

UNIVERZITA PALACKÉHO V OLOMOUCI

PEDAGOGICKÁ FAKULTA

Katedra technické a informační výchovy

Diplomová práce

Bc. Petr Felner

**Kuchařka nápadů pro tvorbu výukových aplikací ve Visual
Basic .NET**

Olomouc 2016

Vedoucí práce: PhDr. Jan Lavrinčík, DiS., Ph.D.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a použil jen uvedené bibliografické a elektronické zdroje.

Olomouc 15. dubna 2016

podpis

Rád bych na tomto místě poděkoval PhDr. Janu Lavrinčíkovi, Dis., Ph.D. za odborné vedení, ochotu a trpělivost při vedení mé diplomové práce. Dále mé poděkování patří rodině, která mě podporovala po celou dobu mého studia.

Obsah

Úvod.....	6
1 Software ve škole	7
1.1 Historie využívání počítačů a programů ve vzdělávání	7
1.2 Současný stav využívání počítačů ve vzdělávání.....	8
1.3 Terminologické vymezení pojmů výukový program a výukový software	11
1.4 Členění výukových programů	13
1.5 Dílčí závěr	22
2 Využití výukového softwaru při výuce	23
2.1 Důvody proč využívat výukový software	23
2.2 Funkce zajišťující výukový software	24
2.3 Využití výukového softwaru	25
2.4 Dílčí závěr	26
3 Současné problémy a nedostatky výukového softwaru	27
3.1 Cena výukového softwaru	27
3.2 Omezení licencí softwaru	28
3.3 Nemožnost přizpůsobení obsahu.....	30
3.4 Omezení platformou.....	30
3.5 Požadavky na hardware.....	31
3.6 Využití techniky, kterou škola disponuje	31
3.7 Možnosti zapojení celé třídy	32
3.8 Možnosti samostudia a automatické odesílání úkolů	32
3.9 Možnosti vytvoření vlastního testu	33
3.10 Přizpůsobení motivační složky.....	33
3.11 Dílčí závěr	34
4 Kuchařka nápadů.....	35
4.1 Metody práce.....	35

4.2	Vědomostní kvíz	38
4.3	Skládání obrázků	44
4.4	Pexeso.....	50
4.5	Šibenice	55
4.6	Slepá mapa	60
4.7	Dílčí závěr	66
5	Výsledek nasazení do školy.....	67
5.1	Kvíz	67
5.2	Skládání obrázků	68
5.3	Pexeso.....	69
5.4	Šibenice	69
5.5	Slepá mapa	70
5.6	Shrnutí	71
	Závěr	73
	Seznam zkratk.....	75
	Seznam tabulek.....	76
	Bibliografické zdroje	77
	Seznam příloh	80

ÚVOD

Počítače se dnes stávají běžnou součástí každodenního života všech lidí. Bohužel, ve většině škol, které jsem měl tu čest navštívit, nebyl plně využíván potenciál, který s sebou přináší. Počítače při výuce slouží většinou ke dvěma účelům. Samozřejmě jsou využívány k výuce práce s počítačem v hodinách informační výchovy. Kromě toho slouží počítač často jako „pohon“ dataprojektoru, či v dnešní době již velice populární interaktivní tabule. V pozadí prozatím zůstává využívání výukového softwaru, který přináší nové možnosti do procesu učení a kromě práce ve škole je možné jej použít i pro domácí přípravu žáků.

Důvodem, proč není výukový software více využíván, může být to, že prakticky neexistuje dokonalý výukový program, který by vyhovoval všem učitelům a ne každému učiteli se podaří najít právě ten, který by byl pro jeho nároky ten pravý.

Právě z tohoto důvodu se v druhé části své práce budu věnovat možnostem tvorby vlastních výukových programů. V úvodní teoretické části se budu věnovat pojmu výukový software, jeho vymezení, možnostem jeho dělení a využívání ve výuce. Následně se pokusím stanovit problémy současného výukového softwaru a tím vlastně i udat důvody, proč je vhodné tvořit si svůj vlastní software.

Hlavním cílem této diplomové práce je sestavit ideje na začlenění jednoduchých počítačových her do výukových programů. Prvním dílčím cílem je provést analýzu současného stavu užívání počítačů a výukového software ve vzdělávání, včetně nastínění důvodů, proč je pro učitele vhodné vytvářet si svůj vlastní výukový software. Druhým dílčím cílem práce je realizace našich nápadů pomocí vyššího programovacího jazyku Visual Basic .NET a navrhnout, jak by bylo možné tyto programy modifikovat pro využití i v jiných předmětech. Jako třetí dílčí cíl si kladu otestovat vyrobené programy, během své souvislé pedagogické praxe.

1 SOFTWARE VE ŠKOLE

1.1 Historie využívání počítačů a programů ve vzdělávání

Historie využívání softwaru v edukaci sahá až k samotnému počátku využívání počítačů. Jedny z prvních softwarů, které v té době byly ještě těsně vázané na svůj hardware, využívané k výuce byly letecké simulátory. Simulátor vytvořený již v roce 1943 nesl označení *the type19 synthetic radar trainer*. Jednalo se o analogové počítače, které simulovaly, palubní přístroje (Yu, 2012).

Další vývoj výukového softwaru nastal až v 70. letech. Software byl dále vázán na konkrétní hardware. V této době to byly jednoúčelové sálové počítače. V roce 1963 uzavřela americká společnost IBM (International Business Machines Corporation) společenství s institutem matematických studií v sociálních vědách Standfordské univerzity a společně se snažili vyvinout první všeobecný CAI (computer assisted instruction), v českém překladu koncept počítačem podporované výuky, pro základní školy. Později, přesně řečeno v roce 1967, vznikla společnost Computer Curriculum Corporation. Tato společnost byla založena za účelem distribuce softwaru pro školy, který byl vyvíjen v partnerství ze společností IBM. Bohužel, jejich rané terminály nabízené školám byly velmi drahé a tím pádem pro drtivou většinu škol nedostupné (Farenga, 2015).

V roce 1967 byl vytvořen programovací jazyk LOGO (Yu, 2015). Tento jazyk můžeme označit jako výukový, hlavně díky projektu, který se kolem jazyku vytvořil. Skupina vědců z Massachuttského institutu technologie se snažili o větší začlenění počítačů do výuky. Protože vybavení škol modernějšími přístroji by bylo velmi nákladné, přistoupili vědci k řešení, při němž se snažili vyvinout metody, které by byly použitelné s dostupnými přístroji. Jedna z vyvinutých metod, využívající právě jazyk LOGO, byla takzvaná „světelná želva“. Při této metodě se žáci seznamují s výpočetním procesem, pomocí ovládání želvy, která za sebou zanechává stopu. Želva se ovládá jednoduchými příkazy jako je například „left“ (vlevo), „forward“ (dopředu) atd., které jsou doplněny o čísla, určující počet kroků vykonaných želvou (Štěpánek, 1981).

Další pokrok v pronikání počítačů do výuky nastal s vlnou prvních osobních počítačů. Jedním z prvních počítačů byl Altair 8800, který byl na trh uveden v roce 1975. Jednalo se o mikropočítač, který byl postaven na procesoru Intel 8080 (Alexon, 1997). Zajímavostí

je i to, že využíval programovací jazyk Altair Basic, což byl jeden z prvních produktů společnosti Microsoft. Altair byl v relativně krátkém časovém horizontu následován dalšími mikropočítači. V roce 1977 představila společnost Comodore svůj mikropočítač PET. O další dva roky později, tedy v roce 1979, ukázalo Atari svoje modely 400 a 800. V tom samém roce se objevil i Apple II. Jak Atari, tak i Apple poskytovali již barevné zobrazení. S přibývajícými modely se měnila i cena a tím i dostupnost jednotlivých modelů. Z původně nedostupných přístrojů, které měly ceny v řádu tisíců dolarů, se staly přístroje relativně dostupné, jejichž cena se pohybovala v rozmezí stovek dolarů (Aiken, 1980). Jak uvádí Yu (2012) rozšiřování dostupnosti počítačů, umožnilo vznik novým společnostem, které byly zaměřeny na vývoj vzdělávacího software. Mezi hlavní společnosti zabývající se vývojem výukového software zařadil The Learning Company, Broderbund a MECC (Minnesota Educational Computing Consortium).

Prozatím poslední velký milník v historii výukového softwaru nastal v devadesátých letech. Zde pokračoval vývoj počítače jak po hardwarové, tak i po softwarové stránce. Díky pokrokům bylo možné na počítačích vytvářet multimediální software, který mohl využívat moderní grafiku i možnosti přehrávání zvuku. Tyto nové možnosti pronikly i do tvorby výukového softwaru. V této době se taky začal měnit i způsob přenosu výukového softwaru. Z disket se software nejdříve přestěhoval na CD, které díky větší kapacitě umožnilo přenášet větší soubory. Ty vznikaly díky užívání modernější grafiky a implementaci zvukových stop. Koncem devadesátých let se mimo CD začal výukový software distribuovat i pomocí internetu (Yu, 2012). Tento způsob distribuce je společně s DVD a CD využíván až do dnešní doby.

1.2 Současný stav využívání počítačů ve vzdělávání

Počítače jsou dnes velice běžným zařízením jak ve škole, tak i mimo ni. Bohužel většinou jsou ve škole nimi vybaveny pouze specializované učebny, určené pro výuku informační výchovy, ale začínají pronikat i do běžných učeben. Důvody, proč by měly počítače vstupovat více do škol, zveřejnila zpráva *OECD Learning to Change: ICT in Schools* (in Zounek a Šed'ová, 2009). Důvody tedy jsou:

1) *ekonomické důvody,*

Tento důvod vychází z potřeby dovednosti práce s počítačem. Bez této dovednosti se člověk v dnešní době jen těžko uplatňuje na trhu práce, proto je důležité tuto dovednost rozvíjet již na škole.

2) *sociální důvody,*

Schopnost pracovat s počítačem se dnes považuje za velice důležitou pro sociální život. Informační gramotnost je v dnešní době přiřazována na podobnou úroveň jako gramotnost matematická, či gramotnost čtenářská.

3) *pedagogické důvody.*

Poslední důvod vychází z možností využití počítačů při vyučování a učení. Přínosné jsou také pro management školy.

Počítač může být ve výuce prakticky využit dvěma různými způsoby, a to „*výuka o počítači*“ a „*výuka s počítači*“ (Burianová, 2003).

1.2.1 Role počítačů ve výuce

Výuka o počítači

Prvním způsobem je **výuka o počítačích**. Při tomto způsobu využití počítače se zaměřujeme hlavně na poznatky o hardwaru a softwaru, které předáváme žákům. S tímto stylem využití počítače se setkáváme především v předmětech, které jsou zaměřeny na informační technologie (Dostál, 2007).

Výuka s počítači

Druhým způsobem je **výuka s počítači**. Při výuce s počítači je počítač využíván jako pomůcka učitele a žáka. Může tedy sloužit například k procvičování práce žáků se softwarovým vybavením počítače, či je učiteli schopen posloužit ke tvorbě a představení multimediálních prezentací doplňujících výuku. Výuka s počítači není na rozdíl výuky o počítačích pevně vázána pouze k výuce ICT (Information and Communication Technologies), ale je možno ji využít ve všech předmětech. Dalším rozdílem je úroveň znalosti práce s počítačem. Výuku s počítači dokáže zvládnout i méně zdatný uživatel. Výukové programy jsou totiž většinou uživatelsky přívětivé (Burianová, 2003).

Výuku s počítači můžeme dále rozdělit podle toho, jakým stylem zasahuje počítač do řízení vzdělávání. Výuku poté rozdělujeme na „*počítačem podporovanou výuku*“ a „*počítačem řízenou výuku*“ (Burianová, 2003).

Počítačem podporovaná výuka

Jedná se o typ výuky, kdy počítač používáme pouze jako doplněk ke klasickým výukovým metodám. Splňuje pouze některé didaktické funkce, jako je například testování vědomostí (Burianová, 2003). Do tohoto typu řízení výuky dále zařazuje dva koncepty a to:

- CAI (Computer-assisted instruction) – tedy počítačem podporovaná výuka,
- CAL (Computer-assisted Learning) – tedy učení podporované počítačem.

Rozdílem mezi CAI a CAL je zaměřenost CAL na proces žákova učení (Zounek a Šedřová, 2009).

Počítačem řízená výuka

Počítač v tomto typu výuky slouží k ukládání a vyhodnocování žákova postupu při učení. Kromě informací o žácích má systém i databázi lekcí, soubor testů, pracovní úlohy atd. Tyto části otevírá systém žákovi v závislosti na jeho postupu. V tomto systému se mění postavení učitele. Ten se zde dostává do role konzultanta (Burianová, 2003).

1.2.2 Didaktické funkce ICT

Didaktickými funkcemi ICT se mimo jiné zabývali Zounek a Šedřová (2009). ICT z technologického hlediska můžeme vymezit jako „*termín, který zahrnuje širokou škálu technologií, k nimž řadí počítače; digitální vysílání; telekomunikační technologie jako mobilní telefony umožňující přístup k e-mailu a další formy počítačem zprostředkované komunikace; a elektronické informační zdroje jako web a CD-ROM*“ (Selwyn, 2006 cit. dle Zounka, 2009 s. 12). Z definice jasně vyplývá, že do termínu ICT jsou zahrnovány i počítače, o kterých jsme se zmiňovali v předchozím textu kapitoly.

Autoři vystavěli svou typologii didaktických funkcí na výzkumu, který uskutečnili mezi učiteli. Díky tomuto výzkumu autoři sestavili pět způsobů, kterými je ICT ve výuce využíváno (Zounek a Šedřová, 2009):

1) ICT jako nosič obsahu,

Při tomto způsobu využití technika zastupuje učebnici. Je nositelem informace a slouží k výkladu nové látky. Provedení výkladu nemusí být na rozdíl od tištěné učebnice pouze textové, ale mohou být využity například i videosekvence. Učitel při tomto typu využití může mít dvě funkce. Buď ustoupí do postranní a bude pouze spouštět obsah, který se žáci sami naučí, nebo bude sloužit jako zpestření zobrazené informace, kterou doplní o svůj výklad, přečte ji, či ji doplní o zajímavosti.

2) *ICT jako extenze,*

Zde je počítač využit jako extenze (rozšíření) možností lidského těla. Tím je myšleno například zobrazování věcí, které by za normálních okolností nebyly pouhým okem viditelné, schémata i modely. Obrazy mají oproti těm v učebnici tu výhodu, že s nimi lze manipulovat. Můžeme je tedy otáčet, přibližovat atd., a tím zajistit větší názornost. Kromě rozšíření zraku, které bývá využíváno nejčastěji, můžeme rozšířit i sluch. Příkladem může být zesílení zvuků třeba při výuce hlasů pěvců.

3) *ICT jako pracovní nástroj,*

Při tomto typu využívání žáci používají počítač ke tvorbě výstupů. Autor dále rozlišuje toto využití podle toho, jestli je cíl v postupu práce, či ve výsledku práce. Pokud je cíl v postupu, označujeme jej jako aplikační postupy, a pokud je cíl ve výsledku práce, označujeme postup jako tvorbu. Aplikace je využívána většinou v hodinách informatiky, kde se žáci učí pracovat s počítačem. Při tvorbě žáci nedostanou návod jakým způsobem postupovat, nýbrž pouze zadání práce, kterou musí splnit. Například zjištění informací o některém zvířeti.

4) *ICT jako testovací nástroj,*

Jedná se o běžný způsob využívání počítačů. Žáci jsou testováni pomocí programů k tomu určených. Výhodou, oproti konvenčním testovacím nástrojům, může být hlavně automatická oprava a vyhodnocení práce žáka.

5) *ICT jako kulisa a doplněk.*

Při tomto využití slouží počítač pouze k oživení hodiny. Patří sem využití těch informací, které plní pouze roli doplňku k učivu. Například krátké videosekvence se zajímavostmi, či podkreslení hudbou. Toto využití může hrát důležitou roli při motivaci žáků.

1.3 Terminologické vymezení pojmů výukový program a výukový software

Využití výukových programů ve výuce spadá do způsobu využití počítače výuka s počítačem. I když je možné, že v některých případech budou výukové programy prezentovat učivo týkající se softwarového, či hardwarového vybavení počítačů. V takovémto případě stojí výukový program v průniku mezi výukou o počítači a výukou s počítači.

Definicemi pojmů výukový program a výukový software a jejich vymezení od příbuzných pojmů se v minulosti i v současnosti zabývalo několik autorů, jak domácích, tak i zahraničních. Pro ilustraci na tomto místě uvedeme několik vybraných definic.

Velice jednoduše stanovil definici pojmu výukový program Brdička (1995), ten výukový program definoval jako: „*Každý program, jehož hlavním cílem je někoho něco naučit.*“ Tato definice ovšem vylučuje z výukových programů ty programy, které jsou zaměřeny pouze na zafixování učiva, či evaluaci.

Dalším z autorů zabývajících se tímto pojmem je E. Mazák (1888, s. 8). E. Mazák uvádí, že jsou dvě možnosti vnímání tohoto pojmu.

První možností je vnímání pojmu výukový program jako ekvivalent pojmu výukový software. V této rovině vnímání je za výukový program považován každý software využívání při výuce. Tedy i software, který učitel používá při přípravě na svou výuku.

Jako druhou možnost uvádí vnímání pojmu výukový program jako ekvivalent pojmu didaktický program. V takovém případě vnímání je tedy výukový program definován jako „*program, který něco vyučuje*“.

Myšlenku pana Mazáka dále upravuje J. Dostál ve svém článku Výukový software a počítačové hry – nástroje moderního vzdělávání (Dostál, 2009). Ten se opět zabývá vztahem pojmů výukový software, výukový program a odděluje pojem edukační software. Aby zamezil zařazování softwaru, který slouží například k přípravě učitele na výuku, mezi výukový software upravil definici výukového software takto:

„*Výukový software je jakékoliv programové vybavení počítače, které je určeno k výukovým účelům a dokáže plnit alespoň některou z didaktických funkcí.*“ (Dostál, 2009, s. 23)

Výukový program chápe o něco konkrétněji než předešlí autoři. Podle něj můžeme chápat výukový program jako: „*konkrétní software určený k výukovým účelům*“ (Dostál, 2009, s. 23)

Pro doplnění na tomto místě uvedeme i definici od zahraničního autora, který vytvořil definici pojmu teachware, což je anglický ekvivalent pojmu výukový software. Definice zní (Sariyildiz a kol., 1996, s. 385):

„*Teachware is software that is specially made for a specific teaching task. The main goal is to teach a person without the physical presence of a teacher but with the advances of an interactive learning environment.*“

Pokud definici přeložíme do českého jazyka, zní tedy přibližně takto: Výukový software je software vyvinutý pro specifické učební úlohy. Hlavním cílem je učit osoby bez fyzické přítomnosti učitele, ale s výhodami interaktivního učebního prostředí.

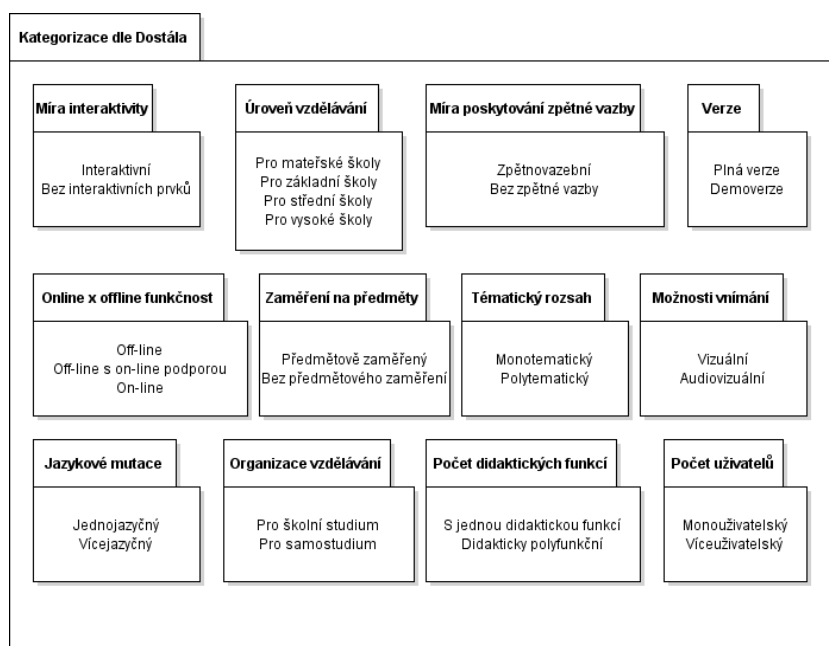
Z uvedených definic tedy můžeme vyvodit závěr, že výukový software je obecně vnímán jako pojem širší, který v sobě zahrnuje i pojem výukový program. Pojmem výukový program poté tedy rozumíme již konkrétní program s cílem něco naučit.

