

Aplikace ovládané hlasem

Bakalářská práce

Lukáš Ostrý

Vedoucí práce: PaedDr. Jiří Vaníček, Ph.D.

Jihočeská univerzita v Českých Budějovicích

Pedagogická fakulta

Katedra informatiky

2011

Abstrakt

V dnešní době je kladen stále větší důraz na alternativní komunikaci s počítačem. Ovládání hlasem je asi nejnadhější cestou, jak počítač ovládat, pro ty, kteří k tomu nejsou jaksi vybaveni. Slovem vybaveni jsou míněny skupiny lidí, pro které je klasická komunikace s počítači nevhodná. Asi nejhlavnějšími skupinami jsou tělesně a duševně postižení, děti a staří lidé. U tělesně postižených a starých lidí problém spočívá v tělesných handicapech, jako je slepota, či problém s končetinami. U dětí a duševně postižených je problém ve složitosti ovládání. Ovládání hlasem je v dnešní době už technicky možné, ale stále se neobjevuje větší zájem o tuto problematiku. Bylo by proto výhodné začít s výukou této problematiky na základních školách a pomoci rozšířit zájem o tuto problematiku.

Abstract

Today, it is placed increasing emphasis on alternative communication with the computer. Voice control is probably the easiest way to control your computer, for those who do not have somehow equipped. Word of equipment are meant groups of people for whom traditional communication with computers inappropriate. The main groups are physically and mentally disabled, children and old people. For disabled and elderly people a problem is physical disabilities such as blindness, or a problem with the limbs. In children and mentally handicapped is a problem in the complexity of control. Voice control is nowadays no longer technically possible, but still appears more interested in the subject. It would therefore be advantageous to start teaching this issue in primary schools and help expand interest in this issue.

Prohlášení

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

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

V Českých Budějovicích dne 22.4.2011

Obsah

1 CÍLE PRÁCE.....	9
2 METODIKA PRÁCE.....	10
3 ÚVOD.....	11
4 ROZPOZNÁNÍ ŘEČI V POČÍTAČÍCH.....	12
4.1 PRINCIP ROZPOZNÁNÍ ŘEČI.....	12
4.2 ROZBOR POSTUPŮ.....	13
4.2.1 Mikrofony u Windows XP.....	14
4.2.2 Rozpoznání řeči u Windows XP.....	15
4.2.3 Mikrofony u Windows Vista a 7.....	15
4.2.4 Rozpoznání řeči u Windows Vista a 7.....	17
4.2.5 Tutoriál pro rozpoznání řeči.....	17
4.2.6 Ověření funkčnosti rozpoznání řeči.....	20
5 OVLÁDÁNÍ HLASEM V IMAGINE.....	21
5.1.1 Pokročilé techniky.....	23
6 NÁVRHY ŘEŠENÍ.....	26
6.1 VÝUKOVÁ APLIKACE – OBRÁZKY.....	26
6.1.1 Fungování aplikace.....	26
6.1.2 Editace obrázků a slovních dvojic.....	29
6.1.3 Přidávání obrázků a slovních dvojic.....	32
6.2 PREZentační PROGRAM.....	34
6.2.1 Tvorba šablon pro prezentační program.....	34
6.2.2 Tvorba prezentačního software.....	39
6.2.3 Efekty prezentace.....	44
6.3 HRA – DRÁHA.....	47
6.3.1 Příprava na hru.....	49
6.3.2 Průběh hry.....	51
7 ROZPOZNÁNÍ ŘEČI V ČEŠTINĚ.....	60
7.1 ALTERNATIVNÍ ŘEŠENÍ PRO ROZPOZNÁNÍ ŘEČI V ČEŠTINĚ.....	61
8 TESTOVÁNÍ.....	64
9 ZÁVĚR.....	66

1 Cíle práce

Cílem první části práce je poskytnout návod na zprovoznění rozpoznání řeči primárně využívaného v Imagine Logo. Následně s fungujícím rozpoznáváním vytvořit aplikace ovládané hlasem v prostředí Imagine Logo. Druhá část práce je zaměřena na sepsání návodů na tvorbu aplikací ovládaných hlasem a na postupy tvorby ukázkových aplikací v tomto prostředí

Dále je součástí práce

Prozkoumat možnosti implementace češtiny pro ovládání hlasem je také jedním cílem práce. Při nalezení takového řešení následuje upravení stávajících nebo vytvoření nových aplikací ovládaných hlasem v češtině. Závěrem je testování samotného rozpoznání řeči a vytvořených aplikací.

2 Metodika práce

Imagine Logo využívá rozpoznání řeči společnosti Microsoft v systémech Windows. Je tedy možné v něm programovat aplikace ovládané hlasem v angličtině. Existují české diktovací programy, které lze využít k vytvoření řešení ovládání hlasem v češtině v Imagine Logo.

Prvním krokem mé práce tedy bylo vytvoření aplikací ovládaných hlasem v angličtině, které by bylo například možno využít při výuce anglického jazyka. Tyto aplikace byly tvořeny k různým činnostem, aby ukázaly využitelnost ovládání hlasem. Po vytvoření těchto aplikací následovalo hledání způsobu, jak implementovat rozpoznání řeči v češtině do Imagine Logo.

Dalším krokem bylo vytvoření návodu na tvorbu jednoduchých aplikací ovládaných hlasem a popis postupů při tvorbě ukázkových aplikací.

Pro testování rozpoznání řeči a vytvořených aplikací byl zvolen způsob osobních sezení s jednotlivými testovanými osobami, aby bylo možné pozorovat pokroky.

3 Úvod

Ovládat počítač hlasem, či diktování a přepis hlasu na text je v dnešní době již snadno uskutečnitelná věc, která má široký potenciál. V první řadě umožňuje ovládat počítač lidem, kteří to jinak dělat nemohou nebo nechtějí. Jiným lidem může ušetřit spoustu práce, jak už při diktování dopisů nebo při přepisu televizního nebo rádiového vysílání.

Výzkum a vývoj rozpoznávání řeči počítačem probíhá už řadu let a výsledky dosažené do dnešní doby mluví samy za sebe. Momentálně se úspěšnost rozpoznání řeči pohybuje někde kolem 90 až 95 procent. Tyto programy jsou schopné se učit a případně i specializovat na uživatele a tím úspěšnost ještě zvýšit.

Rozpoznání řeči se už stalo i součástí systémů Windows. A této vlastnosti využívá i programovací prostředí Imagine Logo, které je určeno pro výuku programování se zaměřením na žáky a studenty. V Imagine je tedy možné naprogramovat ovládání hlasem, ale v nápovědě k programu Imagine není možnost ovládání hlasem šíře propracována a už vůbec se nezmiňuje o potřebě fungování rozpoznání řeči ve Windows. To se tedy stalo primárním cílem bakalářské práce. Prozkoumat problematiku rozpoznání řeči počítačem a následně zjistit a vyzkoušet způsoby ovládání Imagine pomocí hlasových povelů.

Druhou částí práce, ovšem neméně důležitou, se stalo vytvoření několika příkladů aplikací ovládaných hlasem s popisem a vysvětlením syntaxe.

Jelikož rozpoznání řeči u Windows nepodporuje češtinu, je jedním z cílů práce i vytvoření řešení pro možnost ovládat Imagine Logo pomocí hlasových povelů v češtině.

4 Rozpoznání řeči v počítačích

4.1 Princip rozpoznání řeči

Řečí se zabývá celé vědecké odvětví. Pro rozpoznávání řeči za pomoci počítače jsou však nejdůležitější Fonetika a Fonologie. Tyto vědní obory se zabývají tvorbou zvuků pomocí hlasového ústrojí a rozdílů mezi zvuky, které jsou používány k mluvení.

Rozpoznání řeči u počítačů tedy využívá poznatků těchto věd k převedení akustického analogového signálu na informace, se kterými se dá následně pracovat. Tyto informace a také základní jednotky fonologie se nazývají fonémy. Fonémy vlastně nepředstavují jakési základní zvukové prvky, ale jen ty, které mají schopnost změnit slovo, či jeho význam. Každý jazyk má určité množství různých fonémů, ze kterých následně skládá komplikovanější stavby a slova. (5)

Fonémy také mají různé varianty, které se ovšem nemohou zaměnit a tedy změnit smysl slova, či slovo samotné. Tyto varianty se nazývají alofony a mohou vznikat vlivem okolních hlásek. Například v češtině se dá rozlišit jakési otevřené a zavřené N. Ve slově *nastavení* po N následuje A a ústa se přirozeně rozevřou, kdežto u slova *stránka*, kde N je následováno K je tomu naopak. Alofony jsou jedním z důvodů proč mají různé jazyky různé počty fonémů. V jednom jazyce může být foném alofonem a naopak. Ale i s podobným počtem fonémů mohou mít dva jazyky různý počet slov. Jak čeština, tak i angličtina mají okolo 40 fonémů, angličtina je používá k vyjádření přibližně 500 000 slov ovšem čeština má slov mnohem víc, jelikož vytváří mnoho tvarů od jednoho slova. (6)

S teorií o tvorbě slov je možné začít získávat data ze zvukového signálu. Nejlepším způsobem tohoto získávání je Fourierova transformace. Ke každému fonému lze přiřadit určitý vzorek časového

průběhu frekvence a její amplitudy a následně tak rozpracovat vzorek na jednotlivé informace, ze kterých se již dají rozpoznávat slova. Ovšem rozpoznávání slov už je ta těžší část. Využívají se k tomu dva přístupy. Ten složitější je zaměřen na spojitě diktování, v druhém je vstupní signál porovnáván se vzorky. (4)

V obou případech je problém s informacemi, které se ze signálu buďto nedají zjistit, anebo je jejich zjišťování velice komplikované. Mohou to být velká písmena, mezery mezi slovy, či konce vět. Konkrétně mohou nastat případy, kdy mezera lze umístit na různá místa a všechny tyto případy budou smysluplné, ale s rozdílnými významy. To se v případě spojitěho diktování řeší pomocí kontextu. Například mluví-li se o zvířatech, vyplývá tak z kontextu, že použité slovo mýval bude zvíře a ne minulý čas od slova mít. Spousta podobných problémů se vyskytuje i v angličtině, kde se některá slova, či jejich části mohou vyslovovat téměř identicky a správnou možnost je poté nutné doplnit z kontextu (13).

Imagine Logo využívá druhého způsobu, dalo by se říct vzorkového (7, s. 220). Vstupní signál je porovnáván se zadanými vzorky, ze kterých nemusí odpovídat žádný, může odpovídat jeden, ale může jich v jistých mezích odpovídat i více. Při tomto způsobu je potřeba jistých mezer mezi slovy, aby bylo možné určit konec vzorku.

4.2 Rozbor postupů

K samotné realizaci ovládaní hlasem v Imagine Logo jsou nutné tři věci. První je, aby počítač dokázal zaznamenat zvukový projev uživatele, což skrývá řadu zádrhelů. Dalším je nutnost aby počítač také rozuměl tomu, co se uživatel snaží říci. U systémů Windows k tomu slouží Speech Application Programming Interface (dále už jen SAPI)

a rozumí-li Windows, rozumí i Imagine. Na konec je nutné, aby Imagine věděl, jak má zareagovat na jaký povel.

4.2.1 Mikrofony u Windows XP

Každý mikrofon, který je správně zapojený a nainstalovaný (je-li to zapotřebí) a dalo by se tedy říct viditelný, bude vypsán v záložce *Zvuk* v okně *Zvuky a zvuková zařízení*, nacházející se v Ovládacích panelech. V části *Záznam zvuku* je výběrové pole, kde se nacházejí všechny mikrofony počítačem registrované.

Mikrofon lze nastavit v tom samém okně v záložce *Hlas*, kde se pod tlačítkem *Hlasitost...* nachází klasické nastavení hlasitosti. Samotné nastavení je ale lepší dělat pomocí *Testu zařízení*, tedy průvodce určenému k testování zvukových zařízení. Na úvod tohoto testu je oznámeno, jaké zařízení bude testováno. Jsou to výchozí zařízení pro *Přehrávání hlasového záznamu a Záznamu hlasu* vybraná v záložce *Hlas*. Také je vyžádáno vypnutí všech spuštěných programů, ale to není nezbytně nutné. Po stisknutí tlačítka *Další* počítač otestuje zvukový hardware. Po průběhu testu je zobrazeno okno *Testování mikrofону*, kde je vyžadováno nadiktování textu do mikrofону, ale ve výsledku jde jen o hlasitost vstupu, která může být automaticky upravena. Na dalším kroku *Testování reproduktorů* vše vypadá téměř identicky, až na odemčení části *Přehrávání*. Je opět vyžadováno nadiktování krátkého odstavce, ale v tomto případě by už měla být zpětná vazba a mělo by být slyšet, co je říkáno. Hlasitost poslechu se může automaticky upravit i zde, ale hlasitost reproduktorů se musí upravit ručně, pokud nevyhovuje. Na závěr je oznámeno dokončení procesu testování.

