

117_111_231

Univerzitet u Beogradu
Matematički fakultet

Master rad

Algoritamsko komponovanje muzike

Student:
Marko Marković
Indeks: 1062/2009

Mentor:
Prof.dr Filip Marić

Beograd
Oktobar 2012

УНИВЕРЗИТЕТ У БЕОГРАДУ
МАТЕМАТИЧКИ ФАКУЛТЕТ
ИМ. Др.
БЕЛАНОВИЋА

Sadržaj

1	Uvod	- 3 -
1.1.	Istorijski razvoj savremene muzike	- 4 -
2	Osnove muzičke teorije	- 5 -
2.1.	Svojstva muzike, zvuka i tišine	- 5 -
2.2.	Osnove muzičke teorije	- 6 -
3	Formati	- 14 -
3.1.	Audio snimak muzike	- 14 -
3.1.1.	Digitalizacija zvuka	- 14 -
3.1.2.	RIFF formati	- 15 -
3.1.3.	WAVE format	- 15 -
3.1.4.	Maskiranje	- 15 -
3.1.5.	MPEG formati	- 16 -
3.1.6.	MP3	- 16 -
3.2.	Notni zapis muzike	- 17 -
3.2.1.	MIDI format	- 17 -
3.2.2.	MusicXML	- 19 -
	Primer. Na kraj sela žuta kuća	- 24 -
4	Tehnike algoritamskog komponovanja	- 27 -
4.1.	Istorija algoritamskog komponovanja	- 27 -
4.2.	Pre-računarski period	- 27 -
4.2.	Komponovanje uz pomoć računara	- 28 -
5	Predloženi algoritam i implementacija	- 32 -
5.1.	Vodopad model	- 32 -
5.2.	Potrebne informacije za generisanje melodije	- 32 -
5.3.	Predloženi algoritam	- 33 -
5.3.1.	Ulazni podaci	- 34 -
5.3.2.	Izlazni podaci	- 35 -
5.3.3.	Domen	- 35 -
5.3.4.	Skala	- 36 -
5.3.5.	Akordi	- 36 -
5.3.6.	Takt	- 36 -

5.3.7. Ritam melodije	- 36 -
5.3.8. Ritam pratećih akorda	- 38 -
5.3.9. Melodija	- 38 -
5.3.10. Opis melodije	- 39 -
5.3.11. Akordi	- 40 -
5.3.12. Rezultat	- 41 -
5.3.13. Vremenska složenost algoritma	- 41 -
5.4. Implementacija algoritma	- 42 -
5.5. Primeri rada programa i generisanih melodija	- 43 -
6 Zaključak	- 44 -
7 Reference	- 45 -

1 Uvod

Tema ovog rada je prikaz oblasti algoritamskog komponovanja muzike, razvoj jednog novog algoritma za automatsko generisanje muzike i njegova implementacija u jeziku C#.

Algoritamsko komponovanje muzike kao oblast je stara kao i samo muzičko komponovanje. Na oblast algoritamskog komponovanja se često gleda kao na varanje ili jednostavno rešenje kada neki kompozitor traži nove materijale ili inspiraciju. Takođe, na ovu oblast može da se gleda kao na alat koji može da olakša posao kompozitoru prilikom kreiranja svog sledećeg dela. Ovaj rad posmatra algoritamsko komponovanje na drugi navedeni način. Često je slučaj da kompozitori prilikom kreiranja nekog dela isprobavaju različite kombinacije melodija i biraju onu koja im deluje kao najprirodnija, najlepša i odgovarajuća u tom trenutku. Ovakav iterativni pristup se može smatrati nekom vrstom algoritamskog komponovanja.

U osnovi, algoritamsko komponovanje muzike je primena rigidnih, dobro definisanih algoritama u procesu komponovanja same muzike. Jedna od glavnih primena je pomoć kompozitorima da bi proširili njihovu muzičku paletu. Druga primena bi takođe bila pomoć prilikom učenja sviranja nekog instrumenta, na primer vežbanje improvizacije.

Muzika i melodije dobijene algoritamskim komponovanjem često se susreću sa pitanjem o autorstvu same "ideje" dobijene ovim pristupom. Da li je sam kompozitor koji je iskoristio generisanu melodiju autor muzike ili sam tvorac algoritma. Na muziku dobijenu uz pomoć algoritama se uglavnom gleda kao na manje važnu ne zato što je napravljena od strane algoritma, već zato što se smatra da muzika dobijena na taj način pripada nekom drugom, samom tvorcu algoritma a ne korisniku algoritma.

Jedina razlika između kompozitorske kreativnosti i samog algoritma je ta da kompozitor ima više fleksibilnosti prilikom kreiranja neke kompozicije. Definicija algoritma je rigidna, dok sama kreativnost često „krši“ pravila. Primena računara je pogodna za zadatak "kreativnosti kroz težak rad" ali nije sposoban za odstupanje iz strogo definisanih pravila. Pod pojmom "kreativnost kroz težak rad" se misli na prolazak kroz mnoge iteracije kreiranih melodija dok se ne dobije ona koja je zadovoljavajuća za korisnika algoritma. Mnogi kompozitori komponuju na ovakav način u praksi, bez praćenja nekog specifičnog algoritma.

Glavni cilj algoritamskog komponovanja muzike je da reprodukuje kreativnu metodologiju kompozitora prilikom kreiranja muzike. Naravno, moguće je složiti se sa činjenicom da algoritam nikad neće zameniti živog kompozitora, ali može da mu bude pomoć a i donekle konkurencija.

U ovom radu, najveća pažnja je posvećena upotrebi algoritama prilikom komponovanja muzike. Upoznajemo se sa istorijom algoritamske kompozicije, sa različitim pristupima rešavanja ovog interesantnog problema i osnovama muzičke teorije. Takođe, u ovom radu se nalazi i predlog jednog originalnog algoritma koji se bavi generisanjem melodija popularne moderne muzike sa akordima koji se izvode u pozadini same melodije.

Razmotrimo pitanje "Šta je to muzika?". Muzika je za mnoge ljude neverbalna forma komunikacije koja dotiče ljudski intelekt i može da izazove duboke ili burne emocije. Drugi smatraju da je muzika pre svega fenomen prirode, rezultat principa fizike i matematike, a da su ljudi samo otkrili, prepoznali i naučili da manipulišu njom. Možemo takođe reći da je muzika kombinacija zvukova koji su organizovani pomoću tri dimenzije: *ritma, melodije i harmonije*. Po nekim ekstremnim definicijama, muzika je bilo koja kombinacija zvukova koju neko negde uživa da sluša [23].

Muzika 20-tog i 21-og veka je glavna inspiracija za ovaj rad. Budući da se poluprofesionalno bavim muzikom u svoje slobodno vreme, potrudio sam se da spojim svoje dosada stečeno znanje iz programiranja, matematike i muzike u algoritam za generisanje melodija. Danas u svetu postoji mnogo algoritama koji se baziraju na različitim tehnikama obrade podataka. Neki uče iz nota kompozitora i pokušavaju da reprodukuju neku kompoziciju u stilu samog kompozitora, neki koriste matematičke formule da generišu melodije. Dugo postoji težnja da se napravi proceduralni algoritam za generisanje melodija popularnih pesama, zbog njihove jednostavnosti, ali dosta tih projekata se završilo bez uspeha. Ja sam se trudio da uspem u tom pokušaju i da napravim nešto što će moći u budućnosti da se unapređuje i usavršava.

1.1. Istorijski razvoj savremene muzike

Hitovi koji dominiraju top listama poslednjih godina, mnogo su glasniji, sa više puta ponovljenim, istim tonovima u odnosu na one koji su vladali osamdesetih godina prošlog veka. To je rezultat istraživanja sprovedenog nedavno u Bristolu, u Velikoj Britaniji [22]. Do zaključka se došlo analizirajući liste najslušanijih hitova i to u poslednjih pedesetak godina.

Naučnici sa muzičkim, ali i inženjerskim znanjem pokušavali su da objasne koliko neka pesma u sebi nosi „snagu“ da postane hit. Opšti zaključak bi mogao da se izvede u nekoliko rečenica: sedamdesetih godina za dominaciju u slušanosti nije bila potrebna ritmičnost, već su od perioda početka sedamdesetih do kraja osamdesetih godina dvadestog veka znatno popularniji bili spori ritmovi. Onda su zavladao bubanj i bas gitara, s uplivom tehničkih novotarija: ritam mašine i računarskih programa za komponovanje. Oni su omogućili da se i u kućnim uslovima napravi ritmična i glasna, a pritom i popularna melodija.

Uprkos uverenju da su osamdesete bile period vladavine disko ritmova, najslušanije pesme tog doba karakterišu manja učestalost ritma, odnosno tempo od 70-89 otkucaja u minuti. Sredinom osamdesetih, najprodavaniji singlovi u Britaniji su bile balade, ukazuju istraživači.

Takođe, u to vreme za uspeh nije bila potrebna toliko, kako su naučnici naveli, sposobnost pesme da „podstakne na igranje“ što je kako su stizale devedesete postalo trend. Kako smo se približili dvadeset prvom veku, pesme koje su bile na samom vrhu, postaju sve glasnije, bržeg ritma i najčešće u četvoro četvrtinskom taktu. Muzički ukusi se menjaju, a ono što je takođe potvrđeno je da je u određenom vremenu prisutna gotovo uniformna dominacija jedne vrste pesama, barem na čelu top lista. Devedesete su donele računarski generisane „binarne“ ritmove i definitivno glasniju muziku.