1.4 Členění výukových programů

Mezi autory zabývající se určitým rozdělením výukových programů do kategorií patří například E. Mazák (1989), E. Burianová (2003) a J. Dostál (2011). Jedna z posledních a nejobsáhlejších kategorizací byla navržena právě J. Dostálem.

1.4.1 Kategorizace výukových programů dle J. Dostála

Jak tvrdí J. Dostál (2011), svou kategorizaci vypracoval po prostudování 148 výukových programů, které vznikly jak na domácím trhu, tak i zahraničních trzích. Kategorizace obsahuje celkově 12 položek, podle kterých můžeme výukový program rozdělit. Grafické znázornění kategorizace nalezneme na následujícím obrázku.



Obrázek 1 Kategorizace dle J. Dostála (2011)

Výukové programy je tedy možné rozdělit podle následujících kategorií (Dostál, 2011):

1) *dle míry interaktivity:*

- interaktivní,
- bez interaktivních prvků.

Mírou aktivity označujeme žákovu možnost zasahovat do chodu výukového programu. U interaktivního programu žák může ovlivňovat jeho chod, kdežto u programu bez interaktivních prvků pouze přijímá obsah.

Příklady interaktivních výukových programů:

- TS Přírodověda – Terasoft,
- Dopravní výchova – Silcom multimédia,
- Matematika zajímavě – Pachner.

Příklady výukových programů bez interaktivity:

- TS pohádková matematika – Terasoft,
- Technický slovník naučný – Leda s.r.o..

2) *Dle úrovně vzdělávání:*

- pro mateřské školy,
- pro základní školy,
- pro střední školy,
- pro vysoké školy.

Zde se jedná o rozlišení výukových programů podle toho, pro který stupeň systému vzdělávání jsou primárně určeny. Liší se jak didaktickou transformací učiva, tak i zpracovávány okruhy, které vyplývají z cílů vzdělávání.

Příklady výukových programů pro mateřské školy:

- Dětský koutek 1 – 5 – Terasoft,
- Alík – než půjdu do školy – Silicom multimedia,
- Chytré dítě – Multimédia Art s.r.o.,
- Méd'a – Barvy a tvary – Petit.

Příklady výukových programů pro základní školy:

- Školní hrátky – Štefl software,

- Matematika zajímavě – Pachner,
- TS Angličtina – Terasoft,
- Křížem krážem staletími – BSP multimédia s.r.o..

Příklady výukových programů pro střední školy:

- Dynamická geometrie – Pachner,
- Historie – interaktivní průvodce dějinami – BSP multimédia s.r.o.,
- Věda – interaktivní průvodce světem přírodních věd - BSP multimédia s.r.o.,
- TS Sexuální výchova – Terasoft.

3) *Dle míry poskytování zpětné vazby:*

- zpětnovazební,
- bez zpětné vazby.

Jedná se o rozlišení, zda program poskytuje žákovi zpětnou vazbu o zvládnutí učiva. Jedna z možností ověření je například testování žáků programem.

Příklady zpětnovazebních výukových programů:

- Interaktivní matematika – Prodos spol. s.r.o.,
- TS Němčina – Terasoft,
- Veselá matematika – 20 000 mil pod mořem – Silicom multimedia.

Příklady výukových programů bez zpětné vazby:

- Fyzika zajímavě – Animace – Pachner,
- TS Artopedia 3 – Terasoft,
- Vytváříme web v HTML/CSS – Landi Multimedia ČR s.r.o..

4) *Dle organizování vzdělávání:*

- pro školní studium,
- pro samostudium.

Pokud jsou výukové programy primárně určeny pro nasazení ve výuce probíhající ve školním prostředí, jedná se o výukový program pro školní studium. Naproti těmto výukovým programům stojí programy určené pro samostatnou práci žáků, která probíhá většinou v mimoškolním prostředí.

Příklady výukových programů pro školní studium:

- Environmentální výchova – Pachner,
- Česká republika a její kraje – Silicom multimedia,
- Edison 5.0 – Terasoft.

Příklady výukových programů pro samostudium:

- Talk Now! – Naučte se anglicky – Terasoft,
- Technický slovník naučný – Leda,
- Vytváříme web v HTML/CSS – Landi Multimedia ČR s.r.o..

5) Dle on-line x off-line funkčnosti:

- off-line,
- off-line s on-line podporou,
- on-line.

V této kategorii rozlišujeme výukové programy podle toho, kde jsou fyzicky nainstalovány. Mohou být nainstalovány na lokálním disku počítače, na serverovém disku školy a některé výukové programy jsou nainstalovány pouze na vzdáleném serveru společnosti a klienti k nim poté přistupují pomocí sítě internet. On-line podporou se rozumí program, který je nainstalován na lokálním úložišti, ale je možné využít internetovou podporu dodavatele.

Příklady off-line výukových programů:

- Odmaturuj z IVT – Allrightsoftware,
- Psanec – Martin Bajatlon a Miroslav Rakušák.

Příklady off-line výukových programů s on-line podporou:

- TS Přírodověda – Terasoft,
- Einstein junior - Silicom multimedia,
- Rostliny kolem nás - Pachner.

Příklady on-line výukových programů:

- Škoda hrou – Škoda auto a.s.,
- Hravé učení – Idnes.cz,
- Pexeso s obrázky květů – Onesoft.

6) *Dle počtu uživatelů:*

- monouživatelský,
- víceuživatelský.

V tomto bodu rozdělujeme program podle toho, zda ho může v jednom čase používat více uživatelů. J. Dostál dále uvádí, že jsou dvě možnosti víceuživatelských programů. Jedna z možností je, že se více uživatelů střídá u jednoho počítače. Druhou možností může být nainstalování výukového programu na školní server, kde k němu má později právo přistupovat více uživatelů v jednom čase. Monouživatelský výukový program je poté přizpůsobený pouze pro jednoho uživatele.

Příklady monouživatelských výukových programů:

- Dopravní výchova – Silicom Media,
- Peníze kolem nás – Pachner,
- TS Přírodověda – Terasoft.

Příklady víceuživatelských výukových programů:

- Akční čeština – Sarsoft,
- Vlastík – Pacher,
- Word Manager – Marek Vít.

7) *Dle tematického rozsahu:*

- monotematický,
- polytematický.

Toto rozdělení udává, zda obsah pokrývá pouze jeden, či více tematických celků. Monotematický výukový program by například v přírodopisu pokrýval savce a naproti tomu polytematický by se mohl zaměřit na všechny organismy.

Příklady monotematických výukových programů:

- Vegetace České republiky – Lesy – Pachner,
- Matematika zajímavě – Aritmetika – Pachner,
- Média počítá – Petit.

Příklady polytematických výukových programů:

- Věda hrou – BSP multimédia,
- TS Celá rodina milionářem – všeobecný přehled – Terasoft,

- Alík – než půjdu do školy – Silicom multimédia.

8) *Dle možností vnímání:*

- vizuální,
- audiovizuální.

Na tomto místě se rozhoduje dle toho, zda program působí pouze vizuálně, nebo obsahuje i auditivní složku. Auditivní složku velice často využívají programy pro děti v předškolním věku, kde je nutné předčítat text.

Příklady vizuálních výukových programů:

- Vegetace České republiky – Pachner,
- TS Matematika pro 5. ročník – Terasoft,
- MATIK – MATIK Liberec.

Příklady audiovizuálních výukových programů:

- TS Český jazyk 1 (pravopis) – Terasoft,
- Méd'a čte – Petit,
- Hravé učení – Idnes.cz.

9) *Dle jazykových mutací:*

- jednojazyčný,
- vícejazyčný.

Tato kategorie rozděluje výukové programy dle toho, zda obsah je vytvořen jedním jazykem, většinou mateřským, anebo program obsahuje více jazykových mutací, mezi kterými lze volně přepínat. Díky této možnosti může u dítěte dojít i k rozvoji jazykových schopností. Další možností vícejazyčných výukových programů je varianta, kdy program slouží přímo k výuce cizího jazyka a obsahuje mateřský i cizí jazyk zároveň.

Příklady jednojazyčných výukových programů:

- TS Botanika – Terasoft,
- Hrátky s přírodou – Štefl software,
- Český jazyk – pravopis hrou – Pachner.

Příklady vícejazyčných výukových programů:

- Hravé učení – Idnes.cz,

- TS Angličtina – Terasoft,
- Didakta Němčina – Silcom.

10) Podle verze:

- plná verze,
- demoverze.

Některé firmy poskytují kromě plné verze programů, která obsahuje všechny možnosti, i takzvané demoverze. Ty jsou ve většině případů poskytovány zdarma, ale nemají všechny možnosti plné verze, například mohou obsahovat pouze jednu kapitolu. Demoverze slouží k vyzkoušení výukového programu před jeho zakoupením.

Příklady výukových programů, které poskytují demoverzi na vyzkoušení:

- Edison 4 – Terasoft,
- Pravopis – MATIK Liberec,
- Peníze kolem nás – Pachner.

11) Dle počtu didaktických funkcí:

- s jednou didaktickou funkcí,
- didakticky polyfunkční.

Jak vyplývá z dříve uvedených definic, výukový program musí plnit alespoň jednu z didaktických funkcí. Podle toho, zda plní právě jednu nebo více didaktických funkcí, rozdělujeme programy v této kategorii.

Příklady výukových programů s jednou didaktickou funkcí:

- Fyzika zajímavě – Atomistika a astronomie – Pachner,
- Vytváříme web v HTML/CSS – Landi Multimédia,
- Hledej květinu podle jména – Onesoft.

Příklady polyfunkčních výukových programů:

- TS Angličtina – Terasoft,
- Vegetace České republiky – Pachner,
- Alík – než půjdu do školy – Silicom multimédia.

12) *Dle zaměření na jednotlivé předměty:*

- předmětově zaměřený,
- bez předmětového zaměření.

Poslední možností je rozdělení programů dle toho, zda jsou zaměřeny výhradně na jeden vyučovací předmět, nebo je jejich aplikace možná nezávisle na vyučovacím předmětu.

Příklady předmětově zaměřených výukových programů:

- TS Přírodověda – Terasoft,
- Vegetace české republiky – Pachner,
- Fyzika zajímavě – Animace – Pachner.

Příklady výukových programů bez předmětového zaměření:

- TS Celá rodina milionářem – všeobecný přehled – Terasoft,
- Věda hrou – BSP multimédia,
- Hravé učení – Idnes.cz.

1.4.2 Členění výukových programů dle funkčního hlediska dle Burianové

Druhým členěním, které v této práci zmíníme, je členění E. Burianové (Burianová, 2003). Burianová ve své práci navrhla členění výukových programů dle funkčního hlediska. Funkční hledisko je vymezeno základními funkcemi výukových programů. Jako základní dělení udává dělení na monofunkční programy a multifunkční programy. Monofunkční programy plní pouze jedinou funkci, kterou je například výklad nového učiva. Naproti tomuto multifunkční programy plní minimálně dvě funkce, ale mohou jich naplňovat i více - například výklad učiva a procvičení učiva. Dále člení multifunkční výukové programy dle dominantní funkce do těchto skupin (Burianová, 2003):

1) Programy k výkladu učiva,

Tyto programy mívají většinou textový charakter. Jejich nasazení může sloužit jako náhrada za klasickou učebnici. Na rozdíl od klasických učebnic umožňují tyto programy graficky si text přizpůsobit a mnohdy program využívá databáze textů, které svým rozsahem učebnice značně převyšují. Mezi dobře zpracované výukové programy s převahou výkladové složky patří i výukový program Vegetace České republiky, kterou vydala česká společnost Pachner.

2) *Programy k procvičování učiva,*

Výukové programy patřící do této skupiny mají dominantní funkci většinou zaměřenou na testování a hodnocení znalostí žáka. Mezi jednoduché výukové programy patří Fast Test Maker od vývojáře Intri, který slouží ke snadné tvorbě didaktických testů.

3) *Demonstrační a motivační programy,*

Demonstrační programy obsahují možnosti k demonstraci učiva. K demonstraci slouží filmy, animace, nákresy atd. Tyto programy jsou při správném využití ve výuce pro žáky zajímavé a aktivizující a tudíž slouží především k motivaci žáka. Množství animací zobrazujících fyzikální jevy obsahuje výukový program Fyzika zajímavě – Animace od společnosti Pachner.

4) *Simulační programy,*

Tyto programy slouží jako náhrada za fyzicky provedené pokusy, které by například nebylo možné v reálném prostředí školy uskutečnit. Příkladem takového výukového programu může být Edison od společnosti Terasoft.

5) *Programy řízení experimentů a procesů,*

Tyto programy umožní počítači pracovat například jako měřicí přístroj, například osciloskop. Příkladem může být software Rc2000 od společnosti RC Didactic, který umožňuje měření signálů, a to analogových i digitálních (RC Didactic, 2007).

6) *Programy pro zpracování databází,*

Jsou programy sloužící k připojování se k databázím prostřednictvím internetu a mimo to i vytváření vlastních menších databází.

7) *Počítačové hry.*

Jedná se o speciální typ výukových programů. J. Dostál definoval didaktickou hru takto (2009, s. 28): „*Didaktická počítačová hra je software umožňující zábavnou formou navozovat činnosti zaměřené na rozvoj osobnosti jedince.*“ Jedná se tedy o výukový program, který se snaží rozvíjet jedince co možná nejzábavnější formou. Tvorbou didaktických počítačových her se zabývá mimo jiné i společnost Terasoft. Jednu z her, kterou je možno využít, nazvali Pexesa pro celou rodinu.

1.5 Dílčí závěr

V této kapitole došlo k analýze současného stavu užívání počítačů a výukového software ve vzdělávání. Důležitou částí je terminologické vymezení pojmů výukový software a výukový program, které bude dále v práci promítnuto. Taktéž byly představeny 2 způsoby, kterými je můžeme členit a tím pádem je i charakterizovat. Kategorizace J. Dostála (2011) bude v praktické části této práce využívána k charakteristice vytvořených programů.

2 VYUŽITÍ VÝUKOVÉHO SOFTWARE PŘI VÝUCE

Výukový software lze ve výuce využívat nejrůznějšími způsoby. Využití výukového softwaru nemusí být plně vázáno pouze na prostředí školy, ale je možné ho použít i k samostatné domácí práci. Taktéž výukový software není vázaný pouze na předměty zabývající se informační výchovou, ale vznikají dnes prakticky pro všechny předměty vyučované na našich školách.

2.1 Důvody proč využívat výukový software

Výhody, které s sebou přináší výukový software a tedy i důvody, proč jej používat, vytyčili ve své práci J. Dostál (2011) a E. Burianová (2003).

Výukový software může pomoci žákům při domácí přípravě na výuku. Díky tomuto softwaru si mohou žáci vylepšit neuspokojivé školní výsledky, či při dlouhodobé absenci, může výukový software posloužit jako nástroj pro doučení zameškaného učiva (Burianová, 2003).

Pro rodiče může být výukový software také velice přínosný. Zejména v situaci, kdy svým dětem pomáhají s přípravou na výuku. Dítě dokáže, po nastavení základních parametrů rodičem, se softwarem pracovat i samo a rodič díky tomu ušetří cenný čas (Burianová, 2003).

Dále může výukový software posloužit při rozvoji kreativity a myšlení. Vedlejším efektem práce žáků s výukovým softwarem je také to, že se učí zacházet se samotným počítačem. Pokud práci na počítači nepoužívá učitel příliš často, bývá tento prvek výuky pro žáky vzácný a tudíž i zábavný. Ovšem nesmí při práci s výukovým softwarem převažovat zábava a zcela chybět prvek učení (Dostál, 2011).

Díky možnosti aktivování více smyslů při učení pomocí výukového softwaru, se žákům informace lépe zapamatovávají. Využíván je ve většině případů jak zrak, tak i sluch. Výukový software také může pomoci s názorností výuky. Při správném zařazení do výuky dokáže zastoupit řadu pomůcek (Dostál, 2011).

Výhodou je také interaktivnost výukového softwaru. Ta spočívá v tom, že pokud žák udělá při svém postupu chybu, ihned se tuto skutečnost dozví, a buď dostane možnost svou chybu napravit, nebo se dozví správné řešení (Dostál, 2011).

2.2 Funkce zajišťující výukový software

Na tomto místě uvedeme činnosti, které by měl správný výukový software zajišťovat. Ne vždy musí software pokrývat všechny z uvedených činností. Naopak je spousta programů, které jsou zaměřeny pouze na některou z uvedených činností.

Funkce, které by měl výukový software zajišťovat (Dostál, 2011):

- *motivace žáka,*

Motivace je považována za předpoklad úspěšného učení. Jedním z hlavních cílů by tedy mělo být přimět žáky, aby se sami chtěli učit (Petty, 2002). U výukového softwaru se často dělá chyba v tom, že jsou vytvořené ze spousty animací, barev atd., ale i přesto je motivační hodnota pouze minimální.

- *předávání informací žákovi,*

Pomocí této činnosti poskytujeme žákovi nové poznatky. Ve výukovém softwaru můžeme použít, kromě klasického textu, i lepší možnosti strukturalizace textu, či můžeme využívat animace, videa, zvuky a podobně.

- *fixace osvojených poznatků a dovedností,*

Výukový software lze velice dobře využít i k fixaci nových poznatků a dovedností. K tomuto účelu by měl výukový software obsahovat různá cvičení, které slouží k opakování učiva. Tuto funkci mohou dobře plnit i didaktické počítačové hry.

- *kontrola osvojených znalostí a hodnocení.*

Hodnocení slouží k měření úrovně znalostí a dovedností, které byly nabyty při výuce. Hodnocení slouží k mnoha účelům. Pomáhá při klasifikaci žákových výkonů, také při výběru žáků do kurzů, může identifikovat učební obtíže u žáků, také určuje efektivnost práce učitele a může sloužit žákům jako cíl jejich práce (Petty, 2002). Je tedy jasné, že hodnocení je důležitá součást výuky, proto by tato část neměla chybět ani u výukového softwaru. Většinou je tato funkce přítomna ve formě didaktických testů. Jak dále uvádí J. Dostál (2011), nemělo by u softwaru dojít pouze ke konstatování určitého stupně znalostí, ale mělo by být upozorněno i na slabé a silné stránky žáka.

V publikaci *Počítač ve vzdělávání* J. Dostál (2007) uvádí ještě jednu funkci, a to *reakci podle výsledků zpětnovazební informace*. Tato funkce by měla sloužit jako reakce na žákovy výsledky učení a nabídnout mu další postup a případně ho i motivovat k dalšímu studiu dané tematiky.

2.3 Využití výukového softwaru

Možnosti využívání výukového softwaru ve výuce navrhla krom jiných E. Burianová (2003). Ta sestavila celkem čtyři body, které popisují, jaké možnosti máme při využití výukového softwaru:

1) *software jako součást výkladu učiva,*

Software jako součást výkladu učiva. Učitel má možnost využít výukový software jako součást svého vlastního výkladu. Výklad patří k nejpoužívanějším metodám výuky, G. Patty (2002) uvádí, že dokonce 60 % hodiny je zaplněna právě výkladem. Už kvůli této skutečnosti je potřeba výklad čas od času obohatit. Předávání nových informací žákům pomocí výukového softwaru může být pro žáky poutavější a zábavnější, než využití učebnice nebo pouze slovního výkladu učitele. Jak dále Burianová (2003) uvádí, podmínkou tohoto využití je přehled učitele o možnostech výukového softwaru a také jeho dostupnost. Toto využití je kromě doplňku samotného výkladu vhodné i pro slabší žáky, či žáky doplňující si zameškané učivo po dlouhé nepřítomnosti.