4.2.2 Rozpoznání řeči u Windows XP

Aplikaci pro rozpoznání řeči nalezneme na webu Microsoftu, kde je nabízeno několik balíčků, dokumentace a podobně (12). Chceme-li jen rozpoznání řeči v angličtině stáhneme *SpeechSDK51.exe*. Spuštěním *SpeechSDK51.exe* ještě nezačíná instalace, ale pouze rozbalení několika souborů. Ke spuštění instalace slouží *Setup.exe*. Ta probíhá pomocí klasického Wizard průvodce. První stránka oznamuje, *co je vlastně instalováno, tedy Microsoft Speech SDK 5.1*. Poté je nutné souhlasit s licenčními podmínkami. Na další stránce jsou pole pro jméno uživatele a jméno organizace. V tomto případě však není vyplnění těchto polí nutné k pokračování. Dále je možnost změnit cestu složky určené pro instalaci. Poslední krok před instalací vám dává možnost vrátit se k předchozím krokům a učinit změny, nebo spustit instalaci. Na závěr už jen ukončíte průvodce a rozpoznání řeči je připravené k použití.

Závěrem je důležité poznamenat, že rozpoznání řeči u Windows XP je jen pro angličtinu, na rozdíl od rozpoznání řeči u Windows Vista a 7, kde je podporovaných jazyků víc.

4.2.3 Mikrofony u Windows Vista a 7

Jelikož rozpoznávání řeči nelze spustit na jiných jazykových verzích systému, než jsou ty podporované, je tedy nutné mít jazykovou verzi Windows podporovanou, což jsou: angličtina, španělština, francouzština, němčina, tradiční čínština a zjednodušená čínština. Na jiných jazykových verzích systémů Vista a 7 zkrátka není možnost spustit rozpoznání řeči. Kdyby byla takto nastavena čeština, systém by jednoduše oznámil, že v aktuálním jazyku není rozpoznávání řeči podporováno. Jedinou výjimkou je Windows verze Ultimate, která umožňuje změnu jazyka zobrazení.

Windows 7 a Windows Vista mají rozpoznání řeči již implementované, tudíž zde odpadá potřeba instalace jako je tomu u Windows XP. Je tedy zapotřebí mít správně nastavený mikrofon, což může být často problém, je-li systém instalovaný na starším zařízení. Občas se stává, že Windows 7 a Vista nemají plně kompatibilní ovladače. Například na základní desku může být ovladač, se kterým pracuje naprosto normálně, ale jakmile je potřeba použít mikrofon s jack konektorem, je zjištěno, že základní deska nepracuje úplně tak, jak by měla a mikrofon nebude fungovat. Takový problém lze jednoduše vyřešit koupí mikrofonu s USB konektorem.

Každý mikrofon správně zapojený do počítače bude zobrazený v záložce *Recording* v panelu *Sound*, který se dá otevřít buď přes obrázek reproduktoru v pravém dolní rohu, anebo v *Control Panel* ve skupině *Hardware and Sound*.

Označíme-li si konkrétní mikrofon v panelu zvuk, můžeme si vyvolat jeho *Properties*. Tam se nastaví hlasitost poslechu. Windows 7 obsahují průvodce nastavením mikrofonu, toho najdeme na kartě *Sound* pod tlačítkem *Configure*, které nás odkáže v *Control Panel* do *Speech Recognition* v záložce *Ease of Acces*. Po spuštění *Set up microphone* je nutno zvolit druh používaného mikrofonu. V nabídce jsou tři možnosti a to *Headset*, *Desktop* a *Other*. Každý je doplněný obrázkem, takže není nijak těžké vybrat odpovídající typ k momentálně používanému. Další krok je odlišný, pro každý nabízený typ mikrofonu, ale ve výsledku jsou to jen povely pro správné umístění a zacházení se zvoleným druhem mikrofonu. Následující krok vyžaduje nadiktování do mikrofonu dvou vět v angličtině. Tlačítko *Next* se zobrazí až po určitém intervalu a po jeho stisknutí je oznámen výsledek poslechu. Jedna ze tří možností je, že počítač plně rozeznal, co bylo řečeno. Druhá říká, že sice zaregistroval zvukový vstup, ale neporozuměl mu a vhodným

řešením by mohlo být umístění mikrofonu na lepší pozici a podobně. A třetí oznamuje absolutní nezaznamenání vstupu a upozorňuje na možnost mechanických tlačítek a manuálních nastavení hlasitosti na různých typech mikrofonů. Po proběhnutí průvodce, je-li to zapotřebí, upraví počítač hlasitost poslechu, pokud vůbec zaznamenal nějaký vstup.

4.2.4 Rozpoznání řeči u Windows Vista a 7

Na stejné lokaci, jako se nachází průvodce nastavením mikrofonu, tedy *Control Panel/Ease of Acces/Speech Recognition* se nachází i aplikace *Start Speech Recognition*. Po prvním spuštění je nutné nejprve projít průvodcem. Hned na začátku je zděleno k čemu je rozpoznání řeči dobré. Následují ty samé kroky jako v nastavení mikrofonu. Po nastavení mikrofonu je třeba zvolit, zda povolit prozkoumávání dokumentů a tím zlepšit preciznost rozpoznávání mluveného slova. Dál je potřeba zvolit způsob aktivace. Jedna možnost je manuální aktivace, kdy se aplikace rozpoznání řeči uvede z neposlouchajícího módu do módu poslouchajícího stiskem tlačítka s obrázkem mikrofonu v aplikaci, anebo klávesovou zkratkou *Ctrl+Windows key*. K hlasové aktivaci je nutné říct *Start Listening*. V následujícím kroku je možnost prohlédnutí a tisknutí seznamu povelů pro hlasové ovládání. Dále se objeví volba, zda chceme zapínat rozpoznávání řeči po spuštění systému. A následně je nabídnuto spuštění či přeskočení tutoriálu.

4.2.5 Tutoriál pro rozpoznání řeči

Tento tutoriál se nachází ve Windows Vista a 7 na stejné lokaci jako průvodce nastavením mikrofonu a rozpoznání řeči. Celý je plně ovladatelný hlasem. Až snad na jedinou výjimku je na každé stránce tutoriálu tlačítko *Next* a také většinou barevně zvýrazněný návod, co

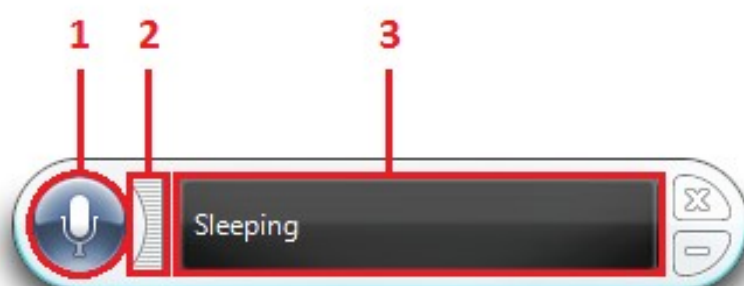
říct, aby se tutoriál posunul na další stránku nebo krok na stránce za použití ovládání hlasem. Tutoriál je rozdělený na několik částí, které se později zaměřují třeba na diktování, ale to v tomto případě není podstatné. Pro využití stačí naučit se pracovat s aplikací rozpoznání řeči a k tomu bohatě dostačuje první část tutoriálu a to *Basics*.



Ilustrace 1: Modul rozpoznání řeči ve Windows 7

Na prvních stránkách jsou vysvětleny základní prvky samotné aplikace. Prvním z nich je tlačítko pro spuštění a ukončení poslouchání, které je v ilustraci 2 označeno 1. Když je aplikace ve stavu *Sleeping* a tedy neposlouchá, respektive poslouchá jen povel pro zahájení poslechu, je tlačítko šedivé. Pokud je aplikace ve stavu *Listening* a tedy poslouchá je tlačítko modré. A pokud aplikace neporozuměla vstupu, je tlačítko dočasně oranžové. Vedle tlačítka je audio měřič, v ilustraci označen 2, který zobrazuje momentální hlasitost vstupu. Je možné vyzkoušet, že aplikace indikuje zvukový vstup, i když je v režimu neposlouchání. Ovšem v tomto režimu se zaznamenaný vstup snaží interpretovat jen na jeden povel (*Start Listening*). Hned vedle audio měřiče je *feedback area*, volně přeloženo prostor zpětné vazby,

v ilustraci 2 je označen 3. Jedná se v jednoduchosti o textové pole, ve kterém se zobrazují stavy a zprávy. Například, když aplikace neporozumí vstupu, tlačítko se jednoduše zobrazí v oranžové barvě a ve *feedback area* se napíše „What was that?“. V poslouchacím stavu je v poli text „Listening“ a při neposlouchání zase „Sleeping“. Občas je možné vidět v pravém dolním rohu *feedback* pole ikonku zaneprázdněnosti, která se standardně ukazuje vedle nebo místo šipky myši v systémech Vista a 7.



Ilustrace 2: Základní prvky modulu rozpoznání řeči u Windows 7

Následně po seznámení s prvky aplikace jsou ještě předvedeny slovní povely jako *Show Speech Options*, který vyvolá menu zobrazitelné i stiskem pravého tlačítka myši na aplikaci. Následně také *Hide Speech Recognition* a *Show Speech Recognition* sloužící k minimalizaci a maximalizaci aplikace. A nakonec povel *What can I say*, který zobrazí *Help* s výpisem povelů.

Během tutoriálu je seznamováno jen s jedním ze dvou nejdůležitějších povelů a tím je *Stop Listening*, který uvádí aplikaci do již zmiňovaného stavu *Sleeping*. Pro opětovné uvedení do stavu *Listening* je nutné využít povelu *Start Listening*, což není povel, který by se nedal domyslet, ale součástí tutoriálu by jistě být měl.

4.2.6 Ověření funkčnosti rozpoznání řeči

U Windows Vista a 7 je ověření, zda rozpoznání řeči funguje, velice jednoduchá věc, díky tutoriálu, který následuje za průvodcem zprovoznění rozpoznání řeči. U Windows XP toto bohužel možné není a je tedy nutné použít nějaký program umožňující ovládání hlasem, jako je tomu u tutoriálu u systémů Vista a 7.

Hledat takový program je vcelku zbytečné, jelikož při potřebě použít rozpoznávání řeči v Imagine Logo, už se předpokládá použití tohoto programu a dá se tedy rozpoznání řeči vyzkoušet pomocí Imagine. Na začátku následující kapitoly je jednoduchý návod, jak vložit do Imagine povel s akcí a pomocí toho lze jednoduše ozkoušet funkčnost rozpoznání řeči.

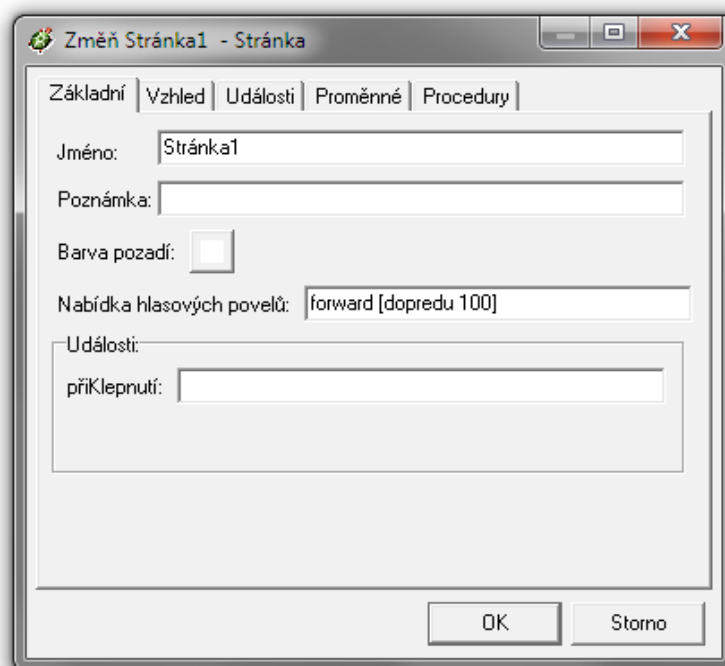
5 Ovládání hlasem v Imagine

V této části je volně čerpáno z článku doktora Tomcsányiho uveřejněného na konferenci Eurologo 2001 (7). V článku jsou popisovány možnosti ovládání hlasem v Imagine Logo.

Imagine je ve své přirozenosti schopný reagovat na anglické hlasové povely. Když je rozpoznán povel, je vyvolána nějaká akce. Povely je nejlépe vybírat jednoznačné a mohou být i víceslovné. Akce jsou vlastně příkazy, či operace a málokdy jsou jednoslovné. Povel s akcí může vypadat následovně:

forward [dopředu 100]

S touto dvojicí akce a povelu, který je zvýrazněn podtržením, by Imagine po rozpoznání řečeného slova *forward* provedl příkaz *dopředu 100*, pro všechny aktivní želvy.



Ilustrace 3: Nabídka hlasových povelů v nastavení stránky

Tento hlasový povel s akcí se dá přiřadit buďto stránce, nebo hlavnímu oknu. V nastavení stránky je pole *Nabídka hlasových povelů*. Po zadání povelu s akcí do tohoto pole nabídky povelů bude Imagine reagovat na hlasový povel *forward*.

