

Licencjacka Pracownia Oprogramowania, grupa 3
Instytut Informatyki Uniwersytetu Wrocławskiego

Marcin Pawłowski

TSAL - Lekka warstwa abstrakcji dźwięku

Dokumentacja biblioteki

18 maja 2012

Spis treści

1	Wprowadzenie	4
1.1	Cel dokumentu	4
1.2	O bibliotece TSAL	4
1.3	Uzasadnienie zadania	4
2	Opis biblioteki	5
2.1	Struktura biblioteki	5
2.2	Możliwości biblioteki	5
3	Szczegóły implementacji	6
3.1	Głośność dźwięku	6
4	Przypadki użycia	6
4.1	Czytanie plików i dźwięki globalne	6
4.2	Wirtualny słuchacz i źródło dźwięku	7
5	Metody klasy TSAL_Manager	9
5.1	TSAL_Manager::TSAL_Manager	9
5.2	TSAL_Manager::~~TSAL_Manager	9
5.3	TSAL_Manager::load_sound	9
5.4	TSAL_Manager::play_global	9
5.5	TSAL_Manager::play_sound	10
5.6	TSAL_Manager::listener_pos	10
5.7	TSAL_Manager::listener_vel	10
5.8	TSAL_Manager::listener_facing	10
5.9	TSAL_Manager::set_volume	11
5.10	TSAL_Manager::set_falloff	11
5.11	TSAL_Manager::set_ref_dist	11
5.12	TSAL_Manager::manage_all_sources	12
5.13	TSAL_Manager::create_source	12
6	Metody klasy TSAL_Source	13
6.1	TSAL_Source::TSAL_Source	13
6.2	TSAL_Source::~~TSAL_Source	13
6.3	TSAL_Source::start	13
6.4	TSAL_Source::stop	13
6.5	TSAL_Source::pause	14

6.6	TSAL_Source::set_pitch	14
6.7	TSAL_Source::set_loudness	14
6.8	TSAL_Source::set_loop	14
6.9	TSAL_Source::set_pos	15
6.10	TSAL_Source::set_vel	15
6.11	TSAL_Source::set_offset	15
6.12	TSAL_Source::set_falloff	15
6.13	TSAL_Source::set_sample	15
6.14	TSAL_Source::is_playing	16
6.15	TSAL_Source::sample	16
7	Słownik	17

1. Wprowadzenie

1.1. Cel dokumentu

Niniejszy dokument opisuje podstawowe założenia biblioteki *TSAL* i jej architekturę, a także dokumentuje jej interfejs programistyczny i do pewnego stopnia wewnętrzną implementację, stanowiąc punkt odniesienia dla każdego, kto chciałby jej użyć w swojej aplikacji lub dalej ją rozwijać. Projekt jest tworzony przez Marcina Pawłowskiego w ramach licencyjnego projektu programistycznego.

1.2. O bibliotece *TSAL*

Nazwa biblioteki to skrót od słów *Thin Sound Abstraction Layer*, co można przetłumaczyć jako *Lekka warstwa abstrakcji dźwięku*. Nazwa ta odnosi się do struktury biblioteki, która warstwą abstrakcji pośredniczy między programistą a znajdującymi się głębiej bibliotekami, na których się ona opiera.

Biblioteka *TSAL* ma ułatwić pisanie przenośnych aplikacji dźwiękowych w języku *C++* dla systemów Windows i Linux. Obsługiwane architektury procesora to x86, 32- i 64-bitowe. Programista korzystający z biblioteki ma do swojej dyspozycji zestaw klas i metod ukrywających bardziej skomplikowane wywołania i mechanizmy, które obsługiwane są automatycznie przez bibliotekę w obu systemach operacyjnych.

Biblioteka *TSAL* jest zbudowana na podstawie otwartych bibliotek *OpenAL* (odtworzenie dźwięku) i *OGG Vorbis* (dekodowanie plików w formacie *ogg*).

1.3. Uzasadnienie zadania

Biblioteka odtwarzania dźwięku *OpenAL* daje programiście ogromne możliwości, w szczególności możliwość symulacji trójwymiarowego dźwięku przestrzennego, efektu Dopplera czy operowania wysokością odtwarzanych dźwięków. Jednak jej interfejs programistyczny napisany w języku *C* jest nieobiektywny i niewygodny w użyciu. Biblioteka zawiera też błędy, a efekty wywołań nie są jednakowe w różnych systemach operacyjnych.