2) *software využitý k simulacím,*

Simulační software může být ve výuce využíván jako doplněk výkladu. Počítačová simulace, může posloužit při vysvětlování některých pojmů či postupů. Simulace pracuje ve většině případů s grafickou podobou simulovaného procesu. Taktéž bývá u simulace běžné nastavování parametrů a okamžité zobrazení výsledků. Pokud je učebna vybavena dataprojektorem, má možnost učitel využít takovou simulaci přímo ve svém výkladu. Druhou možností je, že si žáci se simulací pracují samostatně v počítačové učebně.

3) *procvičování osvojených znalostí a dovedností,*

Využívání k **procvičování osvojených znalostí a dovedností** je nejčastější možnost využívání softwaru v procesu výuky. Podle Burianové (2003), platí pravidlo, že čím je cílová skupina žáků procvičovacího programu mladší, tím více cvičení v něm připomínají spíše hru. Procvičovací software je určen v drtivé většině případů pro samostatnou práci žáků

a to jak ve školní počítačové učebně, tak i v domácím prostředí. Některé výukové programy se snaží podpořit soutěživost mezi žáky použitím výkonových žebříčků. V takovémto žebříčku je poté uložen například čas, nebo skóre dosažené žákem ve cvičení. Žáky posléze k dalšímu opakování motivuje i touha porazit své spolužáky

4) porovnávání a testování vědomostí žáků.

Poslední možností využití podle E. Burianové (2003) je využití k **porovnávání a testování vědomostí žáků**. Programy zaměřené na testování žáků mají své zvláštní místo při využití výukového softwaru. Existují dvě možnosti zpracování softwaru pro testování žáků. Prvním způsobem je generování testů, které jsou posléze vytisknuty a předloženy žákům na papíře. Tady si učitel zvolí látku a počet otázek, software automaticky vygeneruje test, který bude vytištěn. Nevýhodou tohoto způsobu je, že učitel bude muset testy žáků kontrolovat sám. Mezi výhody naopak patří, že učitel nemusí vymýšlet otázky sám a také není potřeba přesun do počítačové učebny. Druhou možností je vyplňování testů žáky přímo na počítači. Tato možnost má tedy tu výhodu, že software většinou umí test sám vyhodnotit. V některých případech je dokonce schopen navrhnout klasifikaci žáka. Nevýhodou tedy je nutnost přesunu třídy do počítačové učebny. Takovéto testování lze využít i při kontrole výsledků během samostatného studia.

2.4 Dílčí závěr

V této kapitole byli analyzovány důvody, které by měli vést k využívání výukového software ve vzdělávání. Taktéž zde byly představeny konkrétní možnosti, které má učitel k dispozici při využívání výukového softwaru při vyučování. Jednotlivé možnosti využití může samozřejmě ve své hodině libovolně kombinovat.

3 SOUČASNÉ PROBLÉMY A NEDOSTATKY VÝUKOVÉHO SOFTWARE

Na základě našeho pozorování při souvislé praxi a rozhovoru s několika učiteli, jsme se pokusili shrnout základní body, které většinou zabraňují širšímu využívání výukového software učiteli při procesu výuky. Mezi tyto problémy je možné zařadit:

- 1) cenu výukového softwaru,
- 2) omezení licencí softwaru,
- 3) nemožnost přizpůsobení obsahu,
- 4) omezení počítačovou platformou,
- 5) požadavky na hardware,
- 6) využití techniky, kterou škola disponuje,
- 7) možnosti zapojení celé třídy,
- 8) možnosti samostudia a automatické odesílání úkolů učiteli,
- 9) možnosti vytvoření vlastního testu,
- 10) přizpůsobení motivační složky.

V následujících podkapitolách podrobněji rozebereme, v čem jednotlivé problémy spočívají a jestli je možné je vyřešit tím, že si učitel bude vytvářet svůj vlastní výukový software.

3.1 Cena výukového softwaru

S financemi je to ve škole vždy obtížné. Výukový software je svou cenou značně variabilní. Existuje sice i software, který je poskytován zdarma, ale často se jedná o méně kvalitní software, neúplný, či jiným způsobem omezený.

Jako další cenový stupeň bychom uvedli výukový software dostupný pro mobilní platformy. Zde se většinou cena pohybuje v řádech desítek korun. Příkladem může být vývojář Antonín Vonderka, který se zabývá tvorbou výukového software pro Android a Windows. Tento vývojář nabízí své produkty v obchodech s aplikacemi obou zmíněných platform (Google Play, Windows Store) průměrně za 25 korun (Učte se hrou, 2007). Způsob

distribuce pro mobilní platformy, který se výhradně praktikuje přes speciální obchody s aplikacemi jednotlivých systémů, s sebou nese pro školy vcelku závažný problém. Problém spočívá v platbě a následné fakturaci požadovaných aplikací. Konkrétně na iTunes, což je obchod s aplikacemi určený pro platformu iOS, není možné v českých podmínkách dostat k zakoupeným aplikacím daňový doklad, který škola potřebuje doložit do svého účetnictví.

Poté přichází již výukový software určený většinou pro stolní počítače. Jeho cena se pohybuje ve stech, či dokonce tisících korun za licenci. Jako příklad zde uvedeme jednu z firem, která na našem trhu s výukovým software působí nejdéle, a to firmu Terasoft. Ceny u této firmy se pohybují okolo 500 Kč za licenci (Terasoft, 2012). Na uvedených příkladech, lze tedy vidět, že pořízení výukového softwaru pro celou třídu není nejlevnější záležitostí. I když je nutno přiznat, že většina firem dodávajících výukový software, poskytuje školám speciální nabídky s balíčky, které obsahují více licencí, za sníženou cenu.

Pro školu a potažmo i pro žáky je nejlepší ten program, za který nemusí platit. A právě proto je výhodné vytvořit si svůj vlastní software, než nemít žádný software, protože vám ho škola neproplatí. Námí vytvořený software je naše dílo a tudíž je zdarma, jak pro nás, tak i pro lidi, kterým tento software poskytneme. Zdarma je tedy pokud pomineme počáteční investice do vývojového prostředí. Ovšem i ty již nejsou zcela nutné, protože Microsoft poskytuje své vývojové prostředí Visual Studio Community 2015 pro samostatné vývojáře, vzdělávání, vědecký výzkum atd. zcela zdarma (VisualStudio, 2016). Jedná se o plnohodnotné vývojové prostředí, ve kterém je možné vyvíjet software pro většinu platform. Po eliminaci investic do vývojového prostředí stojí vývoj výukové aplikace pouze čas.

3.2 Omezení licencí softwaru

Licenci můžeme charakterizovat jako smlouvu mezi autorem softwaru na jedné straně a uživateli programu na straně druhé. V licenční smlouvě, se kterou bychom měli být seznámeni ještě před začátkem využívání softwaru, bývají stanoveny přesné podmínky, za kterých je možné software používat (Štědrón, 2010). Softwarová licence úzce souvisí s cenou, která byla řešena v předchozí podkapitole. Při koupi softwaru totiž neplatíme ani tak za samotný produkt jako právě za licenci k používání softwaru.

Podle licence můžeme rozdělit software na několik základních skupin. Mezi tyto skupiny patří tzv. proprietární software, shareware, freeware, free software a jiné.

Nejvíce omezující bývá pro uživatele proprietární software, jinak označovaný také software komerční. Uživatel při tomto typu licence dostává již zkompileovaný program a nemá přístup do zdrojového kódu. Uživatel nemá právo zasahovat do obsahu softwaru, nesmí software dále šířit a i podmínky užívání jsou značně omezené. Licenci u proprietárního software je navíc nutné vždy zaplatit (Štědron, 2010).

Freeware a shareware licence mají s proprietární licencí společnou nemožnost úpravy obsahu. Rozdílem mezi těmito licencemi je v ceně užívání. Freeware bývá zdarma po celou dobu užívání. Naproti tomu shareware bývá zdarma po určitou zkušební dobu, ale pokud chceme software využívat i po vypršení této doby, je nutné licenci zaplatit. Výhodou pro učitele může ale být, že oba typy licence je možné šířit dále bez souhlasu autora (Štědron, 2010).

Poměrně novým licenčním modelem jsou takzvané freemium aplikace. Dalo by se říci, že software s tímto druhem licence, stojí mezi freeware a shareware softwarem. Základní část programu, která ve většině případů plně postačuje k plnohodnotnému využívání, bývá poskytována po celou dobu využívání zcela bezplatně. Mimo základní program nabízí autor rozšiřující moduly, či obsah za poplatek. Dokupování rozšiřujícího obsahu samozřejmě není pro uživatele povinné a program je i bez něj nadále použitelný zdarma. Tímto se odlišuje licence freemium od shareware (Freemium, 2016).

Nejméně omezující, nejen pro učitele, je licence free software. Jedná se o software, jehož licence s sebou nenese žádná omezení. Majitel má právo software spouštět, studovat jeho kód, předávat software dále, ba dokonce i měnit, či rozšiřovat jeho kód a obsah, to vše lze bez svolení autora. Free software má další výhodu a to tu, že bývá velice často distribuován zcela zdarma (What is free software?, 2016). Tato licence je tedy pro učitele stejně vhodná, jako tvoření si svého vlastního softwaru. Učitel není svazován žádnými pravidly a může si obsah svého programu přizpůsobit podle svých vlastních představ. Bohužel se mezi softwarovými licencemi jedná o výjimku.

Pokud tedy nenarazíme na free software, jsou vždy licence pro učitele svým určitým způsobem svazující. Největší omezení většinou spočívá právě v tom, že licenční smlouva učiteli zabraňuje v upravování obsahu a kódu softwaru. Dále také učitel nemusí mít možnost poskytnout software žákům domů, aby mohli pracovat dále samostatně. Tato omezení

samozřejmě padají s tvořením vlastního software, kdy je učitel sám autorem a tudíž si také sám určuje pravidla nakládání s jeho produktem.

3.3 Nemožnost přizpůsobení obsahu

Tento problém je spjat s předchozí podkapitolou, kde je vysvětleno, proč není možné, aby si učitel sám upravoval obsah výukového softwaru. Tato vlastnost je pro učitele dosti svazující. Učiteli ne vždy musí vyhovovat veškerý obsah, který je v programu přítomen. Příkladem může být výukový program, zaměřený na prezentaci nového obsahu, kde se sice učitel líbí zpracování didaktického textu, ale obrázky mu nepřijdou dosti názorné a má k dispozici vlastní. V případě zakoupeného softwaru nemá učitel většinou možnost sjednat nápravu.

Je samozřejmé, že tento problém odpadá s tvořením si vlastního výukového software, kde si učitel sám volí obsah, který nejvíce mu vyhovuje. Zvolený obsah samozřejmě jako autor smí kdykoliv obměňovat a aktualizovat.

3.4 Omezení platformou

Za platformu můžeme označit pracovní prostředí. Mimo knihovny a programovací jazyky patří mezi platformy i operační systémy (Computing platform, 2001). Operačních systémů je na trhu v dnešní době celá řada. Kromě těch, dnes již tradičních, počítačových operačních systémů, se dostávají do popředí i operační systémy navržené pro mobilní zařízení, jako jsou mobilní telefony a tablety. K těm nejznámějším platformám patří v oblasti počítačů Windows, Mac OS a Linux a v oblasti mobilních zařízení Android, iOS a případně Windows Phone.

Máme tu tedy poměrně velké množství platform, pro které vývojáři mohou vyvíjet své programy. A protože software vyvinut na jednu platformu nemusí být kompatibilní s platformou jinou, musí vývojář své programy upravovat pro jednotlivé platformy. Z tohoto důvodu se ne vždy rozhodne svůj program vydat pro všechny platformy. Pokud tedy škola disponuje méně oblíbenou platformou, pro kterou se vývojáři často nerozhodují, nastane situace, kdy učitel nebude mít dostatek výukového softwaru.

Pokud se učitel rozhodne vytvářet si svůj vlastní výukový software, je logické, že bude vyvíjet takový software, který bude kompatibilní s dostupnou platformou. Visual Studio nabízí doplňky a moduly, pomocí kterých je možné vytvářet software na většinu nejoblíbenějších platform. Samotný jazyk Visual Basic je možné využít při tvorbě programů pro operační systémy Windows a Windows Phone (VisualStudio, 2016). Pro ostatní platformy je nutné zvolit jiný programovací jazyk, jako je například C#.

Konkrétně vyšší programovací jazyk C# .NET je v kombinaci s vývojovým prostředím Visual Studio velice silným nástrojem pro vývoj programů na různé platformy. A to hlavně díky nástroji jménem Xamarin, který je poskytován vývojáři v základní verzi zdarma. Xamarin umožňuje vyvíjet a testovat aplikace napříč mobilními platformami. Xamarin přijímá kód jazyku C#, který je posléze kompilován nativními kompilátory mobilních platform. Jediné, co je vhodné upravit pro jednotlivé platformy, je uživatelské prostředí, a to takovým způsobem, aby zapadalo do filozofie dané platformy. Platformy, které Xamarin podporuje, jsou iOS, Android a Windows Phone (Xamarin, 2016).

3.5 Požadavky na hardware

Avšak ne ve všech případech škola disponuje nejmodernějším hardwarovým vybavením, které by bylo schopné zajistit plynulý chod moderních graficky náročných výukových programů. Proto je učitel omezen ve svém výběru a může nastat situace, kdy výukový program, který by byl přiměřený k hardwaru, je zastaralý a nemusí obsahovat nejaktuálnější informace.

Učitel, který si vyvíjí vlastní výukový software, má možnost přizpůsobit hardwarovou náročnost programu. Jedna z možností je například minimalizace grafických efektů, které hardware počítače velice zatěžují.

3.6 Využití techniky, kterou škola disponuje

Učitelům mnohdy schází ve výukovém softwaru možnosti pro využití stávající didaktické techniky. Pojem didaktická technika chápeme jako „*souborné označení technických zařízení užívaných pro výukové účely.*“ (Průcha a kol., 2003, s. 43)

K nejžádanější technice patří interaktivní tabule, kterou dnes disponuje celá řada škol. Mnoho výukových programů nepočítá s možností ovládní přes interaktivní tabuli a nemá k těmto potřebám přizpůsobené uživatelské prostředí ani obsah.

Visual Basic ve svých nejnovějších verzích počítá i s využíváním zařízení, která jsou ovládána dotykem. Mezi tyto zařízení můžeme počítat i interaktivní tabule. Učitel si díky této možnosti může vyvinout program, který bude možné ovládat pomocí interaktivní tabule.

Samozřejmě do didaktické techniky neřadíme pouze interaktivní tabule, ale učitelé mohou chtít více využívat například i audio zařízení připojené k počítači. Pomocí jazyku Visual Basic je samozřejmě možné pracovat i se zvukovými soubory, které může učitel zapracovat do svého programu a díky tomu využívat audio zařízení.

3.7 Možnosti zapojení celé třídy

Nyní opět dochází s prolnutím s předchozí podkapitolou. Učitelé by si samozřejmě přáli do práce s výukovým programem v hodině zapojit celou třídu. Cesta k této možnosti může vést právě přes využívání interaktivních tabulí. Kdy můžeme například vytvořit program s jednoduchou hrou, při které by žáci museli vybírat hlasováním dalším kroky.

Další možností může být relativně samostatná práce žáků s výukovými programy. Kdy žáci s programem sice pracují samostatně, ale učitel je má plně pod kontrolou pomocí některého z již existujícího softwaru určeného k podpoře výuky. Příkladem takového softwaru je program PC Control 2 (PC Control, 2013), který umožňuje kontrolovat pracovní stanice žáků. Má tedy například možnost předvést úspěšnou práci jednoho ze žáků celé třídy, kontrolovat, zda se žáci věnují své práci, či všem hromadně spustit žádaný výukový program.

3.8 Možnosti samostudia a automatické odesílání úkolů

Výukový program by mohl sloužit jako ideální nástroj pro zadávání domácích prací žákům. Bohužel je pro učitele obtížné nalézt výukový program, který by žákům zpřístupnil přesně ten obsah, který učitel nestihl se žáky probrat.

O přizpůsobení obsahu již byla řeč výše. Učitel si tedy samozřejmě může obsah svého výukového programu upravit tak, aby ho žáci mohli využít při své domácí přípravě.

Další možností může být zadávání domácích úkolů pomocí výukových programů. Učitel má možnost vytvořit ve svém programu cvičení týkající se právě probírané látky. Často výukovým programům chybí možnost automatického zasílání vypracovaných cvičení učiteli.

Automatické odesílání výsledků učiteli je možné pomocí jazyku Visual basic vyřešit například pomocí odeslání emailu programem. Výsledek dokončeného cvičení je poté možné automaticky vyexportovat do přijatelné podoby, například textový soubor, a ten poté opět automaticky odeslat na učitelem zadanou emailovou adresu, popřípadě uložit na webový server.

3.9 Možnosti vytvoření vlastního testu

Samozřejmě, jak jsme již zmiňovali ve 2. kapitole, výukového softwaru zaměřeného na testování je celá řada. A některé dokonce umožňují i zadávání svých vlastních otázek a tvorbu vlastních testů. Příkladem může být program UTP či online tvorba testů na serveru iTesting. U většiny takovýchto programů ovšem narazíme na problém, pokud chceme v testu využít audio nahrávku, například při testu zaměřeném na rozpoznávání zpěvů ptáků, popis obrázku, či videosekvenci. Problém také nastává, pokud máme na testovací software speciální požadavky, jako například tisk šablony správných odpovědí, nebo uložení vypracovaného testu na webový server, či odeslání na email.

Pokud si bude učitel tvořit svůj vlastní software, má možnost přistupovat ke tvorbě testovacího programu značně kreativněji. Kreativita se může plně využít hlavně při tvorbě samotných testů v programu. Učitel nebude omezen na pouhé zadávání otázek a odpovědí textem, ale bude moc využít obrázky, zvuky, videa, animace apod. Jazyk Visual Basic samozřejmě umožňuje s těmito soubory pracovat.

3.10 Přizpůsobení motivační složky

Motivace zastává v učení žáků bezesporu velice významnou roli. Proto by bylo pro učitele vhodné, aby mohl přizpůsobit motivační prvky právě probírané látce. Jak tvrdí ve své publikaci Lokšová (1999), aby byla motivace co možná nejúspěšnější, je vhodné ji přizpůsobit cíli výuky, obsahu výuky a nepochybně i věku žáků.

Motivaci k učení rozdělujeme podle podnětu na vnitřní a vnější motivaci. Vnitřní motivace znamená, že se žák učí, protože se sám učit chce. To znamená, že vykonává činnost jen pro činnost samou. Druhým typem je motivace vnější, při vnější motivaci se žák učí, protože mu to někdo nařídil, ne proto, že sám chce. Pro učitele je samozřejmě vhodnější, aby rozvíjel vnitřní motivaci žáků, protože vnitřně motivovaní žáci podávají kvalitnější výkony (Lošková, 1999).

G. Petty stanovil ve své knize 7 motivů, proč se žáci chtějí učit (2002, s. 40).

1. *Věci, které se učím, se mi hodí.*
2. *Kvalifikace, kterou studiem získám, se mi hodí.*
3. *Při učení mívám dobré výsledky a tento úspěch mi zvyšuje sebevědomí.*
4. *Když se budu dobře učit, vyvolá to příznivý ohlas mého učitele nebo spolužáků.*
5. *Když se nebudu učit, bude to mít nepříjemné důsledky.*
6. *Věci, které se učím, jsou zajímavé a vzbuzují moji zvědavost.*
7. *Zjišťuji, že vyučování je zábavné.*

Výukový software je ve většině případů zaměřený právě na poslední motiv, tedy na vyzdvížení zábavnosti výuky. Zábavnost výuky je možné zajistit pomocí didaktických her, využití soutěží, či správnou volbou příkladů, které by měly být žákům co možná nejbližší.