K zadání nabídky povelů z příkazového řádku lze využít příkaz *nabidkaPovelu!*.

```
nabidkaPovelu! [forward [dopredu 100]]
```

Přiřazení povelu hlavnímu oknu je možné udělat pouze z příkazového řádku pomocí příkazu *spolecnaNabidkaPovelu!*.

```
spolecnaNabidkaPovelu! [forward [dopredu 100]]
```

Jak akci, tak i hlasový povel je nutné při více slovech zapisovat jako seznam. Při více povelích a jejich akcích je i nabídky povelů nutné zapisovat ve tvaru seznamu. V následující ukázce je už komplikovanější nabídka povelů, ve které jsou podtržením zvýrazněny povelé.

```
[left [v1 45] [big left] [ v1 60] [small left][v1 30]]
```

Je vidět, že povelé jsou liché prvky seznamu a akce sudé. S touto nabídkou povelů, by Imagine po rozpoznání povelu *left* otočil aktivní želvy o 45 stupňů doleva, po rozpoznání *big left* o 60 a po rozpoznání *small left* o 30. Při vyšším počtu povelů je lepší volit takové povelé, aby nedocházelo k záměnám.

Pokud nebudou v nabídce hlavního okna akce blíže specifikovány, budou aplikovány na momentálně aktivní stránku, popřípadě momentálně aktivní želvy na této stránce. A obdobně je tomu i u nabídky stránky. Obě nabídky jsou na sobě nezávislé, a proto Imagine při zaznamenání slova *forward* provede akci tohoto povelu

z nabídky povelů hlavního okna a pokud by aktivní stránka měla také tento povel, bude provedena i akce z nabídky povelů stránky. V případě identických nabídek povelů ve tvaru *forward [dopředu 100]* by bylo vykonáno dvakrát *dopředu 100*. Je tedy vhodné umístit do nabídky hlavního okna povely společné všem stránkám a na jednotlivých stránkách zanechat jen neopakující se povely.

5.1.1 Pokročilé techniky

Podstatnou součástí ovládání hlasem je možnost vypnutí, nebo pozastavení poslechu, aby nedocházelo k zadávání nechtěných povelů. Ne vždy jsou podmínky pro ovládání hlasem ideální. Jedny z rušivých elementů mohou být jiné osoby v místnosti s uživatelem, či hluky v místnosti přítomné, nebo do místnosti pronikající. Ovšem v některých případech může být rušivým elementem i samotný uživatel. Má-li potřebu mezi ovládáním hlasem z nějakého důvodu mluvit, je potřeba deaktivovat ovládání hlasem. Jak uvádí doktor Tomcsányi, je dobré použít hardware tlačítko na mikrofonu a v případech, kde jsou používány mikrofony bez takovýchto tlačítek, je nutné použít software řešení (7, s. 222 – 224).

Takové řešení lze udělat pomocí příkazu pro nastavení aktivity nabídky povelů *aktivníNabídkaPovelů!*, který nabývá hodnot ano a ne. Při nastavené hodnotě ne jsou obě nabídky povelů neaktivní, a tudíž nemůže docházet k nechtěným voláním povelů. Hodnotu *aktivníNabídkaPovelů* lze jednoduše nastavovat pomocí přepínacího tlačítka.

Jinou možností je použít ovládání hlasem a využít k tomu stavových menu pro nabídku povelů. Stav pro zapnutý odposlech by tvořila klasická nabídka povelů. Stav pro vypnutý by tvořila jediná dvojice povel a akce. Akce by byla procedura měnící stav, a tedy

nabídku povelů, na zapnutý. Povel k této akci by měl být takový, aby se nedal zaměnit s jinými slovy. Například slova *go* nebo *start* jsou značně nevhodná. Doktor Tomcsányi používá slovo *computer* a já ho osobně také shledávám dostatečně nezaměnitelným. Ke zpětnému přepnutí do stavu vypnutý se také může použít povel. Měl by být opět co nejvíce nezaměnitelný. Ovšem tento způsob je nevhodný pro potřeby častého přepínání stavů. V takovém případě by bylo vhodnější použít automatické přepnutí do stavu vypnutý po časovém limitu.

Procedura pro změnu stavu na vypnutý by tedy mohla vypadat asi takto:

```
příkaz vypnuto
 nabídkaPovelů! [computer [zapnuto]]
konec
```

Procedura pro stav zapnutý s časovým omezením, kdy se po 2 vteřinách opět zavolá vypínací procedura zase takto:

```
příkaz zapnuto
 nabídkaPovelů! [left [vl 45] right [vp 45]]
cekej 2000 vypnuto
konec
```

Vždy je určitá možnost, že se povel nepovede říct, či zrovna během zadávání povelu vznikne nějaké zvukové rušení a následně povel není rozpoznán. Je proto nutnost určité zpětné vazby, která může být řešena změnou tvaru želvy. Vytvoří se nová želva se jménem *ž1* a do procedury *vypnuto* se jednoduše přidá příkaz:

```
ž1'barvaper! "cerna ž1'tvar! [puntik 30]
```

Následně se do procedury *zapnuto* vloží obdobný příkaz, který opět změní tvar želvy na puntík, ovšem s jinou barvou, například červenou. Ve výsledku bude na obrazovce černý puntík, poté, co uživatel poví slovo *computer* do mikrofону, změní se barva puntíku na červenou po dobu dvou vteřin a během této doby bude Imagine reagovat na povely z aktivní nabídky.

6 Návrhy řešení

Jedním z cílů bakalářské práce bylo vytvoření aplikací ovládaných hlasem. Byly zvoleny tři okruhy využití.

- 1) Prvním z nich je program pro výuku, jelikož Imagine samotný je výukový software. Nejlepší spojení výuky a rozpoznání řeči počítačem je výuka cizích jazyků. V tomto konkrétním případě se jedná o program pro výuku angličtiny.
- 2) Další aplikaci bylo třeba zvolit k nějaké praktické činnosti. Jako činnost bylo vybráno prezentování a to z důvodů možnosti využití prezentačního software při obhajobě bakalářské práce. Při vytvoření jakési šablony prezentace by následně byla jednoduchá tvorba dalších prezentací.
- 3) Pro třetí aplikaci byl zvolen okruh her, jelikož Imagine slouží k výuce a nejlepší výuka je hrou.

6.1 Výuková aplikace – Obrázky

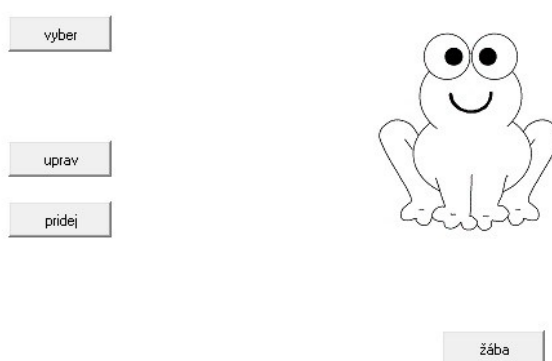
Jedná se o jednoduchou edukační aplikaci typu: poznej obrázek a řekni k němu příslušné slovíčko v angličtině. Aplikací jsou náhodně vybírány obrázky, které uživatel rozpoznává a dále do mikrofonu diktuje, co na obrázku vidí. Je-li diktované slovo správně, aplikace vybere nový obrázek. Součástí aplikace je také nástroj pro správu a nástroj pro přidávání obrázků.

6.1.1 Fungování aplikace

Celý projekt výukové aplikace je umístěn na třech stránkách. Fungování aplikace zajišťuje první stránka, kde se nachází jedna želva a několik tlačítek. Želva zobrazuje generované obrázky. Dvě tlačítka odkazující na další dvě stránky projektu. Na konec je zde tlačítko,

spouštějící výběr a tlačítko s nápovědou. Nápověda je jak vizuální, tedy textová na tlačítku, tak i audio po stisku tlačítka.

Výběr obrázku a jeho povelu se spouští třemi způsoby. Prvním je už výše zmiňované tlačítko, dalším je událost při otevření *stránky1* a nakonec hlasový povel obsahující název věci z obrázku.



Ilustrace 4: První stránka projektu s želvou ve tvaru žáby a hlasovým povelem *frog*

Ve všech třech způsobech je volána procedura *vyber*. Ta si pomocí procedury *vyberHodn* nechá vygenerovat náhodné číslo mezi jedničkou a počtem prvků ze seznamu *seznamSlov*, který obsahuje česko-anglická slovní spojení (například: [auto car]). Vygenerovaná hodnota se kontroluje, zda není stejná s předešlou vygenerovanou hodnotou uloženou v proměnné *cislo*. Pokud není, jednoduše se tato proměnná přepíše, ovšem pokud je, opět se zavolá procedura *vyberHodn*.

příkaz *vyberHodn*

```
dosadZde "hodnota 1 + nahodne ( pocet :seznamSlov )
```

```
kdyzJinak :cislo = :hodnota
  [vyberHodn]
  [dosad "cislo :hodnota]
konec
```

Po vygenerování čísla procedura *vyber* změní tvar želvy *obrazek* na prvek seznamu *seznamObrazu* odpovídajícímu hodnotě proměnné *cislo*. Taktéž dosadí do proměnné *vybPrv* příslušný prvek seznamu *seznamSlov*, což je česko-anglické slovní spojení k odpovídajícímu obrázku. Poté procedura uloží do nabídky hlasových povelů stránky1 seznam *povely*, skládající se z druhého prvku proměnné *vybPrv*, tedy anglického slovíčka, jako povelu a procedury *vyber* jako akce. Na závěr procedury je změněn popis tlačítka nápovědy, který se změní na první prvek seznamu *vybPrv*, tedy české slovíčko.

```
příkaz vyber
  vyberHodn
```

```
dosad "vybPrv prvek :cislo :seznamSlov
```

```
obrazek'tvar! prvek :cislo :seznamObrazu
dosadZde "povely vlozprvni prvek 2 :vybPrv [vyber]
```

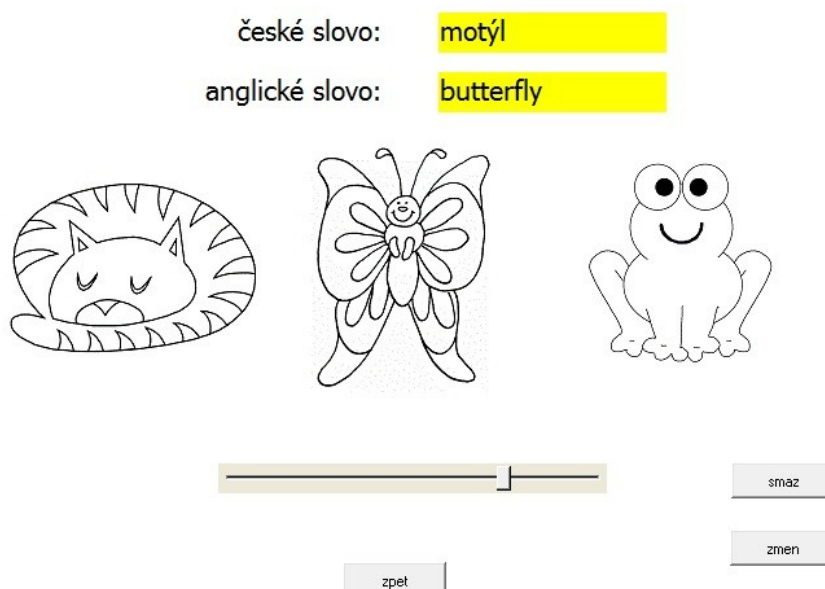
```
napoveda'popis! prvek 1 :vybPrv
nabidkaPovelu! :povely
konec
```

Tlačítko nápovědy jednoduše po stisknutí spustí povězení, nebo převod textu na řeč, druhého prvku seznamu *vybPrv* počítačem.

```
rekni prvek 2 :vybPrv
```


6.1.2 Editace obrázků a slovních dvojic

Druhá stránka projektu je určena k úpravám a mazání. Obsahuje vodorovný posuvník k vybírání z obrázků a tři želvy, jednu pro zobrazování momentálně vybraného obrázku a dvě pro zobrazování předcházejícího a následujícího obrázku ze seznamu *seznamObrazu*. Na stránce jsou také dvě textová pole, pro zobrazení anglického a českého slova k momentálně vybranému obrázku. Změnu obrázků a obsahy textových polí zajišťuje procedura *tahnuti* volaná událostí posuvníku *priTahnuti*.



Ilustrace 5: Editační stránka projektu se zvoleným obrázkem motýla, jehož lze smazat, nebo editovat jeho slovíčka

Procedura *tahnuti* mění tvar prostřední želvy *obrazek2* na odpovídající prvek ze seznamu *seznamObrazu*, podle hodnoty posuvníku. Stejně mění i obsah textových polí *textCZ* a *textEN* na první a druhou hodnotu příslušného prvku ze seznamu *seznamSlov*.