2 Osnove muzičke teorije

Ovo poglavlje će predstavljati neki osnovni uvod u teoriju muzike. Da bi neka svojstva algoritma bila jasna potrebno je upoznati se sa nekim osnovnim pojmovima muzičke teorije.

2.1. Svojstva muzike, zvuka i tišine

Muzika je vrsta umetnosti čiji medijum su zvuk i tišina. Zvuk, odnosno *zvučni signal* predstavlja promenu pritiska vazduha kroz vreme. *Frekvencija* predstavlja broj perioda u sekundi (mereno u hercima (Hz), ciklus/sekunda). Što signal češće menja svoju vrednost to nam se zvuk čini piskavijim. Čovek čuje opseg frekvencija od 20Hz-20kHz (audio) dok je glas približno od 500Hz do 2kHz. Treba napomenuti da je precizni raspon osobina svakog pojedinca. U principu, signale frekvencije iznad 10kHz većina ljudi veoma loše čuje. *Amplituda* zvučnog talasa je pomeranje pritiska vazdušnog talasa od njegove srednje vrednosti ili razlika između maksimalne i minimalne vrednosti signala. Što je amplituda veća signal je jači. Jačina zvuka se izražava u decibelima (dB). Sama jedinica predstavlja varijaciju pritiska i izražava se uz pomoć logaritamske skale

$$\text{Intenzitet (dB)} = 10 \log_{10} (P/P_0)$$

gde P predstavlja jačinu zvučnog pritiska a P_0 donji prag čujnosti. Jačina zvuka koju ljudsko uho može da registruje se kreće od skoro 0dB (prag šuma) do 120dB (prag bola).

Treba se osvrnuti i na odnos pojmova kao što su *zvuk* i *ton*. Većina zvukova koji do nas dopiru iz okoline je mešavina velikog broja zvučnih talasa različitih frekvencija – mi ih registrujemo kao šumove ili buku. Međutim, ako neki zvučni izvor osciluje pravilno, stvarajući zvučne talase tačno određene frekvencije, mi ćemo taj zvuk registrovati kao specijalan (muzički) ton. Ton je dakle zvuk koji ima četiri svoje karakteristike: *visinu* (koja zavisi od frekvencije zvučnog talasa), *intenzitet* (koji zavisi od amplitude talasa), *trajanje* i *boju*. Na osnovu toga se može zaključiti da su muzički tonovi svojevrsni atomi od kojih je sastavljena svaka muzička kompozicija.

Reč muzika potiče od Grčke reči *μουσική* (mousike; "umetnost Muza"). Muzika može da bude glasna ili tiha, note mogu da budu kratke ili duge ili bilo gde između. Muzička fraza može da se izvede nežno ili grubo, neprekidno ili iseckano, i muzika može da se svira u različitim brzinama, od sporog do brzog. Tipični elementi muzike su *visina tona* (koji određuje melodiju i harmoniju), *ritam* (koji je u vezi sa tempom, metrom i artikulacijom), *dinamika* (koja određuje glasnoću i način izvođenja nekog dela), i *zvučni kvalitet* (koji je određen bojom zvuka i teksture).

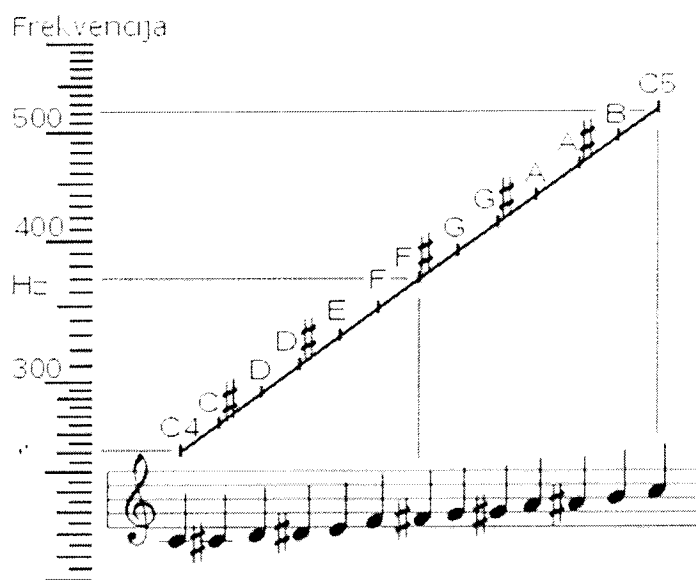
Kao što je kompleksan svetlosni zrak sastavljen od duginih boja, zvuci koje mi čujemo oko sebe su kompleksan skup frekvencija, koji je sastavljen od puno „čistih“ zvukova (zvukova samo jedne frekvencije). Naše čulo sluha radi slično kao *Furijeov analizator* i razdvaja kompleksan zvučni talas na spektar jednostavnih talasa. Boja ljudskog glasa je zbog toga specifična: svako od nas proizvodi svoj „lični“ spektar zvučnih talasa, i kada prepoznamo nečiji glas, mi ustvari uspevamo da detektujemo upravo taj lični, specifični spektrogram.

Na isti način, kada neko odsvira određen ton na flauti, violini ili klaviru, mi čujemo da su tonovi iste visine, ali potpuno različite boje. Za boju tona odgovorni su tzv. *aliquotni tonovi* (viši harmonici), koji se čuju pored osnovnog tona. Ako je osnovni ton frekvencije f , aliquotni tonovi koje stvaraju naši instrumenti jeste spektar tonova manje jačine čije su frekvencije celobrojni umnošci od f tj. $2f, 3f, 4f, \dots$ Od raspodele tih aliquota zavisi boja tona određenog instrumenta. To se sada može objasniti Pitagorinim zakonom malih brojeva, koji je prvi uočio da ton odsviran na duplo dužoj ili kraćoj žici zvuči veoma slično osnovnom tonu tj. radi se o istom tonu u različitoj oktavi.

Danas električni muzički instrumenti (kao što je sintisajzer, električna gitara) ili razni dodaci tim instrumentima, mogu veštački da manipulišu zvučnim talasima različitih frekvencija, da ih razdvajaju ili sabiraju, da ističu ili smanje intenzitet određenih aliquota, i tako sintetišu zvuke novih boja koji se ne bi mogli dobiti na prirodan način. Na neki način, to označava novu eru u muzici, i mi smo, hteli to ili ne, svedoci te velike promene.

2.2. Osnove muzičke teorije

Visina tona je svojstvo zvuka a pogotovu muzičkog tona koji je određen frekvencijom talasa koji ga proizvodi. Što je ton više frekvencije ton ljudskom uhu zvuči piskavije, a što je ton niže frekvencije zvuči dublje. Na slici 2.1. predstavljen je odnos frekvencija i odgovarajućih visina tonova koji se obično izvode.



Slika 2.1. Frekvencija i visine tonova

Raspon se odnosi na rastojanje između najvišeg i najnižeg tona. Uglavnom se predstavljaju celim brojem koji predstavlja broj različitih tonova između njih. Postoje dva načina predstavlja tog celog

broja: broj različitih tonova skale (predefinisani podskup tonova) i broj različitih tonova (u odnosu na ceo skup tonova).

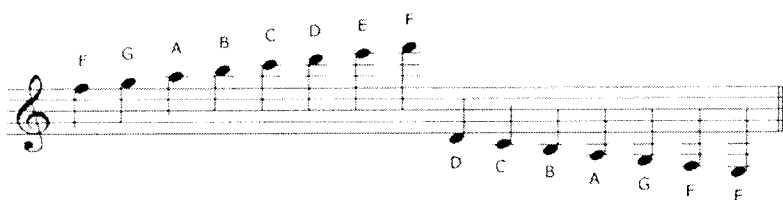
Muzička azbuka (domen) se sastoji od slova: *A, H, C, D, E, F, G*. U nekim sistemima se umesto slova *H* koristi oznaka *B* ali su u suštini to isti tonovi. Takođe u ovom sistemu se malo slovo *b* koristi kao skraćunica za ton *A# (Ais)*. Od ovih 7 oznaka se može napraviti 12 različitih tonova uz pomoć povisilica i snizilica, o čemu će biti reči kasnije, ali je bitno napomenuti da se posle ovih 12 tonova grupa ciklično ponavlja i da te oznake čine kompletan domen. Takođe, bitno je napomenuti da je redosled fiksiran. Svaka ova oznaka predstavlja ton različite frekvencije koje su poredane u rastućem redosledu.

Na kraju muzičku azbuku čine tonovi: *C, Cis, D, Dis, E, F, Fis, G, Gis, A, b i H*.

Često se koristi i solmizacija gde muzičku azbuku čine isti tonovi ali drugačije nazvani i to su tonovi: *Do, Re, Mi, Fa, Sol, La, Si*.

Zapis muzike se vrši u *notnom sistemu* (zapisu) koji se sastoji od 5 horizontalnih linija. Te horizontalne linije prave 4 prostora između njih. Linije i prostori između njih se numerišu odozdo na gore. U takvom notnom sistemu, linije predstavljaju visinu tona, i ako se zapisana nota nalazi na višem nivou ton je viši (veće frekvencije) a ako je zapisana nota na nižem nivou ton je niži (manje frekvencije). Note u notnom sistemu se postavljaju ili samo na liniji ili samo u prostoru između linija. Visina tona se predstavlja pozicijom u notnom sistemu.

Većina nota se zapisuje u notnom sistemu, mada neke note se mogu naći više ili niže nego što notni sistem to može da prikaže. U tom slučaju koriste se male horizontalne *pomoćne linije* i crtaju se na mestima gde bi linija notnog sistema trebala da bude u slučaju da u svom prikazu sadrži više od pet linija i četiri mesta između. Pomoćna linija se uvek koristi zajedno sa notom koju predstavlja u slučaju da se ona nalazi van notnog sistema (kao što je to prikazano na slici 2.2.).