3.11 Dílčí závěr

V kapitole jsme popsali skutečnosti, které byly vyhodnoceny jako problémové, při využívání výukového software učitelem ve svých hodinách. Na zjištěné problémy reagujeme vytvořením kuchařky nápadů, kde učitelé předkládáme své ideje, pro tvorbu vlastních výukových programů. Na základě tvorby svých vlastních výukových programů, může učitel výše zmíněná rizika lépe kontrolovat, případně zcela eliminovat.

4 KUCHAŘKA NÁPADŮ

4.1 Metody práce

Všechny výukové programy, které budeme dále představovat, jsme vytvořili pomocí programovacího jazyku Visual Basic.NET. Jako vývojové prostředí jsme použili nejaktuálnější verzi Visual Studia od firmy Microsoft, a to verzi 2015 určenou pro platformu Windows. Výhody jazyka Visual Basic a vývojového prostředí Visual Studio osobně spatřujeme především:

- intuitivní kód,

Kód jazyka Visual Basic je tvořen anglickými výrazy, které umožní pochopení zapsaného kódu i méně zkušeným programátorům.

- nerozlišování velkých a malých písmen,

Visual basic nerozlišuje, jestli název proměnné, funkce, konstanty atd. napíšeme velkými nebo malými písmeny. Visual Studio dokonce zařídí, aby název použitého objektu, odpovídal velikostí písmen původní deklaraci.

- přehlednost tvořeného kódu,

Visual Studio používá různé nástroje, kterými se pokouší zvýšit přehlednost kódu. Mezi tyto nástroje patří například automatické odsazování kódu, barevné odlišování příkazů, parametrů, textových řetězců atd. i možnost zabalení aktuálně nepotřebných procedur. Všechny tyto možnosti výrazně usnadňují orientaci ve vytvořeném kódu.

- integrovaný našeptávač,

Visual Studio obsahuje integrovaný našeptávač, který uživateli navrhuje, jaké má možnosti při dokončování příkazu. Díky této funkci nemusí uživatel vždy znát přesné znění všech příkazů.

- návrh formulářů.

Práce s formuláři je ve Visual Studiu velice intuitivní. Vývojář umísťuje a upravuje jednotlivé prvky na formulář, jako jsou tlačítka, textboxy, labely atd., pomocí jednoduchého grafického rozhraní.

Právě díky tomuto seznamu vlastností byla zvolena tato kombinace programovacího jazyka a vývojového prostředí, která je díky těmto vlastnostem vhodná nejen pro nepřilíš

zkušené programátory. Zajisté do této skupiny programátorů bude nejspíš patřit i většina učitelů, kteří by se mohli věnovat vývoji vlastního výukového softwaru.

4.1.1 Visual Basic

Název Visual Basic zastupuje v užším pojetí konkrétní programovací jazyk se svými pravidly, které musí programátor dodržovat, aby připravil funkční kód pro kompilátor (Halvorson, 2015).

Za vznikem a vývojem tohoto programovacího jazyka stojí americká společnost Microsoft. První verzi představila tato společnost již v roce 1991 a nesl označení Visual Basic 1.0. Visual Basic se mezi programátory ujal, hlavně díky možnosti umísťování prvků na formuláře a nastavování jejich vlastností a následným doplňováním kódu vázaného na prvky (Halvorson, 2015).

Vývoj pokračoval přes verze 2, 3, 4, 5 až k Visual Basicu verze 6.0, který se stal mezi programátory velice populární. Tato verze byla vydána v roce 1998 a 2 roky po jejím vydání, tedy v roce 2000, začal pracovat Microsoft na nové verzi již s přívlastkem .NET. Visual Basic 6.0 byl tedy poslední verzí bez tohoto přívlastku (Jak je to s verzemi Visual Basicu, 2008).

V roce 2002 vydal Microsoft první verzi s přívlastkem .NET, díky které se stal Visual Basic univerzálním programovacím jazykem, který byl určen pro vývojovou platformu Microsoft.NET Framework, tehdy ve verzi 1.0. Novinkou tohoto vývojového prostředí byla změna přístupu ke kompilaci souborů. Klasický přístup byl převedení kódu vyššího programovacího jazyka přímo do strojového kódu. S nástupem .NET byl tento přístup změněn. Kód vytvořený vyšším programovacím jazykem, byl nejprve převeden do mezikódu, označovaného jako MSIL (Microsoft Intermediate Language) a až tento mezikód je překládán do kódu strojového. Díky tomuto rozdílu byly smazány startovací rozdíly mezi jazyky podporujícími platformu .NET. Kromě Visual Basicu je takovýmto jazykem i C# (Hanák, 2011). Vyjma této změny se zjednodušila i práce s jazykem. Jako příklad uvádíme na obrázcích číslo 2 a 3 rozdíl mezi nakreslením diagonální červené čáry ve Visual Basicu 6.0 a v nové .NET verzi.

```

Private Sub Form_Load()
Line1.BorderColor = vbRed
Line1.BorderWidth = 10
Line1.X1 = 0
Line1.X2 = Me.ScaleWidth
Line1.Y1 = 0
Line1.Y2 = Me.ScaleHeight
End Sub

```

Obrázek 2 Nakreslení čáry ve VB 6.0

Na obrázku číslo 2 lze vidět nastavení souřadnic čáry ve starší verzi Visual Basicu. Čára zde musela být na formulář přidána před psaním kódu, nastavení souřadnic, barvy a šířky pera proběhlo na šesti řádcích. Na obrázku číslo 3 lze vidět stejný úkon v .NET verzi, kde se čára vykreslila přímo při psaní kódu a samotné nastavení proběhlo na pouhých dvou řádcích.

```

Private Sub Form1_Paint(sender As Object, e As PaintEventArgs) Handles Me.Paint
Dim pen As New Pen(Color.Red, 10)
e.Graphics.DrawLine(pen, 0, 0, Me.ClientSize.Width, Me.ClientSize.Height)
End Sub
End Class

```

Obrázek 3 Nakreslení čáry ve verzi .NET

Dnes je Visual Basic, již ve verzi 2015, považován za moderní programovací jazyk s objektově orientovanými základy. Bohužel pro svou rychlost provádění operací není vhodný pro tvorbu náročnějších aplikací, jako jsou například moderní hry. Naopak je vhodný pro méně náročnější aplikace, jako mohou být jednodušší hry, či jednoduché kancelářské aplikace (Suchánek, 2013).

4.1.2 Visual Studio

Visual Studio je integrované vývojové prostředí, které vyvíjí společnost Microsoft. Vývojovým prostředím myslíme místo, které slouží ke tvoření našeho kódu a k úpravě vzhledu aplikací (Halvorson, 2015).

Nejnovější verze tohoto vývojového prostředí nese název Visual Studio 2015. Kromě jiného se změnila i licence, pod kterou je Visual Studio dodáváno. U předchozí verze, Visual Studio 2013, bylo možné dostat takzvané Express edice zdarma. Bylo nutné ovšem mít samostatnou Express edici pro vývoj webových, desktopových a mobilních aplikací (Halvorson, 2015). U nejaktuálnější verze nabízí Microsoft zdarma takzvanou Community

edici, která již všechny tyto moduly obsahuje a není tedy nutné instalovat více programů, pokud chceme vyvíjet aplikace jak pro Windows Mobile, tak desktopové Windows. Novinkou je též možnost vyvíjet takzvané univerzální aplikace pro Windows 10, které musí být schopny přizpůsobit se různým velikostem obrazovky na různých typech zařízení s Windows 10 (mobilní telefon, počítač, tablet, Xbox atd.). Dále je možné pořídit si placené licence, které nesou název Profesional a Enterprise, které nabízejí pokročilé možnosti, jako jsou například nástroje pro spolupráci v týmu (VisualStudio, 2016).

4.2 Vědomostní kvíz

Vědomostní kvíz je výuková hra pro dva žáky, založená na principu známé televizní soutěže AZ – kvíz. Herní pole je složeno z 28 šestiúhelníků. Pod každým z těchto šestiúhelníků se skrývá náhodně vybraná otázka. Pokud se žákovi podaří na otázku odpovědět správně, pole získává. Hra končí v tu chvíli, kdy se podaří žákovi propojit svými navzájem se dotýkajícími poli všechny tři stěny hracího pole.

Žák je motivován k opakování si obsažených témat nejen samotnou hrou, ale velice důležitým faktorem je i soutěživost mezi žáky, kteří mezi sebou soupeří o vítězství ve hře. K tomu, aby žák mohl zvítězit, musí mít přehled v daném předmětu a znát co nejvíce odpovědí.

4.2.1 Popis programu Kvíz

Celý program se skládá celkem z pěti formulářů, databáze s otázkami a pomocného modulu. Z tohoto jsou běžnému uživateli přístupny pouze 4 formuláře, a to:

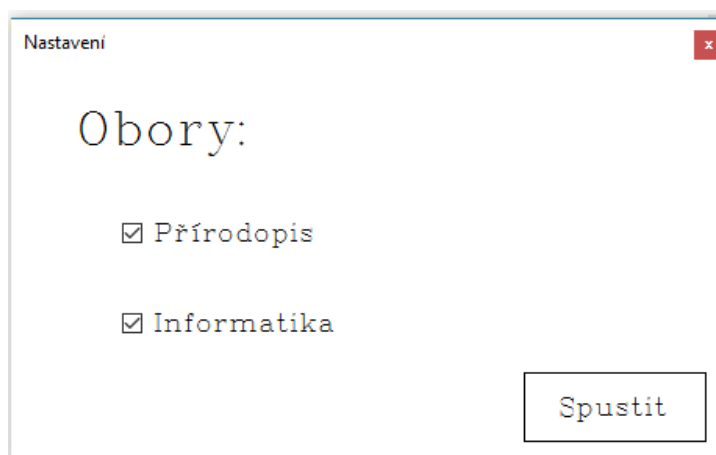
- úvodní formulář,
- nastavení,
- kvíz,
- otázka.

Úvodní formulář se spustí jako první po spuštění programu. Jeho jediným účelem je spuštění dalšího formuláře.



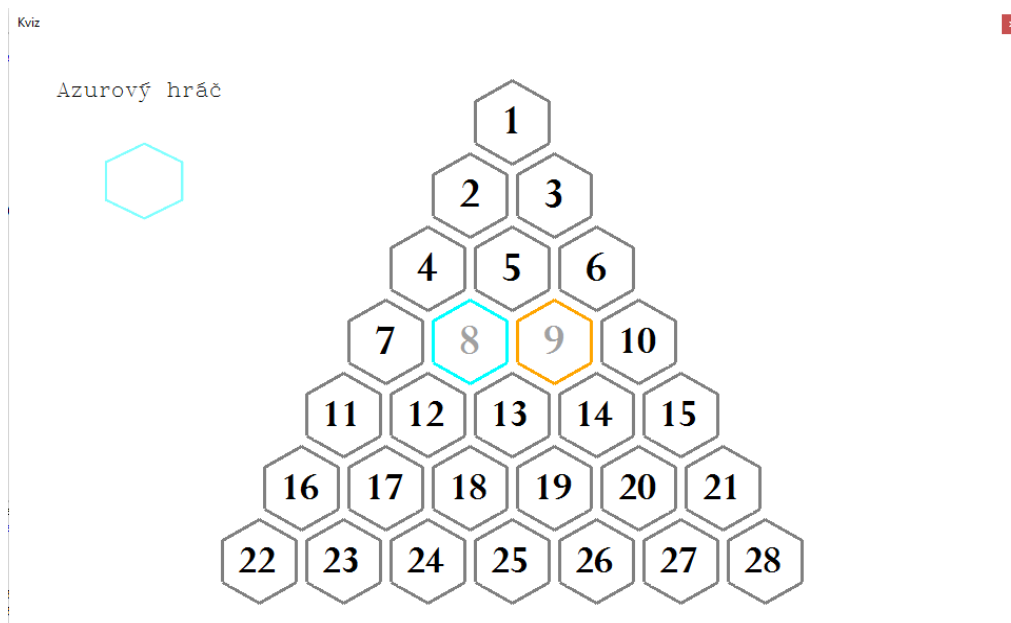
Obrázek 4 Úvodní formulář

Po stisknutí tlačítka určeného ke spuštění kvízu se zobrazí druhý formulář s možnostmi nastavení. Na tomto formuláři si uživatelé mohou zvolit, ze kterých oborů budou vybírány otázky pod políčky. Uživatel musí vždy zvolit nejméně jeden, ale může samozřejmě i více, z nabízených oborů, jinak nebude vpuštěn dále. Po vybrání oboru otázek, lze tlačítkem spustit kvíz.



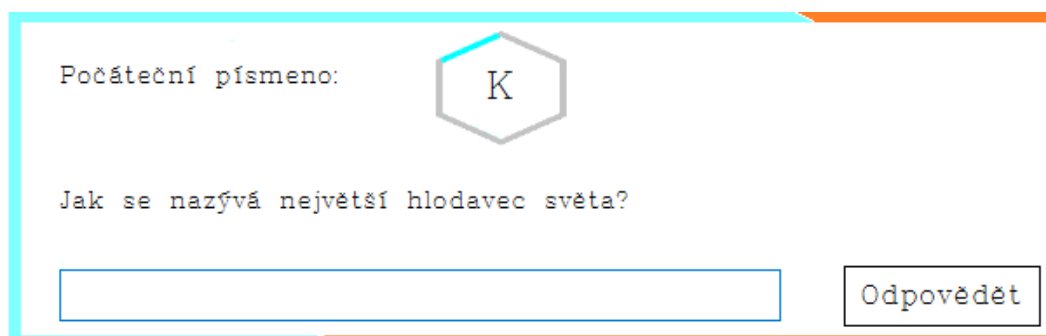
Obrázek 5 Nastavení

Na formuláři s kvízem je kromě indikátoru, který hráč je momentálně na řadě, dominujícím prvkem právě hrací pole. Hrací pole se skládá celkem z 28 labelů, ve kterých jsou vypsána čísla od 1 do 28. Tyto labely jsou obkresleny při spuštění černými šestiúhelníky. Barva těchto šestiúhelníků se poté mění podle toho, který z hráčů, z dvojice azurový a oranžový, si dané pole zabere. Po zabrání pole hráčem již není možné na něj znovu kliknout. Formulář s vybranou otázkou se zobrazí po kliknutí na label vybraného pole.



Obrázek 6 Kvíz

Na posledním formuláři se nachází textbox, do kterého hráči budou vpisovat své odpovědi. Nad textboxem je umístěno náhodně vybrané zadání otázky. Počáteční písmeno očekávané odpovědi je zobrazeno nad zadáním otázky a je obklopeno časomírou. Každý z hráčů má na svou odpověď přesně 30 sekund. Pokud hráč neodpoví do této doby, pokus propadne a na tahu bude druhý hráč. Odpočet času je vyobrazen pomocí šestiúhelníku, který postupně šedne. Zešednutí celého šestiúhelníku značí vypršení času.



Obrázek 7 Otázka

Otázky i s odpověďmi jsou uloženy v externí databázi, do které má možnost přidávat otázky i učitel, který nemá programovací dovednost. Otázky je možné vpisovat do tabulek databáze pomocí jakéhokoliv programu, který je schopný upravovat databáze. Příkladem takového programu je i Access s balíku Microsoft Office. Databáze je spojena s programem díky pomocnému formuláři, který je běžným uživatelům skryt, přes prvek datagridview.

Program nejprve pomocí zvolení náhodného čísla určí, z které tabulky se bude vybírat otázka. Po vybrání tabulky se zjistí počet otázek, který je v dané tabulce uveden. Z nich

je následně vybrána jedna, která bude načtena do formuláře, včetně pomocného prvního písmena. Správná odpověď je uložena do pomocné proměnné, která bude sloužit pro pozdější kontrolu hráčovy odpovědi. Aby nedošlo k opakování stejných otázek během jedné hry, jsou řádky obsahující zvolenou otázku dočasně vymazány z databáze.

```

Sub nacteni(data As Object)
  Randomize()
  pocetradku = data.RowCount - 1
  nahodnyradek = (Rnd() * (pocetradku - 1))
  Label1.Text = CStr(data(2, nahodnyradek).Value)
  Label2.Text = CStr(data(1, nahodnyradek).Value)
  a = CStr(data(3, nahodnyradek).Value)
  data.Rows.Remove(data.rows(nahodnyradek))
End Sub

```

'Určení počtu otázek
'Vybrání náhodné otázky
'Načtení otázky do labelu
'Načtení odpovědi do proměnné
'Dočasně odstranění otázky

Obrázek 8 Výběr otázky

Po každém správném zodpovězení otázky dojde k vyhodnocení programem, zda hráč, který je momentálně na tahu, již dosáhl všech třech stran hracího pole.

Pokud již hráč má obsazeny pole u všech stran, začne program vyhodnocovat, zda došlo k propojení mezi těmito stranami. Jako první začne vyhledávat cestu mezi levou a pravou stranou. V tomto případě prohledává algoritmus pole primárně sousedící na pravé straně od prvního bodu na levé straně. Pokud nalezne sousední pole se stejnou barvou, přesune se ukazatel na něj a algoritmus začíná od počátku.

```

If over <> 1 Or over <> 22 Or over <> 23 Or over <> 24 Or over <> 25 Or over <> 26 Or over <> 27 Or over <> 28 Then
  D = stupen(over)

  If barvy(over + 1) = vys Then
    over = over + 1
    If over = P Then
      LP = 1
      Exit Sub
    End If
    Call hledejLP()

  ElseIf barvy(over - D + 1) = vys Then
    over = over - D + 1
    If over = P Then
      LP = 1
      Exit Sub
    End If
    Call hledejLP()

  ElseIf barvy(over + D + 1) = vys Then
    over = over + D + 1
    If over = P Then
      LP = 1
      Exit Sub
    End If
    Call hledejLP()

  ElseIf barvy(over + D) = vys Then
    over = over + D
    If over = P Then
      LP = 1
      Exit Sub
    End If
    Call hledejLP()
End If

```

Obrázek 9 Ukázka hledání sousedních pozic

Nastane-li situace, že algoritmus nalezne spojení mezi levou a pravou stranou, bude vyvolána obdobná procedura, která tentokrát bude hledat spojení mezi pravou a spodní stěnou. Zde budou tedy upřednostňováni sousedící pole, která jsou položena směrem ke spodní straně hracího pole.

Pokud i zde nalezne program spojení, bude hráč, který je momentálně na tahu, vyhodnocen jako vítěz a hra bude ukončena.

4.2.2 Možnosti modifikace

Přidání nové otázky, z již obsažených předmětů, je velice snadné. Stačí k tomu uživatelské znalosti práce s programem Microsoft Access. Pro přidání otázky stačí otevřít tabulku s příslušným předmětem a přidat otázku vyplněním nového řádku.

Rozšíření na další předměty je již mírně obtížnější. Jako první musí vzniknout v databázi nová tabulka, která bude reprezentovat nově přidávaný předmět. Samozřejmostí je naplnění tabulky otázkami z okruhu tohoto předmětu. Jako další krok musí být nově vytvořená tabulka propojená s programem pomocí prvku datagridview.

Po propojení je nutné ještě zajistit, aby program začal vybírat otázky i z nové tabulky. K tomu je nutné dát uživateli možnost si tabulku vybrat, což zajistíme přidáním checkboxu na formulář s nastavením a k němu navázat příslušný kód, který zajistí výběr otázek beroucí v potaz vybrané okruhy.

```
If informatika = 1 And prirodopis = 1 Then           'Aktivní checkbox s oběma předměty
  Randomize()
  c = (Rnd() * (1))                                'Rozlosování zda bude vybraná otázka z přírodopisu či informatiky
  If c = 0 Then
    nacteni(Form4.DataGridView2)
  Else
    nacteni(Form4.DataGridView1)
  End If
Else
  If prirodopis = 1 Then                             'Aktivní checkbox s přírodopisem
    nacteni(Form4.DataGridView1)                    'Vybrání otázky z tabulky přírodopis
  End If

  If informatika = 1 Then                             'Aktivní checkbox s informatikou
    nacteni(Form4.DataGridView2)                    'Vybrání otázky z tabulky informatika
  End If
End If
```

Obrázek 10 Vybrání tabulky

Tímto způsobem je možné přizpůsobit si otázkami program pro využití v libovolném předmětu.