Biblioteka *TSAL* ma na celu utworzenie dodatkowej warstwy pośredniczącej między programistą a *OpenAL*, a także dodanie brakujących funkcji, które okazują się bardzo potrzebne przy tworzeniu aplikacji interaktywnych. Dzięki temu pisanie aplikacji staje się łatwiejsze, a wynikowy kod – bardziej przenośny.

2. Opis biblioteki

2.1. Struktura biblioteki

Biblioteka składa się z czterech głównych elementów:

zestawu funkcji opakowujących wywołania funkcji biblioteki *OpenAL* w kod sprawdzający poprawność ich wykonania;

klasy *TSAL_Source*, której obiekty reprezentują wirtualne źródła dźwięku;

klasy *TSAL_Priv_Source*, która stanowi wewnętrzną, ukrytą implementację źródeł dźwięku;

klasy *TSAL_Manager* zapewniającej centralne sterowanie i zarządzanie dźwiękiem i obiektami klasy *TSAL_Priv_Source*.

Pierwszy i trzeci z nich są używane wewnętrznie przez bibliotekę, podczas gdy pozostałe dwa stanowią interfejs programowania.

2.2. Możliwości biblioteki

Biblioteka *TSAL* posiada następujące funkcje:

ładowanie plików dźwiękowych w formacie *ogg* z dysku,

tworzenie wirtualnych źródeł dźwięku w liczbie nieograniczonej przez liczbę źródeł sprzętowych,

nadawanie wirtualnym źródłom dźwięku pozycji i prędkości,

nadawanie wirtualnemu słuchaczowi pozycji i prędkości,

trójwymiarowy dźwięk przestrzenny, rozkład na kanały zależy od względnej pozycji słuchacza i źródła dźwięku,

kontrola wysokości i głośności odtwarzanych dźwięków,

odtwarzanie dźwięków globalnych, niezależnych od pozycji źródła i słuchacza,

zarządzanie źródłami dźwięku przez zmianę odtwarzanego pliku, zatrzymywanie, wstrzymywanie i wznowianie odtwarzania.

3. Szczegóły implementacji

3.1. Głośność dźwięku

Głośność dźwięku określana jest przez następujący wzór:

$$dist = \max(ref, \sqrt{(l_x - s_x)^2 + (l_y - s_y)^2 + (l_z - s_z)^2})$$
$$vol = \frac{ref \times vol_g \times vol_s}{ref + (rolloff_s \times rolloff_g) \times (dist - ref)}$$

gdzie l i s to punkty określające położenie odpowiednio wirtualnego słuchacza i wirtualnego źródła dźwięku, ref to dystans odniesienia a vol_g i vol_s to mnożniki głośności odpowiednio globalny i źródła. Analogicznie $rolloff_g$ i $rolloff_s$ to mnożniki słabnięcia dźwięku.

4. Przypadki użycia

4.1. Czytanie plików i dźwięki globalne

Klasa *TSAL_Manager* służy do zarządzania biblioteką *TSAL*. Każdy obiekt tej klasy ma związane ze sobą: wirtualnego słuchacza, zbiór przeczytanych plików dźwiękowych i zbiór wirtualnych źródeł dźwięku.

Metoda `TSAL_Manager::load_sound` służy do czytania z dysku plików dźwiękowych w formacie `ogg`. Przyjmuje ona dwa argumenty typu `string`, pierwszy to ścieżka do pliku (podana względem położenia programu), drugi to nazwa, jaką chcemy nadać dźwiękowi. Nazwa będzie służyła do późniejszego odwoływania się do pobranych danych.

Metoda `TSAL_Manager::play_global` służy do odtwarzania globalnych dźwięków, to znaczy takich, które nie są zależne od położenie wirtualnego słuchacza. Pierwszy argument tej metody to nazwa pliku, który chcemy odtworzyć. Kolejne dwa są opcjonalne i określają głośność i wysokość, jakie chcemy nadać dźwiękowi (domyślne wartości obu z nich to 1).