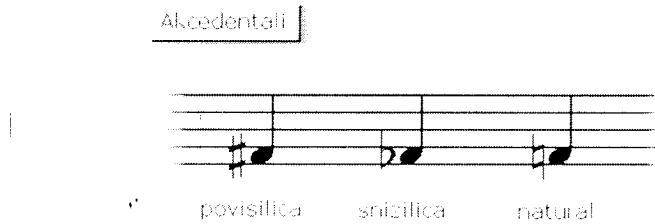


Slika 2.2. - Zapis nota koje se nalaze van notnog sistema

Visina tona može da se promeni na dva načina: da se pomalo povisi i da se pomalo snizi. To se prikazuje uz pomoć posebnih simbola koji se stavljaju ispred note na koju utiču. Te simbole nazivamo *akcedentalima*. Akcedentali predstavljaju te promene visine tonova i postoje tri vrste akcedentala: *povisilica, snizilica i natural*.

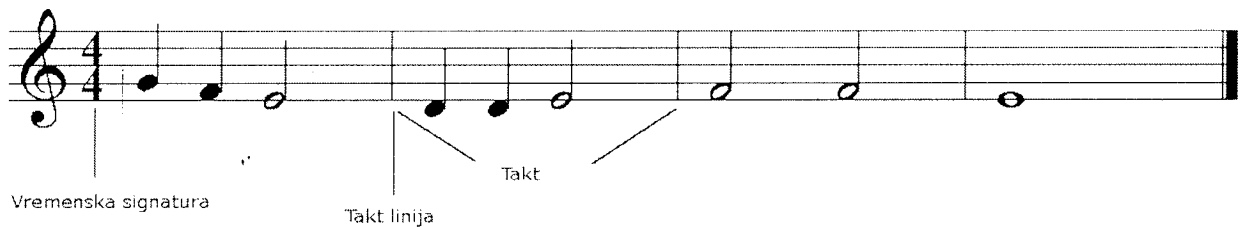
2 Osnove muzičke teorije

Snizilica predstavlja smanjenje visine tona za malu količinu (tačnije za polustepen, taj termin će biti naknadno objašnjen), povisilica predstavlja povećanje visine tona za malu količinu dok natural ne menja visinu tona i njena primena ima smisao poništavanja efekta prethodno definisane povisilice ili snizilice.



Slika 2.3. - Akcedentali

Da bi zapis muzike bio lakši za čitanje, notni zapis se deli na sekcije koje se nazivaju taktovi i završetak takta se obeležava sa vertikalnom linijom. Dvostruka vertikalna linija se koristi kada želimo da označimo kraj neke sekcije ili kraj pesme. Na slici 2.4. prikazan je notni zapis koji se sastoji od četiri takta.



Slika 2.4. - Notni zapis

Dužina nota– Ukratko rečeno, dužina note prilikom izvođenja muzike predstavlja određen broj otkucaja u minuti tj. koliko se neki ton svira bez prekida. Postoji dosta različitih zapisa dužina nota od kojih se najčešće koristi 5 osnovnih.

Naziv	Nota	Pauza	Dužina
Cela nota			4 oktucaja
Polovina			2 oktucaja
Četvrtina			1 oktucaj
Osmina			1/2 oktucaja
Šesnaestina			1/4 oktucaja

Slika 2.5. - Različite dužine trajanja nota

Note sa tačkom – Pored prethodne definicije dužina nota, postoji još jedan pojam a to su note sa tačkom. Pored svake prethodne definisane note može da se stavi tačka. Ta tačka označava da se dužina trajanja note produžava za 50%. Ako je to na primer cela nota, ona će umesto 4 otkucaja trajati $4+2 = 6$ dok će na primer četvrtina umesto 1 otkucaja trajati $1 + \frac{1}{2} = \frac{3}{2}$ itd.

Pauze - Pauze posmatramo kao note koje nemaju nikakav zvuk tj. tišine. One takode imaju svoje dužine trajanja sa tim da imaju svoj specifični način zapisivanja. Taj način zapisa se može videti na slici 2.5.

Ključ je simbol koji se koristi na početku notnog zapisa koji uspostavlja vezu između slova iz muzičke azbuke sa linijama i prostorom između. Postoji više vrsta ključeva i to su: C ključ, violinski ključ, bas ključ i ritam ključ. Za ovaj rad interesantni su violinski i bas ključ.

Violinski ključ je najčešća vrsta zapisa muzike. Uglavnom se koristi za instrumente sa tonovima u višim registrima. *Registar* predstavlja skup oktava koji neki instrument može da izvede. Primer instrumenata koji koristi violinski ključ su: pikolo flauta, violina, klavir, glas, gitara itd.



Slika 2.6. - Violinski ključ

Bas ključ je druga najčešća vrsta zapisa muzike. Bas ključ se koristi za instrumente sa nižim registrom i neki od njih su: čelo, tuba, klavir, vokal, bas gitara itd. Klavir i vokal imaju velike raspone tonova pa se mogu predstaviti sa oba ključa.



Slika 2.7. - Bas ključ

Prostiji oblik muzike se može predstaviti uz pomoć pratnje u vidu akorda (zapisana u bas ključu) i melodije (zapisana u violinskom ključu). Kao što i sama definicija ključeva kaže, pratnja će da sadrži tonove nižeg registra dok će melodija sadržati tonove višeg registra.

Kako bi se izbegla potreba za navođenjem akcedentala ispred velikog broja nota, akcedentale je moguće uvesti i na nivou pojedinačnih taktova a i cele kompozicije, i tada važe sledeća pravila.

1) Kada se akcedentali koriste na početku takta, njegovi efekti traju do kraja takta. To je pravilo namenjeno radi lakšeg zapisa da ne bi za svaki ton ponavljali znak za povisilicu ili snizilicu.

2) Takode mogu se definisati akcedentali za celu kompoziciju i to tako što se zapisuju znakovi na odgovarajućem mestu u prvom taktu posle zapisa violinskog ključa i metra.

Vremenska signatura/metar - Posle definicije ključa zapisuje se *vremenska signatura ili metar*. Svaki takt može ali i ne mora da ima svoju specifičnu vremensku signaturu. Vremenska signatura nam govori informaciju koliko otkucaja ima u jednom taktu. Pored te informacije, dobija se informacija koja dužine note dobija jedan otkucaj. Predstavljanje vremenske signature je u obliku razlomka npr 4/4. U slučaju 4/4 sada znamo da takt traje 4 otkucaja i da se otkucaj ritma meri na svaku četvrtinu tj. na svaki otkucaj.

Tempo je reč koja potiče od italijanske reči *tempus* koja u prevodu znači vreme. Postoje različite vrste tempa i sledeća tabela objašnjava vezu između vrste tempa i otkucaja po minutu (BPM engl. beats per minute). Prostije rečeno, tempo nam govori koliko brzo ili sporo izvodimo neku kompoziciju.

Kategorija	Klasifikacija tempa	BPM(Tempo)
Klasična	Prestissimo	200-208
Klasična	Presto	168-200
Klasična	Vivace	~140
Klasična	Allegro	120-168
Novi talas 80-tih godina	Outfield Playdeep Brzi singlovi	125-135

2 Osnove muzičke teorije

Moderna muzika	Hit pesme optimalni tempo	120
Rok en Rol	Hit pesme Beatles-a	111-145
Rok muzika	Nirvana Nevermind album	90-133
Klasična	Moderato	90-115
Klasična	Adagio	66-76
Klasična	Larghetto	60-66
Klasična	Largamente/Largo	<40

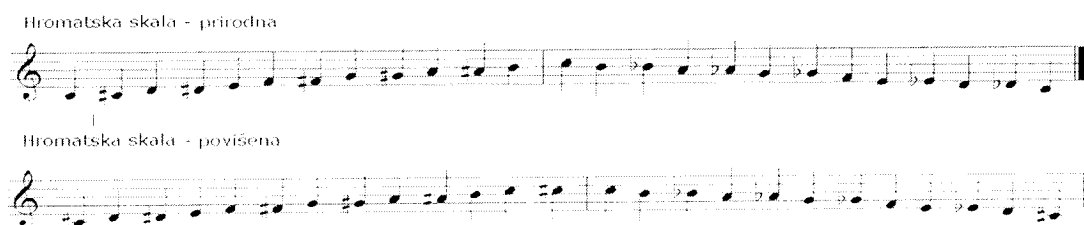
Slika 2.8 - Tabela tempa

Dinamika je koliko glasno ili tiho izvodimo muziku. Osnovne reči za dinamiku su *piano* (italijanska reč tiho) i *forte* (italijanska reč glasno). Dinamika neke note se takode može objasniti visinom amplitude samog signala. Što je amplituda veća nota je glasnija a u suprotnom je tiša.

Polustepen je razlika između bilo koja dva susedna tona u domenu. To je najmanji razmak koji je moguće napraviti između dva elementa domena. Dva polustepena čine *ceo stepen*. Ovi elementi predstavljaju osnovne gradivne blokove muzičkih skala.

Skale – skale predstavljaju određen podskup domena koji se ciklično ponavlja. Postoje razne vrste skala i u nastavku se obraća pažnja na par bitnih za ovaj rad. *Diatonička skala* je izraz koji se koristi za skale koje se sastoje od 7 nota koje se ciklično ponavljaju. Razmak između svakog ponavljanja je oktava.

Hromatska skala (prikazana na slici 2.9.) – od svih skala je najjednostavnija. Ona se sastoji od 12 tonova i sadrži svaki ton iz domena.



