Knihovna zvuků

Bc. Michal Jirouš
V Praze dne 12.12.2008

Obsah

1. Úvod

V multimediálních aplikacích se na celkovém dojmu značně podílí zvuk. Zvláště ve hrách pak přidává na realističnosti prostředí a hráč si tak dokáže lépe představit, jaké by to bylo, stát v prostředí hry. Pro využití ozvučení v těchto projektech je vhodné vlastnit nástroj, který udělá za nás co nejvíce práce sám, aby na samotného programátora zbylo pouze používání několika jednoduchých funkcí, které všechny starosti vyřeší za něj. Pro tyto účely vznikla tato knihovna zvuků, která je postavena na zvukovém aplikačním rozhraní OpenAL. Je multiplatformní, protože je napsaný v c++ bez závislostí na operačním systému a používá pouze knihovnu OpenAL a Vorbis file, které jsou také multiplatformní.

Tento modul je primárně určen pro hry nebo pro aplikace, kde lze využít prostorový zvuk. Umožňuje vytvářet zdroje zvuku, nastavovat jim pozici a směr pohybu v prostoru, přičemž o všechny výpočty výsledné hlasitosti zvuků se stará OpenAL, které je svým stylem funkcí podobný grafické knihovně OpenGL. Podporovanými formáty zvukových souborů je WAV a OGG a jejich data se načtou do paměti vcelku, takže streamování není podporováno. Zvuková data lze sdílet mezi jednotlivými zdroji zvuku ve scéně a tím se sníží paměťová náročnost a redundance, protože každý zvukový záznam v databázi bude unikátní.

2. Proč OpenAL

Samozřejmě OpenAL není jediným prostředkem pro přehrávání zvuku. Existují i další projekty, kde jedním z nejznámějších je asi FMOD, který je stejně jako OpenAL multiplatformní, a který se používá v mnoha herních titulech a to i v těch neznámějších jako Crysis, Bioshock i World of Warcraft. FMOD je pro nekomerční účely zdarma, ale pro komerční účely už je třeba zaplatit nemalou částku, naproti tomu OpenAL je zcela zdarma. Abych upřesnil, proč jsem nepoužil k tvorbě tohoto modulu FMOD, tak proto, že celý tento modul je jakýsi můj FMOD, který sice není tak dokonalý, ale zato je to můj vlastní produkt a na něm se toho člověk naučí nejvíce. OpenAL je totiž nízkoúrovňové rozhraní podobně jako je OpenGL, takže mnoho věcí si programátor musí vytvořit sám, ale na druhou stranu dostane více moci nad svým programem. Tím se dostávám k tomu, že naučit se používat OpenAL není tak jednoduché jako FMOD, který je „high level“ a ovládání je primitivní. Z těchto vlastností by se mohlo zdát, že bych si měl naopak vybrat FMOD i přesto, že není tak úplně zdarma, ale mě přijde OpenAL sympatičtější, protože díky němu dokáži plně ovládat zvukové prostředky a hlavně se setkám s problémy, kterou tahle oblast musí řešit, a se kterými jsem se dosud nesetkal.

3. Vlastnosti OpenAL

OpenAL je multiplatformní 3D audio API, které je vhodné pro použití ve hrách a jiných audio aplikacích. Svým stylem a konvencemi se velice podobá OpenGL, a proto je pro programátory OpenGL jeho používání jednoduché. Projekt vznikl ve firmě Loki Software ve snaze o snadnější portování her na linux. Po zániku firmy se projektu ujala komunita open source vývojářů, ale nyní se projektu ujala firma Creative Technology s podporou firmy Apple.

Zvuková data jsou v paměti uložena ve formátu PCM v 8 nebo 16-ti bitech buď jako mono nebo stereo. Stará se o veškeré výpočty prostorového zvuku jako například o takzvaný Doppler effect, což je jev, který popisuje změnu frekvence a vlnové délky přijímaného oproti vysílanému signálu, způsobenou nenulovou vzájemnou rychlostí vysílače a přijímače. Dále pak vypočítává oslabení hlasitosti zvuku dle vzdálenosti zdroje od posluchače. Pro tento jev poskytuje tři možnosti řešení: Inverzní, linerární a exponenciální (Pro bližší informace přejděte na stránky OpenAL).