Dále je hodnota posuvníku porovnávána s počtem prvků v seznamu *seznamObrazu*, a je-li stejná, je pravé želvě (*obrazek3*) dán tvar z proměnné *bezObr*. Při nerovnosti hodnot, dostává želva tvar hodnoty posuvníku zvětšený o jedna. Obdobně je řešena i levá želva (*obrazek1*), ovšem hodnota posuvníku je srovnávána s hodnotou 1 a při nerovnosti je tvar želvy dle hodnoty posuvníku zmenšené o jedna.

příkaz tahnuti

```
obrazek2'tvar! prvek p1'hodnota :seznamObrazu
textCZ'hodnota! prvek 1 prvek p1'hodnota :seznamSlov
textEN'hodnota! prvek 2 prvek p1'hodnota :seznamSlov
```

```
kdyzJinak p1'hodnota=počet :seznamObrazu
[obrazek3'tvar! :bezObr]
[obrazek3'tvar! prvek (p1'hodnota + 1)
 :seznamObrazu]
```

```
kdyzJinak p1'hodnota=1
[obrazek1'tvar! :bezObr]
[obrazek1'tvar! prvek (p1'hodnota - 1)
 :seznamObrazu]
```

konec

Proměnná *bezObr* obsahuje obrázek, který vypovídá o hranici seznamu, například šedivý obdélník, červený kříž, nebo obrázek odpovídající pozadí a tedy neviditelný. Toto dosazování obrázků by se dalo také snadno vyřešit zneviditelňováním první a třetí želvy.

Na druhé stránce jsou dále také tři tlačítka. Tlačítko *zpět* jednoduše přepne aplikaci na první stránku. Toho efektu se dosáhne vypsáním jména první stránky (*stranka1*) do tlačítka.

Tlačítko *smaz* po stisknutí volá proceduru *smazani*. Tato procedura odebere právě zvolený obrázek a k němu náležící slovní spojení z jejich seznamů. K odebrání slovního spojení lze jednoduše použít operaci *bezPrvku*. Tato operace potřebuje na vstupu dostat číslo a seznam a výstupem je vstupní seznam bez prvku určeného vstupním číslem. Ovšem tuto operaci z nějakého důvodu nelze aplikovat na seznam obrázků, respektive to lze udělat, ale výstupem této operace bude jen první obrázek ze vstupního seznamu. Je tedy nutné toto nějakým způsobem obejít. Je možné vybrat ze seznamu jen prvky před vybraným prvkem a prvky za vybraným prvkem pomocí operace *prvek*, která umožňuje vybrat jen určité množiny prvků ze seznamu. Například *prvek [3 4]* by vrátil čtyři prvky seznamu počínaje třetím prvkem.

příkaz *smazani*

```
dosad "seznamSlov bezprvku p1'hodnota :seznamSlov
```

```
dosadzde "segment1 []
```

```
dosadzde "segment1 vlozposledni 1 :segment1
```

```
dosadzde "segment1 vlozposledni (p1'hodnota - 1)  
:segment1
```

```
dosadzde "segment2 []
```

```
dosadzde "segment2 vlozposledni (p1'hodnota + 1)  
:segment2
```

```
dosadzde "segment2 vlozposledni (p1'maximum -  
p1'hodnota) :segment2
```

```
dosadzde "bezprvkujinak seznam :segment1 :segment2
```

```
dosad "seznamObrazu prvek :bezprvkujinak :seznamObrazu  
konec
```

Po odstranění prvku ze *seznamSlov* za použití operace *bezPrvku* a momentální hodnoty posuvníku jsou tvořeny dva lokální seznamy. *Segment1* obsahuje hodnotu 1 a momentální hodnotu posuvníku zmenšenou o 1. *Segment2* obsahuje momentální hodnotu posuvníku zvětšenou o 1 a maximální hodnotu posuvníku zmenšenou o momentální hodnotu posuvníku. Z těchto seznamů je poté vytvořen seznam *bezprvkuJinak*, který symbolizuje seznam bez konkrétního prvku. Seznam *bezprvkuJinak* je použit jako vstupní hodnota pro operaci *prvek*, jehož výstupem je seznam, kterým je přepsán seznam *seznamObrazu*. Nový obsah seznamu *seznamObrazu* se skládá z prvků před a za prvkem starého obsahu tohoto seznamu, jehož pořadí odpovídalo momentální hodnotě posuvníku.

Procedura *zmena* volaná tlačítkem *zmen* nahrazuje česko-anglické slovní spojení odpovídající hodnotě posuvníku ze seznamu *seznamSlov* slovním spojením, které je získáno z textových polí.

příkaz *zmena*

```
dosad "vybPrv nahrad 1 :vybPrv textCZ'hodnota
```

```
dosad "vybPrv nahrad 2 :vybPrv textEN'hodnota
```

```
dosad "seznamSlov nahrad p1'hodnota :seznamSlov
```

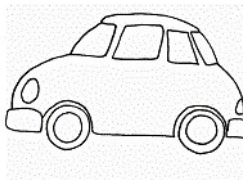
```
:vybPrv
```

konec

6.1.3 Přidávání obrázků a slovních dvojic

Třetí stránka projektu slouží k přidávání obrázků a česko-anglických párů slov. Vyskytuje se na ní jedna želva. Tvar této želvy se mění pomocí tlačítka pro vybrání obrázku, které spustí průzkumníka a uživatel jen najde lokaci, kde má uložené obrázky a jeden vybere. Dále jsou na stránce opět dvě textová pole pro české a anglické slovíčko.

české slovo: auto
anglické slovo: car



vyber obrazek

pridej

zpet

Ilustrace 6: Stránka projektu pro přidávání vybraného obrázku a napsané slovní dvojice

Po vyplnění polí a vybrání obrázku je možné stisknout tlačítko *pridej*, které spustí proceduru *pridani*. Ta uloží obsahy obou polí na konec seznamu *seznamSlov* a momentální tvar želvy na konec seznamu *seznamObrazu*.

příkaz *pridani*

```
dosad "vybPrv nahrad 1 :vybPrv textCZ'hodnota  
dosad "vybPrv nahrad 2 :vybPrv textEN'hodnota  
dosad "seznamSlov vlozposledni :vybPrv :seznamSlov  
dosad "seznamObrazu vlozposledni ž1'tvar :seznamObrazu  
konec
```

Tlačítko *zpet* přepíná na první stránku projektu a umožňuje vrátit se k výuce.

6.2 Prezentační program

U prezentačních programů bývají využívány objekty určené pro nadpisy, různé texty, obrázky a podobně. V Imagine Logo je možné takové objekty také vytvářet. Před tvorbou aplikace je dobré se nejprve zaměřit na vytvoření šablon jednotlivých typů těchto objektů. Imagine umožňuje tvorbu tříd, podle kterých jsou objekty tvořeny. Může se jednat o stránky s předem danou velikostí a pozadím, ale také o želvy, které budou reprezentovat různé druhy, styly a velikosti písma. Dá se tedy říct, že se jedná o šablony, které se využijí při tvorbě prezentace. Důvod pro používání šablon u prezentace je velice jednoduchý. Nemusí se nastavovat každému novému objektu parametry a vlastnosti, které se budou u objektů tvořených z jedné třídy opakovat.

6.2.1 Tvorba šablon pro prezentační program

K tvorbě tříd, a tedy i šablon, slouží příkaz *novaTrida*. Tento příkaz očekává tři vstupy. První vstup je jméno rodičovské třídy, od které bude nová třída odvozena a od které převezme všechna základní nastavení. Rodičovskou třídou může být například stránka nebo želva, ale může jí být i třída už takto vytvořená. Druhým vstupem je název nové třídy. Posledním vstupem jsou nastavovány hodnoty vlastností nové třídy. Mohou to být velikost, tvar a podobně. Je také důležité zdůraznit, že při nastavování vlastností se za příkazy nepíší vykřičníky, které se jinak při zadávání hodnot vlastností v Imagine psát musejí. Příkazy pro vytvoření třídy pro stránku a želvu mohou vypadat následovně:

```
novaTrida "Stranka "sablonStranky [  
  nabidkapovelu [next [udelej]]]
```

```
novaTrida "Zelva "sablonZelvy [barvapera "cervena]
```

Jak je vidět u příkazu na vytvoření třídy stránky, lze nastavit nabídku povelů pro hlasové ovládání. Každá stránka vytvořená z této třídy bude mít nastavenou tuto nabídku.

Vlastnosti využitelné u stránek jsou jiné, než ty u želv. V první řadě to bude již zmiňovaná nabídka povelů, kde je ale důležité se zamyslet nad využitím povelů. Je lepší zadat jeden povel pro hlavní okno, než každé stránce zadávat stejný opakující se povel a zadat jen ty, které se neopakují příslušným stránkám. To už ale náleží tvorbě každé jednotlivé stránky. U stránek by měla být také nastavena velikost stránky. Ta se udá pomocí příkazu *velikost [X Y]*, kde X je šířka a Y výška stránky. Pro velikost stránky je však velmi důležité i nastavení pozadí. Nastavením pozadí se totiž změní velikost stránky, je tedy důležité dbát na správnou velikost obrázku sloužícího jako pozadí. Samotné udání obrázku jako pozadí se provede pomocí příkazu *pozadi obrazek*, kde za *obrazek* lze dosadit cestu k obrázku, ale je lepší odkazovat se zde na proměnnou, kde je obrázek už uložený. To hlavně z důvodu přenositelnosti. Je možné nastavit i barvu pro pozadí příkazem *barvaPozadi "barva*. Celkový příkaz pro vytvoření šablony stránky by tedy mohl vypadat takto:

```
novaTrida "Stranka "sablonStranky
[
  velikost [1024 768]
  pozadi :obrazekPozadi
  nabidkaPovelu [...]
]
```

U želv se musejí použít jiné vlastnosti. Ty, které želva nemá, by se uložily v nově vytvořených želvách, jako jejich proměnné. Například při použití nabídky povelů, by si nově vytvořená želva udělala

proměnnou *nabidkaPovelu*, která by obsahovala seznam následující za příkazem při zadávání.

U prezentace je potřeba hlavně text a k tomuto účelu se výborně hodí želvy s tvarem textu. Je proto dobré si připravit želvy s formátováním odpovídající různým částem textu, jako jsou nadpisy, obyčejný text, citace a podobně. V první řadě je tedy nezbytné nastavit styl písma.

```
pismo [[Verdana][32 700 0 0 0 238]]
```

Jak je vidět na příkladu, příkaz pro nastavení písma je dosti komplikovaný. V první části se jednoduše nastavuje font písma. Před použitím je ovšem dobré zjistit, zda je použitý font součástí používaného operačního systému. Druhá část obsahuje seznam několika čísel a je o poznání složitější.

První číslo udává velikost písma a spolu s druhým, udávajícím tloušťku písma, budou asi nejpoužívanější. Třetí číslo v sobě může uchovávat tři informace, těmi jsou kurzíva, podtržení a přeškrtnutí. Není-li žádné takové zvýraznění použito, má číslo hodnotu 0. Při použití kurzívy hodnota čísla stoupne o 1, při použití podtržení o 2 a při přeškrtnutí o 4. Dalo by se to přirovnat k třem dvoustavovým přepínačům, kde je 8 možných kombinací stavů. Při hodnotách 1, 3, 5, 7 bude zapnuta kurzíva, při hodnotách 2, 3, 6, 7 bude zapnuté podtržení a při hodnotách 4, 5, 6, 7 bude zapnuté přeškrtnutí. Další číslice při hodnotě 1 vyjadřuje vyplněnost pozadí, což vytvoří obdélník v barvě písma přes celý text. Předposlední číslice vyjadřuje při hodnotě 1 psaní na řádku, pro představu je to spíše posunutí o řádku výš. Poslední číslo udává údaj označený *kódová stránka*. Jde o označení kódování textu, které nejspíš nebude vůbec měněno.

Po nastavení stylu písma také může nastat potřeba přednastavení barvy písma, ta se na rozdíl od stylu nastaví velice jednoduše pomocí příkazu *barvaPera*. Barva pera může být jedna z šestnácti barev čtyřbitové palety, barva pojmenovaná v Imagine, anebo barva RGB.

```
barvaPera 11
barvaPera "tmavomodra2
barvaPera [100 150 125]
```

Důležitým prvkem u želv s tvarem písma, které se možná budou pohybovat je možnost zvednout své pero, aby při pohybu nemalovaly. Nastavit, aby pero bylo nahoře, lze v šabloně pomocí příkazu *stavPera* nastaveného na hodnotu *ne*. Ovšem jak želvy, tak i jiné objekty mají události, mezi nimiž se nachází i událost při vytvoření, díky této události lze nastavit, aby každá nově vytvořená želva zvedla své pero nahoru. Využitím této události, lze obejít hledání názvů vlastností a zadat jen příkazy, jako do příkazového řádku v Imagine.

```
priVytvoreni [peroNahoru]
```

Téměř každá prezentace je originál a některé želvy bude nutné dovytvořit pro konkrétní případy, ale jako ukázka šablony by měla stačit třída želv sloužících jako nadpisy.

```
novaTrida "Zelva "nadpis
[
  pismo [[Verdana][32 700 0 0 0 238]]
  barvapera! [50 50 50]
  priVytvoreni [pn]
]
```

Po zvládnutí tvorby tříd je možné přejít k tvorbě objektů z tříd. Při tvorbě stránek se nejspíš bude využívat nabídka povelů, ovšem při zadávání nové nabídky povelů se přepíše ta zadaná v šabloně. Je proto lepší jí doplnit dodatečně až po vytvoření stránky.

```
nova "sablonStranky [nabidkaPovelu [left [vl 30]]]
```

Nadpis

- Text na více řádků, první řádek.
Duhý řádek
- Jednořádkový text



Ilustrace 7: Ukázka stránky prezentace s želvami, které jsou vytvořené podle šablony pro nadpisy a šablony text.

