, na které by mohl program být zaměřen, nabízí několik možností. Například dělení dle věkových kategorií uživatelů, jejich dosavadních znalostí, nebo schopností, zaměření na klasickou, či elektrickou kytaru atd. Některé kategorie jsou však buď výjimečné (vzhledem k ojedinělému výskytu programů čistě v dané kategorii), nebo jsou součástí jiné kategorie, či jsou dokonce pro zájemce o výukové programy úplně zbytečné. Poslední zmíněný aspekt tedy naznačuje, že je zbytečné se dalším dělením dále podrobněji zabývat.

3 Koncepce programu

Tato kapitola, potažmo podkapitoly pod ní spadající, pojednává o samotném programu, jehož vytvoření je součástí této bakalářské práce. Zejména se zabývá jednotlivými částmi programu a to jak popisem postupů tvorby těchto částí, tak i vysvětlení jejich funkcí, využití a zaměření. Program (v této práci dále také pod názvem Kytarace) je zaměřen na určitou věkovou skupinu uživatelů a výuka hry na kytaru je v něm rozdělena na několik částí.

Celý program je zaměřen na praktickou výuku hry na kytaru. Jeho primárním cílem tedy je, aby si uživatel opravdu vzal do rukou kytaru a hrál.

Teorii se tedy příliš nezabývá a využívá ji pouze jako úvod do problematiky. Je prakticky nemožné, aby výuka obsahovala vše, co je pro hraní na kytaru nutné. Celá koncepce je tvořena tak, aby uživateli programu poskytla základy hraní, rozvinula jeho zručnost a v budoucnosti umožnila rychleji zvládnout pochopit, i zahrát nové techniky.

3.1 Cílová skupina uživatelů

Celý program je zaměřen zejména na cílovou skupinu lidí, kteří již dosáhli věku alespoň 15 let. Co se týče předchozích zkušeností s hraním na kytaru, tak program není tvořen s určitým zaměřením. Nicméně bylo místy přihlíženo i k předpokladu, že uživatelé budou převážně samouci. Program tedy mohou využívat úplní začátečníci - jako pomůcku pro získání pevných základů pro hraní na kytaru. Stejně jako zkušenější

kytaristé - pro procvičení základů, které by měli ovládat, či rozhýbání prstů i rukou před hraním.

Věkové zaměření striktně neznamená, že program nemohou využívat i mladší začínající hráči na kytaru. Berme tedy tuto věkovou hranici spíše jako orientační.

Důvody zvolení spodní věkové hranice uživatelů programu

- Vzhledem k tomu, že osobně nemám žádnou pedagogickou přípravu, nemohu se stavět do role odborníka, respektive učitele hudební výchovy. Nepokoušel jsem se tedy vytvořit program, kterému by mohli snadno porozumět mladší uživatelé (děti). Naopak jsem se zaměřil na uživatele, kteří jsou schopni intuitivně pochopit funkce, chod i cíle programu a jednotlivých lekcí, aniž by k tomu potřebovali rozsáhlé návody.
- Na kytaru jsem se začal učit v 18 letech a to bez cizí pomoci, takže jako takzvaný samouk. Z vlastních zkušeností tedy vím, co se začínající hráči (v přibližně stejném věku) na kytaru chtějí naučit, čemu by se měli vyvarovat, či na co se naopak zaměřit.
- Začínající hráč, kterému je 15 let a více, nechce ztrácet čas s výukou teorie, nebo not. Chce se naučit hlavně hrát základní akordy a postupy a za tímto účelem je také program vytvořen.
- Oproti dětem, jsou dospělí lidé výrazně zručnější (berme lidi starší 15 let, jako dospělé). Mají lépe rozvinutou motoriku celého těla. Dokáží tedy lépe ovládat i své ruce a prsty a naučit se tak

hrát základní akordy během několika měsíců, nebo dokonce týdnů. A to i v případě, že kytaru nikdy předtím v ruce nedrželi.

Proto je i metoda výuky snadněji realizovatelná a dá se při tvorbě výukového programu předpokládat přibližný vývoj pokroků uživatele.

3.2 Struktura a zaměření výuky

Protože je pevně vymezena věková hranice uživatelů programu, je podle ní také náležitě koncipována výuka. Abych se ujistil, že je mnou zvolený postup vytváření lekcí a tudíž i metoda výuky správná, konzultoval jsem jednotlivé lekce s učitelkou hudební výchovy a profesionálním kytaristou. Ten osobně poskytuje soukromé lekce a přímo se podílel na vymyšlení, nebo upravení několika lekcí, které byly následovně zrealizovány v programu. S oběma odborníky došlo ke vzájemné shodě nad problematikou, která by měla být v programu stěžejní. Jedná se o problém synchronizace pohybů pravé ruky (tj. trsátka) a prstů na ruce levé. Výukové lekce jsou tudíž zaměřeny zejména na synchronizaci.