Zamieszczony poniżej kod ilustruje przeczytanie pliku dźwiękowego i odtworzenie jego zawartości jako dźwięk globalny.

```
#include "tsal.hpp"  
#include <cstdio>  
#include <unistd.h>
```

```

int main()
{
    TSAL_Manager manager;

    // Przeczytaj plik "sound.ogg" i nazwij pobrany dźwięk "sound"
    manager.load_sound("sound/sound.ogg", "sound");

    // Odtwórz globalnie dźwięk "sound" w oryginalnej głośności, ale
    // w częstotliwości dwukrotnie niższej niż oryginalna
    manager.play_global("sound", 1.0f, 0.5f);

    // Poczekaj sekundę na zakończenie odtwarzania
    sleep(1);

    return 0;
}

```

4.2. Wirtualny słuchacz i źródło dźwięku

Do tworzenia wirtualnych źródeł dźwięku służy metoda `TSAL_Manager::create_source`, która zwraca nowoutworzony, zarejestrowany w wewnętrznej strukturze danych klasy *TSAL_Manager*, obiekt klasy *TSAL_Source*. Metoda ta przyjmuje cztery argumenty. Pierwszy z nich jest wartością typu `string` i określa nazwę dźwięku, który źródło będzie odtwarzać. Kolejne trzy są typu `float` i określają pozycję początkową źródła w trójwymiarowej przestrzeni.

Metoda `TSAL_Source::set_loop` określa, czy przypisany sobie dźwięk źródło ma odtwarzać w pętli, czy tylko jednokrotnie.

Metoda `TSAL_Source::set_pos` przyjmująca jako argumenty trzy wartości typu `float` pozwala zmienić pozycję wirtualnego źródła.

Metody `TSAL_Manager::listener_pos` i `TSAL_Manager::listener_facing` pozwalają ustalić pozycję i zwrot wirtualnego słuchacza.

Metoda `TSAL_Manager::manage_all_sources` pozwala przenieść właściwości wirtualnego źródła dźwięku na faktyczne źródło i powinna być wykonywana z częstotliwością odpowiadającą zamierzonej częstotliwości odczuwalnych dla słuchacza zmian we właściwościach wirtualnych źródeł dźwięku.

Poniższy kod ilustruje utworzenie wirtualnego źródła dźwięku i jednoczesne odtwarzanie przypisanego mu dźwięku w pętli i przemieszczanie źródła względem wirtualnego słuchacza.

```

TSAL_Manager manager;
manager.load_sound("sound/sound.ogg", "sound");

// Utworz wirtualne źródło dźwięku i nadaj mu początkową pozycję
TSAL_Source source = manager.create_source("sound", 100.0f, 0.0f, 0.0f);

// Niech dźwięk źródła będzie odtwarzany w pętli
source.set_loop(true);

// Nadaj pozycję i kierunek wirtualnemu słuchaczowi
manager.listener_pos(0.0f, 0.0f, 0.0f);
manager.listener_facing(0.0f, 1.0f, 0.0f, 0.0f, 0.0f, 1.0f);

// Rozpocznij odtwarzanie dźwięku ze źródła
source.start();

for (int i = 0; i <= 10; i++)
{
    // Funkcja zarządzająca źródłami, powinna być wykonywana regularnie
    manager.manage_all_sources();
    // Przesuń wirtualne źródło wzdłuż osi OX i poczekaj pół sekundy
    source.set_pos(100.0f - i*20.0f, 0.0f, 0.0f);
    usleep(500000);
}

```


5. Metody klasy `TSAL_Manager`

5.1. `TSAL_Manager::TSAL_Manager`

Deklaracja

```
TSAL_Manager::TSAL_Manager();
```

Opis

Domyślny, bezargumentowy konstruktor klasy.

5.2. `TSAL_Manager::~~TSAL_Manager`

Deklaracja

```
TSAL_Manager::~~TSAL_Manager();
```

Opis

Domyślny, bezargumentowy destruktor klasy.

5.3. `TSAL_Manager::load_sound`

Deklaracja

```
void TSAL_Manager::load_sound(std::string file, std::string name);
```

Opis

Czyta plik dźwiękowy w formacie ogg ze ścieżki wskazywanej przez argument `file` i zapamiętuje go pod nazwą `name`.

5.4. `TSAL_Manager::play_global`

Deklaracja

```
void TSAL_Manager::play_global(std::string name, float loudness = 1,  
float pitch = 1);
```

Opis

Odtwarza globalnie dźwięk o nazwie `name`. Głośność dźwięku określana jest przez argument `loudness` a jego wysokość przez argument `pitch`.

