

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

GENEROVÁNÍ HERNÍHO PROSTŘEDÍ NA ZÁKLADĚ  
HUDBY

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

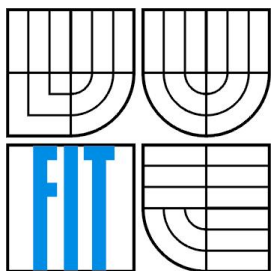
AUTOR PRÁCE  
AUTHOR

BC. JIŘÍ VANĚK

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ  
FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# GENEROVÁNÍ HERNÍHO PROSTŘEDÍ NA ZÁKLADĚ HUDBY

GAME ENVIRONMENT GENERATION FROM MUSIC

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

BC. JIŘÍ VANĚK

VEDOUCÍ PRÁCE

SUPERVISOR

ING. MICHAL HRADIŠ

BRNO 2009

## **Abstrakt**

Tématem této práce je generování herního prostředí na základě hudby. Zabývám se zde především analýzou hudby. Řeším otázku, jaké informace jsou pro generování hry užitečné a jak je z hudby můžeme získat. Návrh systému pro analýzu hudby, předložený v této práci, vychází z teorie zpracování signálů a statistické klasifikace. Navržená analýza hudby je zaměřená především na detekci rytmu, rozpoznávání hudebních žánrů a segmentaci písně. Druhá část práce se zabývá návrhem hry, která své prostředí generuje na základě dat, získaných při analýze hudby. Implementoval jsem kompletní systém pro analýzu hudby a prototyp hry, na kterém je možné zhodnotit výsledky vytvořené analýzy. Implementaci a dosažené výsledky popisuji v závěru práce.

## **Abstract**

The topic of this work is game environment generation from music. The main subject of this work is the music analysis. I am dealing with the problem of finding relevant information in music, which would be useful for game generating. The design of the system for music analysis, presented in this work, is based on the theory of signal processing and statistical classification. The proposed analysis of music is focused mainly on a beat detection, musical genre recognition and song segmentation. The second part of the work deals with the design of a game, which generates its environment from the data obtained in the music analysis. I have implemented the complete system for the music analysis and a prototype of the game, in which it is possible to evaluate the results from the analysis. The implementation and the achieved results are described in the conclusion of this work.

## **Klíčová slova**

Hudební hry, generování herního prostředí, digitální hudba, vnímání hudby, analýza hudby, zpracování signálů, klasifikace hudebních žánrů, detekce rytmu, segmentace písně

## **Keywords**

Music games, game environment generation, digital music, music perception, music analysis, signal processing, music genre classification, beat detection, song segmentation

## **Citace**

Vaněk Jiří: Generování herního prostředí na základě hudby, diplomová práce, Brno, FIT VUT v Brně, 2009.

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Michala Hradiše.

Další informace mi poskytli Doc. Dr. Ing. Jan Černocký a Bc. Lukáš Polok.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Jiří Vaněk  
1.6.2009

## Poděkování

Děkuji svému vedoucímu, Ing. Michalu Hradišovi, že se ujal vedení této práce a poskytnul mi mnoho užitečných informací a podnětných konzultací. Dále chci poděkovat Doc. Dr. Ing. Janu Černockému za poskytnutou konzultaci a za celý předmět SRE – Systémy zpracování řeči. Poslední významné poděkování patří Lukáši Polokovi, za poskytnutí grafického frameworku pro tvorbu hry a pomoc při jeho použití.

© Jiří Vaněk, 2009.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

1	Úvod.....	2
2	Digitální hudba a vnímání.....	5
2.1	Vnímání hudby člověkem.....	5
2.2	Audio signály a hudba.....	9
2.3	Klasifikace hudby.....	11
3	Současné hry založené na hudbě.....	14
3.1	Hry založené na paměti.....	14
3.2	Hry založené na rytmu.....	15
4	Návrh systému pro analýzu hudby.....	18
4.1	Detekce rytmu.....	18
4.2	Klasifikace hudebních žánrů.....	22
4.3	Segmentace písně.....	25
5	Návrh generátoru herního prostředí.....	26
5.1	Generování tvaru tratě.....	26
5.2	Objekty ve hře, styly zobrazení.....	30
6	Implementace.....	32
6.1	Analýza hudby.....	32
6.2	Generátor herního prostředí.....	36
7	Výsledky.....	39
7.1	Testy a výsledky klasifikace hudebních žánrů.....	39
7.2	Generování hry.....	40
8	Závěr.....	42
	Literatura.....	44
	Seznam příloh.....	46

# 1 Úvod

Hudba je významnou součástí našich životů. Pomáhá nám relaxovat, ale také představuje jednu z mála forem umění, kterou je dnešní člověk ochoten akceptovat. Díky prudkému rozvoji počítačů a digitální multimediální zábavy se hudba stala dostupnější a rozšířenější. Největší podíl na tom pravděpodobně má MPEG komprese (mp3), která redukuje paměť potřebnou pro digitální uložení hudby přibližně na desetinu a zachovává přijatelnou kvalitu. Lidé si díky tomu zvykli používat počítač mimo jiné jako přehrávač hudby. Hudbu ve formátu mp3 dnes umí přehrát kapesní mp3 přehrávače, mobilní telefony, audio systémy v autech a další zařízení. Lidé tak poslouchají hudbu na cestách, při sportu nebo při práci. Poslech hudby se stal jakousi doplňkovou záležitostí, zvukovou kulisou pro jiné činnosti.

Nezbývá než respektovat způsob, jakým se dnes hudba poslouchá. Jen těžko si lze představit člověka, který sedí u počítače, poslouchá hudbu a nepracuje zároveň s jiným programem. Při poslechu hudby je dnes běžné procházet internet, případně hrát jednoduché hry. Moje práce si klade za cíl nabídnout alternativní činnost, která by vrátila hudbu do středu zájmu. Mým záměrem je doplnit hudbu o vizuální vjem, a nabídnout posluchači interakci. Vizualizace hudby bývá přijímána pozitivně, důkazem je popularita hudebních videoklipů, která byla masivně zvýšena díky internetu. Samotné sledování videa však nezapojuje posluchače do dění. Interakci lze snadno zajistit, pokud bereme v úvahu, že posluchač sedí u počítače. Ideálním interaktivním spojením hudby s obrazem by byla hra, která by svým obsahem přímo souvisela s přehrávanou hudbou.

Existuje mnoho her založených na interakci s hudbou, většina z nich se dnes orientuje na rytmus přehrávané hudby. Typickým zástupcem je série Guitar Hero (hra popsána v kapitole 3.2). Zaměření hry na rytmus je pro běžného hráče vyhovující, většině lidí nedělá problém rytmus rozpoznat a dělat podle něj jednoduchou činnost. Vnímání rytmu je přirozené a hra, která svou dynamikou opisuje rytmus hudby, je díky tomu velmi dobře hratelná. Rytmus tedy musím zahrnout i do své práce, ale chci se věnovat i dalším charakteristickým vlastnostem hudby. Proto se v této práci zabývám otázkou, jaké další informace jsou v hudbě relevantní pro běžného posluchače a jak je vizualizovat.

Zásadní problém většiny dnešních hudebních her vidím právě v jejich hudebním obsahu. Ten totiž bývá omezen pouze na předem připravené písně, ke kterým je pevně přiřazeno herní prostředí. Například pokud hra spočívá v mačkání správných tlačítek ve správnou dobu (do rytmu hudby), vizuální prostředí pro danou píseň musí být předem vytvořené autorem hry, musí mít pevně definované časové úseky, kde se očekává reakce od hráče. Hráči jsou tak omezeni pouze na tu hudbu, kterou mu předloží autor hry. Do popředí se tím pádem dostává hra, předkládající hudbu, kterou vybral někdo jiný. Nejdůležitější by však měla být samotná hudba, kterou chce hráč opravdu poslouchat. Hra by měla být pouhým doplňkem libovolně zvolené hudby. Proto chci ve své práci

vytvořit systém, který umožní vygenerovat herní prostředí automaticky ze zvolené písně. Takové hry se sice již objevily, ale jednalo se o malé, nepříliš rozšířené tituly, typicky zaměřené pouze na rytmus hudby. Příkladem takové hry je Vib-Ribbon (popsaný v kapitole 3.2).

Aby uživatel mohl ve hře použít libovolnou píseň ze své digitální hudební sbírky, je potřeba jako vstup pro moji aplikaci uvažovat pouze audio signál. Generovat herní prostředí z notového zápisu skladby (například z MIDI souboru) by mohlo být snazší, ale v této formě nikdo běžně poslouchanou hudbu nemá, k poslechu ani není určena. Je tedy potřeba se na problém podívat z hlediska zpracování audio signálů, zjistit jaké možnosti v této oblasti máme, které vlastnosti hudby můžeme takto získat. Analýza hudby a extrakce informací, které by byly použitelné ve hrách, je hlavním předmětem této práce. Druhým úkolem této práce je navrhnout, jak získané informace využít a jak podle nich vygenerovat herní prostředí. Na základě tohoto návrhu by měl vzniknout prototyp hry, která bude generovat své prostředí ze zvolené písně.

## Členění práce

Ve druhé kapitole předkládám teorii související s touto prací. Popisuji zde lidské vnímání hudby. Dále se zabývám základní teorií zpracování signálů, související s hudbou. Významným tématem je psychoakustika, která řeší vztah audio signálů a lidského vnímání. Nakonec se zde zabývám statistickou klasifikací hudby.

Ve třetí kapitole jsou popsány dosavadní výsledky snahy o spojení hudby s interaktivní grafikou – existující hudební hry. Prezentuji zde významné koncepty v hudebních hrách, jejich zástupce a principy.

Čtvrtá kapitola se věnuje návrhu systému pro analýzu hudby. Systém musí detekovat rytmus, nejen průměrné tempo celé písně, ale i konkrétní časy významných rytmických událostí. Získání těchto údajů popisují v první části kapitoly. Mimo to musí systém rozpoznat z dané písně další informace, potřebné pro vizualizaci. Které informace to jsou a jak je získat, řeší tato kapitola dále. Jedna z významných vlastností hudby, která najde v mém systému využití, je přiřazení písně k určitému hudebnímu žánru. Klasifikace hudebních žánrů je jedním z hlavních problémů, kterými se celá práce zabývá. S využitím některých postupů, použitých při klasifikaci hudebních žánrů, můžeme rozdělit zadanou píseň na souvislé úseky s jednotným charakterem. K tomu použijeme metodu shlukování, popsanou v závěru kapitoly.

V páté kapitole předkládám návrh generování herního prostředí. Popisuji zde, jak by měla hra vypadat a které prvky je potřeba do prostředí zahrnout. Především se zabývám otázkou, jak využít informace získané z hudby ke generování všech potřebných součástí herního prostředí.

Šestá kapitola popisuje implementaci navrženého systému. V první části je popsána implementace systému pro analýzu hudby, se všemi jeho součástmi. Druhá část se zabývá implementací výsledné hry. V obou částech uvádím použitá řešení, nástroje a knihovny.

V sedmé kapitole uvádím dosažené výsledky. Objektivně se dá zhodnotit pouze klasifikace hudebních žánrů. Testování a výsledky této části systému podrobně popisují v této kapitole. Ostatní části hudební analýzy se dají subjektivně hodnotit na výsledné hře. Celá hra se pak potýká s několika technickými problémy, na které v této kapitole upozorňuji.

Osmá závěrečná kapitola shrnuje a hodnotí celou tuto práci. Nakonec uvádím náměty na úpravy a další vývoj mé práce.

## **Návaznost na semestrální projekt**

V semestrálním projektu, který předcházel této práci, jsem se zabýval pouze analýzou hudby. Vytvořil jsem návrh celého systému pro analýzu hudby, z tohoto návrhu vychází čtvrtá kapitola této diplomové práce. Související teorii, kterou popisují v první kapitole, jsem nastudoval také v rámci semestrálního projektu. Dále jsem se seznámil s existujícími hudebními hrami, které popisují ve třetí kapitole této práce.

V rámci semestrálního jsem implementoval pouze jednu z metod detekce rytmu, konkrétně metodu založenou na sledování změn energie, popsanou v úvodu kapitoly 4.1. Tuto implementaci jsem zahrnul do systému pro analýzu hudby, vytvořeného v této diplomové práci.



## 2 Digitální hudba a vnímání

Hudba je ve své fyzikální podstatě pouze zvuk – akustické vlny. Mechanismy tvorby a slyšení zvuku studuje akustika. Zvukové vlny můžeme poměrně přesně zaznamenat a zpracovat digitálně, na základě teorie zpracování signálů. Můžeme tak sledovat fyzikální vlastnosti signálu, jako jsou například frekvence nebo energie. Hudbu lépe charakterizují vlastnosti jako melodie a harmonie, které s fyzikálními vlastnostmi zvukového signálu úzce souvisí. Vnímání hudby člověkem přesto nelze popsat pouhým souhrnem fyzikálních vlastností. Některé vlastnosti lidský sluchový systém zanedbává, a naopak přidává jiné informace, které v signálu obsaženy nejsou. Při analýze hudby je tedy potřeba vzít v úvahu, jakým způsobem hudbu slyší a vnímá člověk, jak ucho přenáší akustické vlny do mozku a jak je mozek vnímá. Tímto vztahem mezi zvukovými signály a lidským vnímáním se zabývá psychoakustika [1]. Podrobněji tuto problematiku popisuje například [2], odtud částečně čerpám i já v této kapitole.

### 2.1 Vnímání hudby člověkem

Pro pochopení fyzikální podstaty zvuku a slyšení je potřeba seznámit se alespoň stručně se způsobem, jakým funguje ucho. Jedná se totiž o velmi složitý orgán, který akustické vlny transformuje na elektrické signály, kterým rozumí mozek. Tato transformace signál silně modifikuje. Abychom mohli modelovat systém lidského vnímání zvuku a hudby, potřebovali bychom vědět, jak mozek přicházející signály interpretuje. Vědci v této oblasti bohužel nemají příliš jasno, jedná se o velmi komplexní systém a k jeho úplnému pochopení máme zatím velmi daleko [3]. I přesto se dostupné znalosti o psychoakustice využívají ve zpracování digitální hudby.

#### Ucho

Ucho provádí transformaci akustických vln na signály, vedoucí do mozku. Zatímco zrak, čich a chuť jsou založeny na chemických procesech, sluchové ústrojí funguje čistě mechanicky [3]. Ucho se skládá ze třech částí: vnější, střední a vnitřní ucho.

Vnější ucho sbírá akustické vlny a přenáší je zvukovodem k bubínku. To je tenká blanka, která se rozechvěje změnou akustického tlaku. Vysoké tóny působí rychlejší chvění, hlasitější zvuky způsobují větší výchylku. Bubínek dokáže filtrovat velmi nízké a intenzivní zvuky, přičemž zůstává citlivý na střední a vyšší frekvence. Díky tomu odlišíme například řeč od okolního hluku, a také jsme chráněni před zvuky, které by náš sluch mohly poškodit. Plyne z toho první významný poznatek pro digitální zpracování zvuku: vnímání zvuku není lineární, zvuky určitých frekvencí a intenzit jsou utlumeny.

Z bubínku je zvuk přenášen do středního ucha. To obsahuje tři pohyblivé sluchové kůstky, propojující bubínek s vnitřním uchem, které se nazývají se kladívko, kovadlinka a třmínek. Bubínek transformuje akustické vlny na mechanický pohyb a kůstky přicházející signál zesilují.

Nejvýznamnějším orgánem ve vnitřním uchu je takzvaný hlemýžď. Je to zakroucená trubice, vyplněná tekutinou. Do této tekutiny jsou přenášeny zesílené vlny, které rozkmitávají membránu s vlasovými buňkami. To jsou nervy, z nichž každý reaguje na jinou frekvenci. Tyto nervy vysílají elektrické signály do mozku. V podstatě tak v tomto orgánu probíhá frekvenční analýza přicházejícího signálu, mozek dostane informaci o tom, jak jsou jednotlivé frekvence zastoupeny.

## Psychoakustika

Psychoakustika zkoumá jak fyzikální jevy spojené se zvukem, tak i způsob, jakým člověk zvuk vnímá. Hledá vztah, mezi objektivními fyzikálními vlastnostmi, a subjektivním lidským vnímáním.