Obecně je výuka primárně zaměřena na praktická cvičení, jejichž prvky se při hraní na kytaru opravdu využijí. Mnoho samouků, amatérských, ale i profesionálních kytaristů, ve své praxi vůbec nevyužívá noty. Problematika not tedy součástí výuky není. Stejně tak program neobsahuje výuku hraní (rozeznívání) strun prsty, ale zaměřuje se pouze na hraní trsátkem. Výuka tím pádem není orientována čistě ani na akustickou, ani na elektrickou kytaru. Učit se tak podle něj mohou začátečníci bez rozdílu, jaký druh kytary mají k dispozici.

V programu je výuka pomyslně rozdělena do tří částí. Pomyslné rozdělení v praxi znamená, že si jej uživatel programu příliš

neuvědomuje, tzn. postupuje dle svých potřeb a není nijak ovlivňován. Program je nicméně navržen tak, že jsou kroky uživatelů předpokládány. Naprostá většina uživatelů postupuje tedy dle předpokladů, od první části po třetí. První část se zabývá teoretickým úvodem, druhá jednoduchou výukou základních akordů. Třetí část je zaměřena zejména na synchronizaci pohybů pravé a levé ruky, obsahuje i výuku některých stupnic, nebo cvičení na širší rozchod prstů.

3.2.1 Teoretický úvod

Pro úplné začátečníky, kteří se s hraním na kytaru teprve seznamují, je teoretický úvod velice důležitý. Úvod pomůže začátečníkům pochopit, jak program pracuje a naučí je číst, respektive hrát zobrazované akordy, nebo přehrávané lekce.

Pro znázorňování akordů i lekcí jsem zvolil postup, z části založený na klasické metodě, jež se využívá při psaní tabulatur.

Tabulatury neznázorňují konkrétní tóny a jejich délku, jako je tomu u not, ale zobrazují pouze číslo pražce, nad kterými se daná struna má zmáčknout. Struny jsou však vyobrazeny tak, jako by byl celý hmatník obrácený o 180 stupňů, oproti očekávání většiny začátečníků. Vše je znázorněno na obrázku 5.



Obrázek 5: Srovnání: řazení strun v tabulaturách vs. špatná představa začátečníka.

Úvodní teorii by však měl absolvovat i uživatel, který se již s tabulaturami setkal. Vyobrazování tónů, je totiž v programu sice na tabulaturách založeno, ale ve výsledku řešeno odlišně. Hráč dostává vizuální informaci, které tóny má hrát, vyobrazením konkrétní pozice na virtuálním hmatníku. Číslo, které se na dané pozici objeví, však neinformuje o prázce, nad kterým se má struna zmáčknout, ale o prstu, kterým se tento tón má hrát. Jedinou výjimkou je vyobrazení akordu, na základě uživatelem vepsané sekvence čísel. Zde je postup při zadávání takřka totožný, jako při psaní tabulatur.



Obrázek 6: Vizualizace poloh jednotlivých prstů (program Kytarace).

Program záměrně neobsahuje vysvětlení některých teoretických problematik týkajících se hraní na kytaru. Existují totiž takové, které se velmi těžko popisují. Bez názorné praktické ukázky, předvedené zkušeným kytaristou na skutečné kytare, je de facto nemožné je vysvětlit,

nebo v pozici studenta naopak pochopit. Mezi tyto problematiky patří například správné tlumení strun, které by mohly při hraní přeznívat, ačkoliv nejsou vůbec hrány. Prakticky není možné tuto, nebo podobné problematiky, realizovat pouze pomocí grafických prvků s textovým, nebo zvukovým doprovodem.

3.2.2 Výuka akordů

Ačkoliv by výuka akordů mohla být součástí výukových lekcí, rozhodl jsem se tyto dvě problematiky v programu oddělit a pojmout odlišným způsobem. Naučení se základních akordů, následně i těch obtížnějších, považuji, z hlediska uživatele, za poměrně snadné. Uživateli stačí vědět, které prsty má použít a na jakých konkrétních pražcích je rozmístit. Proto není výuka akordů realizována formou lekcí, ale pouze grafickým zobrazením požadovaného akordu na virtuálním hmatníku.

Výuka akordů je tedy řešena tak, že se na konkrétních pozicích virtuálního hmatníku zobrazují čísla prstů, které akord mají hrát. Současně s vyobrazením daného akordu, se přehraje i odpovídající audio interpretace tohoto akordu (v klasickém kytarovém lazení, tj. E, A, D, G, H, e).

Součástí tohoto oddílu programu je také převod textu na grafické znázornění polohy prstů na virtuálním hmatníku. I tento převod je vlastně způsobem výuky, avšak netýká se přímo akordů, ale spíše tabulatur. Pochopením jednoduchého principu převodu, získá uživatel jasnou představu o způsobu zápisu a čtení tabulatur. Funkce je jednoduchá. Uživatel napíše sekvenci šesti znaků, kde každý znak reprezentuje jeden tón, jež se zobrazí na virtuálním hmatníku. Pořadí znaků je dané pořadím strun, tj. například druhý znak, pokud je to číslo, odkazuje na pozici

zobrazení tónu na struně lazené do A. Pokud znak není číslo, znamená to pro program, že na dané struně nemá zobrazit nic (na dané struně se žádný tón nehraje).