OpenAL lze používat na většině platforem jako například MS Windows, GNU/Linux, MAC OS X,

Solaris, BSD, Xbox, iPhone a další. A stejně jako FMOD, tak i OpenAL je použit v některých známých hrách jako například: Doom 3, Quake 4, Prey, Battlefield 2 či Unreal engine 2 a 3.

V OpenAL se vyskytují primárně tři typy objektů, se kterými pak pracuje tento modul a jsou to: Source (zdroj zvuku), Buffer (zvuková data) a Listener (posluchač).

4. Vlastnosti modulu

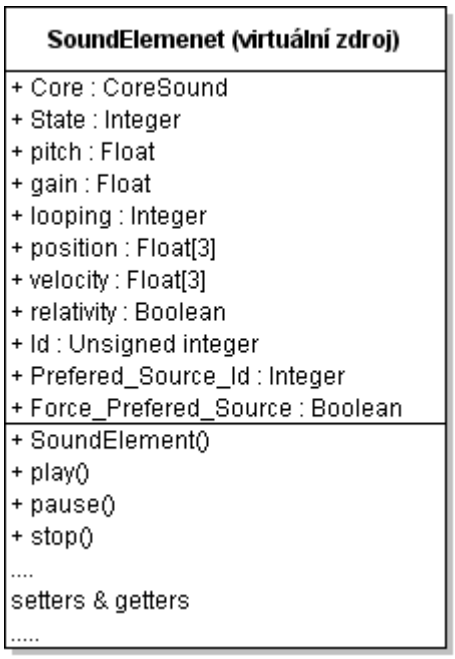
- Umožňuje načítat zvukové záznamy ve formátu WAV a OGG (algoritmus pro načítání Ogg byl přejat z ukázkového přikladu, který lze nalézt na [4]).
- Dokáže vytvořit až 32 hardwarových zdrojů zvuku a neomezeně virtuálních zdrojů (počet je omezen paměťovými možnostmi počítače). Počet najednou přehrávaných zvuků je závislý na počtu hardwarových zdrojů.
- Sdílí zvuková data mezi virtuálními zdroji zvuku.
- Poskytuje sadu jednoduchých funkcí, kterými se modul ovládá.
- Podporuje 3D zvuk díky OpenAL.

5. Přidělování hardwarových zdrojů

V minulé kapitole jsem se zmínil o hardwarových a virtuálních zdrojích zvuku. Počet hardwarových je totiž omezen počtem kanálů zvukové karty, a proto bylo třeba navrhnout řešení, které by dokázalo v případě požadavku virtuálního zdroje o přehrávání, nalézt vhodného kandidáta z dostupných hardwarových zdrojů. Při použití této knihovny zvuků je ale také možné nechat si nějaký zdroj rezervovat, a pak se dárce nevyhledává. Naopak když nemám zarezervovaný zdroj, tak se musí projít postupně všechny nerezervované zdroje a najít ten nejvhodnější. Pro tento výběr jsem použil algoritmus, který se používá v paměti cache při výběru vhodného dárce k uložení dat: LRU (least recently used). U každého zdroje mám počítadlo přístupů a ten, který se využívá nejméně a zrovna nepřehrává žádný zvuk, se vybere jako vhodný dárce. Jelikož procházím maximálně 32 zdrojů a nastavení zdroje dle virtuálního je záležitostí nastavení pár atributů, je operace časově nenáročná, čili asymptotická složitost je konstantní $\Theta(1)$. Samozřejmostí systému je, že při výběru dárce nejprve zkontroluje zdroj, který virtuální zdroj naposledy použil, a pokud se od minulého použití nezměnil, tak nemusím hledat nového dárce, ale využiji tento předchozí, jelikož ho nemusím nijak nastavovat a můžu rovnou přehrávat.