### Hranice slyšitelnosti

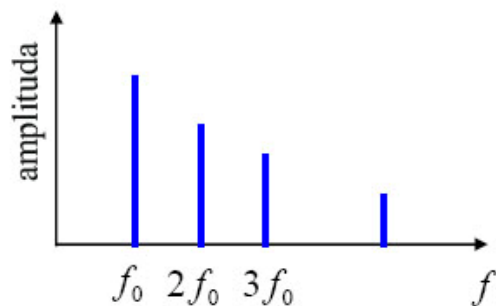
Mezi základní poznatky patří omezení frekvencí, které je člověk schopen slyšet. Pod 16Hz zvuk nevnímáme jako souvislý, výšky nad 20kHz jsou neslyšitelné. Zatímco dolní hranice sluchového pole je celkem stálá, horní hranice s věkem klesá [4]. Tato informace je důležitá především při digitálním záznamu hudby, kdy se spojitý signál vzorkuje. Vzorkovací frekvence se určí právě na základě horní hranice slyšitelnosti podle vzorkovacího teorému, uvedeného ve vzorci 1.

$$f_{\max} < \frac{f_{vz}}{2} \quad (1)$$

Ze vzorce vyplývá, že vzorkovací frekvence  $f_{vz}$  musí být dvakrát větší, než nejvyšší frekvence  $f_{\max}$  obsažená v signálu, která má zůstat zachována i po vzorkování. Pokud tedy bereme v úvahu hranici slyšitelnosti 20kHz, vzorkovat bychom měli na frekvenci 40kHz. Na audio CD se běžně používá vzorkovací frekvence 44,1kHz, takže máme přibližně 2050Hz rezervu nad hranici slyšitelnosti. Vzorkování tedy z pohledu lidského posluchače neovlivňuje audio signál, kvalita digitální hudby zůstává oproti skutečnému spojitému signálu nezměněna.

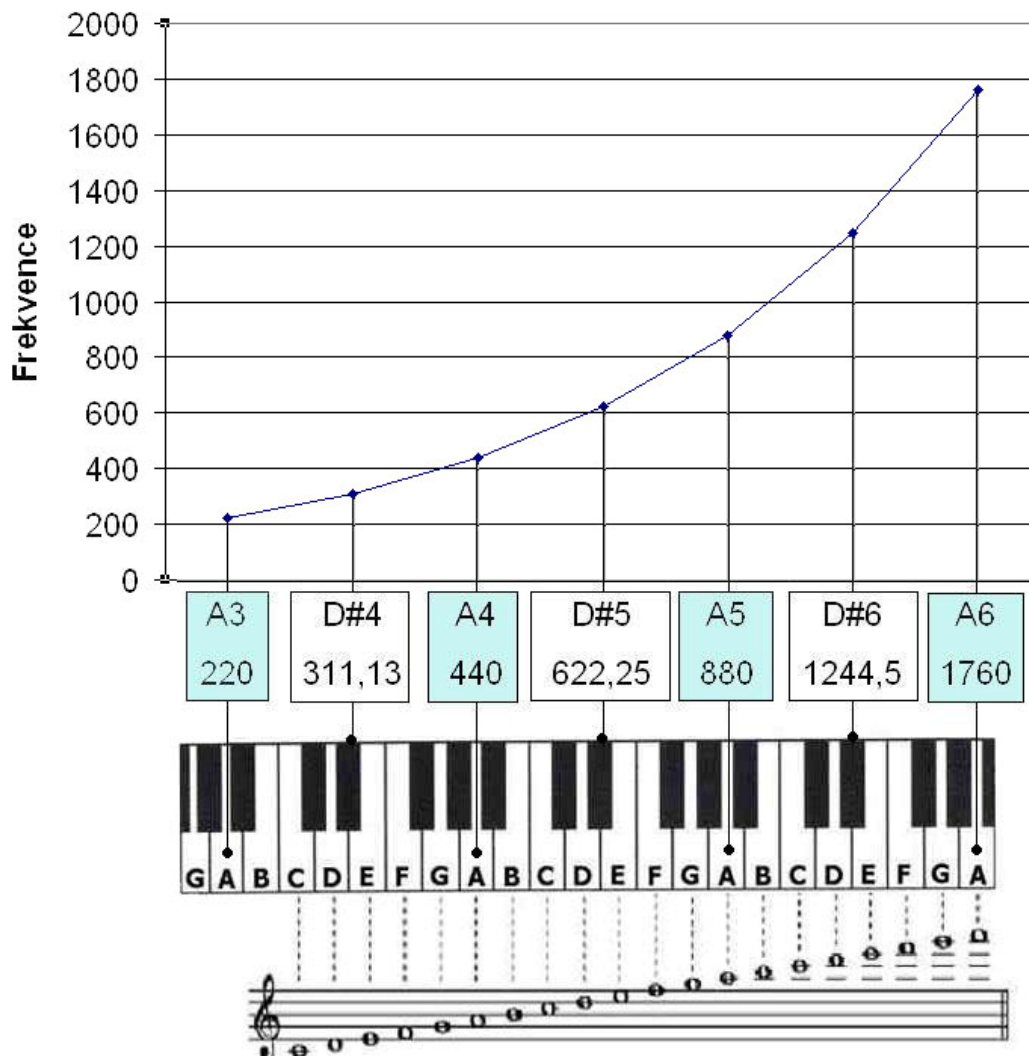
### Vnímání výšky tónu

Nejdříve je nutné ujasnit vztah mezi frekvencemi akustických vln, a hudebními tóny. Tón zahráný hudebním nástrojem je periodickou funkcí času, má čárové spektrum, znázorněné na obrázku 1. Člověk vnímá výšku tónu jako základní frekvenci obsaženou v jeho spektru  $f_0$ .



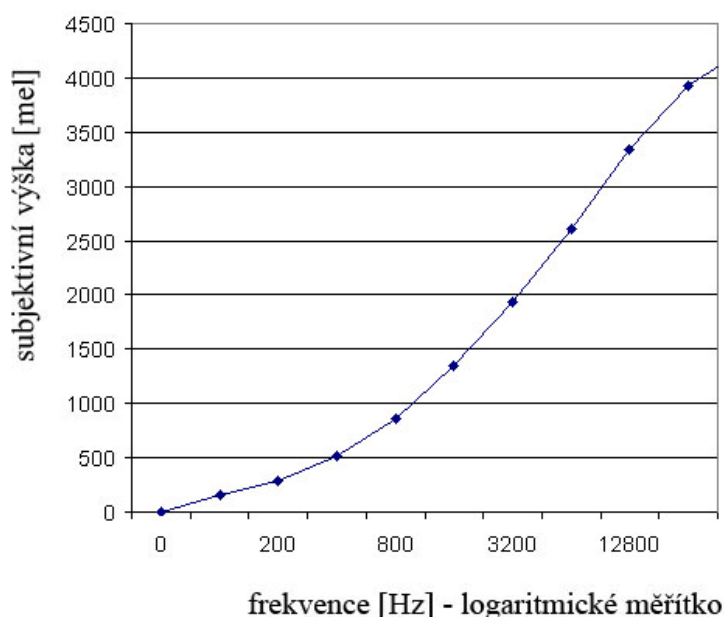
Obrázek 1. Spektrum tónu

Rozložení základních frekvencí jednotlivých tónů není lineární. Pokud se zdvojnásobí frekvence tónu, v hudbě se jedná o rozdíl jedné oktávy (12 půltónů). Frekvenci každého následujícího půltónu dostaneme z předcházejícího násobením dvanáctou odmocninou dvou. Frekvence tedy na stupnici tónů rostou exponenciálně, jak ukazuje obrázek 2.



Obrázek 2 Frekvence tónů

Lidské vnímání výšky tónů však neodpovídá přesně ani tomuto modelu. Subjektivní vnímání výšky závisí také na intenzitě zvuku. Proto při stanovení subjektivní výšky nějakého tónu musíme srovnávat tento zkoumaný tón s normálem, který má standardní hladinu hlasitosti, a to 40 dB [5]. Jednotka subjektivní výšky se nazývá *mel* (zkratka slova melody). Tón 1000 Hz (v intenzitě 40 dB) má výšku 1000 melů. Pro tóny stejné intenzity je závislost subjektivního vnímání výšky na frekvenci (v logaritmickém měřítku) vyjádřena grafem na obrázku 3. Z grafu je zřejmé, že citlivost na změnu výšky tónu klesá na krajích slyšitelného spektra – směrem k vysokým i nízkým tónům. Tato citlivost dále závisí na intenzitě tónů, rozlišovací schopnost se zlepšuje při vzrůstání intenzity asi k 60dB, pak zůstává přibližně stejná [5].



**Obrázek 3** Závislost subjektivní výšky na frekvenci, pro tóny stejné intenzity

### Vnímání hlasitosti

Tak jako výška vnímaného tónu souvisí s frekvencí zvukových vln, vnímání hlasitosti je spojené s intenzitou zvuku (amplitudou vlny). Intenzita se udává v decibelech (dB). Decibel je logaritmická jednotka, udávající poměr dvou veličin. V případě měření hlasitosti se jedná o vztah zkoumané intenzity  $I$  k intenzitě referenčního tónu  $I_0=10^{-12}\text{W/m}^2$ . Hladina intenzity  $L_I$  je dána rovnicí 2.

$$L_I = 10 \log \frac{I}{I_0} [\text{dB}] \quad (2)$$

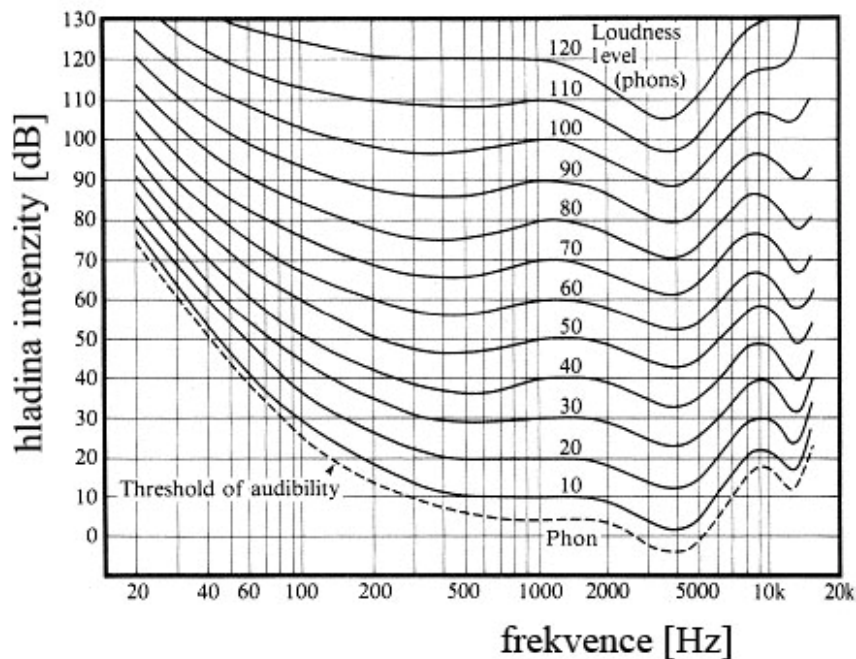
Pro měření intenzity využíváme akustický tlak  $p$ , hladinu intenzity můžeme tedy určit přímo z něj, dle rovnice 3. Referenční hodnota tlaku  $p_0 = 2 \cdot 10^{-5} \text{N/m}^2$ .

$$L_I = 20 \log \frac{P}{P_0} [dB] \quad (3)$$

Hladina intenzity nebo akustického tlaku neumožňují vyjádření subjektivního vjemu zvuku. Funkci, podle které probíhá vnímání hlasitosti, je možné stanovit empiricky [5]. Pro číselné vyjádření subjektivního vjemu hlasitosti byl zaveden pojem hladina hlasitosti  $L_H$ , jednotkou je jeden *fon*. Hladinu hlasitosti zjistíme srovnáním zvuku s referenčním tónem na frekvenci 1000Hz. Číselně odpovídá hladině intenzity referenčního tónu, který se nám zdá stejně hlasitý. Experimentálně byly zjištěny křivky se stejným vjemem subjektivní hlasitosti, které ukazuje obrázek 4. Spodní křivka představuje práh slyšitelnosti – má hodnotu přibližně 4 fony.

Na základě hladiny hlasitosti můžeme vyjádřit samotnou hlasitost  $H$ . Ta respektuje nelinearitu subjektivního vnímání intenzity zvuku [6]. Zvuk s dvojnásobnou hlasitostí se bude pozorovateli jevit dvakrát silnější. Jednotkou hlasitosti je *son*. Hlasitost je dána rovnicí 6.

$$H = 2^{(L_H - 40)/10} [son] \quad (4)$$



Obrázek 4 Křivky stejné hladiny hlasitosti [15]

## 2.2 Audio signály a hudba

Člověk v hudbě nevnímá přímo její fyzikální vlastnosti, jako jsou frekvence nebo energie signálu. Místo toho rozpoznává například melodii a rytmus, mluvíme o takzvaných perceptuálních vlastnostech. Ty však mají vždy základ ve fyzikálních jevech, není možné je oddělit. Zatímco

psychoakustika se snaží zjistit, jak přiřadit fyzikálním vlastnostem zvuku odpovídající vjemy, samotnými fyzikálními jevy se zabývá zpracování signálů. To pokrývá mnohem širší oblast, než jen zvukové signály. Například zpracování obrazu staví na technikách zpracování signálů, protože obraz můžeme chápat jako dvourozměrný signál. Já se v této práci zabývám pouze zpracováním jednorozměrných zvukových signálů.

## Fourierova transformace

Jednou ze základních technik zpracování signálů je fourierova transformace. Tato matematická metoda provádí dekompozici signálu do základních sinusových vln. Superpozicí těchto vln lze získat původní signál, tuto rekonstrukci provádí inverzní fourierova transformace. Díky fourierově transformaci tedy víme, jak silně jsou v signálu zastoupeny vlny jednotlivých frekvencí (frekvenčních pásem). Podobný proces se děje v lidském sluchovém ústrojí (popsáno výše), mozek také dostává informaci o frekvenčním spektru. Fourierova transformace bude tedy v našem případě velmi užitečný základ pro další zpracování audio signálu. Přehledný úvod do této techniky naleznete například zde [7], z tohoto zdroje čerpám v této kapitole.

Pro spojitou funkci jedné proměnné  $f(t)$  je fourierova transformace definována integrálem zapsaným v rovnici 5.

$$F(f) = \int_{-\infty}^{\infty} f(t)e^{-j2\pi ft} dt \quad (5)$$

Pro vzorkovaný signál použijeme diskretní fourierovu transformaci (DFT), kde místo integrálu figuruje suma, jak je vidět v rovnici 6. Pokud máme  $N$  vzorků signálu  $x$ , výstupem je  $N$  frekvenčních pásem. Vzorky jsou komplexní hodnoty, tedy  $x_i = x_{real} + x_{imag}$ .

$$X(n) = \frac{1}{N} \sum_{k=0}^{N-1} x(k)e^{-jk2\pi n/N} \quad \text{pro } n=0..N-1 \quad (6)$$

Transformace počítaná podle tohoto vzorce může být aplikována na signál o libovolné délce, pracujeme přitom s kvadratickou složitostí. Existuje však rychlejší algoritmus, realizující tento výpočet – FFT (Fast Fourier Transform). Jeho omezením je počet vzorků na vstupu, ten musí být mocnina dvou. Složitost výpočtu je pak  $N \cdot \log_2(N)$ . V praxi je vhodné využít právě tento algoritmus. Signál nebudeme totiž zpracovávat vcelku, ale budeme ho dělit na okna zvolené velikosti – volíme tedy mocninu dvou a můžeme bez omezení použít FFT. Při velikosti okna 1024 vzorků se rychlost výpočtu zvýší přibližně stokrát. Konkrétní zápis algoritmu FFT v jazyce C naleznete například v [7].

## Autokorelační funkce

Další ze základních technik zpracování signálů, která může být užitečná při práci s hudbou, je autokorelační funkce. Tato funkce zjišťuje, jak je diskretní signál  $s$  podobný sám sobě, pokud ho posuneme o  $k$  vzorků [8]. Její koeficienty se spočítají podle rovnice 7.

$$R(k) = \sum_{n=0}^{N-1-k} s(n) * s(n+k) \quad (7)$$

Index koeficientu s maximální hodnotou udává periodu, na které je možné v signálu najít jistou pravidelnost. Typicky se tedy využívá tato perioda pro určení základního tónu. Její hodnotu máme ve vzorcích, říká se jí *lag*. Pro určení časové periody  $T_0$  stačí *lag*  $L$  podělit vzorkovací frekvencí  $F_s$ .

$$T_0 = \frac{L}{F_s} [s] \quad (8)$$

Frekvence základního tónu je pak převrácená hodnota periody:

$$F_0 = \frac{1}{T_0} [Hz] \quad (9)$$

Takto určená frekvence může sloužit k určení tónů v monofoní hudbě – tj. takové, kde v jednom okamžiku hraje pouze jeden nástroj jediný tón. V našem případě chceme pracovat s reálnou hudbou, kde hraje mnoho nástrojů a mnoho tónů zároveň, proto nám tento přístup určování frekvence příliš kvalitní informaci nepodá. Nicméně hledání periodičnosti, které je podstatou autokorelační funkce, se nám může hodit pro vyšetření jiné důležité vlastnosti hudby – rytmu. Rytmus bývá přibližně periodický, takže nám autokorelační funkce pomůže v jeho nalezení. Jako vstup však nepoužijeme samotný audio signál, ale mohli bychom použít například energii signálu v jednotlivých frekvenčních pásmech. Dále detekci rytmu proberu v kapitole 4.1.