3.2.3 Výukové lekce

Výukové lekce jsou zaměřené převážně na synchronizaci pohybu trsátka a prstů hrajících jednotlivé tóny. Tvoří hlavní část celé výukové koncepce a ve větší, či menší míře se objevuje ve všech výukových lekcích.

Lekce jsou zaměřeny na následující konkrétní problémy: výuka užitečných postupů a vybraných stupnic, cvičení na širší rozchod prstů.

Řazení lekcí je uspořádáno dle obtížnosti od nejjednodušší (1. lekce), po nejsložitější (10. lekce) a je doporučeno výuku absolvovat vzestupně.

3.2.3.1 Synchronizace

Synchronizace je zdánlivě nenápadný problém. Vyskytuje se však téměř ve všech problematikách týkajících se hraní na kytaru. Pokud se jí hráč v začátcích dostatečně nevěnuje, s největší pravděpodobností bude mít v budoucnu problém, například při hraní rychle za sebou jdoucích tónů. Naučit se synchronizaci až po několika letech hraní, obvykle vyžaduje o mnoho více úsilí, než když se jí hráč učí v začátcích.

Jednou z nejčastějších a současně nejvíce slyšitelných chyb je nechtěně zahraná struna, o kterou hráč zavádí při hraní tónu nacházejícího se o jednu, či více strun. Neúmyslně zahraná struna pak může vydat rušivý

zvuk, který upozorní i méně pozorné posluchače na kytaristovu nepřesnou hru.

3.2.3.2 Výuka užitečných postupů a vybraných stupnic

Užitečný postup může být stupnice, nebo zdánlivě nesmyslná sekvence tónů, která má však určité zaměření. V každé lekci je obsažen jeden postup, pokaždé s jiným výukovým cílem. Některé postupy se dají, při zvolení vhodné tóniny, využít při improvizaci, sólech, nebo jako rozcvičení před hraním.

Do sekce užitečných postupů spadá i výuka, zaměřená na oblíbený, avšak velmi náročný (zejména co se týče čisté hry) hrací postup zvaný arpeggio. Tato technika je obsažena až v poslední, tj. desáté lekci. Je tomu tak z důvodu již zmíněné vysoké náročnosti.

3.2.3.3 Cvičení na širší rozchod prstů

Lekce zabývající se cvičením na širší rozchod prstů, je vhodná i pro zkušenější hráče, kteří se ale nenaučili používat při hraní malíček. Pomocí tohoto cvičení takoví hráči získají v malíčku větší cit, prst bude obratnější, na vzdálenější pražce se dokáže pohotověji natáhnout a strunu lépe zmáčknout. U začátečníků by tedy mělo být samozřejmostí absolvování i této lekce.

4 Metodika

V této kapitole jsou popsány způsoby, kterými se prováděl průzkum zjišťující zaměření výukových programů, požadavky uživatelů i hodnocení programů uživateli. Kapitola je rozdělena do tří hlavních částí, z nichž se každá zabývá jiným vývojovým stupněm, respektive časovým úsekem vývoje, nebo testování programu Kytarace.

4.1 Před vytvořením programu

Před vytvořením programu, bylo nutné zjistit, jakými konkrétními problémy se výuka v daných programech obvykle zabývá a jaké požadavky mají samotní začínající hráči. Pro srovnání a znázornění, jaká mají konkrétní programy zaměření, byla vytvořena srovnávací tabulka (viz. Tabulka 1). Jednotlivé požadavky uživatelů byly zjišťovány pomocí dotazníku č. 1.

Do tabulky bylo ke srovnání vybráno pět subjektů včetně programu Kytarace. Tato tabulka zobrazuje jednotlivé prvky výuky a znázorňuje, zda je srovnávané programy obsahují. Aby bylo srovnání komplexnější, byl do tabulky zařazen i výukový soubor video lekcí s názvem vGuitarLessons a program Guitar Pro (verze 5.2), který není tvořen se zaměřením na výuku hry na kytaru, ale spíše na zápis, nebo tvorbu hudby. Guitar Pro se totiž stal velice populárním a jeho uživatelé ho začali využívat i při zdokonalování svých schopností. Tvůrci programu poté některé výukové lekce sami vytvořili a nyní jsou tyto soubory