U stránek se tedy moc nastavení během jejich tvorby provádět nemusí. Želvy jsou na tom podobně. Po vhodném vytvoření šablony není nutné výrazně měnit vytvářené želvy, tedy až na tvar, který bude mít každá želva jiný, jelikož bude obsahovat jiný text. Nová želva se zadáním tvaru vypadá následovně:

```
nova "nadpis [tvar [text[Text nadpisu]]]
```

Důležitým faktem u tvorby želv s tvarem textu je, že jim nelze změnit nastavení *zvětšení*, respektive lze, ale nebude to mít viditelný efekt.

6.2.2 Tvorba prezentačního software

Samotná aplikace je obstarávána procedurou pro posun, která zobrazí stránku podle hodnoty uvedené v proměnné *cisloPrvku*. Zapotřebí je k tomu operace *slovo* a příkaz *ukazStranku*. Operace skládá jedno slovo ze dvou slov. V tomto případě ze slova *stranka* a hodnoty proměnné *cisloPrvku*. Příkaz *ukazStranku* vybírá nově aktivní stránku, dle jejího názvu, který je výstupem operace *slovo*. Procedura předpokládá, že stránky se budou jmenovat *stranka1*, *stranka2* až *strankaN*.

příkaz posun

```
ukazStranku slovo "stranka :cisloPrvku
```

konec

Změnu hodnoty proměnné *cisloPrvku* provádí procedury *vpred* a *zpet*, které následně provádějí akce uložené v seznamu *prikazy*. Tento seznam obsahuje v každém prvku různé příkazy a operace a při každém zavolání jedné z posunovacích procedur se zavolá prvek odpovídající hodnotě *cisloPrvku* z tohoto seznamu. Je tedy zapotřebí, aby jedním z příkazů v prvku byla i procedura *posun*. Seznam *prikazy* může vypadat takto.

```
[[posun] [posun] [dosad "cisloPrvku 1 posun]]
```

Je vidět, že když prezentace dojde na konec, je automaticky přesměrována na první stránku změnou hodnoty proměnné *cisloPrvku*. Procedura pro posun *vpred* může tedy vypadat takto:

příkaz *vpred*

```
dosad "cisloPrvku soucet :cisloPrvku 1
pro [ž1] prvek :cisloPrvku :prikazy
konec
```

Proměnná *cisloPrvku* se zvýší o jedna a následně je vyzvána želva *ž1*, aby provedla akce z prvku *cisloPrvku* v seznamu *prikazy*. Fakt, že akce provádí *ž1*, se ve výsledku neprojeví, ale je nutné v případě nutnosti ovládní jiné želvy opět použít příkazu *pro*. Procedura předpokládá existenci objektu *ž1*, ale může být využita i jakákoliv jiná želva, která na stránce dozajista bude. Procedura *zpet* má problém, že nejnižší prvek seznamu *prikazy* je prvek 1, a proto nabude-li *cisloPrvku* hodnoty menší než 1 nebude se nic dít. To je sice v pořádku, jenže při záporné hodnotě této proměnné by bylo nutné opětovně volat proceduru *vpred*, a tedy několikrát opakovat slovní povel sloužící k jejímu zavolání, aby se záporná hodnota proměnné *cisloPrvku* zvýšila na kladná čísla. Z toho důvodu bude v proceduře *zpet* podmínka, zda je *cisloPrvku* menší než 1 a při kladné hodnotě této podmínky se *cisloPrvku* nastaví na hodnotu 1.

příkaz *zpet*

```
dosad "cisloPrvku rozdil :cisloPrvku 1
když cisloPrvku < 1 [dosad "cisloPrvku 1]
pro [ž1] prvek :cisloPrvku :prikazy
konec
```

Toto řešení umožňuje provádět akce po provedení procedury *posun*, například nějaké efekty objevení textu. Efekty před procedurou *posun* provádět lze, ale při pohybu zpět by se akce vztahovaly na špatné stránky.

```
[[posun] [posun] [skryj posun] [posun]]
```

Z příkladu si lze odvodit, že pokud je aktivní druhá stránka a zavolá se procedura *vpred*, jsou nejprve skryty aktivní želvy a teprve poté proveden samotný posun. Problém tkví v obousměrném procházení a bude-li aktivní čtvrtá stránka a zavolá se procedura *zpet* dojde ke stejnému efektu na čtvrté stránce. Ovšem tento problém lze obejít událostí stránky při otevření. A za předpokladu častějších využití těchto stránek lze vytvořit šablonu stránky z již vytvořené šablony a tuto událost do ní přímo vložit. Do tříd bohužel nelze vložit všechna nastavení a je proto potřeba využít události při vytvoření.

```
novaTrida "sablonStranky "sablonPriOtevreni  
[udalost! priOtevreni skryj]
```

Dalším řešením by také mohlo být vytvoření zpětnému seznamu k seznamu *prikazy*. V tomto řešení by seznam *prikazy* byl využíván k posunování na konec prezentace a druhý seznam k posunování na začátek.

Takových řešení jsou spousty a záleží spíše na tom, jaké požadavky jsou na prezentaci kladeny. Pomocí prezentace by se dalo vytvořit něco na způsob testu. Daly by se vytvořit šablony pro různé styly otázek, jako výběr jedné možnosti, nebo výběr všech pravdivých tvrzení. Takový test by neměl potřebu vracet se k zodpovězeným otázkám, které by byly na jednotlivých stránkách. Odpadla by tedy potřeba procedury pro vracení se na předchozí stránky a tedy problém s akcemi v seznamu *prikazy*. Završením by mohlo být generování otázky na každou stránku. Do seznamu *prikazy* lze vložit i popis pro vytvoření nové želvy, což by pro generování otázek bylo velice praktické. Takový popis pro vytvoření může vypadat následovně:

```
[nova "nadpis [  
  pozice [5 112]  
  tvar [text [Otázka č.1]]  
]]
```

Tvar želvy v podobě textu není nijak moc prozkoumán a bylo tedy nutné se této problematice věnovat, jelikož je text v prezentaci velice důležitá složka. Zadávání tvaru želvy bylo ukázáno v předchozích částech, ale vkládán byl jen krátký text. Pro vložení více řádků je už nutné pracovat s návodem na kreslení, což se velice podobá ovládání želvy. V návodu na kreslení tedy lze kreslit obrázky, které budou nastaveny jako tvar želvy. Při použití příkazu *text* bude ve tvaru vykreslen text. Tento text se bude vykreslovat z levého horního rohu. Dalo by se říct, že se na tomto místě nalézá kreslicí bod, či neviditelná želva, která tento text vykreslí. Po vložení druhého textu do návodu, se tento text bude opět vykreslovat z kreslicího bodu jehož pozice je stále stejná, druhý text tedy bude vykreslen přes první. Je proto nutné před zadáním druhého textu bod posunout. To se provede stejně, jako pohyb u želvy. Tvar se dvěma texty by tedy mohl vypadat takto:

```
tvar [  
  pn  
  text[text prvního řádku]  
  smer! 180  
  do 10  
  smer! 0  
  text[text druhého řádku]  
]
```

Na příkladu si lze povšimnout zvednutí pera, jelikož by docházelo k psaní při pohybu jako u klasické želvy. Taktéž je velice důležité

zachovat původní směr kreslicího bodu, tedy směr odpovídající dvanácti hodinám. Při nedodržení správného směru by i vypsáný text byl ovlivněn momentálním směrem. V případě *směr 180* by text byl vzhůru nohama a vypisoval by se z kreslicího bodu zprava doleva a zdola nahoru.



Ilustrace 8: Tvar želvy v podobě textu vykresleného při směru 180, kde červená tečka znázorňuje kreslicí bod

Zajímavým faktem je, že tvar pomocí návodu na kreslení je oříznut shora, zdola, zleva i zprava v linii prvních vykreslených bodů v tomto směru. Jako příklad lze uvést tento tvar:

```
tvar [  
  pn  
  směr! 180  
  do 10  
  směr! 0  
  text[text druhého řádku]  
]
```

Po prvním pohledu to vypadá, že text by měl být umístěn 10 pixelů od horního okraje, ale ve skutečnosti bude text přímo u tohoto okraje. Při odstranění příkazu *pn* bude změna jasně vidět. Toto je skutečně zapeklitý problém, ale díky pokusům se povedlo najít řešení v podobě příkazu *puntik*. Tento příkaz vytvoří vyplněný kruh o zadaném průměru a v momentální barvě pera. Ovšem při použití průměru 0 může být příkaz *puntik 0* brán jako krajový bod, přestože ve tvaru není

vykreslen ani pixel. Tvar textu posunutého od horního okraje by mohl vypadat třeba takto:

```
tvar [  
  pn  
  puntik 0  
  vz 10  
  text[text druhého řádku]  
]
```



Ilustrace 9: Rozdíly mezi identickými tvary želv, které se liší pouze v příkazu *puntik 0*

6.2.3 Efekty prezentace

Součástí prezentací jsou i jisté grafické efekty, které většinou animují změnu prezentace. Samotná změna stránek by se v Imagine Logo dala řešit pomocí objektu papír. Při použití papíru je nutné vytvořit celou prezentaci na jedné stránce, což není nejvhodnější, ale možné to je. Na rozdíl od papíru jsou želvy pro efekty přímo stvořené. Jeden z nejpoužívanějších je třeba efekt příjezdu z poza okraje.

```
příkaz příjezdP :XK :XS  
  kdyz :XS >= :XK [  
    xsour! :XS  
    cekej 5  
    příjezdP :XK (:XS-1)
```


]
konec

Procedura prijezdP vytváří animaci příjezdu textu z pravé strany obrazovky. Procedurou jsou vyžadována dvě čísla, kde XK je hodnota souřadnice x koncové pozice a XS je hodnota souřadnice x startovní pozice. Prvním krokem procedury je kontrola, zda je souřadnice x startovní pozice větší, nebo rovna koncové. V případě splnění se momentální souřadnice x želvy změní na hodnotu XS. Procedura počká 5 milisekund a znovu zavolá proceduru prijezdP se stejnou koncovou hodnotou souřadnice x a hodnotou startovní zmenšenou o 1. Obdobně by vypadaly procedury pro příjezd zleva, zdola a shora. Ze všech čtyř směrů budou při tvorbě prezentace nejvhodnější zprava a zdola, jelikož text bude čten zleva doprava a shora dolů.

Další efekty mohou být určeny ke zvýraznění textu. Metro efek vytváří střídavý pohyb doleva a doprava. V efektu je nejprve po jedné zmenšována souřadnice x želvy, poté zvětšována a na konec opět zmenšována, aby se želva dostala na svou původní pozici.

příkaz metro

opakuji 2 [
opakuji 50 [xsour! xsour-1 cekej 10]
opakuji 100 [xsour! xsour+1 cekej 10]
opakuji 50 [xsour! xsour-1 cekej 10]
]
konec

Není dobré nechat efekt vícekrát opakovat, jelikož je při pohybu značně nečitelný, ovšem pro upoutání pozornosti je malý počet opakování velice vhodný. Upozornit lze i změnou barvy.

```

příkaz blikaniBarevne
dosadZde "puvodniBarva barvaPera
opakuj 5 [
  barvaPera! " cervena
  tvar! tvar
  cekej 100
  barvaPera! :puvodniBarva
  tvar! tvar
]
konec

```

Procedura mění barvu pera želvy a poté znovu vykreslí tvar, který je vykreslen v nové barvě. Původní barva je uložena do lokální proměnné, takže po skončení procedury je text vykreslen ve své původní barvě. Cyklus změny barev se opakuje pětkrát, aby nedocházelo k zhoršení čitelnosti, jak už bylo řečeno.

Další efekty by mohly být se změnou velikosti textu. Například jakési velikostní blikání, či zvětšení nějaké nejmenší velikosti. V takových procedurách je nutné měnit pozici želvy, aby střed textu, který reprezentuje, byl stále na stejném místě. Jelikož není způsob, jak zjistit velikost želvy, tak jediný způsob, jak získat nějaké hodnoty je využití operace *velikostTextu*. Tato operace po aplikování na text vrátí seznam se čtyřmi prvky. První dva vyjadřují šířku a výšku textu, další dva vzdálenost základní a akcentové linky a mezeru mezi řádky, ale ty nebudou zapotřebí. S velikostí a šířkou textu by bylo tedy možné vytvoření procedury se změnou velikosti, ovšem tato operace má značný nedostatek. Operace *velikostTextu* totiž udává hodnoty jen pro font Times New Roman, a proto je-li zvolen jiný font, tyto hodnoty budou nepoužitelné. Z tohoto důvodu nejsou procedury v této práci ukázány. Ovšem pro zviditelnění textu je možné například zvětšovat

část textu. Kupříkladu na hlasový povel by se postupně zviditelňovaly jednotlivé body na stránce prezentace a při zviditelnění dalšího, by se vracela původní velikost již zviditelněných.