4.2.3 Charakteristika programu Kvíz podle kategorizace J. Dostála (2011)

Podle kategorií vytvořených J. Dostálem (2011) můžeme program označit jako:

- *interaktivní,*

V tomto případě se jedná o interaktivní program. Žáci totiž mají možnost zasahovat do chodu programu, a to několika způsoby. Zaprvé hned u volby okruhu otázek a zadruhé si sami volí strategii hry.

- *určený pro základní školy,*

Otázky, které jsou nyní v databázi připraveny, jsou určeny především pro užití na druhém stupni základních škol. Ovšem obměnou otázek lze kvíz přizpůsobit pro libovolnou úroveň vzdělávání. Problém může nastat pouze u školy mateřské, kde děti většinou neumí číst a byla by například potřeba integrovat předčítání.

- *zpětnovazební,*

Žák dostává zpětnou vazbu okamžitě po zodpovězení otázky. Vidí, zda otázku zodpověděl správně, či nikoliv. Pokud byla otázka zodpovězená nesprávně, bude zobrazená správná odpověď.

- *plná verze,*

Svůj program poskytujeme pouze v plné verzi, nikoliv v demoverzi.

- *offline,*

Kvíz pracuje na lokálním počítači a není možné ho umístit na Internet. Taktéž k programu neposkytujeme on-line podporu.

- *bez předmětového zaměření,*

Kvíz nemá pevné zaměření na jeden školní předmět. V současné době jsou vybrány otázky z přírodopisu a informatiky.

- *polytematický,*

Kvíz svými otázkami zasahuje do různých témat. Ovšem je možné, aby byl učitelem změněn na monotematický, a to pouhou změnou otázek v databázi.

- *vizuální,*

Kvíz působí na žáka pouze prostřednictvím obrazu. Zvuk není prozatím použit. Bylo by možné pokládat i zvukové otázky, například při zaměření programu na hlasy pěvců. Ovšem otázky by nemohly být uloženy v externí databázi, ale například by mohly být nahrány v resource souborech.

- *jednojazyčný,*

Celé uživatelské prostředí i otázky obsahují prozatím pouze jednu jazykovou mutaci. Je samozřejmě možné zadávat otázky i v jiných jazycích, než je právě jazyk mateřský.

- *pro samostudium,*

Program je spíše vhodný pro samostatnou práci žáků. Kdy si žáci zábavnou formou opakují své znalosti z různých oblastí.

- *víceuživatelský,*

Hra je určena pouze pro dva hráče. Není v současné době možné, aby hrál hráč sám. Tuto možnost by bylo možné realizovat doděláním automatických tahů počítače, kdy by si počítač náhodně volil pole a náhodně by se volilo, zda by na otázku odpověděl, či nikoliv.

- *didakticky polyfunkční.*

Program se zaměřuje jednak na motivaci žáka, ale také na fixaci již dříve osvojených poznatků

4.3 Skládání obrázků

Hra skládání obrázků vznikla na motiv populární dětské hry pro jednoho hráče. Účel této hry spočívá v tom, že se obrázek rozložil na 9 částí, z nichž je jedna část odstraněna. Zbýlých 8 částí je promícháno a hráč má za úkol složit obrázek do původní podoby, kdy může využívat k prohazování částí pouze sousedící volné místo, které vzniklo díky odstranění jedné z částí obrázku.

Samotný program je cílený na opakování vzhledu organismů, konkrétně z řádu pěvců. Žák si při skládání obrázku detailně prohlédne vzhled jednotlivých pěvců. Navíc, aby mohl být žákův čas zapsán do tabulky vítězů, musí být zodpovězeno, jaký pěvec se na obrázku vyskytuje.

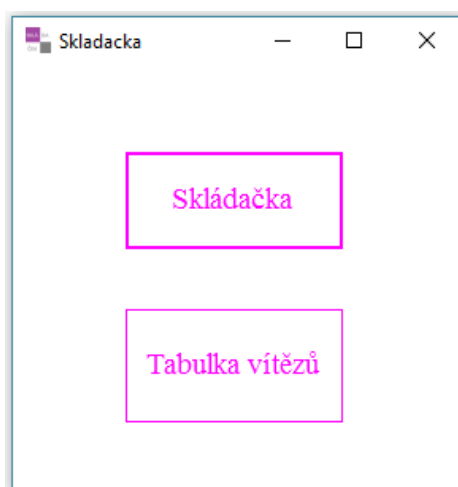
Žák je tímto programem motivován k učení rozeznávání organismů pomocí jednoduché hry. Jako další motivační prvek je vložena tabulka vítězů, která působí na žákovu potřebu podat co možná nejlepší výkon. Žák musí nejdříve složit obrázek v co nejkratším čase a po složení zodpovědět otázku.

4.3.1 Popis programu Skládání obrázků

Program skládání obrázků se skládá pouze ze 3 formulářů a ke všem má uživatel přístup. Program se skládá z těchto formulářů:

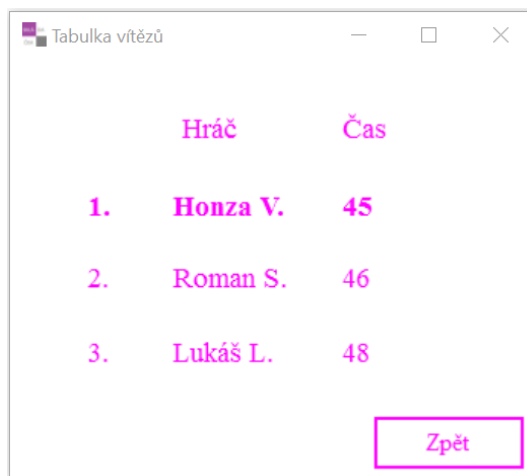
- úvodní formulář,
- tabulka vítězů,
- skládání obrázků.

Po spuštění programu bude zobrazena úvodní obrazovka, která slouží pouze jako rozcestník mezi ostatními formuláři. Máme tedy možnost z tohoto místa přejít do tabulky vítězů, anebo přejít rovnou k samotné hře.



Obrázek 11 Úvodní formulář

Na formuláři s tabulkou vítězů, jsou uvedeny 3 nejlepší hráči. Každý hráč je uložen pod jménem, které si sám zvolí a bude u něj uložen i jeho čas. Časem se myslí čas, za který se hráči podařilo složit obrázek a zároveň správně odpovědět na otázku, který živočich se na obrázku ukrývá. Čas je uveden v sekundách.



Obrázek 12 Tabulka vítězů

Informace o třech nejlepších hráčích, jsou uloženy v externím textovém souboru. Tento soubor je po spuštění programu nahrán do pomocných proměnných typů pole. Tento krok je nutný provést z důvodu práce s těmito údaji. Po úspěšném dokončení hry je nutné porovnat dosažené časy a v případě překonání některého z vítězných časů, musí dojít k záměně pořadí. Ta je realizována vložením aktuálního času na jeho pozici a posunutím všech hráčů, kteří se ocitnou za aktuálním časem. Hráč, který se svým časem posune na 4. pozici, bude zapomenut a již se v tabulce neobjeví. Po obměnění pozic, je nové pořadí opět zapsáno do textového souboru.

```

Sub vyhodnoceni(tvuj As Integer)

    If cas(0) >= tvuj Then                                     'Ujistění nového času, pokud je aktuální nejlepší
        vitez(2) = vitez(1)
        vitez(1) = vitez(0)

        cas(2) = cas(1)
        cas(1) = cas(0)

        cas(0) = tvuj
        vitez(0) = jmeno
        prepis()
    ElseIf cas(1) >= tvuj Then                                'Volání procedury na přepsání textového souboru
        vitez(2) = vitez(1)                                  'Ujistění nového času, pokud je aktuální druhý nejlepší
        cas(2) = cas(1)

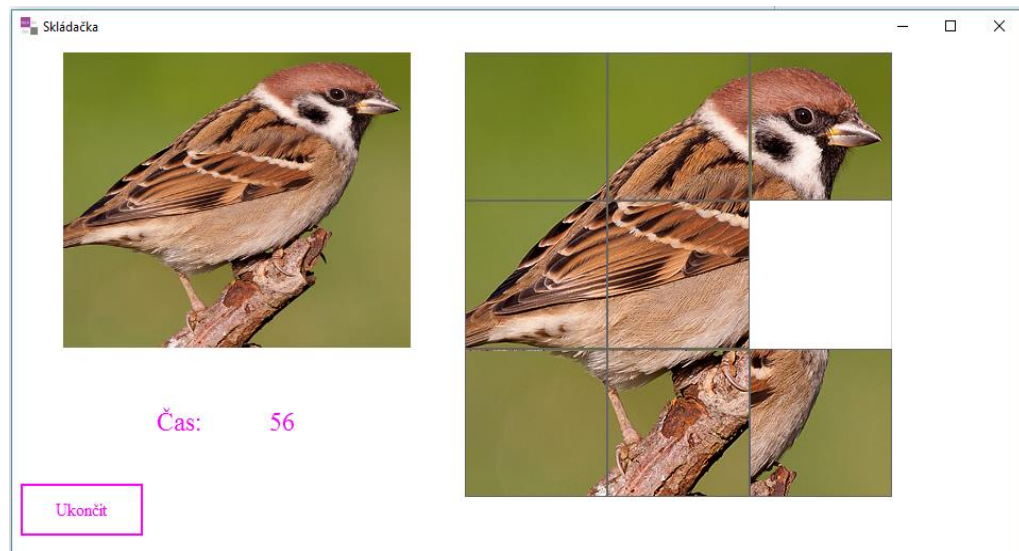
        cas(1) = tvuj
        vitez(1) = jmeno
        prepis()
    ElseIf cas(2) >= tvuj Then                                'Ujistění nového času, pokud je aktuální třetí nejlepší
        cas(2) = tvuj
        vitez(2) = jmeno
        prepis()
    End If
    Form3.Show()
End Sub

Sub prepis()
    FileOpen(soubor, "vitezove.txt", OpenMode.Output)       'Načtení textového souboru
    For i As Integer = 0 To 2
        WriteLine(soubor, vitez(i))                         'Zapsání přezdívky souboru
        WriteLine(soubor, cas(i))                           'Zapsání času do souboru
    Next
    FileClose(soubor)
End Sub

```

Obrázek 13 Práce s tabulkou vítězů

Na posledním formuláři se již nachází samotná hra skládání obrázků. Nachází se zde samotné hrací pole s náhodně vybraným rozloženým obrázkem. Fragmenty obrázku jsou v hracím poli samozřejmě promíchány náhodně. Předloha, ve formě původního nerozloženého obrázku, a to včetně části, která na rozloženém obrázku chybí, se nachází na levé straně formuláře. Dále je zde časomíra, na které si žák může kontrolovat svůj výkon. Čas je tu opět zobrazován pouze v sekundách a je spuštěn při prvním posunutí obrázku na hracím poli.



Obrázek 14 Formulář se hrou

Obrázek v hracím poli je rozložen na 8 částí a 9. pole je prázdné. Úkolem uživatele je poskládat obrázek do původní podoby právě pomocí prázdného pole. Vždy po kliknutí na část obrázku, která sousedí s prázdným polem, dojde automaticky k přesunutí na pole prázdné. Prázdné pole je u složeného obrázku zpravidla na dolní pravé straně.

```

Sub prubeh(box As Object)
lokace = box.Location
Call kontrola()
box.Location = lokace
Call konec()
End Sub
Sub kontrola()
Timer1.Start()
'Kontrola, zda je v okolí obrázku prázdné místo. Pokud ano dojde k přehození lokace prázdného místa s obrázkem, na který bylo kliknuto.
If (lokace.X - 134 = prazdne.X And lokace.Y = prazdne.Y) Or (lokace.X + 134 = prazdne.X And lokace.Y = prazdne.Y) Or (lokace.X = prazdne.X And lokace.Y + 134 = prazdne.Y)
pomocne = prazdne
prazdne = lokace
lokace = pomocne
End If
End Sub

```

Obrázek 15 Průběh prohození částí obrázku

Po úspěšném složení obrázku, které program sám rozpozná, bude muset uživatel splnit ještě jednu podmínku, aby mu bylo umožněno zapsání do tabulky vítězů. Podmínkou je rozpoznání a napsání správného rodového jména organismu, který se nachází na skládaném obrázku. To znamená, že ani sebelepší čas nezaručuje zapsání do tabulky, pokud žák nerozpozná organismus, který skládal dohromady.

4.3.2 Možnosti modifikace

Program lze upravit samozřejmě změnou obrázků, které jsou v něm uvedeny. Provedením této úpravy můžeme docílit jednak rozšíření oboru z ptáků na celé obratlovce, tak i rozšíření na jiné předměty. Zde jsou příklady, s jakým typem obrázků můžeme rozšířit program na jiné předměty:

- matematika,

V programu lze obrázky upravit například na výuku násobků, použitím obrázků, které budou obsahovat násobky některého čísla a žák je bude mít za úkol seřadit od nejmenšího po největší.

- dějepis,

Pro výuku dějepisu je například možné naplnit pole určené na skládání obrázků místo jednoho rozloženého obrázku, obrázky 8 našich prezidentů, které bude mít žák za úkol chronologicky seřadit.

- český jazyk.

Pro výuku českého jazyka, konkrétně literatury, může být program naplněn fotografiemi spisovatelů. Po poskládání podoby spisovatele, by žák musel vypsát alespoň jedno jeho dílo.

Přidání nového obrázku je trochu zkomplikované tou skutečností, že obrázek musí být před přidáním upraven. Konkrétně je nutné ho rozdělit na 9 stejně velkých dílů, z nichž 8 bude nahráno.

Připravené obrázky je vhodné nejprve nahrát do resource souborů a z nich teprve k obrázkům přistupovat. Po přidání nového obrázku je nutné, ve formuláři se hrou, vytvořit novou větev s tímto obrázkem, aby mohl být programem vybrán. Při nahrávání do pictureboxů je třeba dodržovat pořadí nahrávaných obrázků. Levá horní část musí být v 1. pictureboxu a poté se pokračuje směrem doleva a dolů. Poslední obrázek tedy prostřední vespod bude nahrán v 8. pictureboxu. Samozřejmě je nutné i rozšířit množinu čísel, ze kterých bude vybíráno náhodné číslo tak, aby odpovídala počtu obrázků.

```

Randomize()
nahoda = Rnd() * (4)
Select Case nahoda
    Case 0
        PictureBox1.Image = My.Resources.Resource1.jedna0
        PictureBox2.Image = My.Resources.Resource1.dva0
        PictureBox3.Image = My.Resources.Resource1.tri0
        PictureBox4.Image = My.Resources.Resource1.ctyri0
        PictureBox5.Image = My.Resources.Resource1.pet0
        PictureBox6.Image = My.Resources.Resource1.sest0
        PictureBox7.Image = My.Resources.Resource1.sedm0
        PictureBox8.Image = My.Resources.Resource1.osm0
        zodpoved = "pěnkava"
        PictureBox9.Image = My.Resources.Resource1.cely0
    'Výběr náhodného obrázku
    'Nahrání levé vrchní části
    'Nahrání prostřední vrchní části
    'Nahrání pravé vrchní části
    'Nahrání levé prostření části
    'Nahrání prostření prostření části
    'Nahrání pravé prostření části
    'Nahrání levé spodní části
    'Nahrání pravé spodní části
    'Správná odpověď
    'Nápověda ve formě nerozloženého obrázku

```

Obrázek 16 Přidání nového obrázku

Tímto postupem je možné program rozšířit například o námi výše navrhované možnosti.

4.3.3 Charakteristika programu Skládání obrázků podle kategorizace J. Dostála (2011)

Výukový program Skládání obrázků, je možné charakterizovat pomocí kategorií J. Dostála (2011) takto:

- *interaktivní,*

Program se dá označit za interaktivní díky samotné hře, kterou uživatel ovládá pomocí kliků a ta se podle toho chová.

- *určený pro základní školy,*

Program je v současné době naplněný obrázky živočichů, které se hodí pro opakování na základní škole. Samozřejmě je možné program vhodnou volbou obrázků přizpůsobit například i pro mateřské školy. Ovšem bylo by zároveň vhodné předělat i způsob dotázání po dokončení skládání.

- *zpětnovazební,*

Zpětnou vazbu uživatel dostane po složení obrázku a zodpovězení otázky. Ihned po zodpovězení se dozví, zda odpověděl správně, či nikoliv. Pokud žák zodpoví otázku špatně, bude mu ihned zobrazená správná odpověď.

- *plná verze,*

I tento program je poskytován pouze v plné verzi. Demoverze k dispozici není.

- *offline,*

Hru skládání obrázku je možné mít nainstalovanou pouze na lokálním disku, případně na síťovém disku. K našim programům neposkytujeme ani žádnou formu online podpory.

- *předmětově zaměřený,*

V současné době je program zaměřen na přírodopis. Toto zaměření vyplývá z obrázků, kterými je program naplněn. Všechny se týkají rozpoznávání organismů, které samozřejmě patří do přírodopisu. Úpravě i na jiné předměty jsme se již věnovali výše.

- *monotematický,*

Všechny obrázky, které musí uživatel poskládat a zároveň určit jaký organismus se na nich vyskytuje, jsou pouze z oblasti ptáků. Na celý předmět přírodopis, lze program rozšířit přidáním i organismů z jiných říší, jako jsou rostliny a houby.

- *vizuální,*

Hra skládání obrázků je založena pouze na vizuálním působení na žáka. Už ze své podstaty se zde vždy bude primárně jednat o skládání obrázku.

- *jednojazyčný,*

Celé uživatelské prostředí obsahuje pouze jednu jazykovou mutaci. Pro opakování jazyků, by bylo možné například požadovat po uživateli odpověď po poskládání obrázku v cizím jazyku.

- *pro samostudium,*

Program je vhodný pro samostatné studium, kdy si uživatelé opakují podobu organismů.

- *monouživatelský,*

Hra je určena pouze pro jednoho uživatele. Jediný prvek, díky kterému přijde uživatel do interakce s jinými hráči, je tabulka vítězů, ve které se mohou uživatelé snažit mezi sebou soutěžit o nejlepší čas.

- *didakticky polyfunkční.*

Program je zaměřen na fixaci dříve nabytých vědomostí a rozeznávání ptáků. Kromě toho s sebou samozřejmě nese i funkci motivační.

4.4 Pexeso

Pexeso je velice známá stolní hra, v původní podobě vyhrazená pro 2 a více hráčů, která je určena hlavně pro děti. Ke hře je potřeba sada hracích karet, které mají z jedné strany různé obrázky a z druhé, rubové strany, jsou u všech hracích karet obrázky totožné. Hra probíhá tak, že se karty pečlivě promíchají a rozloží se po pracovní ploše rubovou stranou směrem nahoru. Hráči se poté střídají v otáčení karet. Pokud hráč nalezne dvojici stejných obrázků, započítává si bod. Vyhrává hráč s větším počtem nalezených dvojic.

Žáci jsou motivováni k učení pomocí hry, která má za účel je nejen pobavit, ale i nenásilně něčemu přiučit. Dalším motivačním faktorem je implementace hry pro 2 hráče, kdy žáka pohání k lepším výkonům touha být lepší.