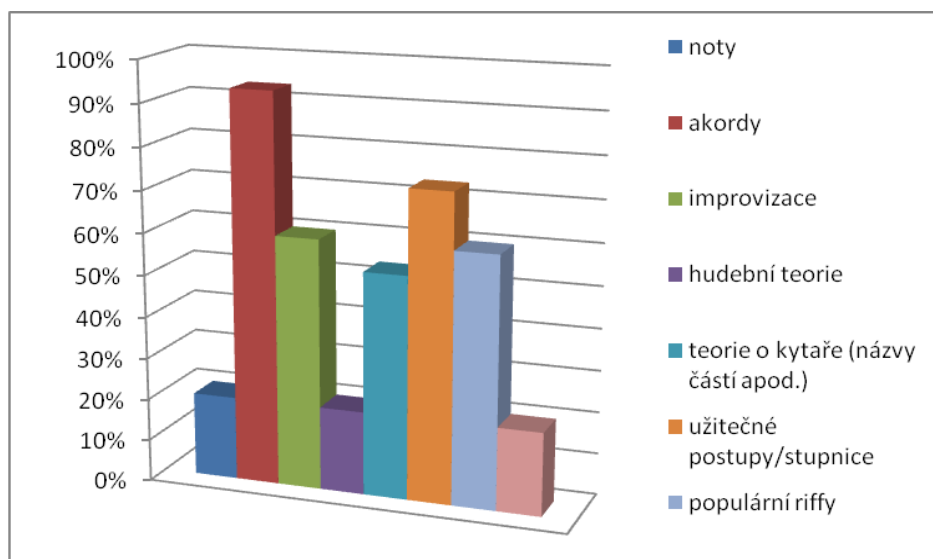
součástí ukázkových příkladů. Tvoří i samostatnou sekci na oficiálních stránkách, kde jsou umístěny i lekce, vytvořeny uživateli programu.

	vGuitarLessons	GBET	Edittab	Kytarace	Guitar Pro
Noty	×	✓	×	×	✓
Akordy	✓	×	✓	✓	✓
Improvizace	✓	×	×	×	×
Hudební teorie	×	✓	×	×	×
Teorie o kytáře	✓	×	×	✓	×
Užitečné postupy/stupnice	✓	✓	×	✓	✓
Populární riffy	✓	×	×	×	✓
Zpěv při hraní	✓	×	×	×	×

Tabulka 1: Srovnání výukových programů

- *Edittab je zkratka názvu Editor Tabulatur*
- *GBET je zkratka názvu Guitar & Bass Ear Trainer*

Dotazník obdrželi pouze lidé, kteří projevíli vážný zájem, pokusit se naučit hrát na kytaru s dopomocí programu Kytarace a spadali do cílové skupiny uživatelů (tj. věk +15, samouci atd.). Vyplněných dotazníků tedy bylo ve výsledku pouze patnáct. O to více však byly výsledky (viz. Graf 1) konkrétnější. K výsledkům bylo částečně přihlíženo i při tvorbě výukové koncepce programu Kytarace. Některé požadavky, plynoucí z průzkumu, byly záměrně vynechány. Buď z důvodu nemožné realizace (improvizace je vlastnost, nelze ji tedy z principu naučit, populární riffy jsou chráněné autorskými právy atd.), nebo nezapadnutí do koncepce výuky.



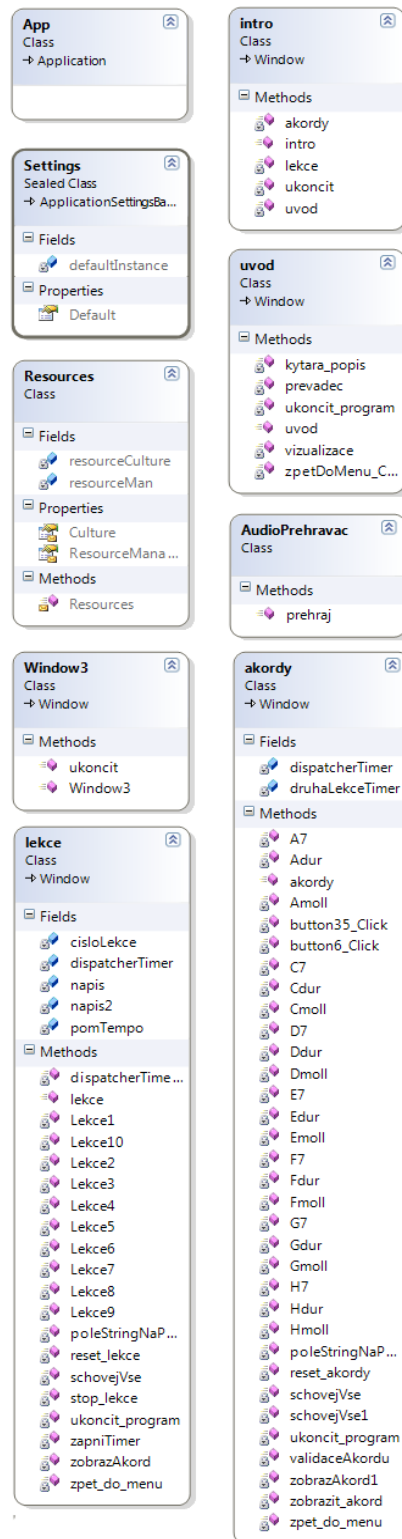
Graf 1: Výsledky dotazníku č. 1.

4.2 Tvorba programu

Při tvorbě programu byla navrhnutá výuka postupně aplikována do programu. Všechny výukové postupy byly realizovány pomocí grafických prvků. Následně se vše opět konzultovalo s odborníky. Nakonec jsem se rozhodl, v rámci zachování jednoduchosti programu, že se všechny akordy a lekce budou zobrazovat na virtuálním hmatníku o pěti pražcích. Zkrácený hmatník poskytuje přehlednější zobrazení jednotlivých poloh prstů. Všechny základní akordy se hrají na prvních pěti pražcích, takže zde délka virtuálního hmatníku postačuje. Lekce jsou navrženy pro pohyb všech čtyř používaných prstů, tudíž i zde pět pražců vyhovuje. Všechny lekce jsou navíc koncipovány tak, aby si jejich postupy hráč mohl libovolně posouvat po hmatníku. Pokud se tedy hráč

danou lekci naučí podle zobrazení v programu, může si její postup posunout například na dvanáctý pražec a procvičovat ho zde.