U prezentací ovládaných hlasem je dobré mít i náhradní způsob ovládání. Ať už z jakéhokoliv důvodu se může stát, že nebude možné prezentaci ovládat hlasem a bude proto nutné použít klávesnici. Nabídka kláves je velice podobná nabídce povelů. Je také tvořena seznamem pro povel, tedy klávesu, a akci. Jako klávesy mohou být použity klasické znaky nebo číslice z klávesnice, ale také jejich kombinace s klávesami Shift, Ctrl a Alt. Krom písmen a číslic mohou být použity i speciální klávesy, jako mezerník, Enter, Insert, Home, či směrové šipky. Tyto klávesy mohou být opět stisknuty spolu s klávesami Shift, Ctrl nebo Alt.

```
ž1'nabidkaKlaves! [Mezernik [...] ctrl.F5 [...] 5 [...]  
  shift.ctrl.4 [...] shift.ctrl.alt.SipkaDolu [...]]
```

Jak je vidět z ukázky, nabídka kláves může být přiřazena i jiným objektům. Než jen stránkám, nebo hlavnímu oknu. Pro prezentaci se bude postupovat obdobně, jako u nabídky povelů a pro posunování jsou nejlepší tlačítka vlevo a vpravo v nabídce hlavního okna. Což právě vylučuje použití těchto kláves v nabídce jiného objektu.

6.3 Hra – Dráha

Hra je založena na principu vyhýbání se překážkám a díky tomu jednoduchému principu se může objevit v nespočtu variací. Namátkou bylo vybráno auto, které se vyhýbá překážkám na vozovce. Další možnost může být třeba loď vyhýbající se ostrůvkům a útesům, nebo také letadélko vyhýbající se ptákům a podobně.

Dalším krokem pro vylepšení aplikace může být přidání činností objektům, kterým je nutno se vyhnout. První činností by určitě měla být změna směru pro objekty schopné takovouto změnu provést. V případě s lodí by změnu prováděly protiplující lodě a ostrůvky by se pohybovaly stále stejným směrem.

Jinou činností mohou být různé formy útoků. Tyto útoky mohou být plošné, či přímo zaměřené na vozidlo, či dopravní prostředek ovládaný hráčem. Jako příkladu můžeme využít vesmír, kde hráč řídí raketu. Po hrací ploše se budou pohybovat různé kameny, které budou mít stálý směr, popřípadě měněný při kolizi. Dále se zde budou objevovat i jiné rakety, které budou vystřelovat různé projektily. Tyto střely mohou mít neměnný směr. Útočící raketa se po přiletu na obrazovku zaměří na hráčovu raketu a vypálí projektil. Nebo po přiletu vypálí tři projektily, jeden přímo před sebe a dva o několik stupňů na každou stranu.

Také je možné, že raketa vyše řízenou střelu, která bude sledovat hráčovu raketu, či sama útočící raketa se bude snažit o kolizi. V tom případě je ale nutné umožnit určitý druh obrany, což vede k přidání další činnosti, ale tentokrát k hráčem řízenému vozidlu. V příkladě s raketou by hráčova raketa mohla také střílet a tím odstranit rakety, či projektily snažící se o kolizi.

Princip aplikace je v horizontálním, či vertikálním pohybu pozadí, což tvoří efekt pohybu vozidla. V této konkrétní aplikaci je pohyb horizontální a hráčem ovládané auto se pohybuje vertikálně, aby se mohlo vyhýbat překážkám. Překážky se pohybují stejně rychle jako pozadí a také horizontálně, což vyvolává iluzi, že jsou umístěny na povrchu dráhy.

6.3.1 Příprava na hru

Před zahájením samotné hry je nutné vygenerovat dráhu. K vygenerování dráhy slouží procedura *vytvorDrahu*, jejímž vstupem je číslo určující délku dráhy. Výstup této procedury je číslo s počtem číslic rovnajícím se vstupu procedury, a tedy délce dráhy. Toto číslo je uloženo do proměnné *draha*. Každá číslice z tohoto čísla zastupuje jeden obrázek ze seznamu *cesty*. Ne všechny tyto obrázky na sebe navazují, a proto musí být řazeny dle určitých pravidel. Například obrázek zastoupený 1 nesmí být před či za 4 a podobně. Procedura *vytvorDrahu* tedy vygeneruje číslice reprezentující seznam obrázků na sebe navazujících tak, aby tvořili jednolitou dráhu.

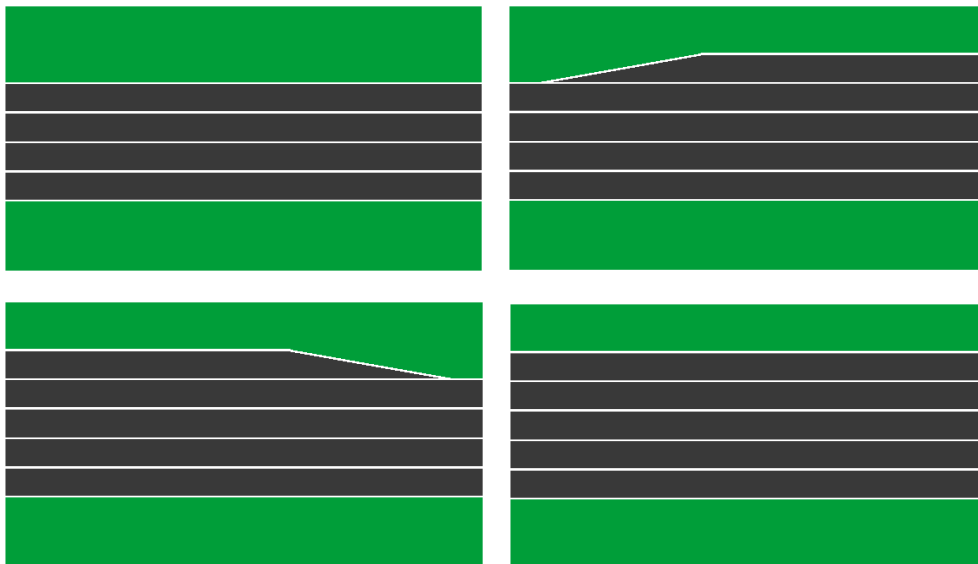
```
příkaz vytvorDrahu :delka
dosad "draha 1
opakuji (:delka - 1)
[
  kdyz posledni :draha = 1 nebo :draha = 3
  [dosadzde "vybranyPrv vyberPrv 1 2]
  kdyz posledni :draha = 2 nebo :draha = 4
  [dosadzde "vybranyPrv vyberPrv 3 4]

dosad "draha vlozposledni :vybranyPrv :draha
]
konec
```

Když obrázky přirovnáme k silnici, tak první je čtyřproudá a čtvrtý pětiproudá, druhý je na začátku čtyř a na konci pětiproudá a třetí naopak. Na začátku procedury je jako první do seznamu *draha* uložena jednička, takže všechny dráhy budou začínat stejně. Poté je zopakován cyklus vybírání dalších obrázků. Cyklus je zopakován podle proměnné

delka, ale zmenšený o jedna z důvodu prvního před uloženého obrázku. Cyklus obsahuje 2 podmínky, pro dva obrázky končící čtyř a dva pětiproudě. Podmínky kontrolují poslední číslo v seznamu a podle toho, kterému odpovídá je zavolána procedura *vyberPrv* s příslušnými hodnotami. Návrátová hodnota z procedury *vyberPrv* je poté uložena do lokální proměnné *vybranyPrv*, jejíž hodnota na konci procedury přidána na konec seznamu *draha*. Procedura *vyberPrv* vybírá náhodně ze dvou čísel, tedy 0 a 1, a podle toho jestli je vybrané číslo rovno 0 je vrácena první nebo druhá vstupní hodnota.

```
příkaz vyberPrv :prvni :druhy  
dosadzde "vybranyPrv nahodne 2  
kdyzjinak :vybranyPrv = 0  
  [vysledek :prvni]  
  [vysledek :druhy]  
konec
```



Ilustrace 10: Obrázky použitelné pro pozadí

6.3.2 Průběh hry

Samotný pohyb pozadí a překážek dělá procedura *simJizdy*. V první části procedury je hra uvedena do startovního stavu.

příkaz *simJizdy*

```
ž1'xsour! :X
```

```
ž1'tvar! prvek prvek 1 :draha :cesty
```

```
ž2'domu
```

```
ž2'tvar! prvek prvek 2 :draha :cesty
```

```
auto'domu
```

```
zrusobjekt vsechnyod "prekazkaP
```

```
zrusobjekt vsechnyod "prekazkaD
```

```
vytvorPrekazky "ne prvek 2 :draha
```

```
dosadzde "pocProjeti 3
```

```
odted veta [ž1 ž2] vsechnyod "prekazkaD
```

Želvy ž1 a ž2 tvoří již zmíněné pozadí, pohybují se horizontálně a mají tvar vozovky. Na začátku procedury jsou tyto želvy přesunuty na startovní pozice a jsou jim dány tvary obrázků ze seznamu *cesty*, ve kterém jsou uloženy obrázky vozovky. Tvary jsou z tohoto seznamu vybrány podle hodnot prvního a druhého prvku v seznamu *draha*. Přesunuta je i želva *auto*, která reprezentuje vozidlo ovládané uživatelem. Po přesunutí a změně tvaru u těchto želv jsou odstraněny všechny želvy vytvořené podle šablon *prekazkaP* a *prekazkaD*. Z důvodu odstranění překážek z předešlé hry. Poté následuje vytvoření nových překážek procedurou *vytvorPrekazky*, dosazení do lokální

proměnné *pocProjetí* hodnotu 3 a zvolení želv aktivních na stránce. Aktivními se stanou ž1, ž2 a želvy nově vytvořené procedurou *vytvorPrekazky* podle šablony *prekazkyD*.

Procedura *vytvorPrekazky* vytváří nové želvy podle jedné ze zmiňovaných šablon. První vstupní hodnota právě rozhoduje, která šablona bude použita a druhá určuje, který obrázek bude v pozadí, na čemž závisí, jak se přesně budou želvy generovat.

```
příkaz vytvorPrekazky :prvniPrekazka :sirkaDrahy
  dosadzde "pozicePrekazek vygPoz
```

```
opakuji pocet :pozicePrekazek [
  dosadzde "momentPoz []
  dosadzde "momentPoz vlozposledni prvek poc
    :pozicePrekazek :momentPoz
  dosadzde "momentPoz vlozposledni prvek
    ( 1 + nahodne pocet :Y) :Y :momentPoz
```

```
kdyz nahodne 2 = 0
```

```
[
```

```
  kdyz :sirkaDrahy = 4
```

```
    [dosadzde "momentPoz nahrad 2 :momentPoz
      (67 + prvek 2 :momentPoz)]
```

```
  kdyz aZaroven :sirkaDrahy = 3
```

```
    prvek 1 :momentPoz < 750
```

```
    [dosadzde "momentPoz nahrad 2 :momentPoz
      (67 + prvek 2 :momentPoz)]
```

```
  kdyz aZaroven :sirkaDrahy = 2
```



```

prvek 1 :momentPoz > 750
[dosadzde "momentPoz nahrad 2 :momentPoz
(67 + prvek 2 :momentPoz)]
]

kdyzjinak :prvniPrekazka
[nova "prekazkaP [pozice :momentPoz]]
[nova "prekazkaD [pozice :momentPoz]]
]
konec

```

Pro vytvoření želvy je nejprve nutné mít nějaké souřadnice, na které je umístíme, ty se získají pomocí procedury *vygPoz*, která vrátí seznam souřadnic x pro nové želvy. Následně se spustí cyklus, podle délky seznamu *pozicePrekazek*, ve kterém je uložen výsledek z procedury *vygPoz*. V každém průběhu cyklem se vytvoří lokální dvouhodnotový seznam *momentPoz* obsahující tolikátý prvek z *pozicePrekazek*, kolikátý probíhá cyklus a náhodný prvek ze seznamu Y, který obsahuje souřadnice y drah, nebo proudů, z obrázku čtyřproudé silnice. Tento seznam je nutné vytvořit podle používaných obrázků. Na konci cyklu je podle první vstupní hodnoty procedury vygenerována želva na pozici z proměnné *momentPoz* podle jedné z šablon. Tato proměnná se však může ještě náhodně změnit. Pokud je obrázek pozadí, na který je želva vytvářena aspoň částečně pětiproudý, může být druhá hodnota *momentPoz*, tedy souřadnice y zvýšena o takovou hodnotu, aby se želva po vytvoření vyskytovala o dráhu výš. Probíhá tedy kontrola, zda je na pozadí druhý třetí nebo čtvrtý obrázek. Číslo obrázku je druhá vstupní hodnota procedury, tedy *sirkaDrahy*. U jen částečně pětiproudých obrázků, tedy druhého a třetího, je kontrola druhé vstupní hodnoty procedury doplněna kontrolou

souřadnice x, aby nedocházelo k posunutí želv do míst, kde už, či ještě není pátá dráha.