Slika 2.9. - Hromatska skala

Durska skala – jedna od najzastupljenijih skala u muzici. Može se primetiti u raznim žanrovima muzike: od hip-hopa, do klasične i jazz muzike, pa i do rock i popularne muzike. Definicija durske skale je gradiran niz muzičkih tonova u rastućem ili opadajućem redosledu prema specifičnoj šemi njihovih intervala. Kod primera hromatskih skala intervali su u razmaku polustepena dok je kod durske skale šema intervala niz polustepena i stepena.

Svaka skala ima svoje ime i opisno ime. Prvo je ime donjeg tona skale a i pritom i gornjeg tona skale. Opisno ime govori koji je tip skale kao što su durska, molska, hromatska itd. Na primer D Dur je skala koja počinje i završava se u tonu D i skala je durskog tipa.

Pre nego što se definiše šema durske skale trebalo bi uvesti nekoliko oznaka. Neka je oznaka za polustepen **h** (engl. Half step) i neka je oznaka za ceo stepen **w** (engl. Whole step). Tada šema intervala durske skale glasi: **w w h w w h**

Sada ova šema može da se primeni na bilo koji početni ton i da se dobije durska skala. Ako uzmemo na primer ton C i konstruišemo skalnu iz domena dobijamo **C dur** skalnu



Slika 2.10. - Cdur skala

Na kraju skalnu nam definiše niz tonova: **C, D, E, F, G, A, H, C.**

Molska skala – molska skala se može prepoznati po osećaju. Uglavnom izaziva osećanje tuge i melanholije. Postoje 3 različite molske skale: *prirodna, harmonijska i melodijska*. Kao i durska skala, svaki tip molske skale ima različite šeme intervala polustepena i stepena.

Prirodna molska skala ima šemu: **w h w w h w w**. Neka se za primer uzme C prirodna molska skala. Tonovi bi bili: **C, D, Dis, F, G, Gis, b, C.**

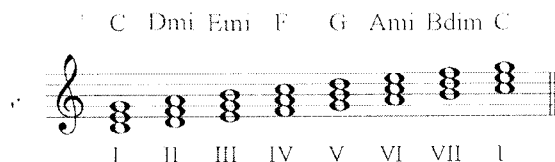
Harmonijska molska skala je pomalo izmenjena prirodna molska skala. Dobija se tako što se 7 ton prirodne molske skale povisi za polustepen tako da šema izgleda ovako: **w h w w h w w+h**, gde w+h označava stepen i po. Kada primenimo ovu šemu na ton C dobijamo niz: **C, D, Dis, F, G, Gis, H, C.**

Melodijska molska skala se dobija kombinacijom ove dve skale. Kad se melodija kreće u rastućem smeru koristimo harmonijsku molsku skalnu, a kada se melodija kreće u opadajućem smeru koristimo prirodnu molsku skalnu.

Osnovna vrsta izvođenja muzike je monofona melodija tj. melodija gde se u svakom trenutku izvodi samo jedan ton. Nasuprot ovoga, polifona muzika podrazumeva izvođenje više tonova odjednom.

Akord čine 3 ili više tonova koji se izvode istovremeno. Postoji mnogo vrsta akorda koje su kao i u slučaju skale durski, molski, prekomerni itd. Kvalitet akorda je određen intervalom tonova unutar samog akorda. Akordi se dobijaju od tonova skale koji su u razmaku dve pozicije od bilo koje početne pozicije u skali (gde se pod pozicijom podrazumeva redni broj tona u skali). Tako na primer C durski akord čine tonovi na prvoj, trećoj i petoj poziciji skale tj. tonovi C E G. Bitno je napomenuti da su akordi koji se prave od prve, četvrte i pete pozicije durski, dok druga, treća i šesta pozicija čine molske akorde.

Akord na sedmoj poziciji je prekomeran i on se skoro nikad ne koristi. Ne treba da zbunjuje činjenica da u okviru skale jednog tipa imamo više različitih tipova akorda. Oni se deklarišu prema zvuku. Molski akordi zvuče tužnije dok đurski akordi zvuče veselije ali su svi u tonalitetu sa skalom u kojoj se nalaze. Na slici 2.11. prikazani su svi akordi koji se mogu napraviti od tonova iz C dur skale.



Slika 2.11. - Akordi - Cdur triade

Diatonička progresija akorda definiše skup pravila koja definišu koji akord može da sledi posle akorda koji se trenutno izvodi. Ovaj skup pravila je karakterističan za modernu popularnu muziku, mada se može videti i u klasičnoj i džez muzici. Progresija akorda je nastala posmatranjem *diatoničkog kruga*. Diatonički krug je vizuelni prikaz odnosa između 12 tonova hromatske skale, njihovih odgovarajućih signatura ključa, i odgovarajućih đurskih i molskih tipova ključa. Svaka muzička skala ima svoju vizuelnu reprezentaciju diatoničkog kruga. Dva susedna tona na tom prikazu su uvek u razmaku pet tonova skale kojoj odgovaraju. Prostiše rečeno, svaki sledeći akord koji treba da se odabere je u razmaku 5 tonova skale u odnosu na prethodni. Ovako grubo definisana pravila su korisna, ali treba imati na umu da u praksi nije uvek tako. Na slici 5.8. je prikazana jedna od struktura pravila koja se koriste za progresiju akorda. Ta pravila su proširena u odnosu na standardna pravila diatoničke progresije akorda. Problem je taj što ne postoji zvanični skup modifikovanih pravila koji se koristi i skoro svaki autor literature muzičke teorije uvodi neku svoju varijaciju pravila. Iz tog razloga, u ovom radu je uvedena posebna struktura pravila.

Nakon što su definisani neki osnovni pojmovi može se opisati šta je uopšte melodija. *Melodija* (potiče od Grčke reči *μελωδία* (meloidia; "pevanje, uzvikujući")) takode poznata je kao glas ili linija. Glas ili linija je linearni niz muzičkih tonova koji se doživljava kao jedan entitet. U doslovnom smislu, melodija je kombinacija visine tona i ritma, dok je više figurativno, ovaj termin proširen da obuhvati sukcesiju drugih muzičkih elemenata kao što je boja tona. Melodiju možemo posmatrati kao nešto što je u prvom planu dok slušamo neku pesmu, kompoziciju ili slično. Melodije se često sastoje od jedne ili više muzičkih fraza ili motiva, i obično se ponavljaju kroz pesmu ili komad u različitim oblicima. Melodija se može opisati kao melodijsko kretanje, promene visine tonova, kao intervali između visina tonova (uglavnom kao konjukcija ili disjunkcija ili sa nekim dodatnim ograničenjima), tenzija i opuštanje, kontinuitet i koherencija, kadenca i oblik. Kadenca je harmonijska veza dva ili tri akorda različitih funkcija, kojom se završava (zaokružuje) muzička misao. Takode, kako nam neka melodija zvuči zavisi takode i od skale u kojoj se izvodi. Ne može bilo kakav niz tonova iz skale dati „slušljivu“ melodiju, tu moraju postojati neka pravila.

3 Formati

Format ili tip datoteke određuje na koji način se vrši memorisanje računarskih podataka. Te datoteke su predstavljene jednodimenzionim redovima bajtova. Takve ograničene i proste linearne strukture se mogu iskoristiti da bi opisali različite vrste kompleksnijih podataka uz pomoć konvencija. Sve konvencije za jednu vrstu datoteka poznate su kao format datoteke.

Čuvanje audio signala je težak zadatak sam po sebi. Razlog je taj što je sam audio signal neprekidan, dok su podaci u računaru u diskretnom obliku. Potrebno je izvršiti digitalizaciju analognog signala zvuka tj. primeniti postupak pretvaranja analognog signala u oblik razumljiv računaru. Digitalizacija zvuka predstavlja tehnologiju koja snima, smešta i reprodukuje audio signal u digitalnoj formi.

Muzika se na računaru može čuvati na dva načina i to kao:

1. audio snimak (zapis) muzike/zvuka koja se izvodi
2. notni zapis muzike koju želimo da izvedemo

3.1. Audio snimak muzike

Prvi način predstavlja reprodukovanje zvuka na računaru onakvog kakvog ga mi čujemo u prirodi. Analogni signal se pretvara u oblik koji je razumljiv računaru i koji on može da reprodukuje. Posebnim konvencijama zapisujemo te podatke tako da računar može da ih razume i izvede. Problem ovakvog načina zapisa je taj što snimak nije podložan izmeni u toku njegovog izvođenja. Takođe ovakav zapis zvuka zauzima dosta prostora na računaru. Neki od poznatijih formata su WAVE, Mp3, OGG itd.

3.1.1. Digitalizacija zvuka

Sempliranje predstavlja proces pretvaranja analognog signala (kao što su na primer zvuk ili bilo koja neprekidna funkcija) u numeričku sekvencu (diskretnu funkciju).

Tipičan metod uz pomoć kojeg se analogni signal predstavlja u digitalnom obliku je poznat kao modulacija impulsnog koda (engl. Pulse Code Modulation ili PCM). Niz podataka dobijen ovim metodom predstavlja digitalni oblik analognog signala, gde se analogni podatak semplira na uniformnim intervalima, gde je svaki odsečak kvantizovan do najbliže vrednosti.

Najkvist-Šenonova (Nyquist-Shannon) teorema sempliranja predstavlja jednu od osnovnih teorema u informacionim tehnologijama i to u oblastima telekomunikacija i obrada signala.