Struktura programu je vidět na Obrázku 7.



Obrázek 7: Class diagram (program Kytarace).

4.3 Po vytvoření programu

Po dokončení byl program nejdříve předán k otestování několika lidem z řad programátorů i kytaristů. Ti testovali celkovou funkčnost, tzn., zda je program stabilní a pracuje tak, jak má. Postupně se řešily připomínky od programátorů. Ty se obvykle týkaly drobných problémů ohledně chodu programu, jako třeba špatně se měnící poloha tlačítek, při změně velikosti okna. Veškeré chyby týkající se funkčnosti programu byly následně opraveny. Testeři také úmyslně zkoušeli program přetížit (například vepsáním velmi dlouhého textu do převaděče textu na akordy). K žádným nečekaným kolapsům nedošlo, všechny možné výjimky byly předem ošetřeny klauzulí *try-catch*.

Na požadavky kytaristů, kteří program testovali, se následně dodatečně dodělávaly a upravovaly některé grafické prvky, zejména vzhled virtuálního hmatníku.

4.3.1 Hodnocení

Stejní respondenti, kteří odpovídali na dotazník č. 1, se zúčastnili i druhého průzkumu (dotazník č. 2), ve kterém hodnotili jednotlivé programy. Hodnocené programy byly tytéž, jako v Tabulce 1. Respondenti měli na vyzkoušení programů dva týdny a předem nedostali informaci, že Kytarace je program, který budou posléze testovat. Vzhledem k faktu, že byli všichni účastníci průzkumu začátečníci, žádný z nich neměl předchozí zkušenosti s jakýmkoliv z hodnocených

programů. Dotazující hodnotili konkrétní vlastnosti jednotlivých programů.

	vGuitarLessons	GBET	Edittab	Guitar Pro	Kytarace
Přehlednost	2	5	3	2	2
Využití v praxi	1	4	1	1	1
Komplexnost	2	4	4	2	2
Další využití	5	5	3	1	5
Design	4	3	3	2	2
Celkový dojem	3	4	3	2	2

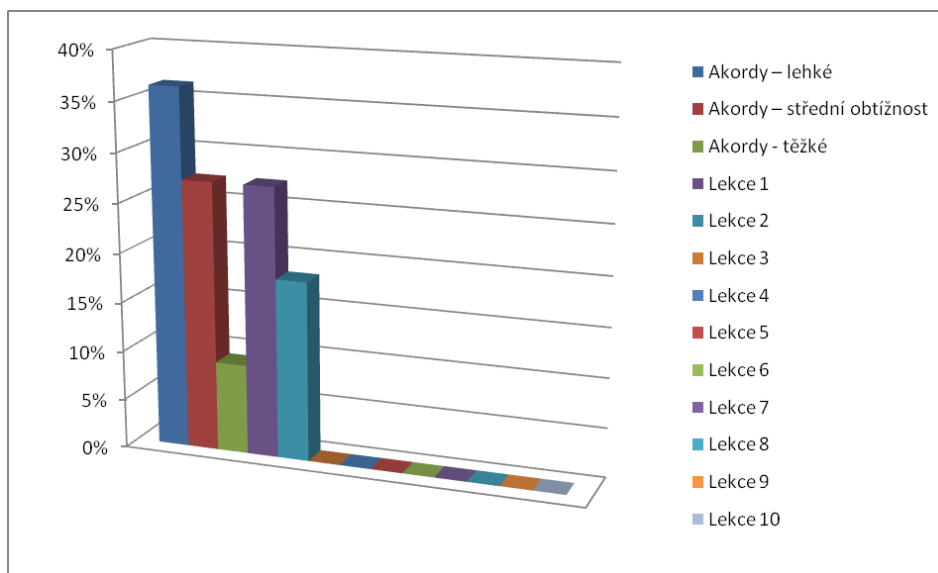
Tabulka 2: Hodnocení uživatelů

- *Hodnocení jako ve škole (1 = nejlepší, 5 = nejhorší)*
- *Výsledky zaokrouhleny na celé číslo*
- *Edittab je zkratka názvu Editor Tabulatur*
- *GBET je zkratka názvu Guitar & Bass Ear Trainer*

4.3.2 Testování

Po vytvoření hodnotící tabulky, obdrželi uživatelé dotazník č. 3, kam zapsali své konkrétní dovednosti týkající se hraní na kytaru. Jako měřítko byl využit program Kytarace. Pomocí Kytarace dotazující vyzkoušeli, které akordy a lekce zvládnou zahrát. Hráč do dotazníku zapsal, zda umí zahrát alespoň nějaký akord z dané kategorie obtížnosti. U všech lekcí bylo při zkoušení stanoveno tempo na 70 bpm (beats per minute, tzn. počet úderů/tónů za minutu). Pokud hráč v tomto tempu lekci bez chyby zahrál, zaznamenal tuto skutečnost do dotazníku. Na vyzkoušení a zapsání svých hráčských schopností, měli respondenti pouze jeden den.