6. Implementace

Pro správnou funkčnost celého systému jsem navthnul několik objektů, které dohromady utvářejí výsledný celek. Z hlediska uživatele knihovny zvuků je nejdůležitější objekt SoundElement, což je virtuální zdroj. K hardwarovům zdrojům není umožněn přístup a může s nimi pracovat pouze objekt knihovny zvuků. Tento objekt CSoundLibrary je objektem, který se stará o vytváření nových virtuálních zdrojů, o výběr vhodného hardwarového zdroje při požadavku na přehrávání a o správu databáze zvukových dat, kde je implementována i funkce sdílení.



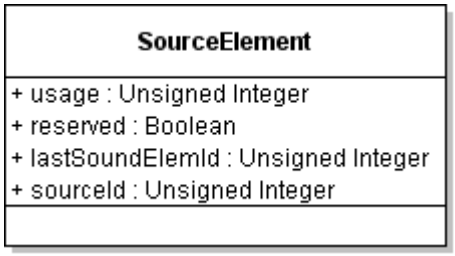
Virtuální zdroj

Virtuální zdroj udržuje všechny informace, které hardwarový zdroj potřebuje ke korektnímu přehrávání zvuku. Pokud dojde k požadavku na přehrávání, tak se všechny tyto atributy musí předat hardwarovému zdroji a teprve potom může dojít ke správnému přehrávání. Většina atributů, které jsou v objektu implementovány vyplývají z vlastností objektu Source v OpenAL, což jsou pozice, směr pohybu, výška tónu, hlasitost, opakování přehrávání či relativní pozice vzhledem k posluchači a stav přehrávání. Kromě těchto vlastností potřebujeme pro jednoznačné rozpoznání virtuálního zdroje identifikační číslo. Pro hledání volného hardwarového zdroje pro přehrávání je důležité znát číslo zdroje, který byl při předchozím přehrávání vybrán. Pak lze nastavit, aby se použil ten stávající hardwarový zdroj, ale při špatném použití pak může dojít k nežádaným efektům, jako například opakované přepisování jediného hardwarového zdroje, což bude mít za následek, že se žádný zvuk nebude moci přehrát do konce.

Objekt SoundElement si uživatel vytváří sám a data se do něj nahrají až při požadavku o načtení zvukového souboru knihovnou zvuků.

Hardwarový zdroj

Při inicializaci knihovny zvuků se vytvoří maximální počet hardwarových zdrojů a vznikne tak množina, ze které je možné vybírat vhodné zdroje pro přehrávání. Pro každý takový objekt si OpenAL generuje jednoznačné identifikační číslo hardwarového zdroje, které se uloží do objektu SourceElement, což v tomto modulu představuje hardwarový zdroj.

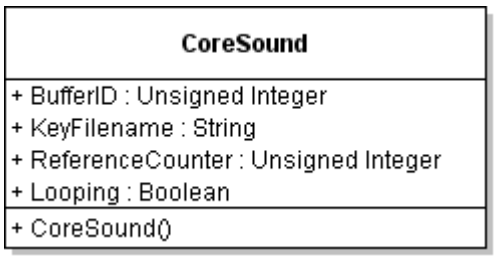


Kromě tohoto atributu potřebujeme znát počet použití tohoto zdroje, což přispěje ke správnému přidělování. Když je hodnota atributu reserved rovna True, tak to znamená, že tento zdroj je rezervovaný a při výběru volného zdroje se bude ignorovat. Poslední atribut, který udává

identifikační číslo naposledy přehrávaného virtuálního zdroje (lastSoundElemId), slouží k určení, zda má provést kompletní nastavení hardwarového zdroje dle virtuálního. Pokud je totiž hodnota lastSoundElemId stejná jako identifikační číslo virtálního zdroje, který chce pomocí tohoto hardwarového provést přehrávní, tak není potřeba nic nastavovat, protože nastavení setrvalo nezměněno od posledního přehrávání.