Jak už bylo řečeno o vytvoření souřadnic se stará procedura *vygPoz*. Ta dále využívá procedury *JednaPoz* ke generování čísel v určitém rozmezí, které záleží na velikosti stránky a velikosti obrázků na pozadí.

příkaz *jednaPoz*

```
dosadZde "poziceZ (397 + nahodne 1091)
```

```
vysledek :poziceZ
```

konec

Vygenerovaná pozice je kontrolována procedurou *kontrolaPoz*, zda už taková, či jí blízká neexistuje. Je ale velice důležité dbát na dostatečný prostor pro tvorbu želv, jelikož z logiky programu nemohou být dvě želvy vedle sebe ve vertikálním směru. Proto nemohou být na úseku 100 pixelů více jak tři želvy, při dodržení horizontálního rozestupu 40 pixelů.

příkaz *kontrolaPoz* :poziceZ :pozicePrekazek

```
kdyzjinak pocet :pozicePrekazek = 0
```

```
[vysledek :poziceZ]
```

```
[
```

```
opakuj pocet :pozicePrekazek [
```

```
kdyzjinak xor
```

```
(:poziceZ > ( 40 + prvek poc :pozicePrekazek ))
```

```
(:poziceZ < ( - 40 + prvek poc :pozicePrekazek ))
```

```
[]
```

```
[vysledek kontrolaPoz jednaPoz :pozicePrekazek]
```

```
]
```

```
vysledek :poziceZ
```

```
]
```

```
konec
```

Pokud je seznam *pozicePrekazek* prázdný, je vstupní hodnota kontrolované pozice ihned vrácena jako výsledek. Není-li tomu tak, je spuštěn cyklus o počtu kroků rovnajících se velikosti tohoto seznamu. V každém průběhu cyklu je kontrolovaná pozice porovnávána s hraničními hodnotami od pozic ze seznamu *pozicePrekazek*. Hraniční hodnoty jsou pozice tohoto seznamu zvětšené a zmenšené o 40. Pokud je hodnota kontrolované pozice mezi těmito hraničními hodnotami, je vráceno jako výsledek opětovné zavolání procedury pro kontrolu, jejímiž vstupními hodnotami je výsledek opětovného zavolání procedury pro vygenerování jedné pozice a seznam *pozicePrekazek*. Skončí-li cyklus bez nalezení kolize se souřadnicemi předchozích želv, je jako výsledek vrácena kontrolovaná souřadnice.

```
příkaz vygPoz
```

```
dosadzde "pozicePrekazek []
```

```
opakuj 7 [
```

```
dosadzde "poziceZ jednaPoz
```

```
dosadzde "poziceZ kontrolaPoz :poziceZ :pozicePrekazek
```

```
dosadzde "pozicePrekazek vlozposledni :poziceZ
```

```
:pozicePrekazek]
```

```
vysledek :pozicePrekazek
```

```
konec
```

Procedura *vygPoz* tedy obsahuje cyklus s generováním souřadnice procedurou *jednaPoz*, s kontrolu této souřadnice procedurou *kontrolaPoz* a s uložením zkontrolované hodnoty do seznamu, který je na konci procedury vrácen jako výsledek.

Druhá část procedury *simJizdy* obsahuje cyklus, který se stará o běh hry. Tento cyklus každých pět milisekund posune všechny aktivní želvy dopředu o jeden pixel. Směr dopředu je pro vybrané želvy směr 270, tedy vlevo. Po posunutí jsou kontrolovány dvě podmínky, pro obě želvy na pozadí se kontroluje, jestli jejich souřadnice *x* se nerovnájí hodnotě v proměnné *X*. Tato proměnná obsahuje hodnotu souřadnice *x* za levým okrajem stránky. Pokud se souřadnice želvy shoduje s hodnotou proměnné, je zkontrolována proměnná *pocProjet*, zda nepřevyšuje počet prvků v seznamu *draha*. Pokud převyšuje, je zastaven celý průběh hry. Jinak je želva, jejíž souřadnice nebyla shodná s hodnotou proměnné *X* přesunuta na svou domovskou pozici, kterou mají obě želvy za pravým okrajem stránky. Po přesunutí je želvě dán nový tvar ze seznamu *cesty*, podle hodnoty prvku ze seznamu *drahy*, odpovídajícímu hodnotě v proměnné *pocetProjeti*. Na závěr jsou zrušeny všechny želvy vytvořené podle jedné z šablon, po zrušení jsou nově vygenerovány a je zvolen nový seznam aktivních želv na stránce. Ve zkratce tedy procedura přesunuje objekty, které jsou již mimo stránku před stránku, aby mohl pohyb pokračovat.

Kazdych 5

[

do 1

kdyz ž2'xsour = :X

[kdyz :pocProjeti = 1 + pocet :draha [zastavse]

ž1'domu

dosadzde "tvarDrahy prvek :pocProjeti :draha

ž1'tvar! prvek :tvarDrahy :cesty

dosad "pocProjeti :pocProjeti + 1

zrusobjekt vsechnyod "prekazkaP

```

vytvorPrekazky "ano :tvarDrahy
odted veta [ž1 ž2]
  veta vsechnyod "prekazkaD vsechnyod "prekazkaP
]

kdyz ž1'xsour = :X
[kdyz :pocProjeti = 1 + pocet :draha [zastavvse]
  ž2'domu
  dosadzde "tvarDrahy prvek :pocProjeti :draha
  ž2'tvar! prvek :tvarDrahy :cesty
  dosad "pocProjeti :pocProjeti + 1
  zrusobjekt vsechnyod "prekazkaD
  vytvorPrekazky "ne :tvarDrahy
  odted veta [ž1 ž2]
  veta vsechnyod "prekazkaP vsechnyod "prekazkaD
]
]
konec

```

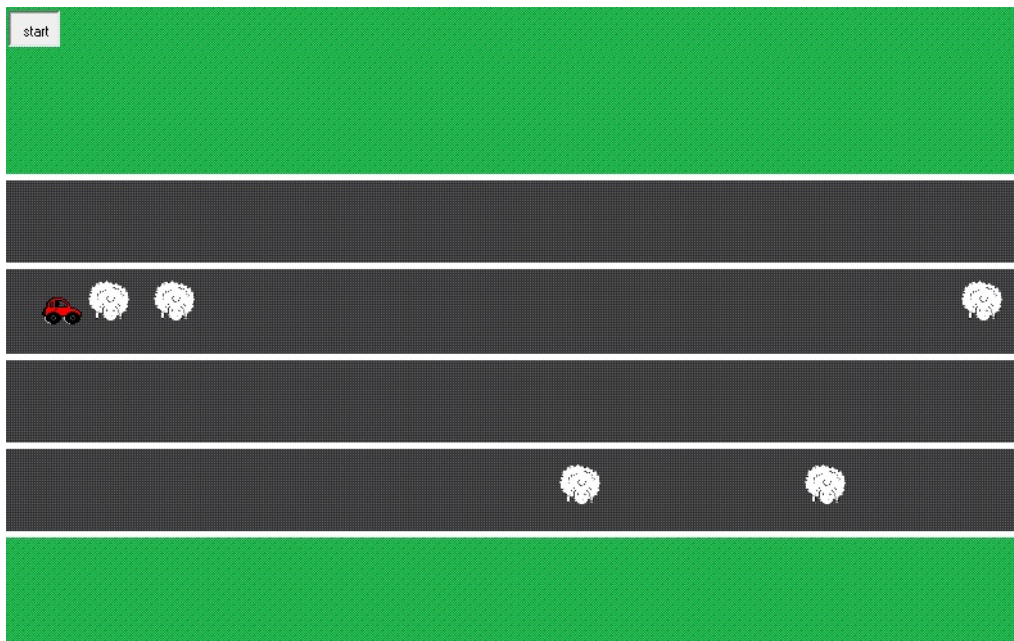
Želvy tvořící pozadí se nejlíp nastaví ručně. Stačí dát pero nahoru, nastavit směr 270 a typ oblasti bez hranic v nastavení obou želv. To znamená, že nebudou kreslit, budou se pohybovat doleva a mimo stránku. Při tvorbě šablon pro přákázky to bude velice podobné.

```

novaTrida "zelva "prekazkaD [
  predpona prekazkaD smer 270 stavpera pn tahnuti ne
  tvar :tvarPrekazky zvetseni 0.3
  typoblasti bezhranic reagujeNaSrazku ano
  priVytvoreni [udalost! "priSrazce zastavVse]
]

```

Je ale nutné nastavit želvám, aby reagovali na srážku a jak na ní mají reagovat. Pro zastavení běhu hry stačí využít příkaz *zastavVse*, při události *priSrazce*. Bohužel nelze v šabloně nastavit událost při srážce stejně, jako událost při vytvoření. Je proto nutné zapsat do události při vytvoření zadání nové události při srážce, což je vidět na příkladu.



Ilustrace 11: Ukázka hry s překážkami ve tvaru ovcí

Na závěr je nutné přidat uživatelem ovládané vozidlo, kterému je opět nastaveno reagování na srážku a je pojmenováno *auto*. Pohyb vozidla řídí procedura *prejed* volaná z nabídky hlasových povelů.

```
left [prejed 1] right [prejed -1]
```

```
příkaz prejed :smer
```

```
opakuje 70 [auto'ysour! auto'ysour + :smer cekej 10]
```

```
konec
```

Díky této proceduře se vstupní hodnotou 1 nebo -1 je postupně měněna souřadnice y vozu a dochází tak k přejetí do jiné části vozovky.

Start hry by mělo být možné provést jak hlasovým povelům, tak pomocí přepínače, který při zapnutí vykoná *vytvorDrahu 6 simJizdy* a při vypnutí *zastavVse*. Stejně tak budou vypadat hlasové povely. Použity mohou být slova *start* a *stop*.

start [vytvorDrahu 6 simJizdy] stop [zastavVse]

7 Rozpoznání řeči v češtině

K ovládní Imagine Logo hlasem pomocí češtiny bylo v první řadě nutné prozkoumat nabídku programů pracujících s rozpoznáním češtiny. Rozpoznání češtiny je v porovnání s angličtinou komplikovanější, jelikož jedno slovo může být v mnoha tvarech. Možná tento problém vede k ne příliš velkému počtu aplikací rozpoznávajících češtinu. Nejznámější společnosti poskytující software pro rozpoznávání češtiny jsou Consulting Company Novasoft a Newton Technologies. Tyto společnosti nabízejí software ke spojitému diktování češtinou do počítače (10, 11). Bohužel tento způsob diktování není vhodný pro použití s Imagine Logo.

Problém je v přesunu textu z diktovacího programu do Imagine. Ten sice umí načítat obsahy textových souborů a pracovat s nimi, ale diktovací programy jistě nezvládnou ukládání po každém nadiktovaném slově. U spojitého diktování je totiž o konečném tvaru slova, které bylo nadiktováno, rozhodováno pomocí kontextových technik a právě bez kontextu ve formě dalšího textu by mohlo docházet k záměnám slov.

Řešení těchto společností tedy byla shledána nevhodnými pro použití s Imagine Logo. Bylo proto nutné pokračovat v hledání programu pro diktování v češtině, ovšem bez technologie spojitého diktování.

Hledaný program by měl fungovat na principu porovnávání se zadanými vzorky. Jedno takové řešení bylo vyvinuto laboratoří počítačového zpracování řeči Technické univerzity v Liberci. Tato laboratoř se nazývá SpeechLab a úzce spolupracovala se společností Newton Technologies při vývoji diktovacího programu NewtonDictate (9). SpeechLab vyvinul aplikaci MyVoice, která slouží k ovládní počítače jen za pomoci hlasu. MyVoice simuluje počítači

vstup z klávesnice a myši. Program obsahuje akce, které mohou simulovat stisky jednotlivých tlačítek klávesnice. A to je právě ta vlastnost, která dokonale koresponduje s Imagine Logo, jelikož Imagine může mít nastavené k tlačítkům na klávesnici akce, které se mají při stisknutí provést. Při kontaktu vedoucí SpeechLabu profesor Nouza zdělil, že je tento program velice flexibilní a je tedy možné přidávat nové povely.

Po nalezení vhodného programu, který je schopen spolupracovat s Imagine Logo a je sám dostatečně variabilní bylo nutné se obrátit na distributora programu firmu Fugasoft. Firma byla požádána o poskytnutí programu MyVoice v jakékoliv formě, či podobě, která by umožnila vyzkoušet ovládání Imagine Logo hlasem v češtině. V odpověď na tuto žádost byla firmou Fugasoft nabídnuta možnost za nákup jedné licence povolit aktivaci programu MyVoice na třech počítačích. Této velkorysé nabídky bohužel nebylo využito z důvodů cenových, jelikož hodnota jedné licence je 5.500,-Kč bez DPH.