Teorema (Najkvist-Šenon) Frekvencija sempliranja bi trebala da bude barem dva puta veća od najveće(maksimalne) frekvencije koja se nalazi u signalu.

$$f_s \geq 2f_c$$

gde je f_s frekvencija sempliranja (koliko često se semplovi uzimaju po jedinici vremena) i f_c je maksimalna frekvencija koja se nalazi u signalu.

Opšte prihvaćen CD audio standard se zasniva na učestalosti sempliranja od 44.1 kHz. DAT kasete, ranije korišćene za profesionalno snimanje, koriste učestanost od 48kHz. Većina zvukova u video igrama je semplirana na 11 ili 22kHz.

Iako se ranije za digitalizaciju koristilo 8 bitova (jedan bajt), danas je standardno da se za zapis svakog sempla koristi 16 bitova (2 bajta). Ovo omogućuje zapis 65536 različitih nivoa jačine zvuka, što daje dinamički raspon od nekih 96dB, što je prilično zadovoljavajuće.

Da bi se bolje dočarao prostorni raspored zvuka, koristi se stereo tehnika. Za digitalizaciju stereo zvuka potrebno je najmanje 2 mikrofona (dva kanala). Ukoliko jednostavno zapišemo niz brojeva dobijenih digitalizacijom zvuka, dobijamo tzv. sirovi zapis (PCM – Pulse Code Modulation). Za zapis jednog minuta zvuka u stereo tehnici, potrebno je: $44100 * 2 \text{ bajta} * 2 \text{ kanala} * 60 \text{ sekundi} = 10,5 \text{ MB}$. U novije vreme koriste se zapisi i sa više kanala (najčešće 5 + 1).

3.1.2. RIFF formati

RIFF je grupa formata za zapis mnogih tipova podataka a pre svega multimedijalnih (zvuk, video itd). Neki od najpoznatijih RIFF formata su WAVE, AVI, DIVX, MP3 itd. Svi RIFF formati se sastoje od parčića (engl. chunks). Svako parče ima svoj tip (engl. header), koji se zapisuje pomoću 4 karaktera (npr 4 ASCII bajta), pa zatim slede 4 bajta koja označavaju veličinu parčeta i zatim sadržaj. U svakoj datoteci može da bude mnogo parčića. RIFF datoteka je sama za sebe jedno parče čiji sadržaj počinje oznakom tipa RIFF datoteka, a zatim nizom drugih parčića.

Microsoft i IBM su 1991. godine uveli ovaj format i predstavljen je kao osnovni format za Windows 3.1 operativni sistem. On se bazira na Electronic Arts-ovom formatu koji je uveden 1985. godine na Amigi 1000.

3.1.3. WAVE format

WAV spada u grupu RIFF formata i namenjen je isključivo za zapis zvuka. Zapis u WAV formatu se sastoji od parčeta sa oznakom „fmt“ i parčetom sa oznakom „data“. WAV format omogućava i nekoliko tipova kompresije, mada se najčešće koristi za zapis nekompresovanog zvuka, tj. parče „data“ sadrži PCM zapis. Problem sa WAV zapisom je taj što zauzima previše memorijskog prostora. Pošto je zvuk objekat koji se veoma nepredvidivo menja, većina algoritama kompresije koji se zasnivaju na ponavljanjima podataka (ZIP, RAR, ARJ) pokazuju loše rezultate.

Zbog toga se pristupa primeni tzv. psihoakustičnih algoritama koji uglavnom spadaju u grupu algoritama sa gubitkom (engl. lossy). Jedan od najpoznatijih formata koji koristi algoritme sa gubitkom je mp3 format.

3.1.4. Maskiranje

Većina algoritama se zasniva na činjenici da se tihi zvuk u blizini mnogo glasnijeg neće ni primetiti i da se na njegovo kodiranje ne isplati trošiti dragocene bajtove. Slično, ako u bliskom trenutku postoje zvuci bliske frekvencije, jači zvuk može da maskira drugi. Ovi podaci se najčešće dobijaju eksperimentima i to sa ljudima koji slušaju zvukove i daju svoj sud. U prevodu to bi značilo da se maskiranje zvuka primenjuje samo na frekvencijski bliskim zvukovima. Iz toga sledi da je dopušteno

podizanje nivoa šuma u blizini jakih zvukova, a čim je nivo dopuštenog šuma veći potrebno je manje bitova za zapis.

Još jedan značajan vid maskiranja je osobina da se vremenski bliski zvukovi maskiraju. Premaskiranje kaže da se tih ton koji se javi do 5 milisekundi pre glasnog neće čuti. Postmaskiranje ima još duži efekat i traje do 100 milisekundi posle završetka jakog zvuka. Najprostije rečeno, kompresija zvuka se zasniva na ograničenjima (slabostima) ljudskog uha.

3.1.5. MPEG formati

MPEG (Moving pictures experts group) je ekspertska organizacija koja je pod pokroviteljstvom ISO standarda napravila nekoliko standardnih formata za zapis zvuka, filma i ostalih multimedijalnih sadržaja.

Neki od poznatijih MPEG formata su:

1. MPEG1 je standard na kome su zasnovani formati kakvi su video CD i MP3
2. MPEG2 standard na kome se zasniva digitalna televizija i DVD format
3. MPEG4 je standard multimedije za fiksni i mobilni web
4. MPEG7 standard za opisivanje i pretragu audio i vizuelnog sadržaja

Slojevi unutar MPEG čine oznake podstandarda koji se odnosi samo na zapis audio signala. Jedan od najpoznatijih MPEG-ovih slojeva je *MPEG layer 3*, poznatiji kao MP3. MP3 je najčuleniji dok su ostali formati skoro zaboravljeni. MPEG layer-i su niže kompatibilni što znači da programi koji mogu da tumače MP3 mogu da tumače i niže layer-e.

Audio MPEG deli celokupni zvučni pojas na 32 podpojasa. Ovi pojasevi kod layera 1 i 2 su bili po 625Hz, dok se kod layera 3 uvode pojasevi različite frekvencijske širine.

3.1.6. MP3

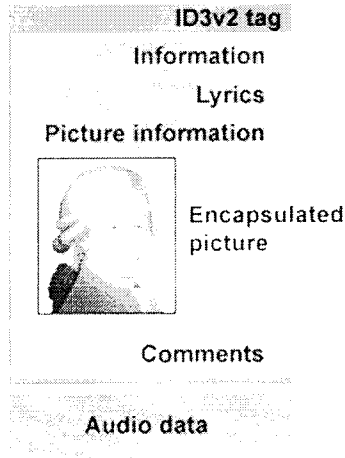
MP3 je danas jedan od najkorišćenijih formata za razmenu i reprodukciju muzike. U današnjem dobu brzog interneta ovaj format sve više i više zamenjuje kompakt diskove, ploče i kasete. Ne zauzima mnogo prostora u memoriji a pritom ima zadovoljavajući kvalitet u odnosu na PCM formate. Svaka MP3 datoteka se sastoji od više delova koji se nazivaju okviri (frames). Svaki okvir se sastoji od 32bitnog zaglavlja (header) i sadržaja. Jedan okvir služi za zapis 1152 uzorka.

Zaglavlje se sastoji od 32bita: AAAAAAAAAA AAABBCCD EEEFFGH IJJKLMM.

A bitovi se koriste za sinhronizaciju, B predstavlja oznaku MPEG kompresije, C je oznaka sloja, D predstavlja oznaku kontrole, E se koristi za bit stopu (označava se u kbps), F se koristi za stopu sempliranja, G se koristi za mogućnost poravnanja bitova, H je rezervisani bit, I predstavlja broj kanala koji se koristi, J je dodatna oznaka načina zapisa (koristi se kod stereo sistema), K ukazuje da li postoje autorska prava, L ukazuje da li je snimljeni materijal original ili kopija i konačno M je oznaka pojačanja.

Iako u zaglavlju svakog okvira nije eksplicitno zapisana njegova dužina, ona se može rekonstruisati na osnovu ostalih podataka i to kod Layera 3 po formuli:
$$\text{Dužina} = 144 * \text{bitrate} / \text{samplerate} + \text{padding}$$

MP3 format se može proširiti dodatnim informacijama o muzici, izvođaču, tekstu pesme i slično. Standard koji ovo opisuje se zove ID3 i trenutno nosi oznaku verzije 2.4.0. Pošto se ovaj standard pojavio posle standardizacije MP3, ovakve dodatne oznake su se pisale na kraju MP3 datoteke. Tek od verzije 2, su oznake premeštene na početak.



Slika 3.1. – mp3 ID3 v2 oznaka

3.2. Notni zapis muzike

Drugi način predstavlja zapis notnog zapisa muzike koja se izvodi. Od potrebnih informacija su uglavnom visina tona koja se izvodi i njeno trajanje. Zauzimaju mnogo manje prostora u memoriji nego prethodni format i notni zapis se može menjati u toku izvođenja. Mana ovakvog zapisa podataka je taj što je ovakav format primenljiv na instrumente ili objekte koji izvide zvukove u skladu sa zapisom nota u muzičkoj teoriji. Računar čita ovakav format i za svaki ton koji pročita pušta odgovarajući kratak snimak (semp) koji stvara utisak izvođenja tog tona. Zbog toga, izvođenje može zvučati pomalo veštački.

3.2.1. MIDI format

Microsoft je uveo novu varijantu postojećeg MIDI formata koji se koristio za smeštanje informacija o pesmi koja treba da se izvede na nekom elektronskom instrumentu. „Novi“ MIDI format je u suštini standardna MIDI datoteka koja je enkapsulirana sa RIFF „omotom“ i imala je ekstenziju .RMI. Pošto je već postojeći MIDI format datoteke podržavao ugrađene oznake (engl. tags), prednosti novog formata nisu bile očigledne.