Časový limit byl určen z důvodu zaznamenání opravdu aktuálních stavů hráčských schopností respondentů. Výsledný Graf 2 obsahuje průměry ze všech vyplněných dotazníků, respektive kolik procent ze všech účastníků dokázalo zahrát konkrétní lekci, či nějaký akord z dané obtížnosti.



Graf 2: Schopnosti uživatelů před absolvováním výuky.

Ukončením průzkumu zabývajícím se dosavadními zkušenostmi uživatelů programu s hraním na kytaru, začala fáze samotné výuky. Doba výukového kurzu, tj. využívání programu Kytarace pro výuku hraní na kytaru, byla stanovena na dva měsíce. Přičemž v těchto dvou měsících měli účastníci kurzu cvičit minimálně dvě hodiny denně.

Doporučené časové rozvržení (plán) kurzu

1. - 2. den – úvodní teorie

3. - 10. den – akordy

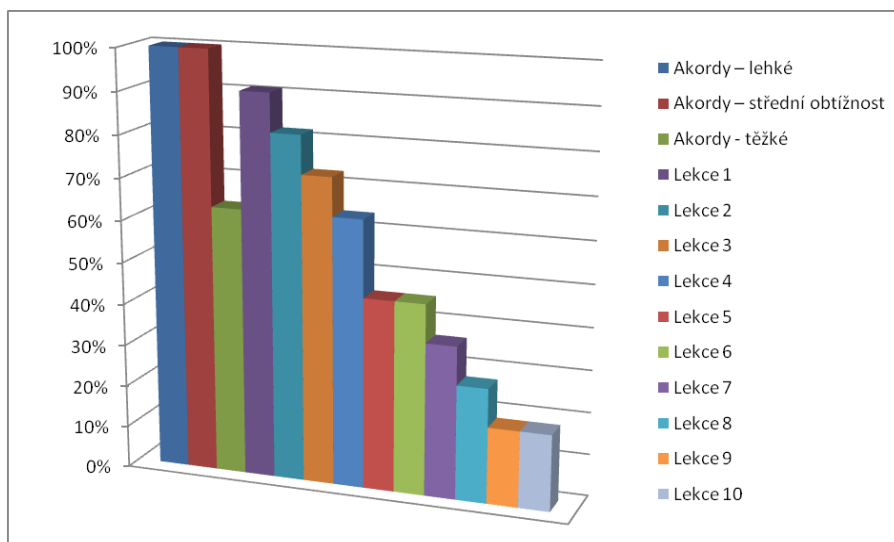
11. - 60. den – lekce

Jelikož jsou stěžejní částí celé výuky hlavně lekce, bylo na naučení úvodní teorie a akordů vyhrazeno pouze prvních deset dnů. Z těchto deseti dnů byly první dva určené pro úvodní teorii a pro akordy zbylých osm. V praxi to znamená, že se každý den (počínaje třetím) musí hráč naučit průměrně tři nové akordy. Lekcím bylo vyhrazeno zbylých 50 dnů. Na každou lekci tak v průměru připadá 5 dnů, neboli 10 hodin cvičení. Doporučený časový plán hráči nemuseli striktně dodržovat. Každý mohl začít na úrovni, která odpovídala jeho výsledkům z testování dosavadních zkušeností a postupovat dle svých aktuálních potřeb. První, tj. jednodušší lekce mohou zabrat času méně a ušetřený čas může zůstat na poslední, obtížnější lekce. Pokud se však hráč bude držet doporučeného časového plánu a prvním lekcím věnuje veškerý jim určený čas, získá tak pevné základy, které může zhodnotit při hraní obtížnějších lekcí. Na ty tím pádem nebude potřebovat čas navíc a vystačí si vyhrazeným časem.