Vyzkoušení s Imagine Logo tedy nebylo provedeno, ale vlastnosti obou programů napovídají, že by neměl být v jejich spolupráci větší problém.

Na webu SpeechLabu je alespoň možnost zhlédnutí krátkého komentovaného videa s ukázkou serfování na internetu pomocí programu *MyVoice*. V ukázce je vidět úspěšnost rozpoznání hlasu a praktické využití ovládání hlasem pomocí tohoto programu.

7.1 Alternativní řešení pro rozpoznání řeči v češtině

Jelikož SAPI funguje na principu rozpoznávání frází, nebo povelů, je možné používat slova stejně znějící, jak v češtině, tak i v angličtině. Bohužel takových slov mnoho není. Dále je také možné používat zkomoleniny anglických slov, které znějí jako česká slova.

K tomuto způsobu se hned namane řešení anglického diktovacího programu, do kterého se budou diktovat česká slova. Text, který bude výstupem programu, se použije jako povely a bude tak možné ovládnání češtinou. Toto řešení ale není možné, jelikož programy pro rozpoznávání pracují na principu porovnávání vzorků. Programy tak řečeno příkazové, jako je Imagine Logo, buďto na slovo, které nepoznají, nezareagují, nebo nějakým způsobem oznámí, že nebylo rozpoznáno, co bylo řečeno. Tyto programy totiž nejsou určeny pro převod řeči na text. Nereagování nebo oznámení o nerozpoznání slova se u programů pro spojitě diktování, které jsou na druhou stranu určeny výhradně pro převod řeči na text, nenastává, ale slovo je vybráno ze slovníku slov. V takovém vybírání je důležité umístění slova ve větě, kontextové informace a podobně. Důležité je, že vždy je vybráno nějaké slovo.

Jednou možností je postup vymýšlení zkomolenin a jejich případné zkoušení pomocí programu Speech u systémů Windows, kterého využívá i Imagine Logo. Pomocí převodu textu na řeč je možné najít nějaké zkomoleniny, které jsou v angličtině vyslovovány jako česká slova. U Imagine se používá jednoduchý příkaz *rekni*, který spustí převod textu na řeč, nebo by se dalo říct povězení.

```
rekni "slovo
```

```
rekni [text z více slov]
```

Při tvorbě zkomolenin je lépe vybírat slova, která neobsahují česká písmenka, jelikož některá se v angličtině nepoužívají. Kupříkladu písmeno ř nemá přepis do angličtiny, který by tomuto písmenu odpovídal.

Následuje tabulka s několika příklady, které by se daly využít v aplikacích tvořených v Imagine Logo. Například slova *další* a *zpět* by se hodila pro posunování stránek prezentace.

české slovo	anglické slovo
vpravo	wbravo
vlevo	wlevo
další	dalshee
zpět	zpyet
čekej	che key
vyčisti	wicheesti
jed'	yed

Ilustrace 12: Ukázka přepisu českých slov do angličtiny

8 Testování

Pro testování byla zvolena skupina osmi osob ve věku 15 až 50 let. Osoby z testované skupiny měly různé stupně znalostí angličtiny, počínaje téměř nulovými. Samotné testování probíhalo formou konzultací s jednotlivými osobami a využito bylo mikrofonu headsetového typu.

V každé konzultaci bylo nejprve na jednoduchém příkladu ukázána a testovanou osobou odzkoušena funkčnost. K ukázce a odzkoušení bylo využito virtuálního přepínače z článku doktora Tomcsányiho (7, s. 223) popsaného v kapitole 3 (3.1.1 *Pokročilé techniky*).

Dalším krokem bylo hlasové ovládání u vytvořených aplikací. První z nich byl prezentační software s využitím příkazů *next a dalshee*. Druhou zkoušenou aplikací byl výukový program a na závěr hra pomocí příkazů *left a right*.

Poslední částí testování bylo vyzkoušení několika zkomolenin, pro využití v češtině.

Z pozorování testovaných osob vyplynuly zajímavé výsledky. Ve většině případů se neznalost, či jen minimální znalost angličtiny, při hlasovém ovládání ukázala výhodou. U osob s vyšší znalostí angličtiny totiž docházelo k naučeným a zažitým výslovnostním chybám, které u osob neznalých šli opravit opětovným nadiktování povelu. Zažité chyby ovšem bylo velice těžké odstranit, k čemuž se ale osvědčil výukový program s tlačítkem pro vyslovení slova počítačem. Testované osoby se následně pokoušely co nejpřesněji zopakovat slovo a tím zlepšovaly svou výslovnost.

Další vyzorovanou zajímavostí je problém konkrétní testované osoby s konkrétním slovem. Pro představu se u dvou testovaných

vyskytl problém se správným vyslovením slova *computer*. Z těchto dvou osob se jedné po několika pokusech podařilo dosáhnout alespoň částečného rozpoznávání povelu, ovšem úspěšnost nepřekročila 50 procent. Druhé osobě se dosáhnout rozpoznání povelu téměř nepodařilo.

Při využití zkomolenin k možnosti používání českých povelů dělaly problémy nepřesné přepisy českých slov do zkomolenin anglických. Slovo *dalshee*, jako přepis českého slova *další*, použité u prezentační aplikace bylo shledáno vcelku úspěšně přepsané. Úspěšnost rozpoznání povelu se ještě dala zvýšit protažením písmenka *í*, které je v přepisu reprezentováno jako *ee*. Na druhou stranu slovo *zpyet*, jako přepis slova *zpět*, se tak úspěšně neukázalo. Bylo nutné ho vyslovovat spíše jako *spijet*, aby docházelo k rozpoznání povelu.

Ovládání pomocí hlasu a tedy i rozpoznání řeči pomocí SAPI se shledalo s kladnými ohlasy a nebyl mu vytýkán žádný výrazný nedostatek. Aplikace použité při testování byly označeny jako výrazně využitelné. Hlavně tedy výukový program pro zlepšování výslovnosti a prezentační program nejenom k prezentování. Nejlepší ohlasy na aplikace ovládané hlasem byly od osob, které počítač téměř nevyužívají. Dovedly by si například představit program pro vyhledávání dopravních spojů ovládaný hlasem a podobně.

9 Závěr

Pro ovládání Imagine Logo hlasem bylo zkoumáno rozpoznávání řeči u systémů Windows. Ze zkoumaných systémů je rozpoznání nejvyužitelnější u Windows XP, jelikož v tomto operačním systému není rozpoznání řeči závislé na jazykové verzi. V této nezávislosti se skrývá velká výhoda, jelikož v systémech Vista a 7 s českou lokalizací není možné zaktivovat rozpoznání řeči, tedy pokud systémy nejsou ve verzi Ultimate, která umožňuje změnu jazyka zobrazení.

Důležitým faktorem u rozpoznání řeči je vliv okolního prostředí a tedy i kvalita mikrofону. Při použití obyčejného stolního mikrofónu je potřeba omezit či úplně zamezit hluky prostředí. Lepších výsledků lze dosáhnout použitím směrového mikrofónu, který hluky prostředí nezaznamená, anebo jen v malé míře. V mnoha případech je také výhodou mechanické tlačítko mikrofónu, které slouží k vypnutí mikrofónu v případech, kdy uživatele potřebuje mluvit a nediktovat při tom povely.

S fungujícím mikrofónem a rozpoznáním řeči lze začít pracovat s ovládáním hlasem v Imagine Logo. Se znalostí tvoření hlasového ovládání byly vytvořeny tři aplikace ovládané hlasem. Témata aplikací byla volena, aby pokryla co nejširší tematický okruh. Cílem aplikací je poskytnout návod na použití ovládání hlasem v Imagine Logo.

Vytvořené aplikace jsou tedy aplikace pro výuku, fungující na principu karet, hra založená na jednoduchém vyhýbání překážkám a prezentační program, tedy spíše šablona pro tvorbu prezentací.

Jedním z cílů práce bylo i vytvoření řešení pro ovládání hlasem pomocí češtiny. Pro toto řešení byl nalezen vhodný program MyVoice, který simuluje stisky klávesnic na klávesnici a tlačítek myši. Na tyto stisky může Imagine navázat, jelikož je schopný k stiskům kláves

přiřazovat akce. Bohužel se nepodařilo program MyVoice získat a vyzkoušet jeho fungování s Imagine Logo.

Zprovoznění ovládání hlasem tedy spočívá ve fungování mikrofону a instalaci rozpoznání řeči, pokud to operační systém umožní. Posledním krokem pro ovládání hlasem je už jen zadání povelů a akcí, které tyto povely vyvolávají, do Imagine.

Literatura

- [1] Blaho, A., Kalš, I. Imagine reference guide, Logotron, 2002.
- [2] Blaho, A., Kalš, I., Tomcsányi, P. Imagine... Nová generacia tvořivého prostredia pre výučbu. In: Poškole 2000: Sborník semináře. Praha: ČVUT, 2000. – P. 88-93- - ISBN 80-01-02172-6.
- [3] Blaho, A., Kalš, I. Imagine Logo, učebnice programování pro děti. Brno: Computer Press 2006. 48 s. ISBN 80-251-1015-X.
- [4] Generation5 [online]. James Matthews. 23/10/2002 [cit. 2011-04-15]. How Does Speech Recognition Work?. Dostupné z WWW: <<http://www.generation5.org/content/2002/howsrworks.asp>>.
- [5] KRČMOVÁ, Marie . Fonetika a Fonologie [online]. 2008 [cit. 2011-04-16]. Foném a jeho vymezení. Dostupné z WWW: <<http://is.muni.cz/do/1499/el/estud/ff/js08/fonetika/ucebnice/ch08s01s01.html>>.
- [6] KRČMOVÁ, Marie . Fonetika [online]. 2007 [cit. 2011-04-16]. Foném a jeho realizace. Dostupné z WWW: <<http://is.muni.cz/elportal/estud/ff/js07/fonetika/materialy/ch08s02.html>>.
- [7] Tomcsányi, Peter. Using Speech Input and Output in Logo Programming [online]. 2001 [cit. 2011-04-16]. Proceedings Eurologo 2001. Liny, 2001, p. 219/227. Dostupné na WWW: <<http://www.ocg.at/activities/books/volumes/band%20156/P73PeterTom.doc>>
- [8] HUNT, Andrew. Comp.speech, frequently asked questions. [online]. poslední změna 05-Sep-1997 [cit. 2011-04-15]. Speech Applications Group, Sun Microsystems Laboratories, USA. Dostupné na WWW: <<http://mi.eng.cam.ac.uk/comp.speech/>>

- [9] Speechlab [online]. [cit. 2011-04-12]. stránky Laboratoře počítačového zpracování řeči. Liberec: TUL. Dostupné na WWW: <<http://www.ite.tul.cz/speechlab/>>
- [10]Hlasové technologie [online]. [cit. 2011-04-14]. stránky programu NEWTONDictate. Dostupné z WWW: <<http://www.diktovani.cz/>>.
- [11]CC Novasoft [online]. c2006 [cit. 2011-04-14]. NovaVoice - řešení pro převod mluvené řeči na text. Dostupné z WWW: <<http://www.ccnovasoft.cz/cz/novavoice-prevod-mluvene-rci/>>.
- [12]Speech SDK 5.1 [online]. c2011 [cit. 2011-04-25]. Microsoft Download Center. Dostupné z WWW: <<http://www.microsoft.com/downloads/en/details.aspx?FamilyID=5e86ec97-40a7-453f-b0ee-6583171b4530&displaylang=en>>.
- [13]NEKOLNÁ, Magdalena . Automatický přepis mluvené řeči na mobilní telefon [online]. [cit. 2011-04-15]. Dostupné z WWW: <<http://www.generation5.org/content/2002/howsrworks.asp>>.

Seznam ilustrací

Ilustrace 1: Modul rozpoznání řeči ve Windows 7.....	17
Ilustrace 2: Základní prvky modulu rozpoznání řeči u Windows 7.....	18
Ilustrace 3: Nabídka hlasových povelů v nastavení stránky.....	20
Ilustrace 4: První stránka projektu s želvou ve tvaru žáby a hlasovým povellem frog.....	26
Ilustrace 5: Editační stránka projektu se zvoleným obrázkem motýla, jehož lze smazat, nebo editovat jeho slovíčka.....	28
Ilustrace 6: Stránka projektu pro přidávání vybraného obrázku a napsané slovní dvojice.....	32
Ilustrace 7: Ukázka stránky prezentace s želvami, které jsou vytvořené podle šablony pro nadpisy a šablony text.....	37
Ilustrace 8: Tvar želvy v podobě textu vykresleného při směru 180, kde červená tečka znázorňuje kreslicí bod.....	42
Ilustrace 9: Rozdíly mezi identickými tvary želv, které se liší pouze v příkazu puntik 0.....	43
Ilustrace 10: Obrázky použitelné pro pozadí.....	49
Ilustrace 11: Ukázka hry s překážkami ve tvaru ovcí.....	57
Ilustrace 12: Ukázka přepisu českých slov do angličtiny.....	62