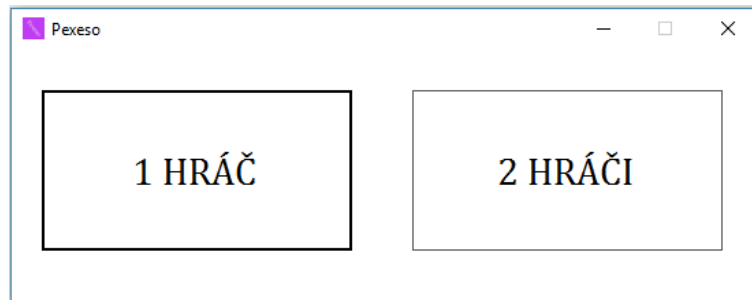
4.4.1 Popis programu

Uživatel má v programu pexeso přístup celkem ke 3 formulářům. Program obsahuje tyto formuláře:

- výběr počtu hráčů,

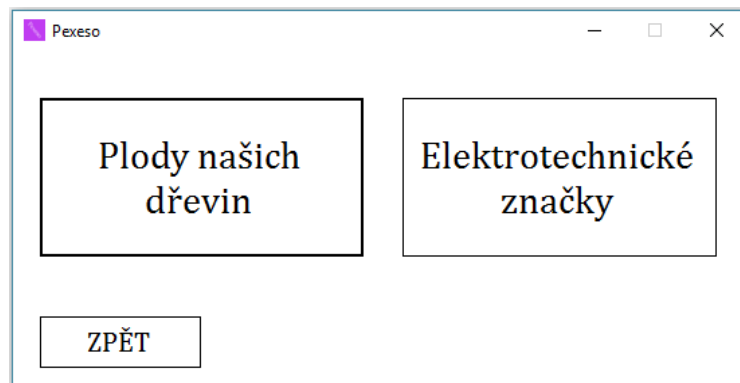
- výběr verze pexesa,
- pexeso.

Po spuštění programu bude zobrazen první formulář s výběrem počtu hráčů. Pexeso v tomto programu může hrát buďto jeden hráč, nebo hráči dva, kteří se střídají po jednotlivých tazích.



Obrázek 17 Výběr počtu hráčů

Po vybrání, kolik hráčů se zúčastní hry, bude uživateli zobrazen formulář, na kterém si musí vybrat, jakou verzi pexesa si chce zahrát. V současné době jsou v programu přítomné 2 verze. Verze „Plody našich dřevin“ je zaměřená na předmět přírodopis, kde se o našich dřevinách vyučuje. Cílem v této verze není nalézt dva stejné obrázky, nýbrž nalézt k sobě název dřeviny a fotografii jejího plodu. Druhá verze, která se nazývá „Elektrotechnické značky“, se zaměřuje na výuku elektrotechnických značek. Cílem této verze je přiřadit dohromady elektrotechnickou značku s reálnou fotografií dané součástky.



Obrázek 18 Výběr verze

Na posledním formuláři se již nachází samotná hra pexeso. Hrací pole obsahuje celkem 16 hracích karet. Pod každou kartou se skrývá jeden obrázek. Obrázky jsou vždy před začátkem hry náhodně promíchány, aby se zabránilo tomu, že by se každou hru vyskytovaly ty samé obrázky na totožném místě. Karty se otáčejí prostým kliknutím na jejich rub. Pokud uživatel otočí 2 karty, které k sobě patří, zůstanou již otočené a hráči se započítá bod.

V případě, kdy hráč otočí 2 karty, které k sobě nepatří, karty se automaticky po 2 sekundách opět otočí.

V případě hry 2 uživatelů musí samozřejmě docházet k jejich střídání. Hráči jsou vystřídáni vždy, po neúspěšném tahu. Pokud při tahu dojde ke spojení karet, které k sobě patří, k výměně hráčů nedojde.



Obrázek 19 Pexeso

4.4.2 Možnosti modifikace

Program je samozřejmě možné rozšířit pomocí přidání pexes, které jsou zaměřeny na jiné oblasti, či předměty. Pexeso může být přizpůsobeno například pro následující předměty:

- matematika,

V matici může být pexeso využito tím způsobem, že by k sobě patřily obrázky, kde by na jednom obrázku bylo zadání příkladu a na obrázku druhém výsledek.

- informatika,

Ve výuce informatiky lze pexeso například využít tak, že žák by k sobě hledal obrázky sběrnic a zařízení, které pomocí této sběrnice můžeme připojit.

- chemie.

Chemie je dalším příkladem předmětu, kde je možné pexeso využít. Možností může být k sobě vyhledávat chemické vzorce a názvy sloučenin.

Pro přidání nové verze pexesa je nutné samozřejmě si předpřipravít dvojice obrázků a ty nahrát do resource souborů. Aby bylo možné novou verzi pexesa si vybrat, je zapotřebí

na druhý formulář přidat tlačítko zastupující tuto verzi. Toto tlačítko bude mít za úkol spustit formulář s pexesem a nastaví proměnou „vyber“ na hodnotu odpovídající nové verzi. Poté již schází pouze rozšířit nahrávání obrázků o novou větev, ve které budou nahrávány obrázky nové verze pexesa.

```

If vyber = 1 Then                                'Rozcestník podle verze pexesa

    Select Case obr
    Case 0
        If prepnuti(0) = 0 Then
            obraz = My.Resources.Resource1.buk    'Přidání prvního obrázku z dvojice
            prepnuti(0) = 1
        Else
            obraz = My.Resources.Resource1.buk1   'Přidání druhého obrázku z dvojice
        End If

    Case 1
        If prepnuti(1) = 0 Then
            obraz = My.Resources.Resource1.dub    'Přidání prvního obrázku z dvojice
            prepnuti(1) = 1
        Else
            obraz = My.Resources.Resource1.dub1   'Přidání druhého obrázku z dvojice
        End If
    
```

Obrázek 20 Nahrání nové verze pexesa

Na obrázku lze vidět na příkladu 2 dvojic obrázků, jak je nutné postupovat, při zadávání nové verze pexesa. Pexeso je přizpůsobeno celkem na 8 dvojic obrázků a tento počet by měl být dodržován.

4.4.3 Charakteristika programu Pexeso podle kategorizace J. Dostála (2011)

Podle kategorií vytvořených J. Dostálem (2011), můžeme výukový program Pexeso označit jako:

- *interaktivní,*

Program může být označen jako interaktivní. Zaprvé si žáci mohou zvolit verzi hry. Samotná hra je samozřejmě také interaktivní. Hráč ji ovládá pomocí kliků, na které hra okamžitě reaguje otáčením karet.

- *určený pro základní školy,*

Program je v současné době svým obsahem zaměřený především na základní školu. Svou povahou by ovšem mohl být velice vhodný i pro mateřskou školu. Samozřejmě by muselo dojít k přidání verze pexesa, která by obsahovala takové obrázky, které by byly přínosné právě pro tento stupeň školství.

- *zpětnovazební,*

Žák dostává zpětnou vazbu pomocí hracích karet. Například ve verzi „Plody našich dřevin“, se žák dozví, jestli dobře přiřadil plod ke dřevině, díky tomu, že karty zůstanou obrácené a bude hráči započítán bod.

- *plná verze,*

Opět i u tohoto programu je poskytována pouze plná verze.

- *offline,*

I tuto hru je možné mít nainstalovanou pouze na lokálním disku. K našim programům neposkytujeme ani žádnou formu online podpory.

- *předmětově zaměřený,*

V současné podobě je program zaměřený na 2 konkrétní předměty. Těmito předměty je přírodopis a technické výchova. Přizpůsobením obrázků je možné program zaměřit i na jiné předměty.

- *polytematický,*

Verze pexesa, které jsou v programu obsaženy, jsou zaměřeny na více než jedno téma. Konkrétně je obsaženo téma elektrotechnické značky a plody dřevin.

- *vizuální,*

Program pexeso vychází z tradiční stolní hry, která je založena pouze na vizuálním vjemu. Proto nejsou ve hře obsažené zvuky. Bylo by možno udělat například pexeso na rozeznávání hlasů pěvců, kde by ve dvojici byl jeden obrázek nahrazen zvukovým souborem, který by obsahoval nahrávku hlasu pěvce.

- *jednojazyčný,*

Uživatelské prostředí i verze pexesa s plodinami dřevin, kde je dřevina zadána písemně, obsahuje pouze jednu jazykovou mutaci. Samotné pexeso by mohlo být vhodné i pro výuku jazyků. Objekt na obrázku by v tomto případě byl spojován s anglickým pojmenováním tohoto objektu.

- *pro samostudium,*

Program je vhodný především pro samostatnou práci žáků. Žáci si mohou formou hry zopakovat plody dřevin a elektrotechnické schématické značky.

- *monouživatelský i víceuživatelský,*

Hru může hrát uživatel jak sám, tak je možné ji hrát i ve dvojici. Mezi těmito volbami je možné si vybrat ještě před spuštěním hry. Pokud si uživatelé zvolí hru pro 2 hráče, budou se střídát po jednotlivých tazích.

- *didakticky polyfunkční.*

Program je určen pro opakování a fixaci učiva. Další funkcí je nepochybně motivace žáka.

4.5 Šibenice

Šibenice je hra, která je v klasické podobě, kdy se hraje pomocí papíru a tužky, určená pro 2 hráče. Jeden z hráčů zadává slova, která má druhý hráč za pomoci indicie uhádnout. Další hráč se snaží slovo uhádnout za pomoci zadávání písmen. Pokud se trefí do některého z písmen, které zadané slovo obsahuje, první hráč ho zapíše na jeho pozici ve slově. Naopak, není-li zadané písmeno ve slově obsaženo, je hráči započítán trestný bod. Trestné body jsou zde realizované kreslením šibenice s oběšencem, podle čeho také dostala tato hra název. Hra končí ve dvou případech. Prvním je uhádnutí zadaného slova, druhým případem je nakreslení celé šibenice i s oběšencem.

V mém zpracování je hráč, který zadává slova a kontroluje postup, nahrazen počítačem. Uživatel na sebe tedy bere roli hráče, který se snaží uhodnout zadané slovo.

Hráč je při této hře motivován nejen hrou samotnou, při které se snaží porazit počítač, ale potřebou soutěžit s ostatními. Tato potřeba nastává pomocí tabulky, do níž je ukládáno skóre nejlepších hráčů.

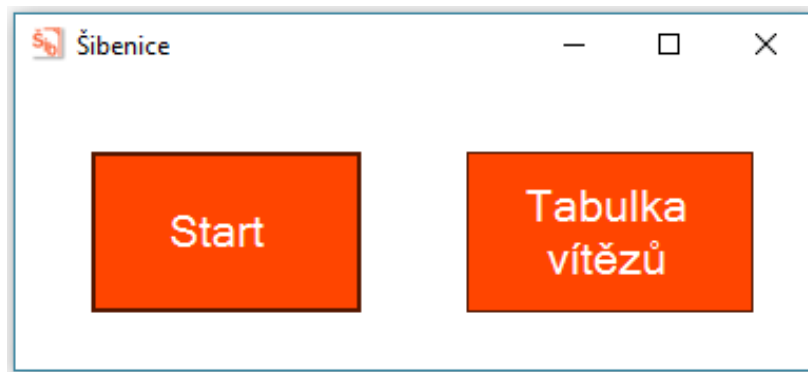
4.5.1 Popis programu

Program šibenice je zaměřen na opakování anglické slovní zásoby. Samotný program se skládá celkem ze 3 formulářů, ze kterých jsou všechny přístupné uživateli.

Tyto formuláře jsou:

- úvodní formulář,
- šibenice,
- tabulka vítězů.

Program začíná spuštěním úvodního formuláře, který slouží pouze jako rozcestník mezi dalšími prvky programu. Můžeme z tohoto místa spustit šibenici, anebo můžeme nahlédnout do tabulky vítězů.



Obrázek 21 Úvodní formulář

Tabulka vítězů je zpracována stejným způsobem, jako tomu je u programů Skládání obrázků a Slepá mapa. Program si tedy pamatuje první trojici nejvyšších dosažených bodů. Samozřejmě je ke každému skóre přiřazeno i jméno hráče, kterým ho bylo dosaženo. Všechny tyto informace jsou uloženy v externím textovém souboru, ze kterého jsou ihned po spuštění programu načteny.

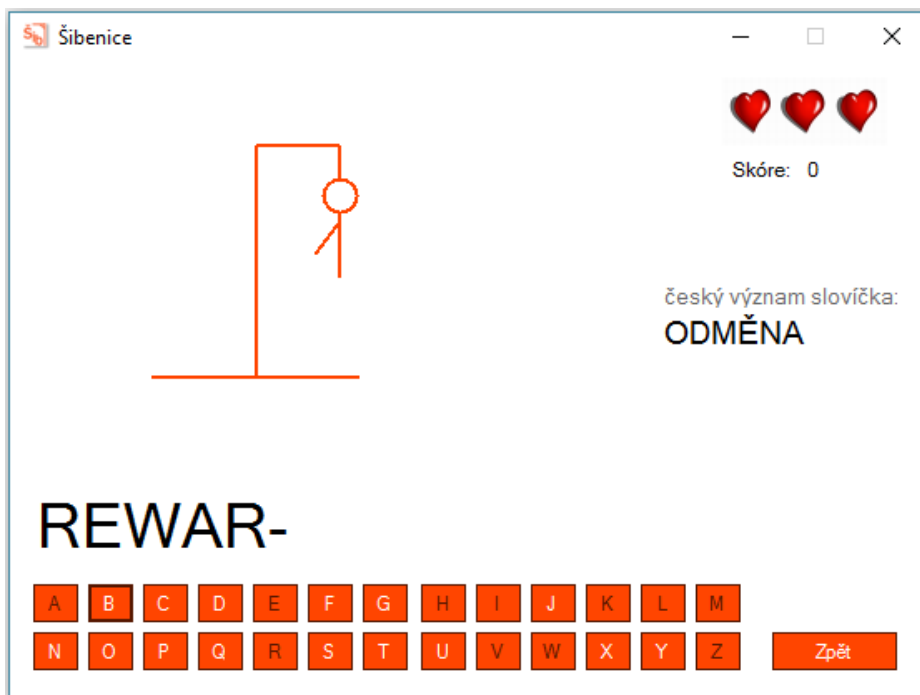


Obrázek 22 Tabulka vítězů

Na posledním formuláři je již zpracována samotná hra šibenice. Na formuláři se vždy nachází zadání českého ekvivalentu hledaného slova, label, který nese nápovědu ve formě počtu písmen, soubor tlačítek, kdy každé odpovídá jednomu písmenu abecedy a ukazatel počtu životů.

Label, nesoucí nápovědu o počtu, je po zadání nového slova naplněn takovým počtem pomlček, které odpovídá počtu písmen ve slově. Uživatel poté zadává písmena. Vždy, když uživatel zadá písmeno, které se ve slově nachází, je pomlčka na pozici písmene nahrazena oním písmenem. Při každém nesprávně zadaném písmenu je postupně malována

šibenice s oběšencem. Po sestavení celého slova, je přičteno 20 bodů k aktuálnímu skóre a pokračuje se dalším slovem.



Obrázek 23 Šibenice

Slova jsou do programu náhodně vybírána z externí databáze, která je s programem spojena přes prvek datagridview. Po vybrání je český ekvivalent nahrán přímo do pro něj určeného labelu a anglický ekvivalent do pomocné proměnné, která slouží k porovnávání zadaného písmena. Aby nedošlo k míchání velkých a malých znaků, jsou všechny prvky v řetězci převedeny na znaky velké.

```

Sub vyber()
  Randomize()
  vyb = CInt(Int(((Form1.DataGridView1.RowCount() - 1) * Rnd()))
  Form1.Label2.Text = CStr(Form1.DataGridView1(0, vyb).Value).ToUpper
  word = CStr(Form1.DataGridView1(1, vyb).Value).ToUpper
  Form1.Label1.Text = ""
  For i As Integer = 0 To word.Length - 1
    Form1.Label1.Text += "-"
  Next
End Sub

```

'Výběr náhodného řádku
'Český ekvivalent
'Anglický ekvivalent
'Vymazání labelu
'Nahrání pomlček do labelu

Obrázek 24 Nahrání vybraného slova

Při kontrole musí být porovnané se zadaným písmenem všechny znaky v řetězci. K tomu bylo použito cyklu For Each, který provede tělo cyklu tolikrát, kolik je znaků v kontrolním řetězci, což je v našem případě hledané slovo. Při každém průchodu je v pomocné proměnné uložen jeden prvek kontrolního řetězce, který je porovnán se zadaným písmenem. Pokud se písmena shodují, bude dosazeno na jeho pozici. Pokud shoda není nalezena, je na aktuální pozici navrácena pomlčka.

```

Sub vyber()
  Randomize()
  vyb = CInt(Int(((Form1.DataGridView1.RowCount() - 1) * Rnd()))))
  Form1.Label2.Text = CStr(Form1.DataGridView1(0, vyb).Value).ToUpper
  word = CStr(Form1.DataGridView1(1, vyb).Value).ToUpper
  Form1.Label1.Text = ""
  For i As Integer = 0 To word.Length - 1
    Form1.Label1.Text += "-"
  Next
End Sub

```

'Výběr náhodného řádku
'Český ekvivalent
'Anglický ekvivalent
'Vymazání labelu
'Nahrání pomlček do labelu

Obrázek 25 Kontrola řetězce

Po konci, situace, kdy dojdou hráči životy, je získané skóre porovnáno s těmi, které jsou v dané době zapsané v tabulce vítězů. Pokud skóre odpovídá svou hodnotou zápisu do tabulky, je vloženo na příslušné místo a zapsáno do externího souboru.

4.5.2 Možnosti modifikace

Výukový program je naplněn pouze anglickými slovy a jejich českými ekvivalenty. To znamená, že v současné době ho lze využít pouze k výuce anglického jazyka. Jednoduchou úpravou je možné zaměnit slova v databázi tak, aby odpovídala právě probírané látce. Databázi je možné upravovat přes Microsoft Access, kde se záměny provádí prostým přepisováním řádků.

Stejným principem je možné program i přizpůsobit pro využití v jiných předmětech. Samozřejmě je využití i ve výuce jiných jazyků, než pouze toho anglického. V tomto případě zůstává princip stejný a zaměníme pouze jazyk očekávaných slov.

V jiných, než jazykovědných předmětech, jako je například přírodopis, či dějepis, lze zaměnit nápoředu ve formě českého ekvivalentu za zadání otázky. Hádané slovo je poté odpověď na zadanou otázku. K tomuto kroku by bylo nutné přizpůsobit i tlačítka určená k zadávání znaků, která neobsahují českou diakritiku.

4.5.3 Charakteristika programu Šibenice podle kategorizace J. Dostála (2011)

Výukový program Šibenice lze podle kategorií J. Dostála (2011) charakterizovat takto:

- *interaktivní,*

Šibenice lze označit jako interaktivní z toho důvodu, že uživatel sám volí písmena z nabídky. Program na zvolené písmeno reaguje a podle toho, zda bylo zvolené správné písmeno, volí další postup.

- *určený pro základní školy,*

Databáze, ze které jsou vybírána slova do hry, je naplněna slovy, která odpovídají náplni výuky na základní škole. Rozšíření na vyšší stupně je možné použitím například odborných výrazů.

- *zpětnovazební,*

Uživatel dostane zpětnou vazbu ihned po stisknutí tlačítka, které představuje některé z písmen. Pokud je odpověď správná, bude písmeno zobrazeno na jeho pozici. Pokud není, bude vykreslena další část šibenice.

- *plná verze,*

Opět i u tohoto programu je poskytována pouze plná verze.

- *offline,*

I tuto hru je možné mít nainstalovanou pouze na lokálním disku. K našim programům neposkytujeme ani žádnou formu online podpory.

- *předmětově zaměřený,*

Program šibenice je zaměřený na opakování anglických slovíček, tudíž je pevně vázaný na předmět anglický jazyk. Jednoduchou a uživatelsky přívětivou úpravou databáze jej lze upravit na opakování slovní zásoby z jakéhokoliv jazyka.

- *polytematický,*

Slova obsažená v databázi nejsou zaměřena na jednu oblast, ale jsou vybrána z různých výukových oblastí.

- *vizuální,*

Program působí pouze vizuálně. Možnost, jak využít i auditivní složku, by vedla skrze nápovědu hledaného slova. Tato nápověda by mohla být ve formě zvuku reprezentující například nějaký předmět. Další možností by mohlo být zařazení nahrávky se správnou výslovností po uhodnutí slova.

- *vícejazyčný,*

Uživatelské prostředí včetně instrukcí a nápovědy je v českém jazyce. Druhý jazyk, v tomto případě angličtina, je vyžadována po uživateli jako odpověď.