4.3.3 Výsledky testování

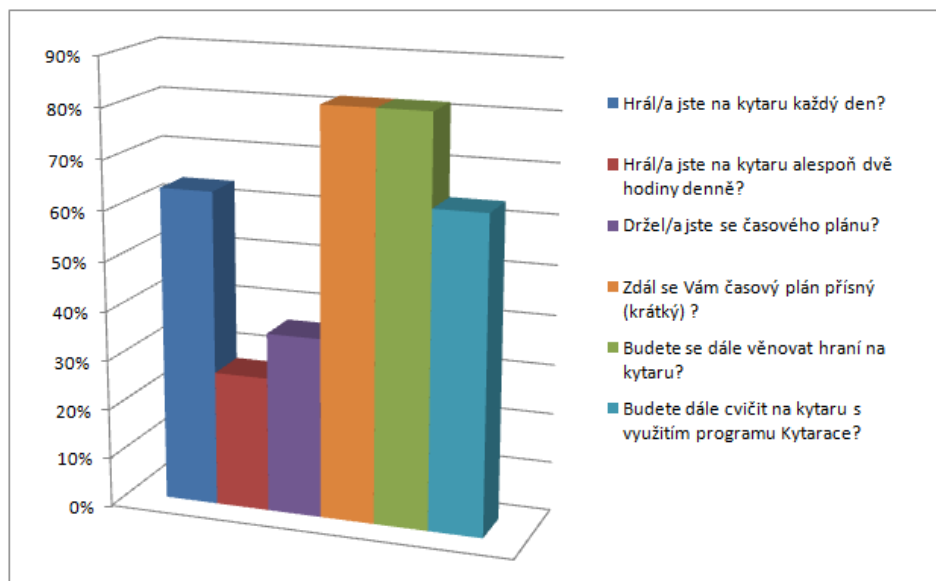
Konečný počet uživatelů, kteří se zúčastnili kurzu, v jeho plném časovém rozsahu, byl 11. Po uplynutí vymezeného času každý vyplnil opět dotazník č. 3, tj. stejný, jako před zahájením kurzu. Výstupem zpracovaných výsledků ze všech dotazníků je Graf 3, ze které je možné

(srovnáním s Grafem 2) zjistit efektivitu celého kurzu. Z Grafu 3 je patrné, že většina účastníků kurzu nezvládla dodržet časový plán.



Graf 3: Schopnosti uživatelů po absolvování výuky.

Aby bylo možné zjistit, zda je za tímto výsledkem fakt, že většina účastníků nevěnovala cvičení na kytaru dostatečný čas. Nebo byl časový plán zvolen příliš přísně a dodržet se jej podařilo pouze talentovaným hráčům. Byl dodatečně vymyšlen a účastníkům kurzu rozdán závěrečný dotazník (č. 4). Zde bylo zjišťováno, zda dodržovali minimální denní dobu pro cvičení a řídili se doporučeným časovým plánem, nebo jestli se dále chtějí věnovat hraní na kytaru. Na Grafu 4, jež vychází z odpovědí z dotazníku č. 4, je například vidět, že většina uživatelů necvičila denně dostatečně dlouho. Z toho samozřejmě vychází i problém nedodržení časového plánu.



Graf 4: Výsledky dotazníku č. 4.

5 Závěr

Cílem práce bylo seznámit se s frameworkem Windows Presentation Foundation a programy zabývajícími se výukou hry na kytaru. Seznámením s WPF se zabývá kapitola 1.2. V kapitole 2 je obecné dělení výukových programů a popis různých realizací výuky. Dále navrhnout a vymyslet vlastní výuku a vytvořit pomocí technologie WPF program, který bude tuto výuku poskytovat. Program pak srovnat s konkurenčními programy a otestovat na malém počtu žáků. Popis výuky, obsažené v programu, je v kapitole 3, testováním programu se věnuje kapitola 4.3.2.

Dle testujících uživatelů si vedl program, ve srovnání s konkurenčními, velmi dobře. Výuka i celkové zpracování je v některých srovnávaných programech sice propracovanější, to ale odpovídá nesrovnatelné době tvorby i počtu vývojářů, kteří se na těchto programech podíleli. Obecně však výuku ve svém programu považují za ucelenou a komplexní. V plném rozsahu ji většina účastníků testování nezvládla stihnout. To z části způsobil poměrně krátký čas, vyhrazený na absolvování výuky. Ale také, jak vyplývá z Grafu 4, nedostatečné věnování se cvičení na kytaru, ze strany účastníků.

Program jsem se snažil udělat přehledný a jednoduchý na ovládání. Při pohledu na jeho vzhled i funkce se tak může zdát, že možnosti technologie WPF nebyly příliš využity. Z vlastních zkušeností však mohu konstatovat, že tvorba byla, ve srovnání se starší technologií WinForms, v mnoha směrech snadnější, rychlejší i příjemnější. Také požadavky programu na výkon procesoru jsou určitě nižší.

Pokud bych měl doporučit výukový program pro hraní na kytaru, zvolil bych na prvním místě jednoznačně Guitar Pro (v současnosti již verze s označením 6). Tento program sice není implicitně koncipován jako výukový, po celém světě je však natolik rozšířený, že je možné dohledat nespočet užitečných souborů (výukové lekce, známé riffy i celé skladby). Jeho další funkce, jako jsou například zápis, tvorba a následný export hudby, jsou pro uživatele také velmi užitečné.

S konečnou podobou programu Kytarace i s výsledky, plynoucími z jeho testování jsem spokojený. Program splňuje všechny požadavky, které byly uvedeny v zadání práce. Věřím, že program bude i nadále využíván začínajícími i pokročilejšími kytaristy pro získání, či procvičení základů ve hraní na kytaru. I přesto bych však doporučil začátečníkům i osobní konzultace se zkušeným kytaristou. Lidský faktor je přeci jen nenahraditelný a nelze předat veškeré znalosti pouze formou programu.