## 2.3 Klasifikace hudby

V mém systému chci z hudby získávat i jiné vlastnosti, než je rytmus nebo melodie. Pro vizualizaci by bylo užitečné znát například hudební žánr dané písně, nebo aktuální „náladu“ přehrávané hudby. Takové vlastnosti jsou velmi významné, a přitom se dají těžko popsat. S analytickým přístupem si tedy nevystačíme, bude nutné použít metody statistické. Ty spočívají v přiřazení objektů do tříd, na základě statistik dostupných dat. Snaží se najít obecné modely, které co nejlépe popisují jednotlivé třídy dat. Objekty pro klasifikaci mohou být celé písně nebo jejich segmenty, podle konkrétního použití. Celý tento problém se dá rozdělit na dvě oddělené části: extrakce příznaků a klasifikace.

## Extrakce příznaků

Pro klasifikaci je nutné popsat jednotlivé objekty výčtem významných vlastností – takových, které jsou podobné uvnitř tříd, mezi různými třídami se však výrazně liší. Extrakce příznaků ze zadaného objektu získá tyto vlastnosti, vyjádřené číselnými hodnotami. Výstupem je pro každý objekt jeho vektor příznaků.

Pro účely klasifikace hudby není snadné zvolit vhodné příznaky. Můžeme použít fyzikální vlastnosti, jako je energie, průchody nulou atd. Stejně tak je ale vhodné využít i vlastnosti perceptuální, tj. ty, které jsou založeny na lidském vnímání, například rytmus nebo barva zvuku. Oddělit fyzikální vlastnosti od perceptuálních by bylo velmi obtížné, protože perceptuální vlastnosti jsou založeny na fyzikálních, jak bylo popsáno v kapitole 2.1. Najít mezi nimi jasnou hranici není možné, a vzhledem k jejich úzké souvislosti by to nebylo ani užitečné.

Příznakový vektor pro náš systém tedy složíme z vybraných vlastností obou těchto druhů. Které konkrétní vlastnosti je vhodné použít navrhuje například práce [9], já popisuji moji volbu v kapitole 4.2.

## Klasifikace

Klasifikace je proces, který rozděluje data do tříd tak, aby se data uvnitř jedné třídy sobě navzájem podobala – měla podobné vlastnosti. Pro statistickou klasifikaci se použijí metody strojového učení, které můžeme rozdělit do dvou skupin: učení s učitelem a bez učitele. Kterou z těchto možností použijeme závisí na tom, jaká máme dostupná data, a především na způsobu, jakým chceme výsledky klasifikace použít.

Učení s učitelem můžeme použít tehdy, máme-li k dispozici dostatek dat, u kterých známe jejich zařazení do tříd. Tato data se použijí pro natrénování systému. Na základě známého přiřazení trénovacích dat do tříd se vytvoří modely tříd, které v systému zůstanou uloženy. Systém je pak schopen určit třídy pro nová data, podle toho, kterému z vytvořených modelů jednotlivá data nejlépe odpovídají. Systém tohoto typu (učení s učitelem) je vhodný například pro klasifikaci hudebních žánrů. Člověk totiž rozpoznává hudební žánry velmi podobným způsobem: zná mnoho písní, u kterých ví, pod jaký žánr se dají zařadit. Pokud má určit žánr nové písně, určí je podle žánru pro něj známé písně, která zní podobně. Jistě by se toto určení dalo udělat i na základě kritérií, jako jsou například hudební nástroje, které v písni hrají, nebo harmonie písně. Mohli bychom pak například říci, že pokud je v hudbě nejvýraznější zvuk elektrické kytary a harmonie se skládá z jednoduchého opakování čtyř akordů, jedná se pravděpodobně o rock nebo punk. To by ale bylo velmi zjednodušující zařazení, navíc by bylo velmi složité popsat takto všechny významné aspekty hudby. Přírozeňším způsobem je tedy opravdu popis na základě podobnosti, což je zřejmé i ze způsobu, jakým popisují hudbu lidé. Častěji se setkáme s popisem typu „zní to podobně jako U2“ než



s popisem taktu a harmonie neznámé písně. Pro rozpoznání hudebních žánrů na počítači se tedy bude hodit strojové učení s učitelem.

Učení bez učitele využijeme, pokud chceme data rozdělit do tříd podle podobnosti, ale nemáme k dispozici trénovací data, u kterých bychom znali správné zařazení. Použijeme metody shlukování (clustering), které dokáží bez znalosti modelů tříd najít na neznámých datech třídy, sdružující podobná data. Tyto metody hledají shluky dat (jejich příznakových vektorů), které jsou si v daném vektorovém prostoru blízké. Kvalita výsledků silně závisí na kvalitě extrakce příznaků a na výběru použitých příznaků. Pro naše účely předpokládáme, že máme k dispozici trénovací data – písně, u kterých známe jejich žánr. Můžeme tedy použít učení s učitelem, experimentovat s různou strukturou příznakového vektoru, a sledovat které příznaky jsou pro klasifikaci hudby nejvhodnější. Kvalitu systému je vhodné ověřit cross-validací. Experimentálně tedy určíme vhodné příznaky, které pak můžeme použít i pro učení bez učitele. Otázkou zůstává, jestli je učení bez učitele k něčemu užitečné, pokud máme k dispozici trénovací data a můžeme dělat učení s učitelem. Jedním z typických využití metod shlukování je segmentace – rozdělení objektu na souvislé části podle obsahu. Často se používá při zpracování obrazu, který se pomocí metod shlukování rozdělí například podle textur v obraze. Stejně tak můžeme segmentaci použít na hudbu, abychom jednu píseň rozdělili na souvislé úseky. Časy, ve kterých píseň přechází z jednoho segmentu do druhého, budou označovat místa, ve kterých se píseň nějak mění, například přechází ze sloky do refrénu. Informace o těchto místech může být velice užitečná pro vizualizaci, proto v mém systému použiji také učení bez učitele, přestože mám k dispozici i data pro učení s učitelem.

## 3 Současné hry založené na hudbě

S vzrůstající popularitou digitálních multimédií se zvýšil i zájem o hudební hry, které byly dříve okrajovým herním žánrem. První hry, které se zaměřovaly na hudbu, byly založené na testování hráčovy hudební paměti. Úkolem hráče je zopakovat sekvenci, která mu byla přehrána. Většina dnešních hudebních her je však založena na rytmu hudby, ten se dá využít různě. Typičtí zástupci rytmických her budou popsáni níže. Poslední významnou skupinou jsou hry, které vyžadují od hráče zpěv a hodnotí jeho intonaci. Tyto „karaoke“ hry nejsou svým zaměřením jakkoli inspirativní pro můj projekt, nebudu je dál probírat. Existují i jiné herní koncepty využívající hudbu jako hlavní motiv, z hlediska tématu této práce však nejsou zajímavé, ani o nich se nebudu dále zmiňovat.

Nejnámější hudební hry dneška najdeme na herních konzolách (například PlayStation, Xbox). Já jsem měl možnost vyzkoušet si jejich alternativy pouze na jiných platformách, přesto se budu snažit v této kapitole popsat především mainstreamové konzolové tituly. Většinu informací jsem čerpal z wikipedie [10], nejlepším zdrojem a ukázkou těchto jsou však videa na youtube – jak oficiální trailery a tutoriály, tak i uživatelské recenze a záběry z hraní.

### 3.1 Hry založené na paměti

Hry tohoto typu ukazují hráči jisté sekvence, které musí hráč opakovat. Pro hráče se typicky jedná o sekvenci tlačítek, které musí stisknout. Ve hře může být tato sekvence reprezentována vizuálně – barvami, objekty, animacemi a podobně, v takovém případě se nemusí jednat o hudební hry, tento koncept je použitelný sám o sobě. Požadovaná sekvence však může být reprezentována také pomocí zvuků, spojení vizuálního vjemu s hudbou je atraktivnější, než zapojovat do hry pouze jeden ze smyslů. Hry založené na hudební paměti vlastně testují hráčovu krátkodobou a eidetickou paměť. Sekvenci je totiž potřeba zopakovat okamžitě, v paměti ji stačí udržet velmi krátkou dobu (v řádu sekund – krátkodobá paměť), je však nutné zopakovat ji zcela přesně (eidetická paměť).

#### Space Channel 5

Jako příklad hudebních her založených na paměti můžeme uvést hru Space Channel 5, vydanou pro platformu PlayStation. Tato hra nevyžaduje opakování sekvencí tónů, ale tanečních pohybů. Ty je potřeba zopakovat nejen ve správném pořadí, ale i ve správném rytmu podle hrající hudby. Hudbu je tedy ve hře významným prvkem, mimo dobré eidetické paměti musí hráč prokázat i smysl pro rytmus. Hra má svůj děj, hlavní hrdinka tančí skrze připravené lokace dle předem daného scénáře, hudba která hru doprovází je tedy také předem určená a uživatel nemá možnost volit vlastní hudbu.

## 3.2 Hry založené na rytmu

Hry založené na rytmu tvoří dnes většinu hudebních her. Rytmus se v nich dá využít různým způsobem, jednotlivé koncepty bude nejnázší popsat na konkrétních příkladech.

### Guitar Hero

Hra Guitar Hero je v dnešní době fenoménem, objevuje se ve filmech a seriálech, stala se populární společenskou zábavou. První titul z této série byl uveden na trh roku 2005 pro PlayStation 2, další verze byly rozšířeny i na většinu dalších platform. Na stejném konceptu staví dnes mnoho podobných her, Guitar Hero je však nejtypičtějším zástupcem této kategorie. Výrazným poznávacím znamením je speciální ovladač ve tvaru kytary. Grafika hry je typická zobrazením přijíždějícího pásu, který připomíná krk kytary. Na něm jsou na pěti strunách rozmístěné barevné body a čáry, které přijíždějí směrem k bližšímu konci pásu, tohoto bodu dosahují v rytmu hudby. Hráč tedy musí do rytmu mačkat tlačítka na ovladači, které odpovídají strunám na zobrazeném pásu. Navíc poloha zobrazených strun přibližně odpovídá výšce tónů hlavní kytarové melodie, směrem doprava se výška zvyšuje. Sekvence tlačítek, kterou má hráč mačkat, tedy není závislá jen na rytmu, ale i na melodii. Tu však nemusí hráč nutně rozeznat sluchem, je zřetelná z vizuálního rozmístění přijíždějících objektů. Provázání melodie, rytmu, a vizuálních vjemů má pozitivní vliv na zážitek hráče, který si tak připadá, jakoby opravdu hrál na kytaru. Herní prostředí ukazuje obrázek 5.



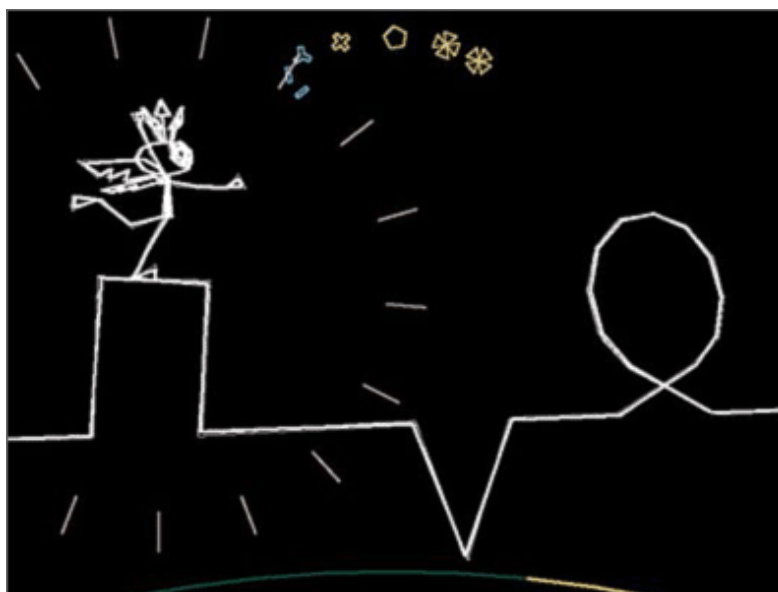
Obrázek 5 Guitar Hero [16]

Hra nedokáže tvořit herní prostředí z libovolně zvolené hudby, je omezená pouze na předem připravené písně. To je chytrá marketingová strategie vývojářů hry, na prodeji dalších písní do hry vydělají další nemalé peníze – speciální edice hry Guitar Hero: Aerosmith vydělala této kapele víc peněz, než kterékoli jejich album [11]. Na druhou stranu tuto závislost na oficiálně vydaných písních vidím jako největší nedostatek hry. Malou kompenzací je možnost nahrávat vlastní písně a sdílet je ostatními uživateli na internetu.

## Vib-Ribbon

Tato hra byla vydána roku 1999 pro PlayStation, tehdy na první pohled šokovala svou grafikou, která byla na tehdejší dobu velmi primitivní. Tato grafická stylizace byla však úmyslem vývojářů, v kontrastu s jednoduchou grafikou hra nabízela skvělou hratelnost a spoustu zábavy. Koncept je jednoduchý – hlavní a jediný hrdina Vibri běží po tenké čáře, na která se místy kroucí a tvoří tak překážky. V rytmu písně Vibri překonává tyto překážky tím, že hráč stiskne tlačítko odpovídající danému typu překážky. Existují překážky několika základních typů, jsou snadno rozeznatelné a dají se kombinovat, pak musí hráč stisknout více tlačítek zároveň. Překonání překážek se děje v rytmu písně, hra je ztížena změnami perspektivy pohledu.

Dosud popsaný systém nepřináší nic převratného, nejvýznamnější vlastností hry byla však možnost nechat si vygenerovat herní prostředí z libovolné písně. Celý program hry se vešel do operační paměti PlayStationu, uživatel mohl pak vytáhnout z mechaniky CD s hrou, a vložit libovolné hudební CD. Ze zvolené písně se pak vygeneroval nový level hry, jeho náročnost samozřejmě odpovídala tempu a rytmice zvolené písně. Tato možnost volby vlastních písní spolu s jednoduchou a přitom velmi zábavnou hratelností učinily z této hry legendu tohoto žánru. Herní prostředí ukazuje obrázek 6.



Obrázek 6 Vib-Ribbon [17]

Pro mě byla tato hra první inspirací pro tuto práci. Chci, aby se moje hra držela jednoduchosti a snadné hratelnosti stejně jako Vib-Ribbon, generování prostředí z libovolné hudby je nutný požadavek. Chci však, aby prostředí nebylo generováno jen na základě rytmu, ale i dalších vlastností hudby. Také se chci pokusit o pestřejší grafiku ve 3d.

## Audiosurf

Tato hra podobně je jako Vib-Ribbon založena na rytmu a umí generovat prostředí z libovolné písně. Uvádím ji zde jako možnost modernějšího přístupu ke stejné koncepci. Grafika odpovídá mé představě o hře, kterou chci tvořit. Hráč ovládá raketu letící po tříproudové dráze, která se vlní do rytmu hudby. Tím je rytmus vizuálně zdůrazněn, ale tato skutečnost neovlivňuje samotnou hru. Hudební žánr je zde skombinován s „puzzle“ hrou – na dráze jsou rozmístěny barevné kvádry, kterým hráč buď vyhýbá, nebo je sbírá, tak aby složil co největší barevný blok. Tyto manévry a kolize s kvádry se typicky dějí do rytmu hudby, takže hráč opět tiskne sekvence tlačítek v rytmu, tentokrát se ho však nemusí držet příliš striktně. Herní prostředí ukazuje obrázek 7.

Zatímco Vib-Ribbon je pro mě inspirací po stránce jednoduchosti, hratelnosti a čisté orientace na hudbu, Audiosurf se mi líbí svou zdařilou grafikou, která hudbu dobře kopíruje – nejen vlněním dráhy, ale i efekty na pozadí.



Obrázek 7 Audiosurf [18]

## 4 Návrh systému pro analýzu hudby

Základem mojí práce je systém, který analyzuje požadovanou píseň a extrahuje z ní informace potřebné pro vytvoření herního prostředí. Moje hra je zaměřena na rytmus, je tedy nutné vytvořit systém pro detekci rytmu. Další informace, která může být pro vizualizaci užitečná, je hudební žánr písně. Mimoto můžeme zjistit informace o souvislých segmentech, mezi nimiž se charakter písně mění.