Zbog toga su MIDI proizvođači prihvatili RIFF zasnovan na MIDI formatu datoteke, i koristio se kao osnova za „proširenu MIDI datoteku“ koja takođe sadrži informacije o instrumentu u „DLS“ formatu, ugrađenu u okviru iste .RMI datoteke.

MIDI (Musical Instrument Digital Interface) je standard koji omogućava velikom broju digitalnih muzičkih instrumenata, računarima i drugim srodnim uređajima da se povežu i komuniciraju međusobno. Standardna MIDI datoteka (SMF) je specijalno dizajnirana za smeštanje podataka (zvuka) koje sekvencer može da snima i izvodi. *Muzički sekvencer* je uređaj ili softverska aplikacija koja može da snima, menja ili izvodi (pušta) muziku, tako što upravlja informacijama o notama i izvođenju u nekoliko različitih formi, uglavnom MIDI formatu.

Format sadrži standardne MIDI poruke (MIDI messages) sa vremenskim pečatom za svaku poruku (npr. serija bajtova koja predstavlja koliko otkucaja časovnika treba da se čeka pre nego što se „odsvira“ neki događaj). Ovaj format omogućava čuvanje informacija o tempu, otkucaju po četvrtini trajanja note, ključu, nazivu kanala i šablonima. Takođe može da čuva više različitih kanala i šablona tako da bilo koja aplikacija može da obradi ove strukture kada učitava datoteku. Kanal odgovara jednom delu muzičkog izvođenja, kao što je npr. izvođenje trube u jednom delu. Šablon odgovara svim muzičkim delovima (truba, bubanj, gitara itd.) za pesmu.

Format je dizajniran da bude generički tako da sekvencer može da čita i piše takvu datoteku a da pritom ne izgubi najbitnije podatke, a pritom je dovoljno fleksibilan da svaka aplikacija može da ga obradi. Na MIDI datoteku može da se gleda kao muzička verzija ASCII tekstualne datoteke (osim što MIDI sadrži i binarne podatke).

Kao što je to osobina i ostalih RIFF formata MIDI datoteka se čuva u parčićima (chunks) koji mogu da se parsiraju, učitaju, preskoče itd. To nam daje mogućnost da smeštamo dodatne informacije u datoteku, a da pritom druge aplikacije koje ne poznaju te informacije mogu da preskoče te delove.

Primer (MIDI parčića) Bajtovi su izraženi heksadekadnim vrednostima: 4D 54 68 64 00 00 00 06

Prva četiri bajta predstavljaju vrednost MThd zaglavlja izraženim u ASCII vrednostima. Sledeća 4 bajta nam govore da treba da očekujemo 6 bajtova informacija pre kraja parčića. Posle tih 6 bajtova tražimo novi parčić.

Sve MIDI datoteke počinju sa ovim MThd zaglavljem i tako znamo da radimo sa MIDI datotekom. Naslov ima dužinu 6 bajtova. On sadrži i informaciju o tipu MIDI datoteke.

Postoje 3 različita tipa formata MIDI datoteka:

1. Tip 0 označava da datoteka sadrži samo jednu traku koja sadrži MIDI podatke na svih mogućih 16 kanala.
2. Tip 1 označava da datoteka sadrži jednu ili više simultanih traka, verovatno svaka na jednom midi kanalu.

3. Tip 2 označava da datoteka sadrži jednu ili više sekvencijalno nezavisnih šablona sa jednom trakom.

Poslednja dva bajta govore informaciju o vremenskom otisku tj koliko otkucaja časovnika po četvrtini note se zasnivaju otisci.

Primer. MIDI MThd

4D 54 68 64 MThd ID

00 00 00 06 Dužina MThd parčeta je uvek 6.

00 01 Tip formata je 1

00 02 Postoje 2 MTrk parčeta u ovoj datoteci

E7 28 Svako uvećanje delta vremena predstavljeno je u milisekundama.

Posle svakog MThd parčeta, trebalo bi doći do MTrk parčeta. MTrk parče sadrži sve midi informacije plus opcione podatke za jednu traku. Opcioni podaci ne moraju biti usko vezani za sam MIDI formatom. Naslov MTrk parčeta počinje sa MTrk, pa zatim ide informacija o dužini (npr broj bajtova dužine). Dužina može biti različita za svaku traku.

Na MIDI traku se može gledati na isti način kao na traku u sekvenceru. Traka sadrži informacije o događajima (events). Npr, prvi događaj u traci može da bude zvuk note C. Svaki događaj ima informaciju o vremenu kada treba da se desi, i događaji su organizovani unutar parčića u redosledu u kome treba da se dese. Takođe MIDI format sadrži informaciju o instrumentu na kome neka traka izvodi muziku. Ova informacija se stavlja na početku MTrk parčeta.

FF 2F 00 označava kraj trake. To mora da bude poslednji događaj svakog Mtrk parčeta.

3.2.2. MusicXML

MusicXML format je osmišljen sa namerom deljenja zapisa muzičkih datoteka između aplikacija i radi čuvanja tih zapisa za korišćenje u budućnosti. Veliki broj aplikacija podržava ovaj format. Kao što su MP3 fajlovi postali sinonim za deljenje snimljene muzike, MusicXML datoteke su postale standard za deljenje interaktivnih zapisa muzike.

Pre MusicXML-a, jedini format za deljenje muzičkih notacija su bili MIDI datoteke. MIDI datoteke su dobar format za aplikacije kao što su sekvenceri, ali nije primenljiv u muzičkoj notaciji. MIDI ne pravi razliku između npr nota F# i Gb, ne zna šta je takt, ponavljanje, mera i ne poznaje mnoge druge pojmove muzičkog zapisa.

Sa pojavom interneta javila se i potreba za deljenjem notnih zapisa putem mreže. MusicXML je nastao prvenstveno od dva akademska muzička formata.

1. MuseData format, razvijen od strane *Valtera Hjuvleta (Waltera Hewlett)* na Stanford univerzitetu.
2. Humdrum format, razvijen od strane *Davida Hurona (David Hurona)* na Ohio State univerzitetu.

Prve beta verzije MusicXML su u osnovi bile XML datoteke za ažuriranje MuseData formata, sa par ključnih koncepata od Humdrum formata. Pošto su oba formata bila uglavnom korišćena za rad sa klasičnom i folk muzikom, MusicXML je proširen izvan tih granica da bi podržavao i savremenu muziku.

MusicXML je dizajniran sa namenom da reši problem međusobne nekompatibilnosti muzičkog softvera. Ima mnogo različitih aplikacija za obradu muzike i svaka ima neki svoj format. Da bi svaka aplikacija komunicirala jedna sa drugom, mora da postoji neki zajednički jezik.

Format je takode dizajniran sa namenom da prikaže kompleksne, strukturane podatke i notne zapise. XML parseri se lako mogu naći za bilo koju platformu tako da je obrada ovog formata laka. Ne mora da se razmišlja o sintaksi jezika, dovoljno je da se samo skoncentriše na semantiku.

Danas postoji mnogo programa za rad sa muzikom i muzičkim notnim zapisom od kojih su najpoznatiji Steinbergovi Cubase i Nuendo, Guitar Pro, MuseScore itd.

3.2.2.1. Struktura MusicXML

Recimo da treba da se izvede neko muzičko delo za koje je potrebno dve ili više osoba da bi je izveli. To delo se sastoji od više delova, svaki deo za jednog izvođača, i delo se izvodi u više različitih taktova. XML predstavlja podatke u hijerarhiji dok se notni zapis gleda u vidu strukture rešetke.

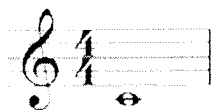
Iz tog razloga MusicXML podržava dva različita tipa obrade notnog zapisa:

- 1) Notni zapis prema vremenu (*<score-timewise>*). U ovo slučaju takt je primarni element, dok su delovi sadržani unutar njega
- 2) Notni zapis prema delu (instrumentu) (*<score-partwise>*). Muzički deo se tretira kao primarni element dok su taktovi sadržani unutar njega

Postojanje dve različite strukture MusicXML datoteke može da predstavlja problem prilikom konverzije iz jednog oblika u drugi, ali srećom na postoje XSLT dokumenti koji vrše konverziju iz jednog oblika u drugi ili obrnuto. Sama aplikacija treba da odluči koji će format koristiti i treba samo da proveri tip dolazećeg dokumenta. Ako nije odgovarajućeg tipa potrebno je samo primeniti XLST stil da bi se napravio novi MusicXML željenog formata.

Iz tog razloga, u daljem tekstu će biti opisan notni zapis prema delu. Zbog kompleksnosti samog formata koristi se jednostavan primer.

Primer. Nota C -Neka za primer treba da se zapiše jednostavna melodija. Neka je to nota C celog trajanja u taktu 4/4. Imamo samo jedan instrument koji izvodi tu melodiju.



Slika 3.2. – Nota C u taktu 4/4

Osnovna i najjednostavnija MusicXML datoteka bi izgledala ovako.