Literatura

- [1] *Wikipedie : Otevřená encyklopedie* [online]. 2009-09-05 [cit. 2009-12-18]. DirectX.
Dostupné z WWW: <<http://cs.wikipedia.org/wiki/DirectX>>.
- [2] ŠTURALA, Aleš. *Vyvojar.cz* [online]. 2007-01-17 [cit. 2009-12-18]. 0. WPF - úvod.
Dostupné z WWW: <<http://www.vyvojar.cz/Articles/445-0-wpf-uvod.aspx>>.
- [3] *Wikipedia : The Free Encyclopedia* [online]. 2009-12-17 [cit. 2009-12-18]. Windows Presentation Foundation. Dostupné z WWW:
<http://en.wikipedia.org/wiki/Windows_Presentation_Foundation>.
- [4] MACDONALD, Matthew. *Pro WPF in C# 2008 : Windows Presentation Foundation with .NET 3.5*. Second Edition. [s.l.] : [s.n.], 2008. 1040 s. ISBN 978-1-4302-0576-0.
- [5] *B.NET(bdotnet.in)* [online]. 2008-06-08 [cit. 2009-12-18]. The Architecture of WPF (Windows Presentation Foundation). Dostupné z WWW:
<<http://bdotnet.in/blogs/lohithbn/archive/2008/05/08/the-architecture-of-wpf-windows-presentation-foundation.aspx>>.
- [6] ŠTURALA, Aleš. *Vyvojar.cz* [online]. 2007-02-13 [cit. 2009-12-18]. 4. WPF - Animace.
Dostupné z WWW: <<http://www.vyvojar.cz/Articles/456-4-wpf-animace.aspx>>.
- [7] ŠTURALA, Aleš. *Vyvojar.cz* [online]. 2007-01-30 [cit. 2009-12-18]. 2. WPF - Styly.
Dostupné z WWW: <<http://www.vyvojar.cz/Articles/449-2-wpf-styly.aspx>>.
- [8] ŠTURALA, Aleš. *Vyvojar.cz* [online]. 2007-02-05 [cit. 2009-12-18]. 3. WPF - Šablony & Trigry. Dostupné z WWW: <<http://www.vyvojar.cz/Articles/451-3-wpf-sablony-trigry.aspx>>.
- [9] ŠTURALA, Aleš. *Vyvojar.cz* [online]. 2007-02-19 [cit. 2009-12-18]. 5. WPF – Data Binding.
Dostupné z WWW: <<http://www.vyvojar.cz/Articles/458-5-wpf-data-binding.aspx>>.

A Instalace a spuštění programu

Program není třeba nějak instalovat. Pro správnou funkci je však nutné spouštěcí soubor (tj. Kytarace.exe) vždy ponechat ve stejné složce, jako složku „zvuky“. Program lze spustit pouze na operačních systémech Windows, konkrétně Windows XP (s nainstalovaným aktualizacím balíčkem Service Pack 2, či novějším), Windows Vista a Windows 7.

Program se spouští spuštěním souboru Kytarace.exe.

B Obsah přiloženého CD

- Kytarace – složka obsahující veškeré soubory programu Kytarace.
 - Kytarace.exe – spouštěcí soubor programu Kytarace.
 - readme.txt – soubor obsahující instrukce ke spuštění.
 - zvuky – složka obsahující veškeré zvuky programu.

C Dotazníky

Dotazník č. 1

Co byste se chtěl/a naučit pomocí výukového programu pro hraní na kytaru (nehodící se škrtněte)?

1. Noty
2. Akordy
3. Improvizaci
4. Hudební teorii
5. Teorii o kytáře (názvy částí apod.)
6. Užitečné postupy/stupnice
7. Populární riffy
8. Zpěv při hraní

Dotazník č. 2

Ohodnoťte vlastnosti jednotlivých programů (hodnocení jako ve škole, tj. 1 = nejlepší, 5 = nejhorší).

	vGuitarLessons	GBET	Edittab	Guitar Pro	Kytarace
Přehlednost					
Využití v praxi					
Komplexnost					
Další využití					
Design					
Celkový dojem					

Dotazník č. 3

Zaznamenejte, zda dokážete zahrát nějaký akord z dané kategorie, nebo určitou lekci (nehodící se škrtněte).

1. Akordy – lehké
2. Akordy – střední obtížnost
3. Akordy - těžké
4. Lekce 1
5. Lekce 2
6. Lekce 3
7. Lekce 4
8. Lekce 5
9. Lekce 6
10. Lekce 7
11. Lekce 8
12. Lekce 9
13. Lekce 10

Dotazník č. 4

Zakroužkujte odpověď (ANO/NE) u následujících otázek.

1. Zdál se Vám časový plán přísný (krátký)?
(ANO/NE)
2. Hrál/a jste na kytaru alespoň dvě hodiny denně?
(ANO/NE)
3. Hrál/a jste na kytaru alespoň dvě hodiny denně?
(ANO/NE)
4. Hrál/a jste na kytaru alespoň dvě hodiny denně?
(ANO/NE)
5. Hrál/a jste na kytaru každý den?
(ANO/NE)
6. Chtěl/a jste se držet časového plánu?
(ANO/NE)
7. Držel/a jste se časového plánu?
(ANO/NE)
8. Budete se dále věnovat hraní na kytaru?
(ANO/NE)