### 4.1 Detekce rytmu

Detekce rytmu na počítači je zajímavý a otevřený problém, protože se jedná o snahu aproximovat lidský vjem, který je těžko popsateľný analyticky. Existuje mnoho algoritmů pro detekci rytmu, konstrukce jednoho z nich je dobře popsána v článku [12]. Z tohoto zdroje vychází můj první návrh, založený na sledování změn energie. Druhá varianta detekce rytmu využívá autokorelační funkce, tato varianta poskytne jiný typ výsledků. Způsob použití autokorelačních koeficientů vychází z [9]. Obě varianty budou mít v koncovém systému jiné využití.

#### Detekce rytmu pomocí sledování změn energie

Rytmus budeme chápat jako sekvenci „beatů“, úkolem této metody je najít konkrétní časy všech takových událostí v hudbě. Pokud známe tyto časy, můžeme je použít pro generování událostí ve hře – tak budou nejlépe odpovídat událostem v hudbě. Abychom se propracovali k finální podobě detekčního systému, musíme začít od jeho jednodušších variant, které budeme postupně optimalizovat.

##### Detekce podle energie signálu

Základním principem je sledování energie zvukového signálu. Vycházíme z předpokladu, že člověk vnímá beat jako prudkou změnu energie. Stačí nám tedy počítat celkovou energii signálu a uchovávat pro ni krátkou historii. Okamžitou energii porovnáme s průměrem uchované historie, pokud ho okamžitá energie přesáhne o jistou konstantu, detekovali jsme beat. Konstantu pro porovnání je potřeba určit experimentálně, délka historie pro určení průměru stačí okolo jedné vteřiny – přibližně v takovém rozsahu porovnává slyšené zvuky i lidský sluchový systém [12].

Tento systém detekce rytmu by mohl poskytnout přijatelné výsledky pro hudbu, kde je změna energie výrazná přes celé frekvenční spektrum – například v taneční hudbě, jako je techno. U jiných hudebních žánrů je situace složitější, musíme použít následující přístup:

## **Detekce podle energie v jednotlivých frekvenčních pásmech**

Například v rockové hudbě často slyšíme v celém středním pásmu dominantní zvuk elektrických kytar, jehož energie se mění jen málo a nepravidelně. Rytmus pak udávají beaty od bicích a od basy, typicky v nízkém frekvenčním pásmu. Metoda sledování energie signálu tyto beaty neodhalí, protože celková energie je více ovlivněna silnou konstantní energií středního pásma. Oproti tomu lidský sluchový systém tyto beaty, „uzavřené“ v určitém frekvenčním pásmu, vnímá bez problémů. Lidské ucho totiž provádí frekvenční analýzu zvukového signálu, popsanou v kapitole 2.1. Aniž bychom si to uvědomovali, náš mozek analyzuje, co se děje v jednotlivých frekvenčních pásmech, a detekuje i události, které jsou z pohledu celkové energie zanedbatelné. Pro detekci rytmu na počítači tedy potřebuje obdobnou frekvenční analýzu.

Převod signálu z časové do frekvenční domény spočítáme pomocí FFT (Fast Fourier Transform). Můžeme experimentovat počtem frekvenčních pásem – 64 nebo 128 se jeví jako vhodné hodnoty. Na vstupu FFT bychom ale museli mít 64 nebo 128 vzorků signálu, což jsou časově příliš malé úseky, beat který hledáme určitě netrvá pouhých 1/344 sekund (128 vzorků při vzorkovací frekvenci 44,1kHz). Řešením je počítat FFT z většího časového úseku – například 1024 vzorků, a ve výsledném spektru sloučit sousední pásma, například každých 8 sousedních pásem zprůměrovat do jednoho – dostaneme 128 nových pásem.

Vzhledem k tomu, že pracujeme se vzorkovaným signálem, musíme vzít v úvahu vzorkovací teorém. Podle něj nejvyšší frekvence  $F_{\max}$  obsažená v signálu musí být menší než polovina vzorkovací frekvence  $F_{vz}$ . Výsledné spektrum z FFT je ale v rozsahu  $0..F_{vz}$ , užitečné informace tedy najdeme pouze v první polovině spektra. V druhé polovině najdeme symetrický obraz první poloviny, protože spektrum reálných signálů (tj. s nulovou imaginární složkou) je symetrické podle svého středu.

Zajímá nás tedy pouze první polovina spektra, pro ni budeme uchovávat historii všech pásem, v rozsahu přibližně jedné vteřiny. Pokud okamžitá energie v jednom pásmu přesáhne průměr historie tohoto pásma o jistou konstantu, detekovali jsme beat. Mimo to máme k dispozici informaci o tom, z kterého pásma beat přišel, a kolik pásem vygenerovalo beat ve stejnou dobu. Tyto doplňující informace můžeme použít pro zajímavější a konkrétnější vizualizaci.

Tato detekce rytmu poskytuje výrazně lepší výsledky, než detekce na základě celkové energie. Přesto je však vhodné udělat další optimalizace:

## **Detekce podle energie v pásmech s proměnnou šířkou**

Lidský sluchový systém má lepší rozlišení v nižších frekvencích, tuto skutečnost je vhodné zohlednit. V předešlé metodě jsme 1024 frekvenčních pásem dělili rovnoměrně – 8 sousedních pásem vytvořilo jedno nové pásmo. Tentokrát sloučíme menší počet pásem v nízkých frekvencích, tím dostaneme užší

pásma a lepší rozlišení. Směrem k vyšším frekvencím bude šířka pásma narůstat a rozlišení klesat. Tak se více přiblížíme lidskému vnímání hudby a detekce rytmu se tím zpřesní. Šířku pásma určíme podle následujících vzorců:

Předpokládáme 1024 pásem na výstupu FFT, používáme polovinu (512), chceme dostat 64 nových pásem. Šířka nového pásma  $w_i$  roste lineárně se vzrůstajícím indexem  $i$  dle rovnice 10.

$$w_i = a * i + b \quad (10)$$

Potřebujeme najít konstanty  $a$ ,  $b$ . Šířku prvního pásma  $w_1$  si zvolíme libovolně, dosadíme do předchozí rovnice a dostaneme rovnici 11.

$$w_1 = a + b \quad (11)$$

Musíme pohlídat, aby součet šířek všech nových pásem zůstal 512. To vyjadřuje rovnice 12. Nových pásem bude  $n=64$

$$\sum_{i=1}^n w_i = 512 = n * b + a * \sum_{i=1}^n i = n * b + a * \frac{n * (n + 1)}{2} = 512 \quad (12)$$

Z rovnic 11 a 12 dostaneme soustavu dvou rovnic o dvou neznámých, kterou můžeme vyřešit ručně. Získáme tak konstanty  $a, b$  pro náš systém. S nimi už snadno spočítáme šířku pásem dle rovnice 10.

Proměnná šířka frekvenčních pásem lépe aproximuje lidské vnímání hudby, detekce rytmu je díky tomu úspěšnější. Nabízí se však ještě jedna optimalizace – proměnný práh.

### Adaptivní detekce s proměnným prahem

Při každém novém výpočtu aktuální energie porovnáváme zjištěnou hodnotu v daném pásmu s průměrnou hodnotou historie (dlouhé 1s). Ve výše popsané metodě jsem uvedl, že nová hodnota musí průměr přesáhnout o jistou konstantu, kterou musíme určit experimentálně. Tato konstanta se však dá upravovat za běhu výpočtu – podle aktuálního rozptylu hodnot v historii. Určíme standardní odchylku  $\sigma$  z hodnot  $x_i$  v historii.  $N$  je počet těchto hodnot,  $\bar{x}$  je jejich aritmetický průměr.

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \bar{x})^2} \quad (13)$$

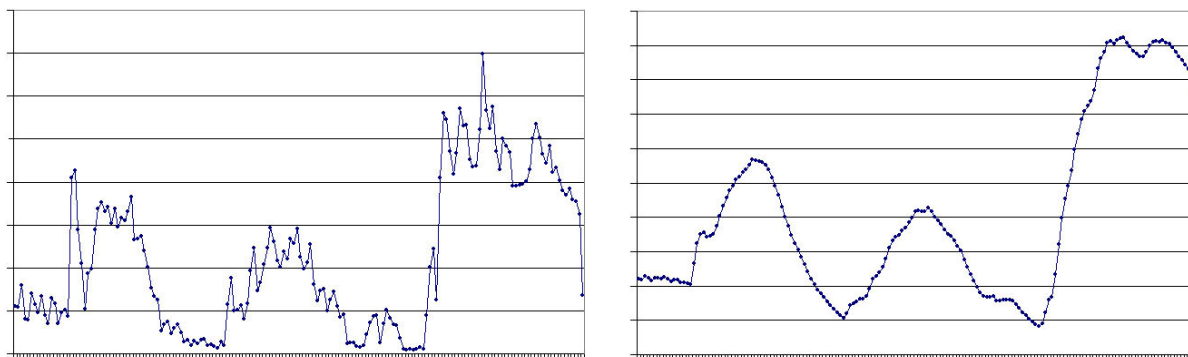
Abychom detekovali beat, musí okamžitá energie v daném pásmu překročit průměrnou hodnotu o spočtenou standardní odchylku, ke které přičteme jistou konstantu. Stále nám tedy zůstává konstanta, kterou musíme experimentálně určit. Ta má ale jinou hodnotu než měla předešlá konstanta pro porovnání a především se porovnávací práh mění podle rozptylu hodnot v historii. Díky tomu se detekce adaptuje na zpracovávanou hudbu, a dostáváme opět lepší výsledky než s pevně nastaveným prahem.



## Detekce rytmu pomocí autokorelační funkce

Tato metoda si neklade za cíl najít konkrétní časy všech beatů v písni, jejím úkolem je zhodnotit celkově rytmickou stránku písně, nebo jejího úseku. Tato rytmická charakteristika poslouží jako část příznakového vektoru pro klasifikaci. K jejímu určení se hodí autokorelační funkce. Tato funkce zjišťuje, jak je signál při posunutí podobný sám sobě. Hledá tedy v signálu určitou periodičnost – tu hledáme i v rytmu hudby.

Jako vstupní signál autokorelační funkce použijeme časový průběh energie jednoho frekvenčního pásma, vyhlazený klouzavým průměrem. Tuto energii před vyhlazením a po něm ukazuje obrázek 8. Indexy největších koeficientů na výstupu autokorelační funkce udávají *lagy* odpovídající periodám mezi nejvýraznějšími beaty (s výjimkou indexu 0, zde je koeficient vždy největší). My vezmeme 3 největší koeficienty, jejich hodnoty uložíme do histogramu k jejich indexu. Opakujeme celý výpočet pro všechna frekvenční pásma. Ve výsledném histogramu nalezneme dvě největší špičky na nenulové pozici. Ty odpovídají periodám dvou nejvýraznějších opakujících se beatů v daném úseku písně. Můžeme z nich určit průměrné tempo těchto beatů a další vlastnosti, které vyjadřují rytmickou charakteristiku úseku písně. Odtud tedy počítáme rytmické vlastnosti, které použijeme v našem příznakovém vektoru.



**Obrázek 8** Energie jednoho frekvenčního pásma v čase před a po vyhlazení klouzavým průměrem

## 4.2 Klasifikace hudebních žánrů

Prvním významným rozhodnutím u klasifikace je výběr příznaků, které se použijí. Neméně důležitá je volba klasifikátoru.

### Volba příznaků

Při volbě příznaků jsem se inspiroval návrhem předloženým v práci [9]. Příznaky jsou zde rozděleny do dvou skupin: Vlastností získané ze spektra a ze samotného signálu, v zahraniční literatuře označované souhrnným termínem „musical surface features“ [13]. Druhou skupinu tvoří vlastnosti rytmické.

### Musical surface features

Budu používat anglické názvy těchto vlastností. Nepodařilo se mi pro většinu z nich najít české ekvivalenty, které vzhledem k úzké specializaci těchto výrazů pravděpodobně ani neexistují. Níže uvedené vlastnosti, mimo poslední (*Low energy*), je vhodné počítat v delším časovém úseku, než je jedno okno signálu, používané pro zpracování. Na druhou stranu není pro určení hudebního žánru nutné slyšet celou píseň, stačí krátký úsek. Jeho délka by měla být minimálně 3 vteřiny – experimenty ukázaly, že člověk je schopen na této délce žánr rozpoznat [9]. Pro všechna okna na tomto úseku spočítáme níže uvedené vlastnosti, dále spočítáme jejich průměr a rozptyl na celém úseku, a teprve tyto hodnoty použijeme do příznakového vektoru pro klasifikaci. Výjimku tvoří příznak *Low Energy*, který se počítá jednou pro celý úsek.

### Spectral Centroid

Jedná se o těžiště spektrálního tvaru. Jeho hodnota je spojena s vnímáním „jasu“ zvuku. Spočítá se ze spektra jako vážený průměr frekvencí  $f_n$  přítomných v signálu, jejich síla  $x_n$  je použita jako váha pro průměr.  $N$  je počet frekvenčních pásem spektra. Výpočet provedeme dle rovnice 14.

$$C = \frac{\sum_{n=0}^{N-1} f_n * x_n}{\sum_{n=0}^{N-1} x_n} \quad (14)$$

### Spectral rolloff

Rolloff je další vlastnost, charakterizující spektrální tvar. Jeho hodnota udává frekvenci, pro kterou platí, že 85% energie ve spektru je pod touto frekvencí. Některé systémy pracují s lehce odlišnou hodnotou, nejčastěji se však setkáme s 85%. Rolloff je označen  $R$ , počet frekvenčních pásem spektra je  $N$ , energie v daném pásmu  $x_n$ .

$$\sum_{n=1}^R x_n = 0.85 * \sum_{n=1}^N x_n \quad (15)$$

### Spectral Flux

Flux  $F$  vyjadřuje míru změny spektrálního tvaru mezi dvěma v čase sousedními okny.  $N$  je počet frekvenčních pásem ve spektru,  $m_n$  je normalizovaná velikost energie v pásmu  $n$ , index okna v čase je  $t$ .

$$F = \sum_{n=1}^N (m_n[t] - m_n[t-1])^2 \quad (16)$$

### Zero Crossing Rate

Vlastnost signálu v časové doméně, používaná ve zpracování řeči a signálů obecně. Známy český název „počet průchodů nulou“ charakterizuje smysl této vlastnosti. Její hodnota  $ZCR$  udává, kolikrát signál projde na sledovaném úseku nulou – to poznáme podle změny znaménka dvou po sobě jdoucích vzorků  $x_n$ .

$$ZCR = \sum_{n=1}^N |\text{sign}(x_n) - \text{sign}(x_{n-1})| \quad (17)$$

Funkce  $\text{sign}$  vrací 0 pro záporný argument a 1 pro kladný argument.

### Low energy

Na rozdíl od předešlých vlastností, které se počítají zvlášť v jednotlivých oknech delšího úseku, low energy neboli nízká energie se počítá jednou pro celý úsek. Vyjadřuje procentuální část všech oken úseku, která mají nižší energii než je průměr v úseku. Hodnota bude větší u písní, které obsahují více tichých částí.

## Rytmické vlastnosti

Pro určení rytmických vlastností použijeme autokorelační metodu popsanou v kapitole 4.1. V histogramu, který uvedená metoda spočítá, nalezneme 2 největší nenulové koeficienty. Ty udávají *lagy* odpovídající periodám mezi nejvýraznějšími beaty, navíc známe relativní sílu těchto beatů, protože jsme ke každému indexu ukládali i hodnotu koeficientu. Na stejném úseku, na kterém jsme počítali „musical surface features“, spočítáme tento histogram a určíme z něj následující rytmické příznaky:

- Relativní síla nejvýraznějšího beatu. Spočteme jako podíl největší hodnoty v histogramu (na nenulovém indexu) se sumou všech hodnot. Výsledek udává, jak výrazný je rytmus – o kolik jsou beaty silnější než zbytek signálu.
- Relativní síla druhého nejvýraznějšího beatu – analogie předchozí vlastnosti.
- Podíl indexů první a druhé největší špičky v histogramu. Vyjadřuje poměr hlavního beatu k druhému nejvýraznějšímu. U písni s výrazným, pravidelným rytmem, se bude hodnota blížit ke 2 nebo k 0,5.
- Perioda hlavního beatu v BPS (beats per second) – spočteme z indexu vrcholu histogramu.
- Perioda druhého nejvýraznějšího beatu v BPS – analogie předchozí vlastnosti.
- Suma celého histogramu. Udává celkovou sílu rytmu.

## Klasifikátor

Pro statistickou klasifikaci hudebních žánrů je vhodné využít učení s učitelem, jak je popsáno v kapitole 2.3. Potřebujeme tedy velké množství trénovacích dat – písni správně přiřazených k hudebním žánrům. Klasifikátor podle těchto dat vytvoří modely jednotlivých tříd, které se uchovávají v systému a slouží ke klasifikaci nových dat. Podoba těchto modelů závisí na použitém klasifikátoru. Pro volbu vhodného klasifikátoru je dobré znát přibližné rozložení dat, která chceme klasifikovat. Podle rozložení dat v daných třídách můžeme vybrat klasifikátor, který využívá modely odpovídající tomuto rozložení.