5.5. TSAL_Manager::play_sound

Deklaracja

```
void TSAL_Manager::play_sound(std::string name, float x, float y,  
float z = 0, float loudness = 1, float pitch = 1, float falloff = 1);
```

Opis

Odtwarza lokalnie dźwięk o nazwie `name`. Argumenty `x`, `y` i `z` określają położenie źródła dźwięku w przestrzeni trójwymiarowej. Głośność dźwięku określana jest przez argument `loudness` a jego wysokość przez argument `pitch`. Argument `falloff` to mnożnik słabnięcia dźwięku wraz z odległością od słuchacza.

5.6. TSAL_Manager::listener_pos

Deklaracja

```
void TSAL_Manager::listener_pos(float pos_x, float pos_y, float pos_z = 0);
```

Opis

Ustala pozycję wirtualnego słuchacza w przestrzeni trójwymiarowej. Argumenty `pos_x`, `pos_y` i `pos_z` to współrzędne punktu, w którym ma znaleźć się słuchacz.

5.7. TSAL_Manager::listener_vel

Deklaracja

```
void TSAL_Manager::listener_vel(float vel_x, float vel_y, float vel_z = 0);
```

Opis

Ustala prędkość wirtualnego słuchacza w przestrzeni trójwymiarowej. Argumenty `vel_x`, `vel_y` i `vel_z` to składowe wektora ruchu słuchacza.

5.8. TSAL_Manager::listener_facing

Deklaracja

```
void TSAL_Manager::listener_facing(float front_x, float front_y,  
float front_z, float up_x, float up_y, float up_z);
```

Opis

Ustala kierunek wirtualnego słuchacza. Argumenty `front_x`, `front_y` i `front_z` to składowe trójwymiarowego wektora określającego zwrot słuchacza. Argumenty `up_x`, `up_y` i `up_z` to składowe trójwymiarowego wektora określającego pion słuchacza.

5.9. TSAL_Manager::set_volume

Deklaracja

```
void TSAL_Manager::set_volume(float v);
```

Opis

Określa globalną głośność dla wszystkich dźwięków odtwarzanych przez bibliotekę. Argument `v` to liczba, przez którą mnożone są indywidualne wartości głośności dźwięków i źródeł.

5.10. TSAL_Manager::set_falloff

Deklaracja

```
void TSAL_Manager::set_falloff(float f);
```

Opis

Określa globalną wartość słabnięcia dźwięku. Argument `f` to liczba, przez którą mnożone są indywidualne wartości słabnięcia dźwięków i źródeł.

5.11. TSAL_Manager::set_ref_dist

Deklaracja

```
void TSAL_Manager::set_ref_dist(float d);
```

Opis

Określa dystans odniesienia dla odtwarzanych dźwięków, dla dźwięków o źródłach oddległych od słuchacza o nie więcej niż dystans odniesienia dźwięk osiąga maksymalną możliwą głośność.

5.12. TSAL_Manager::manage_all_sources

Deklaracja

```
void TSAL_Manager::manage_all_sources();
```

Opis

Funkcja zmusza zarządcę dźwięków do wprowadzenia wszystkich dokonanych przez programistę zmian w życie. Powinna być wywoływana z częstością odpowiadającą częstości dokonywania zmian w konfiguracji wirtualnych źródeł. Częstość około stu wywołań na sekundę jest wystarczająca by dźwięk brzmiał naturalnie niezależnie od częstości zmian.

5.13. TSAL_Manager::create_source

Deklaracja

```
TSAL_Source TSAL_Manager::create_source(std::string sample = "",  
float x = 0, float y = 0, float z = 0);
```

Opis

Funkcja tworzy nowe wirtualne źródło dźwięku i zwraca obiekt klasy `TSAL_Source` stanowiący jego interfejs programistyczny. Argument `sample` określa nazwę dźwięku, który początkowo zostanie przyporządkowany źródłu. Argumenty `x`, `y` i `z` określają początkowe położenie źródła w przestrzeni trójwymiarowej.

6. Metody klasy TSAL_Source

6.1. TSAL_Source::TSAL_Source

Deklaracja

```
TSAL_Source::TSAL_Source();  
TSAL_Source::TSAL_Source(const TSAL_Source& source);
```

Opis

Konstruktory klasy. Bezargumentowy konstruktor domyślny i konstruktor przez kopiowanie.

6.2. TSAL_Source::~~TSAL_Source

Deklaracja

```
TSAL_Source::~~TSAL_Source();
```

Opis

Domyślny, bezargumentowy destruktor klasy.

6.3. TSAL_Source::start

Deklaracja

```
void TSAL_Source::start();
```

Opis

Rozpoczyna odtwarzanie dźwięku przypisanego źródłu.