- *pro samostudium,*

Program je vhodný především pro samostatnou práci žáků. Žáci si mohou formou hry procvičit a zopakovat slovní zásobu.

- *monouživatelský,*

Hra je určena pouze pro jednoho hráče. Jak již bylo řečeno, roli druhého hráče zde přebírá počítač. Víceuživatelský program by bylo možné vytvořit, pokud by hledané slovo zadával druhý hráč, nikoliv počítač.

- *didakticky polyfunkční.*

Výukový program má jak funkci opakovací a fixační, tak i funkci motivační.

4.6 Slepá mapa

Slepá mapa je způsob jak ověřit, zda se žák orientuje na mapě. Žák se snaží na mapě, na které jsou zobrazeny pouze hranice, případně jiné významné orientační body, zaznačit polohu, měst, řek, hor atd.

V našem programu je vložena formou minihry i hra had.

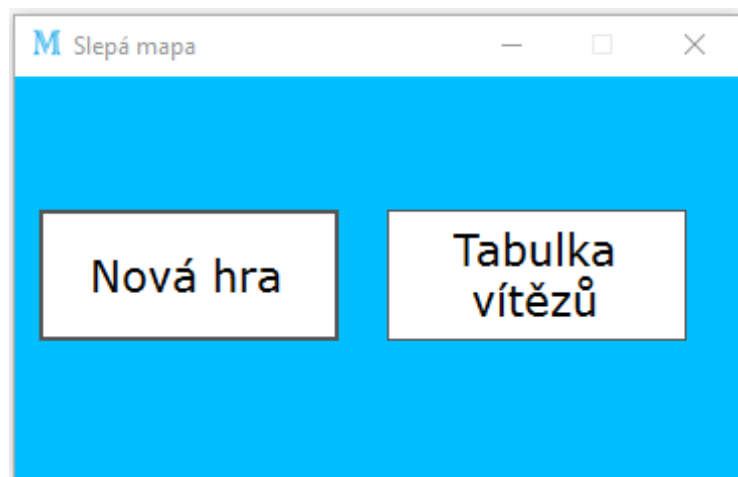
Had je jednou z klasických počítačových her. Jejím cílem je posbírat všechny body, které se vyskytují na obrazovce. Hráč se přitom musí vyhýbat různým překážkám a okraji hrací obrazovky. Při nárazu do jednoho z těchto objektů pro něj hra končí.

4.6.1 Popis programu Slepá mapa

Program slepá mapa je složena ze 4 formulářů, z nichž jsou všechny přístupné uživateli. Program obsahuje konkrétně tyto formuláře:

- úvodní formulář,
- tabulka vítězů,
- slepá mapa,
- had.

Po spuštění programu je zobrazen úvodní formulář, který slouží pouze jako rozcestník mezi dalšími formuláři. Z tohoto místa se můžeme dostat do tabulky vítězů, případně lze spustit přímo slepou mapu.



Obrázek 26 Úvodní formulář

Tabulka vítězů je zpracována naprosto stejným způsobem, jako tomu bylo v případě programu Skládání obrázků. To znamená, že výsledné skóre je včetně jména držitele rekordu, zapsáno a uloženo v externím textovém souboru. Po spuštění programu jsou tyto informace nahrány do proměnných typu pole a je s nimi dále pracováno. Například právě po otevření formuláře s tabulkou vítězů jsou informace nahrány do labelů, které se zde nachází. Do tabulky vítězů se zapíše pouze první 3 nejlepší skóre.

	Hráč	Skóre
1.	Magda	400
2.	Iveta	400
3.	Vědunka	300

Zpět

The image shows a window titled 'M Tabulka vítězů'. The window has a blue background and displays a table with two columns: 'Hráč' and 'Skóre'. The table lists three players: 1. Magda (400), 2. Iveta (400), and 3. Vědunka (300). Below the table is a white button with the text 'Zpět'. The window title bar includes a blue 'M' icon, the text 'Tabulka vítězů', and a close window control.

Obrázek 27 Tabulka vítězů

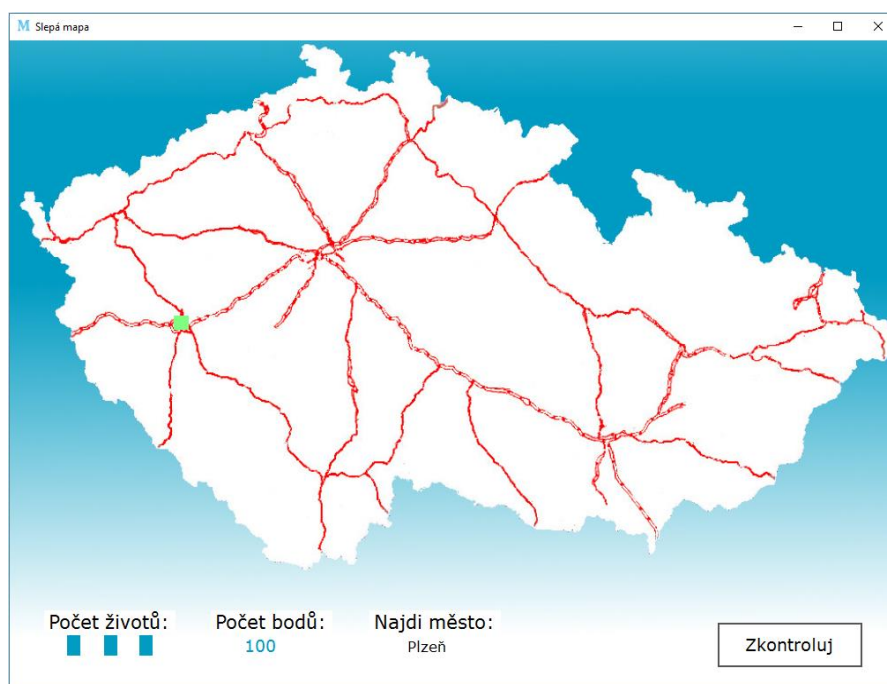
Pokud uživatel na úvodním formuláři zvolí novou hru, dojde k zobrazení formuláře se slepou mapou. Slepá mapa je zaměřena na města České republiky. Tudiž je na formuláři zobrazená mapa, na které jsou vyobrazeny pouze hranice republiky a hlavní dálniční tahy.

Uživatel má za úkol označit zeleným bodem na mapě místo, kde si myslí, že by se mohlo zadané město nacházet. Zelený bod se na mapu umísťuje pouhým kliknutím

na vybranou lokaci. Uživatel má možnost své rozhodnutí samozřejmě změnit, a to do té doby, než svou volbu potvrdí stisknutím tlačítka s nápisem zkontrolovat.

Města, která má uživatel vyhledat, jsou zadána v labelu ve spodní části formuláře. Do tohoto labelu jsou nahrávána z externí databáze, pomocí prvku datagridview. Nejprve je náhodně zvolen řádek, ze kterého bude město vybráno, a posléze bude do proměnných vložen název města a souřadnice, na kterých se správně nachází. Po spuštění kontroly, budou porovnány právě tyto souřadnice, se souřadnicemi zeleného bodu, který označuje místo určené hráčem. Samozřejmě je počítáno s jistou tolerancí.

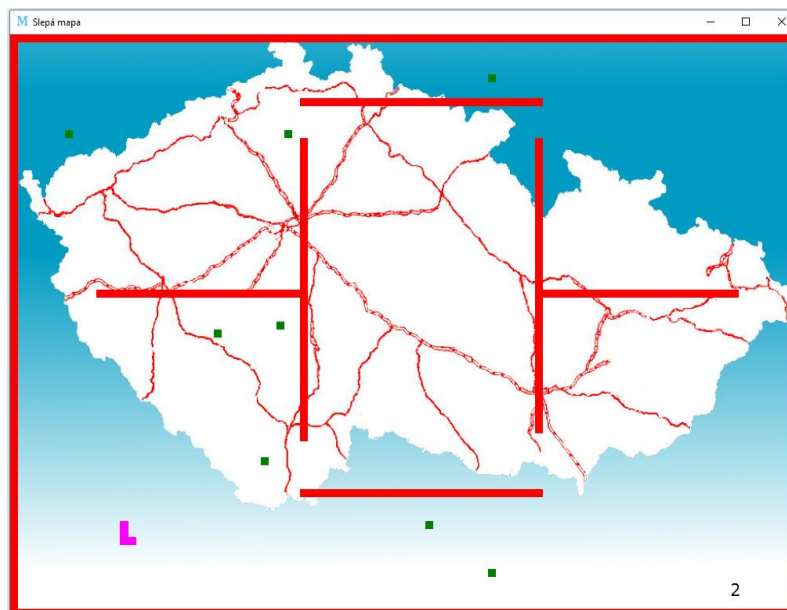
Pokud uživatel zaměří město správně, bude mu přičteno 100 bodů do skóre a bude moci pokračovat s dalším městem. Pokud nastane situace, že uživatel zaměří město nepřesně, bude uživatel nucen splnit minihru, aby mohl pokračovat dále. Zároveň mu bude odečten jen ze 3 životů.



Obrázek 28 Slepá mapa

Jak již bylo zmíněno výše, minihrou je právě hra had. Aby mohl uživatel pokračovat dále a navyšovat skóre díky zaměřování měst, musí posbírat všech 10 zelených bodů, které jsou náhodně rozmístěny po obrazovce. Zároveň jsou na ploše rozmístěny červené překážky. Pokud had narazí do této překážky, hra pro něj automaticky končí. V případě, kdy se podaří hráči sesbírat všechny zelené body, bude hráči zobrazena správná poloha města a bude moci pokračovat. V programu jsou nachystána 3 kola, s různým rozmístěním zábran. Což přesně

odpovídá počtu životů. S každým kolem se také zvyšuje rychlost pohybu hada. Pokud hráč nezaměří správnou lokaci ani na 4. pokus, hra pro něj automaticky končí.



Obrázek 29 Had

Souřadnice překážek jsou uloženy v proměnných typu pole. Do těchto proměnných jsou souřadnice uloženy vždy při samotném vykreslování překážek. Stejným způsobem je proveden i okraj hrací obrazovky, náraz do něho taktéž znamená konec hry. V proměnných typu pole jsou uloženy i souřadnice bodů, které musí hráč sesbírat. Ty jsou ale na rozdíl od překážek vybírány náhodně. Avšak vždy je hlídáno, aby se neshodovaly souřadnice bodu se souřadnicemi překážek.

Select Case kolo

Case 1

```

zde = 140 'Prostřední levá zábrana
For i As Integer = 0 To 37
  poleX(i) = 370
  poleY(i) = zde
  e.Graphics.FillRectangle(Pen, poleX(i), poleY(i), 10, 10)
  zde += 10
Next
zde = 140 'Prostřední pravá zábrana
For i As Integer = 38 To 74
  poleX(i) = 670
  poleY(i) = zde
  e.Graphics.FillRectangle(Pen, poleX(i), poleY(i), 10, 10)
  zde += 10
Next
  
```

Obrázek 30 Tvorba zábran 1. kola

Had je dán do pohybu díky prvku timer. V každém tiku timeru se had přesune o 10 bodů, kdy směr posunu závisí na poslední stisknuté klávese, která slouží k ovládání hada.

Kromě pohybu se při každém tiku musí zkontrolovat, zda had náhodou nenarazil do některé z vytvořených překážek a také zda náhodou had nedosáhl některého z bodů.

```
Private Sub Timer1_Tick(sender As Object, e As EventArgs) Handles Timer1.Tick
    If Klavesa = 87 Then          'Pohyb nahoru postisknutí W
        zpozice()
        pozice2.Y -= 10
    End If

    If Klavesa = 65 Then        'Pohyb vlevo postisknutí A
        zpozice()
        pozice2.X -= 10
    End If

    If Klavesa = 83 Then        'Pohyb dolů postisknutí S
        zpozice()
        pozice2.Y += 10
    End If

    If Klavesa = 68 Then        'Pohyb vpravo postisknutí D
        zpozice()
        pozice2.X += 10
    End If

    PictureBox4.Location = pozice5 'Posun hada
    PictureBox3.Location = pozice4
    PictureBox2.Location = pozice3
    PictureBox1.Location = pozice2
    kontrola()
    kontrolajidla()
End Sub
```

Obrázek 31 Události při tiku Timeru

Pokud had bodu dosáhne, budou souřadnice bodu vymazány, aby nemohlo dojít k jeho opětovnému započítání. Na následujícím obrázku je zachycena kontrola dosažení souřadnic bodu. Stejným způsobem probíhá i kontrola nárazu do překážek.

```
Sub kontrolajidla()
    For i As Integer = 0 To 10
        If pozice2.Y = jidloY(i) And pozice2.X = jidloX(i) Then 'Porovnání pozice hada s prvkem pole
            Label2.Text += 1                                     'Navýšení skóre
            Label2.Refresh()
            jidloY(i) = 0                                       'Vymazání souřadnic
            jidloX(i) = 0
        End If
    Next
End Sub
```

Obrázek 32 Kontrola dosažení souřadnic bodu

4.6.2 Možnosti modifikace

První možností rozšíření je samozřejmě přidání dalších měst do databáze. To můžeme provést pomocí programu Microsoft Access prostým přidáním nového řádku. Ještě před přidáním je ovšem nutné zaměřit ve vývojovém prostředí souřadnice města na mapě.

Kromě tohoto můžeme proměnit zaměření slepé mapy. Pro tento krok je nutné nejprve vytvořit mapu, která by odpovídala tomu objektu, ve kterém chceme žáky naučit se orientovat, poté tuto mapu vložíme jako nový podklad formuláře. A posléze stačí znovu naplnit databázi novými objekty, které lze zaměřit na změněné mapě.

Další možnosti změny se mohou týkat využití hry had v jiném programu. Hra může například sloužit k výuce anglického jazyka, kdy by se na hracím poli objevily vždy dva objekty s anglickými výrazy a žák by měl za úkol sebrat to slovo, které by odpovídalo překladu zadaného českého slova. Tento princip by bylo opět možné přizpůsobit pro využití v ostatních předmětech. Příkladem může být matematika, kde by žák hadem sbíral správné výsledky zadaných příkladů.

4.6.3 Charakteristika programu Slepá mapa podle kategorizace J. Dostála (2011)

Podle kategorií vytvořené J. Dostálem (2011) můžeme výukový program Slepá mapa označit jako:

- *interaktivní,*

Slepá mapa je interaktivní díky tomu, že žák sám přidává bod na mapu, který je automaticky zkontrolován a podle toho, zda je bod umístěn na správném místě, program dále reaguje. Interaktivní je i minihra, která reaguje na stisknutí daných kláves.

- *určený pro základní školy,*

Města, která jsou v současné době přítomna v databázi, jsou určena pro základní školu. Na vyšší stupeň školství by mohla mapa být přizpůsobena například přidáním i menších měst, či jiných podrobnějších informací.

- *zpětnovazební,*

Žák dostává zpětnou vazbu po umístění bodu a stisknutí tlačítka zkontrolovat. Program mu sdělí, zda byl při zaměřování úspěšný, či nikoliv. Pokud nebyl, dozví se správnou lokaci po dohrání minihry.

- *plná verze,*

Opět i u tohoto programu je poskytována pouze plná verze.

- *offline,*

I tuto hru je možné mít nainstalovanou pouze na lokálním disku. K našim programům neposkytujeme ani žádnou formu online podpory.

- *předmětově zaměřený,*

Program je svým charakterem zaměřený na předmět zeměpis. Bylo by možné ho přizpůsobit například i k výuce přírodopisu. Jedna možnost by mohla být umístování zvířat na mapu světa, podle jejich přirozeného výskytu.

- *monotématický,*

Slepá mapa je v současném stavu zaměřena pouze na města ČR. Rozšířit program na další témata by bylo možné například přidáním dalších geografických objektů, jako jsou hory, rybníky apod.

- *vizuální,*

Slepá mapa působí na žáka pouze vizuálně. Orientace na mapě je založena pouze na vnímání zrakem, nikoliv jinými smysly.

- *jednojazyčný,*

Uživatelské prostředí i zadání měst je pouze v jedné jazykové verzi a to v češtině.

- *pro samostudium,*

Program je vhodný především pro samostatnou práci žáků. Žáci si mohou formou hry procvičit orientaci na mapě České republiky.

- *Monouživatelský,*

Slepá mapa je určena pouze pro jednoho uživatele. Jediný způsob, jak se do hry může zapojit více žáků, je díky soutěži o nejlepší skóre.

- *didakticky polyfunkční.*

Výukový program slouží nejen k opakování polohy měst na mapě, ale i k motivaci žáků formou hry.

4.7 Dílčí závěr

Výukové programy představené v této kapitole jsou pouze ukázky toho, co lze s využitím principu jednoduchých her a programovacího jazyka Visual Basic .NET vytvořit. Možnosti jsou samozřejmě mnohem širší. Prakticky je omezuje pouze fantazie učitele, učitelovy programovací dovednosti a samozřejmě čas, který je ochotný do tvorby svého vlastního výukového softwaru investovat.

5 VÝSLEDEK NASAZENÍ DO ŠKOLY

Všechny vytvořené programy byly vyzkoušeny na škole, kde probíhala má 2. souvislá pedagogická praxe. Konkrétně v 6. a 7. ročníku. Žáci měli možnost hodinu programy testovat. Během jejich testování jsem s žáky prováděl rozhovory a nakonec je nechal ke každému programu vyplnit krátký dotazník, který byl zaměřený na vlastnosti programů.

Mimo to jsem požádal o vyjádření k jednotlivým programům i učitele, který mi dělal vedoucího učitele při mé pedagogické praxi.

5.1 Kvíz

Kvíz patřil k těm programům, které žáky bavily nejvíce. Jako zábavnou formu opakování označilo Kvíz celkem 75 % žáků, zbylých 25 % sice neprojevovalo z programu úplné nadšení, ale neoznačili ho ani jako nezábavný. To se projevilo na vůli nasazení programu do výuky, k čemuž se přiklonilo již 83 % žáků.

Ovládání programu včetně principu hry bylo srozumitelné většině žáků. Konkrétně jako srozumitelný označilo program 91 % žáků. Problém nastal pouze u jedné ze dvojic, která nepochopila, že smyslem hry je spojit obsazenými poli všechny 3 stěny hracího pole a očekávala ukončení po zodpovězení určitého počtu otázek.

Se vzhledem byli spokojeni všichni žáci mimo jednoho, který nebyl spokojen se vzhledem žádného z mých programů a tuto skutečnost vždy zapsal i do kolonky určené na poznámky žáků.

Chyba se opět vyskytla opět pouze u jedné dvojice. Po dokončení hry a vyhlášení vítěze, nedošlo k uzavření formuláře s hracím polem. Tento jev byl ojedinělý a nepodařilo se mi bohužel odhalit jeho příčinu.

Nejčastější připomínky směřované k tomuto programu byly zaměřené na obsah otázek. Většina žáků uvedla, že výběr otázek byl moc složitý. Ovšem otázky byly vybrány z okruhů, které jsem sám se žáky probíral během své pedagogické praxe. Dále by žáci uvítali přidání otázek i z jiných předmětů, než těch, které jsou již obsaženy.

Učitel, který shodou okolností vyučuje předměty již v programu obsažené, uvítal tu možnost, že si může sám upravovat otázky, bez toho aniž by musel zasahovat do kódu. Navíc by uvítal možnost výběru odpovědi z několika nabídnutých možností.

5.2 Skládání obrázků

Program skládání obrázků patřil mezi ty méně oblíbené. Ve výuce by ho uvítalo pouze 58 % žáků. Stejný počet žáků shledává tento způsob opakování jako zábavný. Při rozhovorech se žáky vyplynulo z jejich odpovědí, že tato obliba je tak nízká hlavně z důvodu náročnosti samotné hry. Většina žáků měla problém s tím obrázek vůbec poskládat. Problém by bylo možné vyřešit vypnutím náhodné míchání částí obrázků a předpřipravit pouze několik jednodušších variant rozmístění obrázků.