Pro hudbu z hlediska žánrů je však těžké vyvodit jakékoli závěry o rozložení dat uvnitř tříd. Hudební žánry nemají pevně stanovené hranice, tvorba jednoho umělce často zasahuje do více různých žánrů a u některých písni se i lidé těžko shodnou na zařazení. Hudební žánry jdou navíc jen velmi těžko popsat, případný popis je vždy nepřesný. Lidé je tedy často definují pomocí typických představitelů – například pro hard rock to mohou písni od AC/DC, Led Zeppelin, Black Sabbath a dalších. Tomuto způsobu definování tříd pomocí příkladů odpovídá funkčnost klasifikátoru KNN – k Nearest Neighbours. Jeho funkčnost bývá někdy označována jako „learning by example“ [14], tedy učení podle příkladů. Tato funkčnost přesně odpovídá mým potřebám. Klasifikátor KNN nestaví na

žádných předpokladech o rozložení dat. Pouze uchovává polohu všech trénovacích dat ve vektorovém prostoru, a nová data klasifikuje na základě analýzy  $k$  nejbližších sousedů. Analýzou může být například vážený průměr, kde váha je vzdálenost od klasifikovaného objektu. Hodnota  $k$  – tj. kolik nejbližších sousedů bereme v úvahu, se dá určit experimentálně.

Pro klasifikaci hudebních žánrů tedy použijí KNN klasifikátor. Ke stejnému závěru došla i diplomová práce, zabývající se pouze klasifikací hudebních žánrů [9].

## 4.3 Segmentace písně

Výše popsaná detekce hudebních žánrů by se ve výsledné hře dala použít pro segmentaci písně. Žánr se totiž bude určovat pro krátké (přibližně 3s) úseky a je jisté, že klasifikátor nebude všechny úseky písně vyhodnocovat jako jediný žánr. To není dáno pouze faktem, že každá statistická klasifikace vykazuje určité procento chyby. Je to způsobeno především změnami charakteru písně v jejím průběhu. Příkladem může být rocková píseň, která obsahuje pomalé pasáže s výrazným smyčcovým zvukem. I lidský posluchač by mohl malý úsek takové písně označit jako klasickou hudbu. Jiné rockové písně obsahují rychlé, silně rytmické části, snadno zaměnitelné s taneční hudbou. Tyto úseky odlišného charakteru právě potřebujeme vizualizovat. Klasifikaci hudebních žánrů můžeme tedy použít k rozdělení písně na souvislé úseky, z nichž každý budeme zobrazovat jiným stylem.

Existuje však druhá možnost, jak tuto segmentaci písně provést. Využije se metod strojového učení bez učitele. Na všech úsecích písně, původně určených pro klasifikaci hudebních žánrů, provedeme stejnou extrakci příznaků. Na získané vektory aplikujeme některou z metod shlukování. Na výstupu dostaneme souvislé segmenty písně, kde každý má jiný charakter. Důležité je, že nezáleží na hudebním žánru, ke kterému bychom segment přiřadili. K oddělení segmentu stačí, že charakter hudby je zde jiný a není nutné specifikovat, v čem se liší. Proto bude pravděpodobně využití těchto segmentů pro styl vizualizace výhodnější, než segmenty dané klasifikátorem žánrů. Bude však nutné vyzkoušet obě možnosti a ověřit tak tuto teorii.

## 5 Návrh generátoru herního prostředí

Před začátkem tvorby hry je nutné rozhodnout se, jak má hra vypadat. Je potřeba zvolit herní prostředí a navrhnout herní systém. Moje hudební hra bude založena na podobném principu, jako Audiosurf, popsany v kapitole 3.2. Hráč bude v mojí hře ovládat auto nebo raketu na dráze, která bude vygenerována podle hudby. Dráha bude pro každou píseň jiná, proto je nejsnazší celé herní prostředí stylizovat do abstraktně pojatého vesmíru. Do takového prostředí se lépe hodí raketa než auto, po dráze tedy poletí model vesmírné lodě. Aby hra zůstala synchronizovaná s hrající hudbou, musí raketa letět konstantní rychlostí. Hráč tedy nemůže rychlost pohybu rakety nijak ovlivnit, proto se nejedná o závodní hru. Spíše je potřeba hru zpracovat jako arkádu, kde máme na trati rozmístěné určité objekty a cílem hry je některé objekty sbírat a jiným uhýbat. Kolize s nimi nesmí nijak ovlivnit rychlost rakety. Aby bylo prostředí zajímavější, je vhodné umístit nějaké objekty také okolo trati a vytvořit vizuální efekty (například výbuchy).

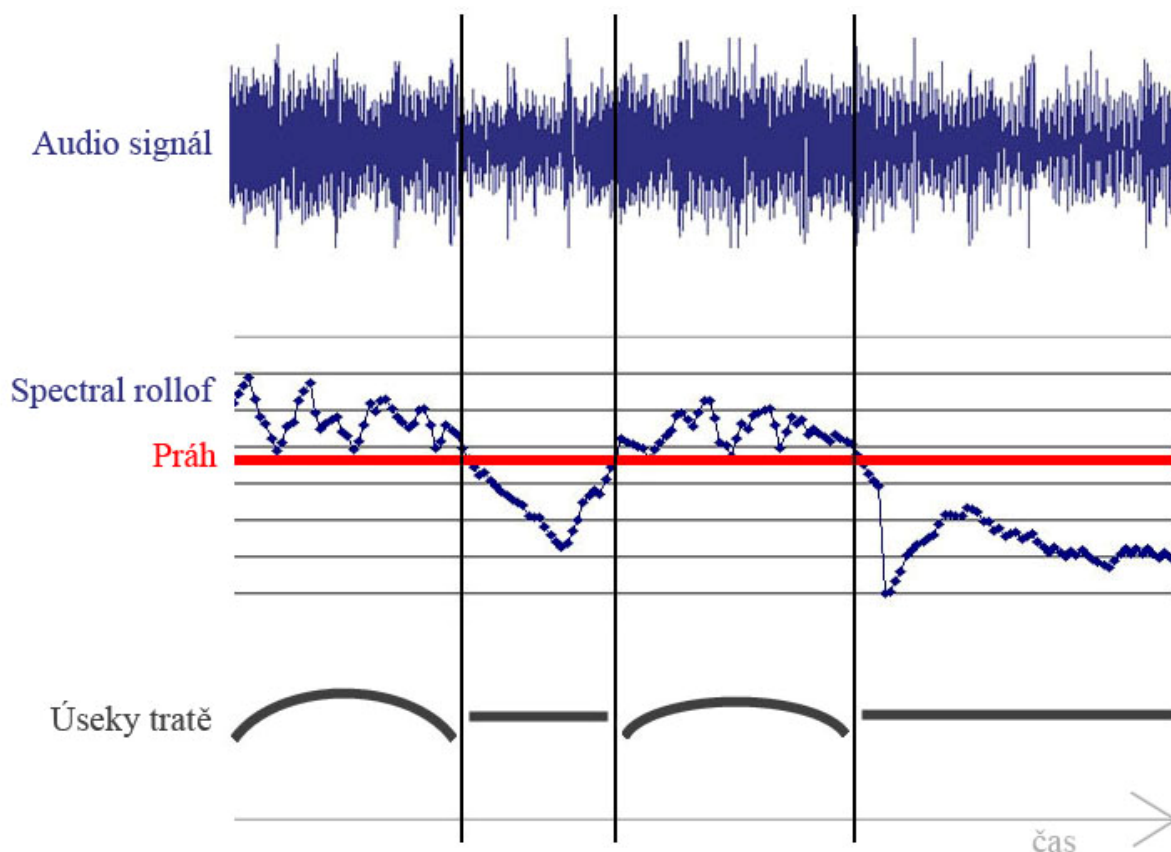
Je tedy potřeba navrhnout způsob generování tvaru tratě, a rozmístění ostatních prvků herního prostředí.

### 5.1 Generování tvaru tratě

Trat' můžeme zjednodušeně chápat jako spojitou linii ve 3d prostoru. Pro její generování tedy budeme potřebovat spojitou veličinu, vzorkovanou s dostatečnou frekvencí, aby mohla trat' svým tvarem zachycovat změny v hudbě. Nelze tedy použít výstupy z klasifikace žánrů ani ze shlukování, odtam dostáváme jednu hodnotu za 4 vteřiny, navíc je výstupem diskrétní veličina s minimálním rozsahem hodnot. Musíme tedy sáhnout po některé z vlastností, které získáváme při extrakci příznaků z hudby a dají se určit pro jediné okno signálu. Průběh jedné takové vlastnosti v čase tvoří dvourozměrnou veličinu, pro trat' však potřebujeme 3 rozměry. Proto použijeme 2 vlastnosti, nezávisle na sobě. Jedna vytvoří výškový profil tratě, může to být například *spectral centroid*. Jeho hodnota udává spektrální těžiště, je tedy spojena s vnímáním výšky a jasu celého zvuku. Proto je tato vlastnost vhodná pro určení výšky tratě. Druhou vlastnost potřebujeme vybrat pro generování zatáček. Mohla by to být opět jedna ze spektrálních vlastností, bude však nutné vyzkoušet různé vlastnosti na výsledné hře a zhodnotit jejich vhodnost. Ze získaných dvou profilů je potřeba složit trat' tak, aby nekolidovala sama se sebou. Toto složení tratě proběhne ve dvou fázích: Nejdříve se vygenerují pouze zatáčky – tj. 2D trat'. Ve druhé fázi se podle výškového profilu vytvoří kompletní 3d linie tratě.

## Zatáčky

Na vstupu generátoru tvaru tratě předpokládáme vzorky dvou vlastností, získaných při extrakci příznaků z hudby. V první fázi vezmeme pouze jednu z nich, podle ní budeme generovat zatáčky tratě. Nejprve potřebujeme identifikovat rovné a točivé úseky trati. Pro tento účel určíme práh, kterým oddělíme úseky s nízkou hodnotou sledované vlastnosti – tyto úseky budou tvořit rovné části tratě. Úseky s hodnotou nad prahem budou tvořit zatáčky, jejich rádius se určí podle velikosti součtu hodnot nad prahem. Vhodným prahem může být průměrná hodnota, tu můžeme případně posunout o nějakou konstantu podle potřeby: Pro získání méně točivé trati práh zvýšíme, pro více točivou trať práh snížíme. Vytvoření rovných a točivých úseků podle zvolené vlastnosti ukazuje obrázek 9. Na obrázku je zvukový signál písňě na čtyřvteřinovém úseku. Pod ním je vidět průběh hodnot vlastnosti, podle které se budou určovat zatáčky. Jako příklad je zvolena vlastnost *spectral rollof*, v praxi to může být i jiná vlastnost. Podle toho, zda jsou hodnoty vlastnosti nad červeně označeným prahem, nebo pod ním, se k danému časovému úseku generují obloukové nebo rovné úseky tratě – v obrázku naznačeny šedou barvou.

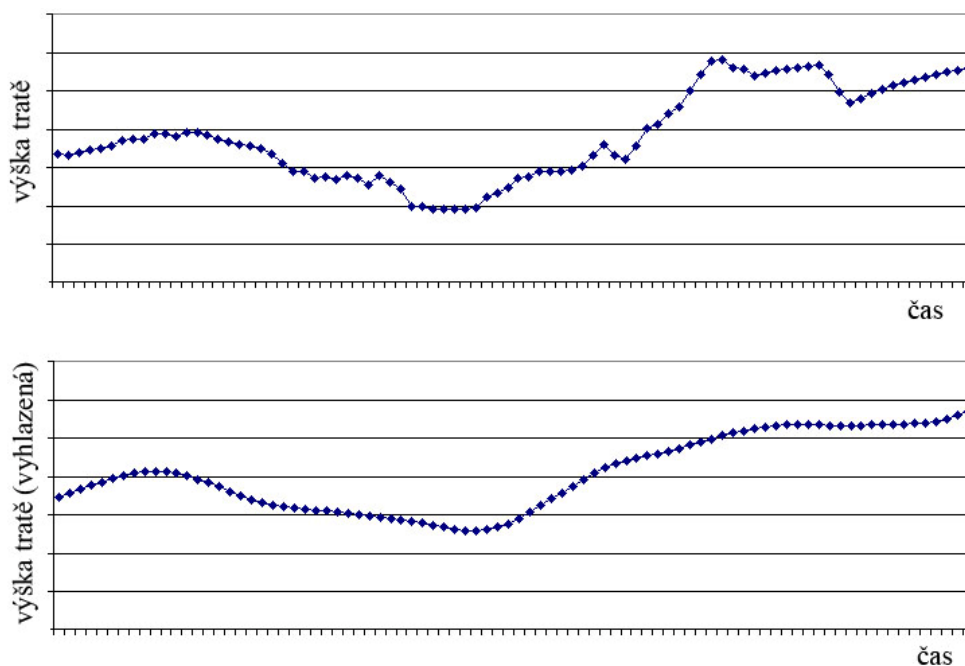


Obrázek 9 Tvorba rovných a točivých úseků tratě

Ve druhé fázi je potřeba vzniklé úseky spojit do souvislé tratě. Budeme tedy úsekům přidělovat pozice v prostoru, prozatím dvourozměrném. Rovný úsek je v této chvíli definován pouze svou délkou, točivý úsek délkou a poloměrem otáčení. V této fázi najdeme počáteční a koncový bod pro rovné úseky. Pro točivé úseky hledáme také počáteční a koncový bod, mimo to potřebujeme ještě střed otáčení – z něj vyplývá poloměr točení a směr zatáčky (vpravo nebo vlevo). Začneme s prvním úsekem, jeho počáteční bod umístíme na startovní pozici v prostoru (možné zvolit libovolně). Podle délky úseku a jeho případného točení spočteme polohu jeho koncového bodu. Na tuto pozici umístíme počáteční bod dalšího úseku, pro tento úsek opět určíme jeho koncový bod. Takto postupně navážeme na sebe všechny úseky. Musíme zkontrolovat, zda trať sama se sebou nekoliduje. Pravděpodobně se při prvním sestavení tratě kolize objeví, protože všechny zatáčky měly stejný směr zatáčení. Proto se náhodně zvoleným točivým úsekům obrátí jejich směr zatáčení a sestavení tratě se provede znovu od prvního úseku. Opakujeme změnu směru náhodných zatáček a sestavíme trať, dokud nedostaneme trať bez kolizí.

## Výškový profil

Pro určení výšky tratě není potřeba trať dělit na rovné a stoupavé nebo klesavé úseky, můžeme výšku plynule měnit v průběhu celé tratě podle vybrané vlastnosti. Ať zvolíme jakoukoli vlastnost, bude potřeba její průběh silně vyhladit, aby i změna výšky tratě byla hladká. K tomu můžeme použít průměrovací filtr, který každému vzorku přiřadí průměrnou hodnotu jeho okolí. Nastavením použité velikosti okolí ovlivníme, jak silně filtr vyhlazuje nerovnosti. Obrázek 10 ukazuje výškový profil tratě na dvoučteřinovém úseku před tímto vyhlazením a po něm, velikost okolí je 10 vzorků na obě strany.