6.4. TSAL_Source::stop

Deklaracja

```
void TSAL_Source::stop();
```

Opis

Zatrzymuje odtwarzanie dźwięku przypisanego źródłu.

6.5. TSAL_Source::pause

Deklaracja

```
TSAL_Source::pause();
```

Opis

Wstrzymuje odtwarzanie dźwięku przypisanego źródłu.

6.6. TSAL_Source::set_pitch

Deklaracja

```
void TSAL_Source::set_pitch(float p);
```

Opis

Ustala wysokość odtwarzanego przez źródło dźwięku. Argument `p` to liczba, przez którą zostanie przemnożona wysokość.

6.7. TSAL_Source::set_loudness

Deklaracja

```
void TSAL_Source::set_loudness(float loud);
```

Opis

Ustala głośność odtwarzanego przez źródło dźwięku. Argument `loud` to liczba, przez którą zostanie przemnożona głośność.

6.8. TSAL_Source::set_loop

Deklaracja

```
void TSAL_Source::set_loop(bool l);
```

Opis

Ustala czy dźwięk przypisany źródłu powinien być odtwarzany w pętli. Argument `l` o wartości `true` wymusza zapętlenie odtwarzania, wartość `false` wymusza jednokrotne odtworzenie dźwięku dla każdego wywołania metody `start`.

6.9. TSAL_Source::set_pos

Deklaracja

```
void TSAL_Source::set_pos(float pos_x, float pos_y, float pos_z = 0);
```

Opis

Ustala pozycję źródła w trójwymiarowej przestrzeni. Argumenty `pos_x`, `pos_y` i `pos_z` to nadawane źródłu współrzędne.

6.10. TSAL_Source::set_vel

Deklaracja

```
void TSAL_Source::set_vel(float vel_x, float vel_y, float vel_z = 0);
```

Opis

Ustala prędkość źródła w trójwymiarowej przestrzeni. Argumenty `vel_x`, `vel_y` i `vel_z` to składowe wektora ruchu nadawanego źródłu.

6.11. TSAL_Source::set_offset

Deklaracja

```
void TSAL_Source::set_offset(float off);
```

Opis

Przewija odtwarzany dźwięk do zadanego położenia. Argument `off` powinien być liczbą z przedziału 0-1, gdzie 0 oznacza początek zapisanego dźwięku a 1 oznacza jego koniec.

6.12. TSAL_Source::set_falloff

Deklaracja

```
void TSAL_Source::set_falloff(float f);
```

Opis

Zmienia słabnięcie dźwięku odtwarzanego przez źródło wraz z odległością od słuchacza. Argument `f` to liczba, przez którą mnożona jest siła słabnięcia dźwięku.

6.13. TSAL_Source::set_sample

Deklaracja

```
void TSAL_Source::set_sample(std::string n);
```

Opis

Zmienia dźwięk przypisany źródłu. Argument `n` to nazwa przypisywanego dźwięku.

6.14. TSAL_Source::is_playing

Deklaracja

```
bool TSAL_Source::is_playing();
```

Opis

Metoda zwraca wartość `true` jeśli źródło obecnie odtwarza dźwięk i `false` jeśli nie odtwarza żadnego dźwięku.

6.15. TSAL_Source::sample

Deklaracja

```
std::string TSAL_Source::sample();
```

Opis

Metoda zwraca nazwę dźwięku aktualnie przypisanego do źródła.

7. Słownik

źródło dźwięku — sprzętowa implementacja odtwarzacza dźwięku, zwykle oznacza pojedynczy kanał na karcie dźwiękowej komputera. Niektóre systemy operacyjne mogą programowo symulować większą liczbę źródeł dźwięku niż pozwala na to karta dźwiękowa, jednak dla biblioteki są one nierozróżnialne od źródeł zaimplementowanych w sprzęcie.

wirtualne źródło dźwięku — obiekt wydający dźwięk w wirtualnym świecie przedstawionym w aplikacji interaktywnej. Jeśli aplikacja interaktywna na przykład jest symulatorem wyścigów samochodowych, to wirtualnym źródłem dźwięku mogą być silniki samochodów.

wirtualny słuchacz — zestaw parametrów określających słuchacza w wirtualnym świecie przedstawionym w aplikacji interaktywnej. Są to na przykład jego pozycja lub kierunek. Decydują one o tym, jak dźwięk zostanie rozłożony na kanały stereofoniczne.