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE score-partwise PUBLIC "-//Recordare//DTD MusicXML 3.0 Partwise//EN"
"http://www.musicxml.org/dtds/partwise.dtd">
<score-partwise version="3.0">
<part-list>
  <part-name>Melodija</part-name>
<score-instrument id="P1-I1">
  <instrument-name>Acoustic Grand Piano</instrument-name>
</score-instrument>
<midi-instrument id="P1-I1">
<midi-channel>1</midi-channel>
<midi-program>0</midi-program>
</midi-instrument>
</part-list>
<part id="P1">
<measure number="1">
  <attributes>
    <divisions>1</divisions>
    <key>
      <fifths>0</fifths>
    </key>
    <time>
      <beats>4</beats>
      <beat-type>4</beat-type>
    </time>
    <clef>
      <sign>G</sign>
      <line>2</line>
    </clef>
  </attributes>
  <note>
    <pitch>
      <step>C</step>
      <octave>4</octave>
    </pitch>
    <duration>4</duration>
    <type>whole</type>
  </note>
</measure>
</part>
</score-partwise>

```

Slika 3.3. – Primer zapisa u MusicXMLu

U nastavku će biti detaljnije analiziran primer na slici 3.3.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
```

Ovo je XML deklaracija koja je potrebna za sve XML dokumente. Navedeno je da se koriste karakteri iz UTF-8 kodiranja. Atribut *standalone* je postavljen na *no* i to znači da definiše dokument sa eksternom definicijom na neki drugi dokument.

```
<!DOCTYPE score-partwise PUBLIC "-//Recordare//DTD MusicXML 3.0
Partwise//EN" "http://www.musicxml.org/dtds/partwise.dtd">
```

Ovde se navodi da koristimo MusicXML i to partitivni zapis što znači da su taktovi neke melodije opisani uz pomoć delova ili instrumenata. URL deklaracija je samo radi reference. Većina aplikacija koje čitaju MusicXML datoteke žele da instaliraju lokalnu kopiju MusicXML DTD (definicije tipa dokumenta) na korisnikovom računaru.

```
<score-partwise version="3.0">
```

Ovaj element govori da je kompozicija načinjena od delova, gde se svaki deo sastoji od taktova. Takođe navodi se verzija MusicXML formata koja se koristi što je u ovom slučaju verzija 3.0.

```
<part-list>
  <part-name>Melodija</part-name>
  <score-instrument id="P1-I1">
    <instrument-name>Acoustic Grand Piano</instrument-name>
  </score-instrument>
  <midi-instrument id="P1-I1">
    <midi-channel>1</midi-channel>
    <midi-program>0</midi-program>
  </midi-instrument>
</part-list>
```

Part-list predstavlja listu delova (instrumenata) koji učestvuju prilikom izvođenja neke muzičke kompozicije. Ovaj segment mora da se navede za svaku MusicXML datoteku. Pre ove deklaracije moguće je i navesti dodatne metapodatke vezane za samo ime kompozicije koje se izvodi, kompozitoru, program koji je generisao datoteku, datum kreiranja i slično. Koristimo deklaraciju *<identification>* da bi definisali ove podatke.

Part-name predstavlja sam naziv koji želimo da dodelimo samom delu.

Score-instrument deklaracija se koristi za informacije vezane za sam instrument koji izvodi melodiju. U ovom slučaju imamo samo definisan naziv instrumenta akustični koncertni klavir (Acoustic Grand Piano).

Opciono je dodati deklaraciju *midi-instrument*. Ona je potrebna da bi se navele informacije koje *generalna MIDI (General MIDI)* biblioteka može da interpretira. Mora da se navede redni broj kanala koji se koristi (podržano je maksimalno 16 različitih kanala) i redni broj tipa instrumenta iz te biblioteke. Podržano je 128 različitih tipova instrumenata u toj biblioteci.

Posle ovog uvodnog zaglavlja sledi opis svakog dela muzike zasebno, i to uz pomoć deklaracije *<part>*. Svaki deo se sastoji od liste taktova koji se nalaze unutar *<measure>* deklaracije. Takt mora da sadrži i redni broj takta kao atribut sa oznakom *number*.

```
<part id="ID">
  <measure number="Broj">
    <note>
      ...
```

```

    </note>
    ...
  </measure>
  ...
</part>
...

```

Svaki takt se sastoji od njegovog opisa i ti podaci se nalaze se pod *<attributes>* etiketom. Tu se nalaze informacije o vremenu takta definisana pod *time* etiketom (u ovom primeru vremensko trajanje takta je $\frac{4}{4}$ i navodi se vrednost imenioca i delioca zasebno), koliko povelica ili snizilica ima u skali (*key* etiketa), tipu violinskog ključa (*clef* etiketa) i vremenskoj podeli u taktu (etiketa *division*). Bitno je napomenuti za *division* etiketu da je to celobrojna vrednost koja predstavlja broj podela po noti koja ima dužinu trajanja $\frac{1}{4}$. Tako je na primer u ovom slučaju ukupna dužina trajanja takta 4, zato što takt $\frac{4}{4}$ može da sadrži u sebi 4 note sa dužinom trajanja $\frac{1}{4}$ sa 1 podelom po toj dužini trajanja. Treba imati na umu ukupan broj podela jednog takta zato što je taj broj usko vezan sa dužinom trajanja same note u taktu.

Posle opisa samog takta sledi lista nota koji opisuju taj takt, i to se može naći pod etiketom *<note>*. Tu se nalaze informacije visini note (*pitch* etiketa), koja se sastoji od stepena u lestvici (*step*) i oktave kojoj pripada (*octave*). Takođe, tu se definiše i celobrojna vrednost trajanje same note. U prethodnom primeru imamo da je nota C ima dužinu trajanja cele note, i da je broj podela po noti dužine trajanja $\frac{1}{4}$ u taktu 4 i zbog toga je vrednost dužine trajanja same note 4. Ako želi da se prikaže nota produženog trajanja tj. nota sa tačkom, dovoljno je samo preračunati tačno trajanje note po podelama. Ako se na primer koristi 4 podele po noti trajanja $\frac{1}{4}$, onda notu sa trajanjem $\frac{1}{4}$ sa tačkom možemo opisati kao notu čija je dužina trajanja $4+2=6$.

Prilikom definisanja akorda potrebno je dodati praznu etiketu *chord* prilikom opisa same note. Mora da se opiše svaka nota od akorda zasebno. Bitno je napomenuti da se etiketa *chord* ne stavlja na prvu notu u akordu, ali da se na svaku sledeću notu postavlja.

```

<part id="P1">
  <measure number="1">
    <attributes>
      <divisions>1</divisions>
      <key>
        <fifths>0</fifths>
      </key>
      <time>
        <beats>4</beats>
        <beat-type>4</beat-type>
      </time>
      <clef>
        <sign>G</sign>
        <line>2</line>
      </clef>
    </attributes>
    <note>
      <pitch>
        <step>C</step>
        <octave>4</octave>

```



```

        </pitch>
        <duration>4</duration>
        <type>whole</type>
    </note>
</measure>
</part>

```

Primer. Na kraj sela žuta kuća

Zbog obima samog MusicXML formata neke kompozicije uzeta su prva tri takta poznate dečije pesmice „Na kraj sela žuta kuća“ sa pratnjom akorda u pozadini.

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE score-partwise PUBLIC "-//Recordare//DTD MusicXML 1.0 Partwise//EN"
"/musicxml/partwise.dtd">
<score-partwise>
<identification>
<encoding>
    <software>Melody Generator</software>
    <encoding-description>MusicXML 3.0</encoding-description>
</encoding>
</identification>
<part-list>
<score-part id="P1">
<part-name>Melodija</part-name>
    <score-instrument id="P1-I1">
    <instrument-name>Acoustic Grand Piano</instrument-name>
    </score-instrument>
    <midi-instrument id="P1-I1">
    <midi-channel>1</midi-channel>
    <midi-program>0</midi-program>
</midi-instrument>
</score-part>
<score-part id="P2">
<part-name>Pratnja</part-name>
    <score-instrument id="P2-I2">
    <instrument-name>Acoustic Grand Piano</instrument-name>
    </score-instrument>
    <midi-instrument id="P2-I2">
    <midi-channel>3</midi-channel>
    <midi-program>0</midi-program>
</midi-instrument>
</score-part>
</part-list>
<part id="P1">
<measure number="1">
<attributes>
    <divisions>1</divisions>
    <key>
        <fifths>0</fifths>
        <mode>major</mode>
    </key>
    <time>
        <beats>4</beats>
        <beat-type>4</beat-type>
    </time>
</attributes>

```

3 Formati

```
<note><pitch><step>C</step><octave>4</octave></pitch><duration>1</duration></note>
<note><pitch><step>D</step><octave>4</octave></pitch><duration>1</duration></note>
<note><pitch><step>E</step><octave>4</octave></pitch><duration>1</duration></note>
<note><pitch><step>E</step><octave>4</octave></pitch><duration>1</duration></note>
</measure>
<measure number="2">
  <note><pitch><step>C</step><octave>4</octave></pitch><duration>1</duration></note>
  <note><pitch><step>D</step><octave>4</octave></pitch><duration>1</duration></note>
  <note><pitch><step>E</step><octave>4</octave></pitch><duration>1</duration></note>
  <note><pitch><step>E</step><octave>4</octave></pitch><duration>1</duration></note>
</measure>
<measure number="3">
  <note><pitch><step>C</step><octave>4</octave></pitch><duration>1</duration></note>
  <note><pitch><step>D</step><octave>4</octave></pitch><duration>1</duration></note>
  <note><pitch><step>E</step><octave>4</octave></pitch><duration>2</duration></note>
</measure>
</part>
<part id="P2">
  <measure number="1">
    <attributes>
      <divisions>1</divisions>
      <key>
        <fifths>0</fifths>
        <mode>major</mode>
      </key>
      <time>
        <beats>4</beats>
        <beat-type>4</beat-type>
      </time>
    </attributes>
    <note><pitch><step>C</step><octave>4</octave></pitch><duration>4</duration></note>
    <note>
      <chord></chord>
      <pitch><step>E</step><octave>4</octave></pitch><duration>4</duration>
    </note>
    <note>
      <chord></chord>
      <pitch><step>G</step><octave>4</octave></pitch><duration>4</duration>
    </note>
  </measure>
  <measure number="2">
    <note><pitch><step>C</step><octave>4</octave></pitch><duration>4</duration></note>
    <note>
      <chord></chord>
      <pitch><step>E</step><octave>4</octave></pitch><duration>4</duration></note>
    <note>
      <chord></chord>
      <pitch><step>G</step><octave>4</octave></pitch><duration>4</duration>
    </note>
  </measure>
  <measure number="3">
    <note><pitch><step>C</step><octave>4</octave></pitch><duration>4</duration></note>
    <note>
      <chord></chord>
      <pitch><step>E</step><octave>4</octave></pitch><duration>4</duration>
    </note>
    <note>
      <chord></chord>
      <pitch><step>G</step><octave>4</octave></pitch><duration>4</duration>
    </note>
  </measure>
</part>
```