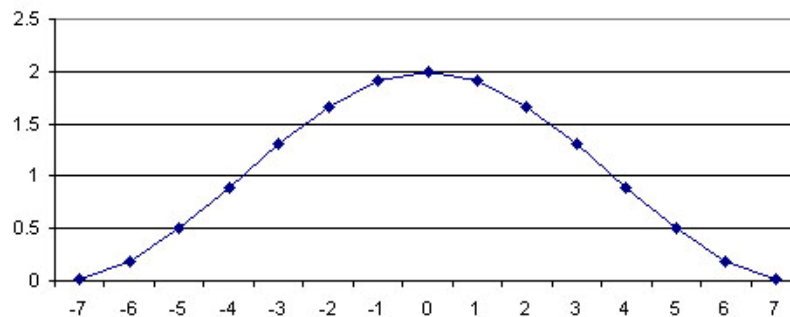
**Obrázek 10** Výškový profil tratě před a po vyhlazení průměrovacím filtrem



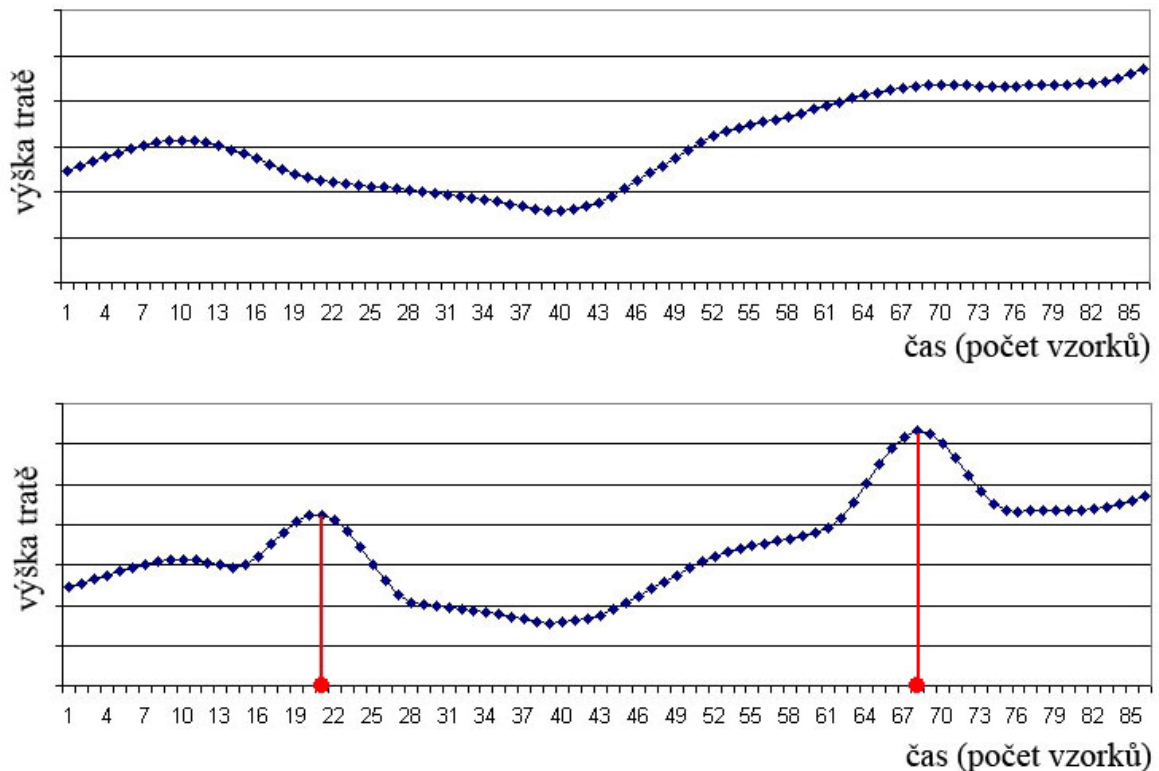
Do výškového profilu bychom mohli také nějakým způsobem promítnout rytmus hudby. Vlnění tratě do rytmu, které se objevuje ve hře Audiosurf, je velmi efektní. K výškovému profilu tedy přičteme „vlnku“ všude tam, kde byl detekován beat. Vlnku vytvoříme pomocí funkce  $\cos(x)+1$  na intervalu  $-\pi.. \pi$ . Abychom vlnku roztáhli do požadovaného času, spočítáme počet vzorků  $n$  odpovídající tomuto času. Argument  $x$  funkce  $\cos(x)+1$  určíme podle indexu vzorku  $i$  tak, jak definuje rovnice 18.

$$x = \frac{i}{n/2} \pi \quad (18)$$

Obrázek 11 ukazuje takovou vlnku, roztaženou na 15 vzorků. Na obrázku 12 je původní výškový profil tratě na dvouvteřinovém úseku, předpokládejme vzorkovací frekvenci 43Hz, úsek má tedy 86 vzorků. Pod ním je ten stejný úsek, kde byly detekovány beats v časech odpovídajících vzorkům 21 a 68 – označeny červeně. Na každou z těchto pozic byla přičtena k výškovému profilu vlnka, zarovnaná svým středem na dané pozice.



Obrázek 11 Vlnka



Obrázek 12 Výškový profil tratě, přičtení vlnek na pozice beatů

Z obrázku 12 můžeme usuzovat, že vlnky budou ve výškovém profilu příliš výrazné. Proto bude potřeba jejich hodnoty snížit – podělit je nějakou konstantou. Při testování výsledného programu by neměl být problém vhodnou konstantu nalézt.

## 5.2 Objekty ve hře, styly zobrazení

Aby měla hra smysl, je potřeba na trat' rozmístit objekty, které má hráč sbírat, nebo jim naopak uhýbat. Dále je vhodné rozmístit nějaké objekty kolem trati a vytvořit vizuální efekty. Zatímco generátor tvaru tratě potřeboval jako vstup pravidelně vzorkovaný signál, objekty na trati budou rozmístěny nepravidelně. Pro každý typ objektů musí tedy být určeny časy všech výskytů daného objektu. Podle těchto časů se objekty zarovnají ke trati.

### Bonusy

Objekty, které má hráč sbírat, můžeme rozmístit podle rytmu. Vzhledem k tomu, že se očekává kontakt rakety s těmito objekty, je vhodné aby ke kontaktům docházelo v časech detekovaných beatů. Sbírané objekty by měly mít různé barvy a sebrání objektu by mělo být doprovázeno probliknutím obrazovky stejnou barvou, jakou měl objekt. Pokud bude hráč objekty sbírat úspěšně, bude obrazovka blikat do rytmu.

### Překážky

Rozmístění překážek, kterým musí hráč uhýbat, není nutné přímo odvozovat z nějaké vlastnosti hudby. Ke kontaktu s raketou by většinou nemělo dojít, těžko tedy rozmístěním překážek vytvoříme nějaký efekt. Můžeme je rozmístit náhodně nebo do větších mezer mezi beaty, případně na místo druhého beatu, pokud jsou dva beaty velmi blízko u sebe.

### Objekty kolem tratě

Objekty kolem tratě mohou být rozmístěny opět podle rytmu. Protože známe sílu jednotlivých beatů (počet frekvenčních pásem, ze kterých byl daný beat vygenerován), můžeme objekty kolem tratě rozmístit pouze k silným beatům. Tak nebudou objekty kolem tratě přímo kopírovat rozmístění bonusů. Druhá možnost je vytvořit kruhový objekt, který tvoří kolem tratě něco jako bránu. Pro rozmístění takových objektů bychom mohli využít shlukování, které nám v písni našlo souvislé úseky jistého charakteru. Takovou bránu můžeme umístit tam, kde píseň přechází z jednoho shluku do druhého.

### Barvy

Shlukování bych chtěl využít také pro výběr barev ve hře. Celkovou barevnost hry silně ovlivňují bonusy, které hráč sbírá – podle barvy sebraného objektu totiž bliká celá obrazovka. Pro každý

nalezený shluk bude vytvořeno „barevné schéma“, to budou dvě nebo tři podobné barvy. Obarvení bonusů na tomto úseku písně se bude vybírat jen z barev aktuálního schématu. Při přechodu do jiného shluku se barevné schéma změní, tím se změní celková barevnost hry a bude tak vyjádřen odlišný charakter nalezených shluků.

### **Hudební žánry**

Podobně jako jsou využity shluky pro změnu barevnosti, můžeme využít detekci hudebních žánrů pro další změny vizualizace. Zde můžeme být konkrétnější, například podle detekovaného žánru vybrat obrázek na pozadí, nebo měnit tvar bonusů. Pro rockovou hudbu bychom mohli zobrazovat například elektrické kytary a pro klasiku housle, buď v obrázku na pozadí, nebo tvarem bonusů.

## 6 Implementace

Systém se skládá ze dvou hlavních částí. V první z nich se provádí analýza hudby, do které patří detekce rytmu, rozpoznání hudebních žánrů a shlukování. Druhá část využívá výsledků této analýzy a generuje podle nich herní prostředí. Implementaci obou těchto částí popisuje tato kapitola. Kompletní výsledný program byl vytvořen v prostředí Microsoft Visual Studio 2005.

### 6.1 Analýza hudby

Pro zpracování zvuku jsem použil rozhraní DirectX, konkrétně knihovny DirectSound. Ty jsou dokumentovány na msdn.com, včetně ukázkových příkladů. S jejich využitím jsem do aplikace integroval jednoduchý přehrávač, umožňující otevřít vybraný zvukový soubor, přehrát jeho obsah a přistupovat k datům ze zvukového bufferu. DirectSound umí pracovat pouze se soubory typu \*.wav, pro moje účely musí být zvuk uložen ve formátu PCM, se dvěma kanály a 16ti bity na vzorek.

První významnou částí hudební analýzy je detekce rytmu, která hledá všechny časy výskytu beatů. Druhá část provádí klasifikaci - extrahuje příznaky, detekuje hudební žánry a provádí shlukování. V této části bylo třeba určit, na jak dlouhých úsecích písně se bude klasifikace provádět. Jak bylo řečeno v kapitole 4.2, úsek by měl být dlouhý minimálně 3 vteřiny, aby bylo možné rozpoznat hudební žánr nebo charakter písně. Na druhou stranu je třeba, aby byl úsek co nejkratší, protože chceme s co největší přesností určit místa, kde se charakter písně mění. Jako dobrý kompromis mezi těmito protichůdnými požadavky se mi jeví délka úseku 4 vteřiny, tu jsem nakonec použil. Při klasifikaci hudebních žánrů na čtyřvteřinových úsecích jsem dosáhl uspokojivé úspěšnosti (detaily v kapitole 7.1) a maximální teoretická nepřesnost určení místa změny jsou přijatelné 2 vteřiny.

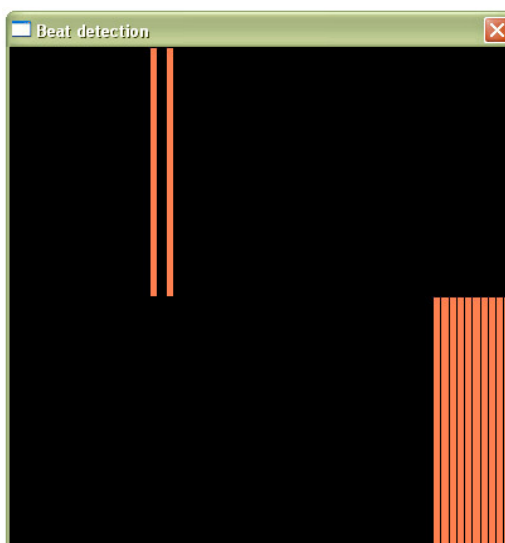
### Detekce rytmu

V této detekci jde o zjištění přesných časů všech beatů v celé písni. Pro tento účel se použije metoda sledování změn energie, jejíž kompletní funkčnost je popsána v kapitole 4.1. Tohoto návrhu jsem se držel při implementaci. Postupně jsem provedl všechny navržené optimalizace, s každou z nich se kvalita systému opravdu zvýšila. Výsledný systém podává uspokojivé výsledky, které jsou pro vizualizaci ve hře použitelné.

Tato metoda přistupuje k bufferu se zvukovým signálem a čte postupně všechny vzorky. Signál je ve formátu PCM, na vzorkovací frekvenci 44100Hz. Jednoho okna pro zpracování signálu má délku 1024 vzorků, z nich se počítá frekvenční spektrum, které se redukuje postupem popsaným v kapitole 4.1 na 64 frekvenčních pásem. Sleduje se průběh energie v těchto pásmech, určuje se

adaptivní práh a podle něj se detekují beaty. Pro klasifikační část budeme potřebovat čtyřvteřinové úseky písně, proto si každých 172 oken uložíme jako samostatný úsek pro budoucí klasifikaci. Ukládáme spektrum i samotný signál, klasifikace využívá obojího.

Pro testování této detekce rytmu jsem si vytvořil jednoduché grafické rozhraní, zobrazené na obrázku 9. Program přehraje zvolenou píseň a zobrazuje přitom detekované beaty. Při každém detekovaném beatu se zobrazí červený pruh, jehož horizontální pozice udává frekvenční pásmo, které beat vygenerovalo: vlevo jsou nízké frekvence, směrem doprava vyšší. Vertikální pozice udává kanál, na kterém se beat objevil: horní polovina okna pro pravý kanál, dolní polovina pro levý. Detekce je dostatečně rychlá pro zobrazování v reálném čase. Detekované beaty tedy vidíme synchronně s přehrávaným zvukem, tak je možné udělat si představu o tom, na kolik je detekce úspěšná.



**Obrázek 13** Detekce rytmu – beaty detekovány ve vysokých frekvencích levého kanálu a nižších středních frekvencích pravého kanálu

## Klasifikace hudebních žánrů

### Extrakce příznaků

Výše popsaná metoda pro detekci rytmu načetla celý zvukový buffer a uložila jednotlivé čtyřvteřinové úseky. V každém úseku máme teď k dispozici 172 oken zvukového signálu i frekvenčního spektra. Můžeme tedy pro úsek spočítat jeho příznakový vektor, tak jak je to popsáno v kapitole 4.2.

Při výpočtu „musical surface features“ spočítáme všechny příznaky (s výjimkou *low energy*) pro každé ze 172 oken (tedy pro celý úsek). Ze získaných hodnot určíme střední hodnotu a standardní odchylku, jejich hodnoty použijeme do příznakového vektoru. *Low energy* určíme jako procentuální část všech oken úseku, která mají nižší energii než je průměrná v úseku. Pro úsek máme tedy těchto 9 příznaků typu „musical surface features“:

```
centroid_avg, centroid_std
rollof_avg, rollof_std
flux_avg, flux_std
zcr_avg, zcr_std
lowEnergy
```

Přípona `_avg` označuje střední hodnotu, přípona `_std` značí standardní odchylku.

Pro výpočet rytmických vlastností daného úseku potřebujeme frekvenční spektrum, které opět máme uloženo ve 172 oknech. Použijeme autokorelační funkci, jako vstup slouží průběh energie jednoho frekvenčního pásma přes všechna okna. Autokorelační koeficienty spočítáme pro všechny frekvenční pásma, špičky ukládáme do histogramu. Z něj pak určíme 6 rytmických příznaků:

```
relative1
relative2
rate
bps1
bps2
beatSum
```

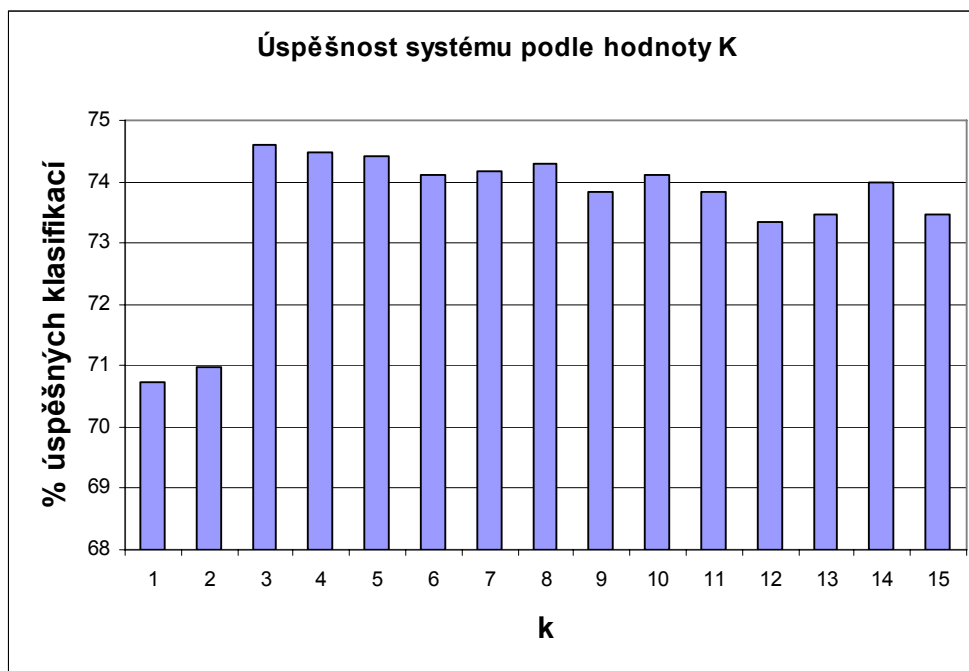
Pro čtyřvteřinový úsek písně máme tedy příznakový vektor o patnácti příznacích. Jedná se o příznaky různých datových typů, navíc se hodnoty různých příznaků pohybují v různých a často velmi odlišných rozsazích. Klasifikátor KNN určuje příslušnost ke třídám na základě vzdálenosti příznakových vektorů, proto by měly příznaky nabývat hodnot ze stejného rozsahu, aby měly stejnou váhu při určení vektorové vzdálenosti. Provedl jsem tedy normalizaci příznaků, která spočívala v dělení hodnoty každého příznaku sumou všech hodnot stejného příznaku na všech dostupných datech.