Celkem 33 % žáků označilo ovládání za nepřehledné. Všichni tito žáci měli problém hlavně po zapnutí hry, kdy nepochopili, že stačí na jednotlivé obrázky klikat a ty se budou sami přesunovat a snažili se obrázky přetahovat. Někteří v kolonce pro vlastní názor uvedli, že by systém změnili tím způsobem, aby bylo možné obrázky vyjímat a znovu vkládat, čímž by se vlastně úplně změnil princip hry a vznikla by hra puzzle.

Vzhled jako líbivý označilo celkem 75 % žáků. Zbylí žáci měli problém hlavně se vzhledem vybraných obrázků.

Žádný žák nenarazil při testování na jakoukoliv chybu, která by ovlivnila chod programu, nebo nějakým způsobem žákům znemožnila bezproblémovou práci s programem.

Na otázku, co by žáci v programu vylepšili, se nejčastěji objevovala odpověď, že by rádi v programu viděli i jiné organismy než pouze ptáky. Konkrétně by rádi viděli v programu savce. Kromě toho se objevila již zmíněná odpověď se změnou principu hry.

Učitel by v programu přivítal jednodušší způsob, jak přidávat nové obrázky.

5.3 Pexeso

Výukový program Pexeso žáky poměrně zaujal. Jeho nasazení do výuky by ocenilo 67 % dotázaných žáků, 17 % žáků nedokázalo odpovědět a stejný podíl žáků by využití při vyučování neuvítalo. Podobně se žáci vyjadřovali i k zábavnosti této hry.

Žáci neměli problém ani s ovládním tohoto programu, kdy pouze 17 % žáků uvedlo, že pro ně není ovládním příliš srozumitelné. Nedorozumění v ovládním se týkalo nepochopení principu propojení obrázků plodu s textem. Tento problém by mohl být řešen lepším ústním vysvětlením, před začátkem práce, případně doplněním další zprávy, která žákovi vysvětlí princip. Proti zprávě hovoří ovšem ten fakt, že většina žáků si je v jiných programech stejně nečetla.

Vzhled programu ocenilo kladně celkem 75 % žáků. Zbylý počet žáků si stěžoval hlavně na volbu barev pexesa a v případě elektrotechnických značek, se některým žáků nelíbily zvolené obrázky.

U tohoto programu se vyskytla chyba poměrně u vysokého počtu žáků. Celkově na chybu narazilo 33 % žáků. U všech se jednalo o tu samou chybu, která byla ve všech verzích. Důvodem, proč ji neodhalili všichni žáci, bylo, že se vyskytovala u hry pro 2 hráče, kde bylo po dokončení hry zobrazeno špatné označení vítězného hráče. Tato chyba byla samozřejmě ihned odstraněna a již by se neměla dále vyskytovat.

Na otázku, co by žáci v programu změnili, přidali, či vylepšili, bylo dosaženo mnoha podmětů, které se všechny bez výjimky týkaly verzí pexesa. Verzi s elektrotechnickými značkami označili žáci jako příliš náročnou, hlavně z toho důvodu, že na škole není tomuto tématu věnována dostatečná pozornost. Mimo to by žáci uvítali přidání dalších verzí pexesa, například pexeso zaměřené na rozpoznávání savců.

Učitel by stejně jako tomu bylo v případě skládání obrázků, uvítal jednodušší způsob jak přidávat nové verze pexesa. Ale velice se mu líbily vymyšlené možnosti využití pexesa při výuce plodů dřevin.

5.4 Šibenice

U programu Šibenice strávili žáci, dle mého odhadu, asi nejdelší čas. To se promítlo i na vyjádření žáků týkajícího se toho, zda by chtěli program zapojit do své výuky. Kladně

se k této otázce vyjádřilo 67 % žáků. Stejný díl žáků i uvedl, že je opakování tímto způsobem zábavné.

Ovládání programu bylo vcelku srozumitelné. Celkem 67 % žáků uvedlo, že je ovládání srozumitelné. Zbytek měl problém hlavně v tom zorientovat se v nápovědě ve formě počtu písmen. Někteří také měli problém s pochopením principu hry, který musel být dovysvětlen. Nedorozumění vzniklo díky nepochopení, že je po žácích požadován anglický ekvivalent uvedeného českého slova.

Vzhled kladně ohodnotilo 83 % žáků. Zbylí 2 mi sdělili, že pro ně byla rušivá oranžová barva, která byla použita na ovládací prvky a také na vymalování šibenice s oběšencem.

Chyba byla odhalena pouze v jednom případě. Nejednalo se ovšem o chybu v programu, nýbrž v externí databázi, ze které si program přebírá slovní zásobu. Chyba nebyla závažná, protože se jednalo pouze o záměnu anglického a českého znění slova na jednom řádku. Ovšem, protože program Šibenice neobsahuje možnost zadání písmen s diakritickými znaménky, znemožnila tato chyba žákovi další pokračování ve hře.

Bohužel v tomto případě žáci nepřišli s žádnými nápady, jak by tento program mohl být vylepšen.

Učiteli se taktéž nelíbila zvolená oranžová barva a pokládal slova, kterými byl slovník naplněn, za příliš jednoduchá. Naopak se mu líbila jednoduchá úprava databáze se slovy a rád by program využil i v jiných předmětech, a to právě formou zadáváním otázek namísto českých ekvivalentů.

5.5 Slepá mapa

Slepá mapa patřila společně s Kvízem k nejoblíbenějším hrám. Shodně jí považovalo za zábavnou a uvítalo ve výuce 83 % žáků. Tato obliba spočívá hlavně v začlenění hry had do programu, kterou žáci shledávali jako velice zábavnou.

S ovládním programu mělo problém pouze 18 % žáků. Ti nepochopili hlavně ovládní hry had. Nepřečetli si totiž zprávu, která jim sděluje, že se had ovládá pomocí kláves W, A, S a D a snažili se hru ovládat šipkami, či jinými klávesami.

S grafickým pojetím bylo spokojeno 67 % žáků. Zbytku žáků nevyhovovalo hlavně barevné ztvárnění překážek a bodů ve hře had. Problém u nich nastával hlavně u hledání bodů, kdy si stěžovali, že zvolená barva splývá s pozadím.

Jeden žák uvedl do dotazníku, že se při hře vyskytla chyba, ale bohužel, mi nesdělil ani osobně, ani v dotazníku, v čem tato chyba spočívala, z tohoto důvodu nemohla být lokalizována, ani napravena.

Při dotazu na návrhy, jak by program žáci vylepšili, žáci označili obtížnost hada, kde se jim nelíbila především jeho rychlost, která jim přišla příliš přehnaná. Stížnost žáků jsem vzal v potaz a rychlost byla snížena.

Učitel sice ocenil možnost přidání nových měst zapsáním do databáze, bez nutnosti otevření kódu, ale nepřišlo mu příliš efektivní získávat souřadnice města, které musí být v databázi samozřejmě uvedeny, přímo z mapy po otevření ve Visual Studiu. Bohužel efektivní řešení tohoto problému nás nenapadlo.

5.6 Shrnutí

Výsledky, které vyplynuly z nasazení vytvořených programů na základní škole, jsou shrnuty v následující tabulce.

	Podíl žáků, kteří by uvítali nasazení ve výuce (%)	Srozumitelnost programu pro žáky (%)	Ocenění vzhledu (%)	Zábavnost pro žáky (%)
<i>Kvíz</i>	83	91	92	75
<i>Skládání obrázků</i>	58	67	75	58
<i>Pexeso</i>	67	83	75	67
<i>Šibenice</i>	67	67	83	67
<i>Slepá mapa</i>	83	83	67	58

Tabulka 1 Shrnující statistiky

Na základě souhrnné statistiky, lze zkonstatovat, že nejvíce žáky zaujal výukový program Kvíz. Tento program byl společně se Slepou mapou nejvíce žádán v otázce nasazení do výuky a mimo to byl pro žáky také nejzábavnější. Kvíz u žáků i v ostatních kategoriích byl pro žáky nejsrozumitelnější ze všech programů a také se jim nejvíce líbil jeho vzhled.

Jako nejméně poutavý žákům připadal výukový program Skládání obrázků. Ten přišel žákům nejméně zábavný a také by jeho nasazení do výuky ocenilo nejnižší procento žáků. Z rozhovoru se žáky vyplynulo, že tato skutečnost, je pravděpodobně způsobena náročností

hry, která je obsahem programu. Vzhledově na žáky nejméně zapůsobil výukový program Slepá mapa, kdy se žákům nelíbil hlavně vzhled minihry had.

Na základě těchto informací, lze tedy vyhodnotit, že jako nejlepší program žáci zvolili výukový program Kvíz. Naopak nejvíce změn by žáci přivítali v programu Skládání obrázků.

ZÁVĚR

Hlavním cílem této diplomové práce bylo sestavit ideje na začlenění jednoduchých počítačových her do výukových programů. Tento hlavní cíl byl naplněn pomocí cílů dílčích, kdy především 2. dílčí cíl již představuje konkrétní zpracování myšlenek.

1. dílčí cíl, který kladl za úkol provést analýzu současného stavu užívání počítačů a výukového software ve vzdělávání, včetně nastínění důvodů, proč je pro učitele vhodné vytvářet si svůj vlastní výukový software, byl řešen v kapitolách 1, 2 a 3. Kapitola 1. se zabývala zmapováním historie nástupu počítačů a výukového softwaru do vzdělávání od svých úplných počátků, až do posledních let. Na historii bylo ihned navázáno současným stavem využívání počítačů ve vzdělávání. Z této problematiky bylo pozvolna přistoupeno k možnostem vymezení pojmů výukový program a výukový software, zde byli vysvětleny rozdíly mezi těmito pojmy a představeny různé možnosti vnímání těchto pojmů. Kapitola byla zakončena možnostmi členění výukových programů, kde byla větší pozornost věnována kategorizaci dle J. Dostála, která byla později využita i v praktické části. Kapitola 2. se zabývala tématem, proč je vůbec vhodné využívat výukový software ve vzdělávání, jaké funkce může výukový software při výuce zajistit a jak je možné ho začlenit do své výuky. Kapitola 3. byla věnována důvodům proč je pro učitele vhodné vytvářet si svůj vlastní software. Zde bylo představeno 10 nejpravděpodobnějších problémů současného výukového softwaru, které zamezují většímu rozšíření do vzdělávání. Všechny body byly postupně rozebrány a byla odhalena jejich konkrétní příčina.

2. dílčím cílem byla realizace našich idejí pomocí vyššího programovacího jazyku Visual Basic .NET a navrhnout možné modifikace těchto nápadů i pro jiné předměty. Pomocí jazyk Visual Basic .NET bylo zpracováno celkem 5 idejí. Jmenovitě se jednalo o Kvíz, Skládání obrázků, Pexeso, Šibenice a Slepá mapa. Všechny námi vytvořené programy jsou popsány v kapitole číslo 4. U každého programu jsou kromě jeho představení uvedeny i možnosti, jak lze program modifikovat, tak aby byl využitelný i v jiných předmětech, než pro které je momentálně připraven. Mimo to byl program i charakterizován podle kategorizace J. Dostála a byly představeny možnosti, jak by v daných kategoriích mohl být program vylepšen.

3. dílčím cílem bylo otestování námi vyrobených programů během souvislé pedagogické praxe. Programy si zde otestovali žáci 6. a 7. ročníku. Na základě jejich

testování došlo v některých z programů k dodatečným úpravám. Byly například objeveny chyby v obsahu programu, jako prohození anglického a českého ekvivalentu slova v programu Šibenice, či prohození označení hráčů v programu Pexeso. Na základě tohoto testování lze zkonstatovat, programy žáky zaujaly a jejich nasazení ve výuce by ocenili.

V průběhu zpracovávání jsme narazili na řadu možností, jak by bylo možné práci dále rozšířit. Možnosti rozšíření naší práce jsou nastíněny již v kapitole věnující se vytvořeným programům. Je totiž samozřejmě vhodné pokračovat ve zdokonalování programů a zapracovat i námi navržené modifikace. Dále by bylo vhodné provést rozsáhlejší výzkum problematiky využívání výukového softwaru ve školách, zaměřený hlavně na důvody jeho nynějšího nevyužívání.

SEZNAM ZKRATEK

CAI - computer assisted instruction

CAL - Computer-assisted Learning

IBM - International Business Machines Corporation

ICT - Information and Communication Technologies

MSIL - Microsoft Intermediate Language

SEZNAM TABULEK

Tabulka č. 1 Shrnující statistiky

BIBLIOGRAFICKÉ ZDROJE

AIKEN, Robert M.; BRAUN, Ludwig. Into the 80's with microcomputer-based learning. *Computer*, 1980, 7: 11-16.

AXELSON, Jan. *The Microcontroller Idea Book: Circuits, Programs & Applications Featuring the 8052-BASIC Microcontroller*. lakeview research llc, 1997.

BURIANOVÁ, Eva. *Matematický a výukový software*. Vyd. 1. Ostrava: Ostravská univerzita, 2003. 74 s. Systém celoživotního vzdělávání Moravskoslezska. ISBN 80-7042-867-8.

BURIANOVÁ, Eva. *Využití aplikačních programů ve výuce*. Vyd. 1. Ostrava: Ostravská univerzita, 2003. 146 s. Systém celoživotního vzdělávání Moravskoslezska. ISBN 80-7042-858-9.

DOSTÁL, Jiří. *Počítač ve vzdělávání*. Vyd. 1. Olomouc: Votobia Olomouc, 2007, 2 sv. ISBN 978-80-7220-295-9.

DOSTÁL, Jiří. "Výukový software a počítačové hry-nástroje moderního vzdělávání." *Journal of Technology and Information Education* 1.1 (2009): 24-28.

DOSTÁL, Jiří. *Výukové programy*. 1. vyd. Olomouc: Univerzita Palackého v Olomouci, 2011. 67 s. Studijní opora. ISBN 978-80-244-2782-9.

FARENGA, Stephen J.; NESS, Immanuel. *Encyclopedia of education and human development*. Routledge, 2015.

HANÁK, Ján. *Programování v jazyce Visual Basic 2010*. Vyd. 1. Kralice na Hané: Computer Media, 2011. ISBN 978-80-7402-112-1.

HALVORSON, Michael. *Microsoft Visual Basic: krok za krokem*. 1. vydání. Překlad Milan Daněk. Brno: Computer Press, 2015. Krok za krokem (Computer Press). ISBN 978-80-251-4412-1.

LOKŠOVÁ, Irena a Jozef LOKŠA. *Pozornost, motivace, relaxace a tvořivost dětí ve škole*. Vyd. 1. Praha: Portál, 1999. Pedagogická praxe. ISBN 80-7178-205-X.

MAZÁK, Eduard. *Počítačové výukové programy a metodika jejich tvorby*. Praha, 1988.

PETTY, Geoffrey. *Moderní vyučování*. Vyd. 2. Praha: Portál, 2002. ISBN 80-7178-681-0.

PRŮCHA, Jan, Jiří MAREŠ a Eliška WALTEROVÁ. *Pedagogický slovník*. 4. aktualiz. vyd. Praha: Portál, 2003. ISBN 80-7178-772-8.

SARIYILDIZ, Sevil; SCHWENCK, Mathias; JANDER, Egbert. *Multimedia teachware in the field of architectural design*. In: Third Design and Decision Support Systems in Architecture and Urban Planning-Part one: Architecture Proceedings, Spa, August 18-21, 1996. 1996.

ŠTĚDRŮŇ, Bohumír. *Ochrana a licencování počítačového programu*. Vyd. 1. Praha: Wolters Kluwer Česká republika, 2010. Právní monografie (Wolters Kluwer ČR). ISBN 978-80-7357-555-7.

SUCHÁNEK, Petr. *Programování ve Visual Basic: pro prezenční formu studia*. Karviná: Slezská univerzita v Opavě, Obchodně podnikatelská fakulta v Karviné, 2013. ISBN 978-80-7248-849-0.

ŠTĚPÁNEK, Petr. Co je LOGO?. *Pokroky matematiky, fyziky a astronomie*, 1981, 26.2: 98-105.

YU, Jui-Chen, et al. Teachers' Professional Development in Free Software for Education in Taiwan. *education*, 2012, 5: 7.

ZOUNEK, Jiří a Klára ŠEĎOVÁ. *Učitelé a technologie: mezi tradičním a moderním pojetím*. 1. vyd. Brno: Paido, 2009, 172 s. ISBN 978-80-7315-187-4.

Elektronické zdroje:

BRDIČKA, Bořivoj. *Učení s počítačem* [online]. UK Praha, 1995 [cit. 2016-02-04]. Dostupné z: <http://it.pedf.cuni.cz/~bobr/ucspoc/>

Computing platform. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2016-03-09]. Dostupné z: https://en.wikipedia.org/wiki/Computing_platform?oldid=291806354

Freemium [online]. 2016 [cit. 2016-03-25]. Dostupné z: <http://www.freemium.org/>

Jak je to s verzemi jazyka visual basic. *Dotnetportal* [online]. 2008 [cit. 2016-03-26]. Dostupné z: <http://www.dotnetportal.cz/blogy/3/Tomas-Herceg/586/Jak-je-to-s-verzemi-jazyka-Visual-Basic>

PC Control [online]. Olomouc, 2013 [cit. 2016-04-01]. Dostupné z: <http://www.pc-control.cz/>

PROGRAM RC2000. *RC Didactic* [online]. 2007 [cit. 2016-02-12]. Dostupné z: <http://www.rcdidactic.cz/cz/item/8-software/46-program-rc2000-kod-701-swrc/>

Učte se hrou [online]. 2007 [cit. 2016-03-05]. Dostupné z: <http://www.vyukovehry.cz>

Terasoft [online]. 2012 [cit. 2016-03-05]. Dostupné z: <http://www.terasoft.cz>

VisualStudio [online]. Microsoft, 2016 [cit. 2016-03-05]. Dostupné z: <https://www.visualstudio.com>

What is free software?: The Free Software Definition. *GNU Operating System*[online]. [cit. 2016-03-06]. Dostupné z: <http://www.gnu.org/philosophy/free-sw.en.html>

Xamarin [online]. 2016 [cit. 2016-03-26]. Dostupné z: <https://www.xamarin.com>

SEZNAM PŘÍLOH

Příloha č. 1

Dotazník

Příloha č. 1

Dotazník

Ocenili byste, zapojení tohoto programu do vaší výuky?

ANO NE NEVÍM

Líbil se vám vzhled programu?

ANO NE NEVÍM

Byla pro vás obsluha programu srozumitelná?

ANO NE NEVÍM

Baví vás opakování znalostí touto formou?

ANO NE NEVÍM

Vyskytla se během vašeho testování nějaká chyba?

ANO NE NEVÍM

Pokud ANO tak jaká?

.....

A na závěr, mi prosím napište vaše nápady, co byste na programu vylepšili.

ANOTACE

Jméno a příjmení:	Bc. Petr Felner
Katedra:	Technické a informační výchovy
Vedoucí práce:	PhDr. Jan Lavrinčík, DiS., Ph.D.
Rok obhajoby:	2016

Název práce:	Kuchařka nápadů pro tvorbu výukových aplikací ve Visual Basic .NET
Název v angličtině:	Cookbook of Ideas for Creating Educational Applications in Visual Basic .NET
Anotace práce:	Tato diplomová práce se zaměřuje na možnosti začlenění jednoduchých počítačových her do výukových programů. Jejím hlavním cílem je přijít s idejemi na začlenění her do výukových programů. Dále předvést možnosti zpracování těchto programů pomocí programovacího jazyka Visual Basic .NET, včetně otestování zpracovaných programů během pedagogické praxe.
Klíčová slova:	Visual Basic, výukové programy, výukový software, počítačem řízená výuka, počítačem podporovaná výuka.
Anotace v angličtině:	This diploma thesis is focused on possibilities of integration of simple PC games into teachware. The main aim is to show ideas how integrate games into teachwares. The next aim is to show possibilities of processing of these programs via programming language Visual Basic .NET including testing of processed programs during pedagogical practise.
Klíčová slova v angličtině:	Visual Basic, teachware, computer-aided instruction, computer-assisted instruction.
Přílohy vázané v práci:	1xCD
Rozsah práce:	80 stran (112 000 znaků)
Jazyk práce:	CZ