Zvuková data

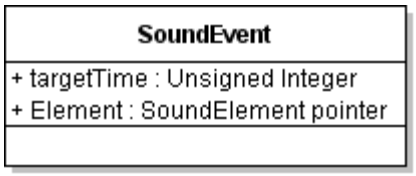
Tato data jsou uložena v databázi, kterou spravuje objekt knihovny zvuků. Pro každý načtený zvukový soubor existuje v databázi jeden objekt CoreSound, který je identifikovaný klíčem, čímž je cesta k datovému souboru, a proto jsou zvuková data vždy unikátní, jelikož v databázi nemohou existovat záznamy se stejným klíčem.



Když podá uživatel požadavek na otevření a načtení souboru se svukovými daty, tak pokud už v databázi takový soubor neexistuje, tak se musí vytvořit nový objekt CoreSound a nastaví se mu unikátní klíč. Teprve poté dojde k vlastnímu načtení zvukových dat ze souboru a vytvoření nového bufferu, kde budou výsledná data uložena. V objektu pak potřebujeme znát pouze vygenerované identifikační číslo tohoto bufferu. Důležitou vlastností je ovšem počítadlo referencí. Na počatku je jeho hodnota rovna nule, ale při každém přiřazení zvukových dat virtuálnímu zdroji se jeho hodnota zvětší o jedničku. Naopak pokud už některý zdroj data nepotřebuje, tak se hodnota o jedničku sníží. Když se dostaneaž zpátky na nulu, tak můžeme zvuková data odstranit z paměti, protože už je nikdo nepoužívá.

Knihovna zvuků

Knihovnu představuje objekt CSoundLibrary. Právě zde je implementován algoritmus přidělování zdrojů. Kromě toho se stará o načítání zvukových souborů a ukládání dat do databáze v podobě objektů CoreSound, přehrávání, pozastavování a resetování přehrávání virtuálních zdrojů a důležitou funkcí je i opožděné spuštění přehrávání zvuků. K tomu slouží jednoduchý objekt SoundEvent, který má pouze dva atrbuty: Odkaz na virtuální zdroj a cílový čas spuštění přehrávání.



Knihovna poskytuje také funkce pro rezervaci hardwarových zdrojů a kontroluje počet referencí jednotlivých objektů CoreSound v databázi a těm, jejichž hodnota je nula, následně smaže zvuková data, protože už nejsou potřeba.

CSoundLibrary funguje částečně jako stavový automat, protože dovoluje nastavovat několik standardních parametrů, které budou automaticky nastaveny nově vytvořeným virtuálním zdrojům. Těmi atributy jsou:

- Zda se má použít relativní pozice zdroje vůči posluchači.
- Preferování přednastaveného hardwarového zdroje při přehrávání

- Výchozí identifikační číslo hardwarového zdroje (vhodné při preferování nastaveného hardwarového zdroje).

Poslední funkcí knihovny je jednoduché nastavování parametrů posluchače. Umožňuje nastavovat pozici, orientaci a pohyb.

7. Závěr

Díky tomuto modulu je práce se zvukem velice jednoduchá. Uživateli této knihovny stačí vytvořit si nový virtuální zvuk, pomocí jedné funkce načíst data a pak už jen zavolat funkci play(). Pro správnou funkčnost bylo důležité překonat problém s omezeným počtem hardwarových zdrojů, což se podařilo díky jednoduchému algoritmu pro přidělování volných zdrojů. Knihovna poskytuje některé funkce, které zjednodušují práci se samotným OpenAL a umožňuje načítat soubory formátu Ogg, které zabírají díky efektivní kompresy mnohokrát méně místa, než standardní Wav.

8. Literatura

[1] Domovská stránka OpenAL (popis, dokumentace, referenční manuál)

http://connect.creativelabs.com/openal/

[2] Wikipedia o OpenAL

http://en.wikipedia.org/wiki/OpenAL

[3] Tutoriál o OpenAL

http://www.devmaster.net/articles/openal-tutorials/lesson1.php

[4] Tutoriál o OpenAL a Ogg vorbis

http://www.gamedev.net/reference/articles/article2031.asp

[5] Stránky Fmod

http://www.fmod.org/