## **Klasifikátor**

Pro klasifikaci hudebních žánrů potřebujeme získat příznakové vektory z mnoha úseků vybraných z mnoha různých písní. Ze všech získaných příznakových vektorů, ke kterým známe správné přiřazení k žánru, se vytvoří modely tříd (žánrů), podle kterých se budou klasifikovat nová data. Tyto modely potřebujeme nějak uložit na disk, aby je měl program při příštím spuštění k dispozici. Vzhledem k tomu, že časově nejnáročnější částí celé této klasifikace je extrakce příznaků, jde nám hlavně o to, aby se příznaky všech trénovacích dat nepočítaly pokaždé znova. Vytvoření modelů tříd je časově zanedbatelné, zvláště v tomto případě, kdy používám klasifikátor KNN. Ten totiž pouze uchovává pozici všech příznakových vektorů v jejich vektorovém prostoru, nezabývá se tedy výpočtem žádného složitějšího modelu a jeho vytvoření je proto velmi rychlé. Proto na disk ukládám pouze spočtené příznakové vektory ve formě textového souboru. Při každém spuštění programu se tyto vektory načtou ze souboru a vytvoří se z nich nový klasifikátor. Případně je možné vynutit nový

výpočet příznakových vektorů, pokud jsou dostupná nová trénovací data nebo se změnil způsob výpočtu příznaků.

Funkčnost klasifikátoru KNN je ovlivněna především parametrem  $k$  – tj. kolik nejbližších sousedů se analyzuje pro určení třídy neznámého vektoru. Optimální hodnota se dá určit experimentálně. V mém případě nejlépe funguje hodnota 3, provedené testy s různými hodnotami ukazují obrázek 10. Pro vytvoření klasifikátoru jsem použil implementaci z knihovny OpenCV.



Obrázek 14 Optimalizace hodnoty  $k$

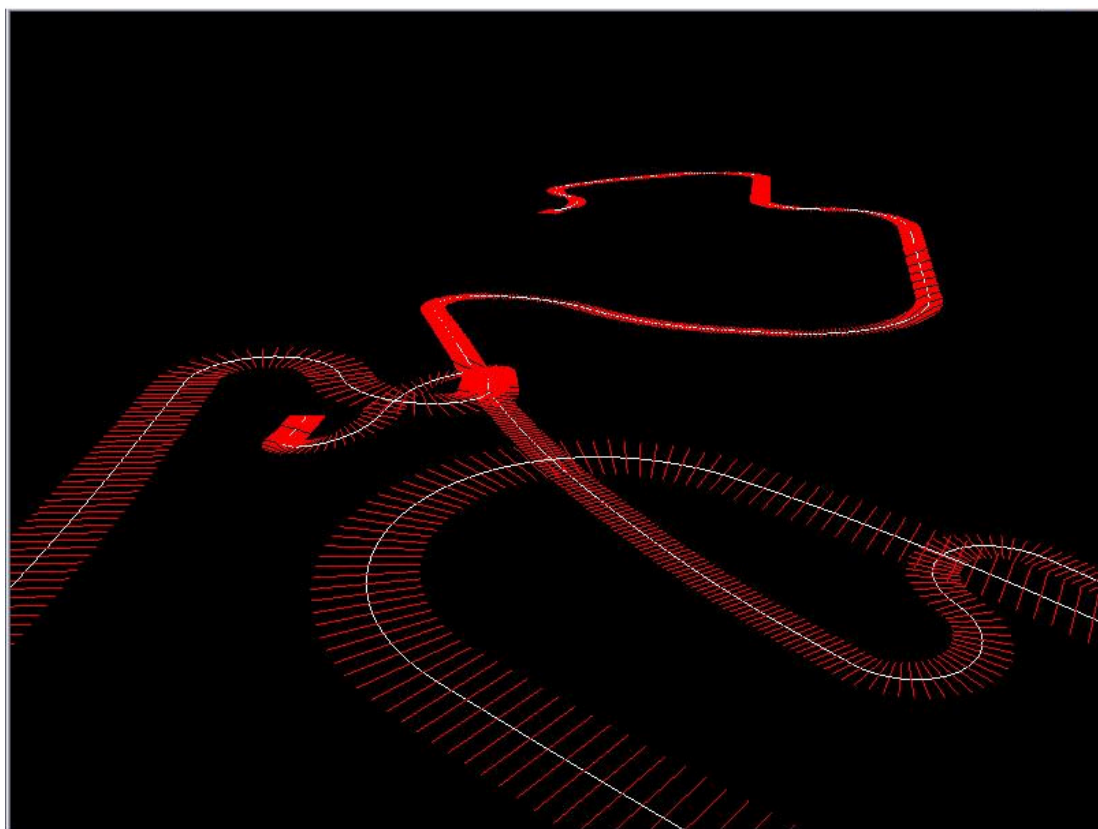
## Shlukování

Shlukování použijeme pro nalezení segmentů písně se odlišným charakterem. Pomocí metod strojového učení bez učitele najdeme v písni shluky podobných úseků, každý shluk bude vhodné ve výsledné hře vizualizovat odlišným stylem. Použil jsem stejné příznaky, které se osvědčily při klasifikaci hudebních žánrů. Jako shlukovací metodu jsem použil KMeans z knihovny OpenCV. Počet shluků, které hledáme, jsem zvolil na 4. Při klasifikaci hudebních žánrů systém dobře fungoval se třemi třídami, proto by mělo podobné číslo fungovat i při shlukování. V tomto konkrétním případě není počáteční volba počtu shluků příliš důležitá: Jedná se o učení bez učitele a nevíme tedy, jakou charakteristikou se daný shluk vyznačuje. Jde nám spíše o zachycení míst, ve kterých píseň přechází z jednoho úseku do druhého. V těchto místech budeme chtít změnit aktuální způsob zobrazování na jiný styl, který musíme vybrat náhodně. Použitý počet shluků tedy ovlivní pouze četnost změn stylu. Při testování výsledné hry bude vhodné upravit tento počet podle potřeby. Pokud se nám bude zdát, že se zobrazovací styl mění příliš často, snížíme počet shluků, naopak při nedostatečném počtu změn zvýšíme počet shluků.

## 6.2 Generátor herního prostředí

Vstupem pro generátor tratě jsou dvě vlastnosti, extrahované z hudby s konstantní vzorkovací frekvencí. Původní audiosignál má vzorkovací frekvenci 44100Hz, jedno okno pro zpracování signálu má 1024 vzorků, proto na jednu vteřinu vychází 43,066 oken. Z každého okna můžeme získat jeden vzorek požadovaných vlastností. Dostáváme tedy signály vzorkované na frekvenci 43,066Hz. To je problém, pro generování tratě bychom potřebovali celočíselnou vzorkovací frekvenci, aby bylo snazší synchronizovat průběh tratě s časem hrající hudby. Nejprve tedy provedeme převzorkování získaných signálů na frekvenci 43Hz. Teprve potom se podle nich generuje tvar tratě tak, jak je popsáno v kapitole 5.1. Pro vytvoření zatáček používám *spectral rollof*, jako práh pro oddělení rovných a točivých úseků používám průměrnou hodnotu. Výškový profil tratě tvořím z hodnot vlastnosti *spectral centroid*, průběh hodnot vyhladím průměrovacím filtrem a přidám „vlnky“ na místa detekovaných beatů. Vznikne tak tvar trati, reprezentovaný spojitou linií ve 3d prostoru.

Vzniklou linii postupně procházíme, posunujeme se po ní s malým konstantním krokem a na každém kroku vytvoříme bod, ze kterého vychází přímka kolmá na směr průběhu tratě. Na rovných úsecích je tato přímka kolmá na linii tohoto rovného úseku, na točivých úsecích je kolmá na tečnu oblouku. Zároveň je přímka kolmá na vertikální osu (y). Tyto přímky pomohou vytvořit základní kostru pro trať. Tuto kostru zobrazuje obrázek 15. Obrázek pochází z raného stadia vývoje hry, kde se trať ještě mohla křížit.



Obrázek 15 Kostra tratě



Na tuto kostru je snadné natáhnout polygony, které vytvoří povrch tratě. Další polygony byly přidány, aby trať dostala objem. Po otexturování, zobrazení skyboxu kolem tratě a zapnutí mlhy vypadá výsledná trať tak, jak ukazuje obrázek 16.



**Obrázek 16** Trať s texturami a mlhou, skybox

Další prvky herního prostředí byly rozmístěny podle návrhu předloženého v kapitole 5.2. Bonusy jsou umístěny do časů všech beatů. Objekty okolo tratě mají podobu brány a jsou umístěné v časech beatů, které vygenerovalo více než 7 frekvenčních pásem zároveň. Barvy bonusů jsou vybírány z aktuálního barevného schématu. Toto schéma se určí na základě výstupu shlukování. Odtam dostáváme 4 shluky, proto existují 4 barevná schémata, která podle shluků přepínáme. Každé schéma se skládá ze dvou barev, které vidíme v tabulce 1. Dokud se nedostane hra do úseku s jiným shlukem, střídají se pravidelně barvy z aktuálního schématu.

Schéma 1	Schéma 2	Schéma 3	Schéma 4
Yellow	Cyan	Red	Blue
Light Yellow	Green	Orange	Purple

**Tabulka 1** Barevná schémata

Grafika hry je vytvořena v OpenGL, s využitím frameworku ÜberLame. Tento framework mi poskytnul Bc. Lukáš Polok a pomohl mi v něm vytvořit zobrazovací systém pro tuto hru. Dále pro mě vytvořil modely rakety a objektů kolem tratě ve 3d Studiu MAX a zajistil jejich import do hry.

# 7 Výsledky

Výsledky systému pro analýzu hudby se dají objektivně zhodnotit pouze v části pro rozpoznávání hudebních žánrů. Úspěšnost detekce rytmu se dá zhodnotit vizuálně v grafickém rozhraní popsaném v kapitole 6.1, a je dobře vidět i z výsledné hry. Výsledky systému pro rozpoznávání hudebních žánrů se dají zhodnotit pomocí cross-validace systému, celé testování této části popíšu v podkapitole 7.1. Úspěšnost rozpoznávání žánrů ukazuje kvalitu extrakce příznaků. Ta je zásadní pro úspěšné shlukování. Pro shlukování jsem tedy použil stejné příznaky, které se osvědčily při klasifikaci žánrů. Samotné výsledky shlukování je možné zhodnotit na změnách barev ve výsledné hře.

Výsledky druhé části – generování herního prostředí – se opět dají hodnotit pouze subjektivně na výsledné hře. V podkapitole 7.2 popisují, jak vytvořená hra vypadá a funguje.

## 7.1 Testy a výsledky klasifikace hudebních žánrů

Vytvořený systém pro detekci hudebních žánrů byl natrénován pro rozpoznávání tří hudebních stylů. Ty byly zvoleny tak, aby byly i pro člověka snadno rozpoznatelné – s výraznou, a pro každý žánr silně odlišnou charakteristikou. Jedná se rockovou, vážnou (klasickou) a taneční hudbu. Použitá data (především jejich kvantita) jsou uvedena v tabulce 2.

	Počet písní	Celkový hrací čas	Výrazně zastoupeni
Rock	62	3,7 hodin	AC/DC, Led Zeppelin, Rolling Stones...
Classic	28	1,6 hodin	Mozart, Tschaikovsky, Vivaldi...
Dance	63	3,9 hodin	Shaun Baker, Mr. Vee...

Tabulka 2 Použitá data

Uvedení umělci jsou zvoleni pouze jako typičtí zástupci, kompletní seznam všech použitých písní je uveden v příloze 2. Pro natrénování a otestování systému se z každé písně vezme několik čtyřvteřinových úseků jako několik samostatných vzorků dat. Není nutné použít celou píseň, dostali bychom velké množství stejných nebo velmi podobných vzorků dat. Proto z každé písně беру jeden úsek z každých dvaceti vteřin. Z celkových 9,2 hodin hudby se tak ve skutečnosti použilo přibližně 1,84 hodin.

Kvalita systému byla vyhodnocena pomocí cross-validace: Všechna data byla rozdělena na pět částí přibližně stejné velikosti, jedna část byla použita jako testovací data, zbylé čtyři jako trénovací. Trénování a testování se opakovalo tak, aby se všech pět částí vystřídal na pozici testovacích dat. Pokaždé se určilo procento úspěšnosti systému, nakonec se vypočítala průměrná úspěšnost.

Nejdříve jsem oddělil příznaky typu „musical surface features“ od rytmických příznaků a otestoval jsem, jakou úspěšnost má systém, využívající pouze jednu z těchto skupin příznaků. V obou případech systém dosáhl úspěšnosti, která přibližně dvakrát převyšuje hranici náhodnosti (pro 3 třídy 33,33%):

- Musical surface features: úspěšnost 71,33 %
- Rytmičné vlastnosti: úspěšnost 64,67 %

Oba tyto typy příznaků jsou tedy pro klasifikaci hudebních žánrů přínosné. Výsledný systém, využívající všechny popsané příznaky, dosaženou úspěšnost dále zvýšil. Drobné zlepšení přinesla normalizace příznaků.

- Výsledný systém před normalizací: úspěšnost 76,67 %
- Výsledný systém po normalizaci: úspěšnost 77,7 %

Matice „confusion matrix“, získaná při cross-validaci výsledného systému, ukazuje například fakt, že nejvíce se při klasifikaci pletla taneční hudba s rockovou. Přitom taneční a klasická hudba byly mezi sebou zaměněny jen v mizivém počtu případů. To je očekávaný výsledek – taneční a rocková hudba mají mnoho společného, v některých pasážích by si posluchač nemusel být jistý, který z těchto žánrů právě hraje. Oproti tomu záměna taneční hudby s klasikou je krajně nepravděpodobná, odlišnosti mezi těmito žánry jsou příliš výrazné i na malém úseku písničky. Výslednou matici „confusion matrix“ zachycuje tabulka 3.

		Skutečný žánr		
		Rock	Classic	Dance
Výstup klasifikace	Rock	601	80	144
	Classic	65	182	22
	Dance	50	7	499

**Tabulka 3** Confusion matrix

## 7.2 Generování hry

Výsledná hra slouží k posouzení kvality provedené analýzy hudby. V tomto ohledu hra plní svůj účel, můžeme říci, že analýza hudby byla úspěšná a získané informace jsou použitelné pro generování her. Tato hra však bohužel zatím trpí technickými nedostatky, které zde musím popsat.

První z nich je spojen s implementací hudební analýzy. Nepříjemné je již samotné omezení vstupního souboru na formát wav. Ten je nevhodný, protože uživatel musí svou hudbu do tohoto formátu konvertovat. Hlavní problém spočívá však v metodě, kterou jsem použil pro otevření hudebního souboru. Soubor otevírám celý naráz a čtu celý zvukový buffer. Proto záleží na použitém

hardwaru, jak velký zvukový soubor mi dovolí otevřít. Při vývoji hry jsem si tohoto omezení bohužel nevšiml, sdílená zvuková karta na mém počítači pravděpodobně mohla alokovat dostatek paměti a mohl jsem tak otevřít písničku téměř libovolné velikosti. Při závěrečných testech na jiném PC se projevilo, že s jinou (starší) zvukovou kartou je silně omezena velikost souboru, který je možné otevřít. V mém případě musela mít vstupní píseň omezena délku pod jednu minutu. V takovém případě je potřeba testovat hru pouze na krátkých úsecích písni – na nich bohužel není příliš dobře možné zhodnotit výsledky segmentace písni. Z pohledu posluchače, který zná píseň celou, by totiž minutový úsek logicky tvořil jeden nebo maximálně dva segmenty, použitá segmentace bude však hledat více shluků. Konkrétní omezení délky vstupní písni je potřeba vyzkoušet na každém HW individuálně. Řešením tohoto problému by bylo použít streamování pro načítání hudebních souborů. To by však vyžadovalo větší zásah do programu a na to mi nezbyl potřebný čas.

Druhým problémem je celkový stav vývoje hry. Do té se prozatím nepodařilo zahrnout všechny plánované vlastnosti. Ve hře se neobjevují překážky, a není nijak zobrazen detekovaný hudební žánr. Pro posouzení hudební analýzy to však není velký problém, rozpoznávání hudebních žánrů totiž bylo zhodnoceno cross-validací a rozmístění překážek nemůže příliš pomoci vyjádření vlastností hudby.

Hru jako celek je tedy nutné brát pouze jako prototyp, určený k vizualizaci získaných dat z hudby. Nedokázal jsem hru dotáhnout do podoby, ve které by mohla být předložena běžnému hráči. K tomu by bylo potřeba více času, než bylo vyhrazeno pro tuto diplomovou práci. I v tom stavu, ve kterém se hra nyní nachází, je však hraní zajímavé. Záběr ze hry je na obrázku 17.



Obrázek 17 Záběr ze hry

## 8 Závěr

Tématem této práce je generování herního prostředí na základě hudby. Zabýval jsem se především analýzou hudby, kde cílem bylo nalezení takových vlastností v hudbě, které se dají využít pro generování hry. Navrhnul jsem systém, který poskytuje několik různých typů užitečných dat pro generování. Analýza hudby je v mém systému zaměřena především na detekci rytmu, rozpoznávání hudebních žánrů a segmentaci písně. Dále jsem navrhnul systém pro generování hry na základě informací získaných z hudby. Při návrhu obou systémů jsem vycházel z vlastností existujících hudebních her, dále bylo nutné prostudovat zpracování audio signálů, lidské vnímání hudby a statistickou klasifikaci.

Navržený systém pro analýzu hudby jsem kompletně implementoval, objektivně se dají vyhodnotit pouze výsledky rozpoznávání hudby. Při cross-validaci rozpoznávače na čtyřvteřinových úsecích písní jsem dosáhl úspěšnosti 77,7 %. To je úspěšný výsledek vzhledem k faktu, že ani člověk nerozeznává hudební žánr se stoprocentní úspěšností [9]. Dále jsem implementoval hru, generující své prostředí podle výsledků provedené analýzy hudby. Tato hra má sloužit pro zhodnocení, nakolik jsou získaná data pro generování hry použitelná a vhodná. Výsledná hra bohužel trpí několika technickými problémy, které jsem v daném čase nedokázal odstranit. Jsou to však pouze implementační nedostatky, které by při dalším vývoji byly jistě odstraněny. Při akceptování těchto nedostatků se ukazuje, že návrh základní koncepce hry je v pořádku a provedená analýza hudby poskytuje vhodná data.

Při pokračování na této práci by prvním krokem bylo odstranění výše problémů, o kterých píšu v kapitole 7.2. Dále by bylo potřeba vytvořit lepší uživatelské rozhraní pro výslednou hru, především vytvořit menu, ve kterém si uživatel vybere písničku pro vytvoření tratě. Největší prostor pro zlepšení mé hudební analýzy vidím v extrakci příznaků. Pro zjištění rytmických příznaků existuje mnoho různých metod, experimentování s nimi by mohlo přinést dobré výsledky. Ostatní příznaky by také mohly být doplněny o úplně novou kategorii – například MFCC, které se používají ve zpracování řeči. Dále by analýza hudby mohla být doplněna o nové vlastnosti, například rozpoznávání „nálady“, které vyhodnotí, jestli píseň zní vesele nebo smutně. Vzhledem k netypickému tématu této práce by se dalo najít mnoho možností dalšího vývoje v této oblasti.

Hodnotu své práce vidím především v části zabývající se analýzou hudby. Přestože existují komerční hry, generující své prostředí z hudby, jejich analýza hudby nezahrnuje tolik aspektů, jako moje. Typicky se tyto hry omezují na pouze na rytmus. Přínos mé práce je tedy v zohlednění dalších informací obsažených v hudbě, které jsou pro posluchače a hráče relevantní. Metody, které jsem použil pro analýzu hudby, vychází z metod běžně používaných při zpracování signálů a především ve zpracování řeči. Při analýze hudby je však potřeba tyto metody použít netradičním způsobem, příkladem může být využití autokorelační funkce pro detekci rytmu. Zatímco metody zpracování řeči

byly a jsou zkoumány a popisovány v mnoha pracích, zpracování hudby je stále málo prozkoumaná oblast. Má práce byla tedy ztížena nedostatkem dostupných studijních materiálů, moje výsledky jsou však o to zajímavější. Výsledky jiných prací zabývajících se analýzou hudby jsou vidět pouze na některých hudebních hrách (Vib-Ribbon, Audiosurf), které jsou však z pohledu komerčního úspěchu v zanedbatelné menšině oproti hudebním hrám bez takové analýzy (Guitar Hero). Celková současná popularita hudebních her však dokládá aktuálnost mé práce.

# Literatura

- [1] Wikipedia: Psychoacoustics  
Dokument dostupný na URL: <http://en.wikipedia.org/wiki/Psychoacoustics> (květen 2009)
- [2] Dorrell, P.: What is music?, 2004-2005, kapitola 8  
Dokument dostupný na URL: <http://whatismusic.info/download.html> (květen 2009)
- [3] Harris, T.: How Hearing Works, 30. března 2001, HowStuffWorks.com.  
Dokument dostupný na URL: <http://health.howstuffworks.com/hearing.htm> (květen 2009)
- [4] Krčmová, M., Batušek, R., Gaura, P.: Fonetika a fonologie, kapitola 6.4  
Dokument dostupný na URL:  
<http://is.muni.cz/elportal/estud/ff/js08/fonetika/ucebnice/ch06s04.html> (květen 2009)
- [5] Centrum protihlukové ekologie, Petýrkova 1997, 14800 Praha 4: Fysiologická akustika  
Dokument dostupný na URL: <http://www.cmail.cz/cpe/clanky/fysiolog/fysiolog.htm#3>  
(květen 2009)
- [6] Novák, J.: Akustické vlnění, 27.listopadu.2006  
Dokument dostupný na URL: <http://webfyzika.fsv.cvut.cz/PDF/prednasky/akustika.pdf>  
(květen 2009)
- [7] Bourke, P.: Discrete Fourier Transform, Fast Fourier Transform, červen 1993  
Dokument dostupný na URL: <http://local.wasp.uwa.edu.au/~pbourke/miscellaneous/dft/>  
(květen 2009)
- [8] Černocký, J.: Zpracování řečových signálů - studijní opora, FIT VUT Brno, 6.prosince 2006
- [9] Kosina, K.: Music Genre Recognition [diplomová práce], Medientechnik und Design in Hagenberg, červen 2002  
Dokument dostupný na URL: <http://kyrah.net/mugrat/mugrat.pdf> (květen 2009)
- [10] Wikipedia: Music video game  
Dokument dostupný na URL: [http://en.wikipedia.org/wiki/Music\\_video\\_game](http://en.wikipedia.org/wiki/Music_video_game) (květen 2009)
- [11] <http://multiplayerblog.mtv.com/2008/09/15/gh-money-for-aerosmith/> (květen 2009)
- [12] Patin, F.: Beat Detection Algorithms, 2003  
Dokument dostupný na URL: <http://www.gamedev.net/reference/articles/article1952.asp>  
(květen 2009)
- [13] Tzanetakis, G., Essl, G., Cook, P.: Automatic Musical Genre Classification Of Audio Signals  
Dokument dostupný na URL: <http://ismir2001.ismir.net/pdf/tzanetakis.pdf> (květen 2009)
- [14] OpenCV Reference, Machina Learning Reference, K Nearest Neighbors
- [15] Wolfe, J.: What is a decibel?, 1998  
Dokument dostupný na URL: <http://www.phys.unsw.edu.au/jw/dBNoFlash.html>  
(květen 2009)



[16] Obrázek převzatý z webu,

URL obrázku: <http://goodbadandugly2.files.wordpress.com/2009/01/guitar-hero-5.jpg>

(květen 2009)

[17] Obrázek převzatý z webu,

URL obrázku:

[http://www.vib-ribbon.com/vibhtml/images/screens\\_big/vib\\_ribbon\\_screen\\_3b.jpg](http://www.vib-ribbon.com/vibhtml/images/screens_big/vib_ribbon_screen_3b.jpg)

(květen 2009)

[18] Obrázek převzatý z webu,

URL obrázku: <http://bonusweb.idnes.cz/pc/recenze/audiosurf-nevidane-hudebni-zavody-d51->

[/fotka.PRQ21584a\\_audiosurf\\_07.jpg.idn](http://bonusweb.idnes.cz/pc/recenze/audiosurf-nevidane-hudebni-zavody-d51-/fotka.PRQ21584a_audiosurf_07.jpg.idn)

(květen 2009)

# Seznam příloh

Příloha 1. Návod k použití vytvořeného softwaru

Příloha 2. Seznam použitých písniček

Příloha 1. CD se zdrojovým kódem programu, jeho binární verzí a tímto textem

# Příloha 1

## Návod k použití vytvořeného softwaru

### Formát vstupní hudby

Jako vstupní data pro vytvořenou hru slouží soubor typu .wav, musí být uložen ve formátu PCM, 44100Hz, 16bitů, stereo. Pro zajištění správného uložení wav souboru doporučuji například utilitu SoX (<http://sox.sourceforge.net/>).

### Omezení délky souborů

Musím upozornit na omezení délky vstupního souboru – to záleží na hardwaru, na kterém je program spuštěn. Při vývoji hry jsem si tohoto omezení bohužel nevšiml, sdílená zvuková karta na mém počítači pravděpodobně mohla alokovat dostatek paměti a mohl jsem tak otevřít písničku téměř libovolné velikosti. Při závěrečných testech na jiném PC se projevilo, že s jinou (starší) zvukovou kartou je silně omezena velikost souboru, který je možné otevřít. V mém případě musela mít vstupní píseň omezena délku pod jednu minutu. V takovém případě je potřeba testovat hru pouze na krátkých úsecích písní – na nich bohužel není příliš dobře možné zhodnotit výsledky segmentace písně. Z pohledu posluchače, který zná píseň celou, by totiž minutový úsek logicky tvořil jeden nebo maximálně dva segmenty, použitá segmentace bude však hledat více shluků. Konkrétní omezení délky vstupní písně je potřeba vyzkoušet na každém HW individuálně.

### Výběr vstupní písně

Program neobsahuje žádné menu, kde by si uživatel mohl vybrat písničku pro vygenerování tratě. Před spuštěním programu je proto nutné soubor se zvolenou písní umístit do složky /test. Důležité je, aby před spuštěním programu v této složce byl vždy jen jeden soubor! Pak je možné program spustit.

### Spuštění programu

Program se spouští souborem Music\_Game.exe. Po spuštění programu načte zvolenou píseň a generuje podle ní trať. Po dokončené generování se otevře okno s hrou, která ihned začíná. Hra se ovládá myší.

# Příloha 2

## Seznam použitých písniček

Zde uvádím seznam všech písniček, použitých pro natrénování a cross-validaci systému pro rozpoznávání hudebních žánrů. Uvádím zvlášť seznamy pro každý z použitých hudebních žánrů. U některých použitých písní neznám celý název nebo autora, proto uvádím pouze názvy souborů, ve kterých mám písně uložené. Z většiny těchto názvů je autor i píseň zřejmá.

Rock	
acdc_ hell ain't a bad place.wav	Queen_13 Now I'm Here.wav
acdc_ high voltage.wav	Queen_21 Hammer To Fall.wav
acdc_ rocker.wav	Rolling Stones_01 Rough Justice.wav
acdc_Back In Black.wav	Rolling Stones_06 Happy.wav
acdc_Hells Bells.wav	Rolling Stones_08 You Got Me Rocking.wav
acdc_problem child.wav	Rolling Stones_10 Oh No Not You Again.wav
acdc_Shoot To Thrill.wav	Rolling Stones_1-01 Jumpin' Jack Flash.wav
acdc_What Do You Do For Money Honey.wav	Rolling Stones_1-03 She Was Hot.wav
acdc_whole lotta rosie.wav	Rolling Stones_1-11 Tumbling Dice.wav
Bad Company_ Too Bad.wav	Rolling Stones_18 honky tonk women.wav
Brian May_Love token.wav	Rolling Stones_19 It's Only Rock'n'Roll.wav
Deep Purple_03 Demon's Eye.wav	Rolling Stones_2-04 Start Me Up.wav
Deep Purple_Into The Fire.wav	Rolling Stones_2-05 Brown Sugar.wav
Deep Purple_Speed King.wav	Rolling Stones_2-06 Satisfaction.wav
Jethro Tull_01 Spiral.wav	Rolling Stones_2-07 Paint It Black.wav
Jethro Tull_04 Hunting Girl.wav	Rolling Stones_2-09 I'm Free.wav
Jethro Tull_08 Hymn 43.wav	U2_01 In The Name Of Love.wav
Joe Cocker_Bathroom window.wav	U2_01-Vertigo.wav
Led Zeppelin Communication Breakdown.wav	U2_02 AudioTrack 02.wav
Led Zeppelin_01 Good Times Bad Times.wav	U2_07 Where The Streets Have No Name.wav
Led Zeppelin_01 Immigrant Song.wav	U2_Desire.wav
Led Zeppelin_01 Whole Lotta Love.wav	U2_1 Will Follow.wav
Led Zeppelin_05 Heartbreaker.wav	
Led Zeppelin_06 Living Loving Maid.wav	
Led Zeppelin_07 Trampled Under Foot.wav	
Led Zeppelin_11 Black Dog.wav	
Led Zeppelin-The Song Remains the Same.wav	
Paul McCartney_13 Nod Your Head.wav	
Pink Floyd_02 What Do You Want from Me.wav	
Pink Floyd_03 Have A Cigar.wav	
Pink Floyd_05 Another Brick In The Wall.wav	
Pink Floyd_09 Young Lust.wav	
Pink Floyd_22 Run Like Hell.wav	
Queen_02 Fat Bottomed Girls.wav	
Queen_02 Tie Your Mother Down.wav	
Queen_03 Headlong.wav	
Queen_03 I'm in Love with My Car.wav	
Queen_03 One Vision.wav	
Queen_04 Seven Seas Of Rhye.wav	
Queen_05 Tear It Up.wav	

<b>Dance</b>	
<p>Bass Brothers - Zookey.wav  beatbox - right here right now.wav  Binum - Bassleader Anthem.wav  Bitch Boys - Power Back.wav  Camouflage - The Launch.wav  Candyman - Hit and Jump.wav  Carlos - The Silmarilla (Jump Mix).wav  Cerla &amp; Manian - Jump Att White.wav  Coffeeshock - Zenzeo.wav  Coone and Ghost - Pitch Up.wav  Da Hustler - My Bass.wav  Dan Winter and Mayth - Dare me.wav  Dark E - Cold Fever.wav  DJ Furax - Shake it jumpstyle.wav  DJ Pat B - Love of My Life.wav  DJ Ruthless - Knock Out.wav  DJ Tony - Scrap.wav  DJ Yorit - First Rebirth.wav  Drunkenmunky - Calabria (Jump Mix).wav  GJ Warez - Aircraft.wav  GJ Warez - Aircraft.wav  Grooveyard - Mary Go Wild .wav  Highstreet Allstars - Rock That Beat.wav  Cherry Moon Traxx - Acid Dream.wav  Chicago Zone - Privat Projection.wav  Indian summer.wav  Jeckyll and Hyde - Freefall .wav  Jeckyll and Hyde - Lost in Time.wav  Jump Faktory - Say Say Say.wav  Jump Masters - Taste Of Summer.wav  Jump Project - Hava Tequila.wav  Junior - Drop That Sound.wav  Kosmonova - Ayla.wav  Looney Tunez vs Doop - Doop 2007.wav  Marlon S. - M.U.G..wav  Mason - Exceeder.wav  MC Junkie - Breakfast In America.wav  Meadow Inferno - Fixed.wav  Meisje Met De Prei - Jumped.wav  Mr Vee - Vinyl Song.wav  Mr. vee - Clap your hands.wav  Noise Provider - Bits and Bytes.wav  ODJ Team - Dominator 2007.wav  Popcorn-Popcorn97.wav  Rocco - Street Knowledge.wav  Rts - Poing 2007 (ODJ Team Mix).wav  shaun_baker-hey_hi_hello.wav  shaun_baker-push.wav  shaun_baker-rectangle.wav  shaun_baker-v.i.p.wav  shaun_baker-v.i.p_feat._maloy.wav  shaun_baker-xplode_2.wav  Speakerz - Hard Drive.wav</p>	<p>Starstylers - Keep On Movin.wav  The Highstreet Allstars.wav  The Jumpclub - See Me Jumping.wav  The Jumpers - Blue (Da Ba Dee).wav  The Klubbheads - Let The Party Begin.wav  The Nasty Trio - Hype My Dick.wav  The Sunclub - Fiesta Reloaded.wav  Tommy Trash,Tom Piper - Fuck To The Bass.wav  Track11.wav  Vorwerk - Cambodia.wav</p>

**Classic**

Bach\_Air.wav  
Beethoven\_Moon Sonata.wav  
Boccherini.wav  
Grieg.wav  
Hayden.wav  
Mozart 01-Nachtmusik1.wav  
Mozart 02-Nachtmusik2.wav  
Mozart 03-Nachtmusik3.wav  
Mozart 04-Nachtmusik4.wav  
Mozart 10-Divertimento2.wav  
Mozart 11-Divertimento3.wav  
Tchaikovsky 01-Schwansee.wav  
Tchaikovsky 03-Schwansee3.wav  
Tchaikovsky 05-Schwansee5.wav  
Tchaikovsky 06-Schwansee6.wav  
Tchaikovsky 07-Schwansee7.wav  
Tchaikovsky 08-Schwansee8.wav  
Tchaikovsky 09-Nussknacker1.wav  
Tchaikovsky 10-Nussknacker2.wav  
Tchaikovsky 11-Nussknacker3.wav  
Tchaikovsky 12-Nussknacker4.wav  
Tchaikovsky 13-Nussknacker5.wav  
Tchaikovsky 14-Nussknacker6.wav  
Tchaikovsky 15-Nussknacker7.wav  
Tchaikovsky 17-Nussknacker9.wav  
Vivaldi\_4 seasons.wav  
Vivaldi\_4 seasons\_part2.wav  
Vivaldi\_4 seasons\_part3.wav