```
</measure>  
</part>  
</score-partwise>
```

4 Tehnike algoritamskog komponovanja

Algoritamsko komponovanje muzike je tehnika korišćenja algoritama da bi se komponovala muzika. Algoritmi (ili u najmanju ruku, formalni skupovi pravila) su se koristili za kompoziciju muzike vekovima. Termin "Algoritamsko komponovanje" je uobičajeno rezervisan za korišćenje niza formalnih procedura koje za zadatak imaju stvaranje muzike različitih oblika bez ljudske intervencije. Ovaj postupak je potpomognut procedurama koje su zasnovane na korišćenju verovatnoće i pomoći računara. Postoji razlika između programa za komponovanje muzike koji koriste nedeterminističke, neodređene (na primer stohastičke) procedure i one koje koriste skup fiksiranih pravila na osnovu kojih se kreira muzika. Dok se prvi uglavnom koristi da bi učili iz već postojećih muzičkih dela i na osnovu toga generisali izlazni rezultat drugi imaju za cilj da grublje opišu pravila uz pomoć kojih je moguće komponovati muziku bez ljudskog uticaja.

Mnogi kompozitori kao kreativnu inspiraciju za njihovu muziku koriste algoritme koji možda nemaju neposrednog muzičkog značaja. Algoritmi kao što su fraktali, L-sistemi, statistički modeli čak i proizvoljni podaci (GIS koordinate, merenja magnetnog polja) se primenjuju u muzičkoj interpretaciji. Uspeh ili neuspeh ovih procedura kao izvora "dobre" muzike uglavnom zavisi od procene ljudskog kompozitora muzike da bi se ne-muzički podaci preveli u niz muzičkih podataka [18].

4.1. Istorija algoritamskog komponovanja

Istorija algoritamske kompozicije se deli na dva razdoblja:

- a) pre-računarski period
- b) komponovanje uz pomoć računara

4.2. Pre-računarski period

Ideja korišćenja formalnih instrukcija i procesa za kreiranje muzike datiraju još iz perioda antičke Grčke. Pitagora je verovao da postoji direktna veza između zakona prirode i harmonije zvuka izraženoj u muzici. Reč muzika je imala šire značenje antičkim Grcima nego što je to danas. U učenju Pitagore i njegovih sledbenika, muzika je bila nerazdvojna od brojeva, za koje se mislilo da su ključ celog duhovnog i fizičkog univerzuma. Tako se za sistem muzičkih nota i ritmova uređenih po brojevima smatralo da predstavljaju harmoniju kosmosa i da je kosmos odgovarao na taj sistem.

Tako, teoretske primene brojeva i raznih matematičkih svojstava su proistekli iz prirode i to su formalizmi ili "algoritmi" sa kojima su stari grčki muzičari konstruisali njihove muzičke sisteme. Ptolomej i Platon su takođe pisali o ovoj praksi.

Ti antički grčki "formalizmi" su ukorenjeni u teoriji, međutim njihova striktna primena na muzičko izvođenje je pod znakom pitanja zato što je grčka muzika uvek bila neka vrsta improvizacije. Te grčke matematičke pretpostavke su sigurno kreirale muzički sistem, intervale i modele kojima muzičar upravlja. Iako te pretpostavke usmeravaju i utiču na muzičara u procesu izvođenja, ne mora da znači da je sam muzičar uklonjen iz procesa odlučivanja. Možda antička grčka muzika nije koristila "algoritamsko komponovanje" ali ima velikog uticaja na samu oblast.

Dodatni nivo apstrakcije će kasnije biti postignut nastajanjem “kanonskih” kompozicija u drugoj polovini petnaestog veka. Preovlađujući metod je bio da se komponuje jedna melodija za vokal i da se daju instrukcije pevačima da izvedu dodatne vokale iz te melodije. Instrukcija ili pravilo po kojima su dalji delovi izvedeni nazivaju se kanoni, što u prevodu znači “pravila” ili “zakoni”. Ta “pravila” imitiranja i manipulacije su itekako “algoritam” prema kojem izvođači izvode muziku. U tom slučaju, što je u suprotnosti sa antičkim Grcima, vidimo eliminaciju kompozitora iz velikog dela kompozicionog procesa: kompozitor samo izmisli jezgro muzike – jednu melodiju ili sekciju – iz koga se cela kompozicija automatski konstruiše.

Mocart je takođe, koristio automatsku kompozicionu tehniku u njegovoj kompoziciji “*Muzika kocke*” (*Musikalisches Würfelspiel*), muzičkoj igri koja je uključivala spajanje različitih malih muzičkih fragmenata i spajajući ih uz pomoć slučajnosti tj. bacanjem kocke i time kreira novu kompoziciju iz tih slučajno odabranih delova. Ovo je veoma jednostavna forma “algoritamske” kompozicije koja ostavlja kreativne odluke u rukama šanse, puštajući da bacanje kocke odluči koje note će se koristiti.

Takođe postoje moderniji primeri algoritamske kompozicije bez korišćenja računara. Kompozitor Džon Kejdž je, na primer, kao i Mocart, koristio slučajnost u mnogim svojim kompozicijama, kao što je to na primer delo “*Reunion*” (1968. godina). Ovo delo je izvedeno igrajući šah na tabli sa foto-receptorima. Svaki potez igrača bi emitovao različite zvukove, i time bi svaka partija šaha proizvela drugačije muzičko delo. Kejdž je takođe za proces komponovanja koristio prirodne fenomene. Na primer, u njegovom drugom delu “*Atlas Eklipticalis*” (*Atlas Eclipticalis*, 1961. godina), iskoristio je list notnog papira postavljen preko astronomskog grafikona. U takvoj postavci same pozicije zvezda na listu notnog papira su predstavljene notama i tim procesom je povezo proces komponovanja sa slučajnošću [19].

U periodu posle drugog svetskog rata kreiran je metod 12-tonova (takođe poznat kao Serijalizam) koji je pokušao u potpunosti da kontroliše sve parametre muzike i da ostvari i apstrahuje kompozicioni proces što je više moguće. Austrijski kompozitor *Arnold Šoenberg* (*Arnold Schoenberg*) je prvi zagovornik i tvorac ovog metoda dvadesetih godina devetnaestog veka. Odluke oko svega, od nota do ritmova za dinamičko markiranje su često predmet pre-komponovanih “serija” i “matrica” vrednosti, koje za efekat imaju “automatizaciju” mnogih parametara. Ti parametri su odlučujući za redosled tonova u toku izvođenja dela. Ove serije i matrice su onda “algoritmi” koji su nasledili ljudski kreativan proces. Serijalizam sa time se može smatrati kao “algoritamska” ili “automatizovana” kompozicija u čistom smislu, pogotovu kada se teži da se integriše što više muzičkih parametara. Na primer, klavirska etida *Olivier Medisana* (*Olivier Meddisaen*, 1949. godina) “*Mode de valeurs et d'insensites*”, je imala 36 visina tonova, gde je svakoj visini tona dodeljena specifična ritmička, dinamička, registarska i napadna karakteristika koja se koristila u kompoziciji [19].

4.2. Komponovanje uz pomoć računara

Računari su uveli veliki broj novih mogućnosti koje se koriste za algoritamsku kompoziciju. *Ada Lovelejs* (*Ada Lovelace*), koja je učestvovala u osmišljavanju prve “računarske mašine”, prethodnika modernih računara, je u vezi sa mogućnošću automatizovane kompozicije u devetnaestom veku rekla [18]:

“Pod pretpostavkom, na primer, da su fundamentalni odnosi visine tona u znakovima harmonije i muzičke kompozicije bili prihvatljivi za takvu ekspresiju i adaptaciju, mašina bi mogla komponovati detaljna i naučna dela muzike do bilo kog stepena kompleksnosti ili obima”.

I tako se desilo, kao što je i Ada predvidela, da će računari približiti naučnike i kompozitore da bi kreirali “detaljna” dela muzike iz novih algoritamskih programerskih metoda.

Najraniji primer računarski generisane kompozicije je izražen u projektu *Ležaren Hillera* (*Lejaren Hiller*) i *Lenarda Isaksona* (*Leonard Isaacson*) na univerzitetu Ilinoisa 1955/56. godine. Kompozicija “*Illiad Suite*” (1957. godina) je nastala tako što je delo komponovano od strane računara i onda transponovano u tradicionalnu muzičku notaciju za izvođenje uz pomoć gudačkog kvarteta. Hiller i Isakson su generisali određene “sirove podatke” uz pomoć računara i onda su modifikovali te materijale prema raznim funkcijama i odabrali najbolji rezultat tih modifikacija prema određenim pravilima. Ta paradigma je kasnije primenjena na MUSICCOMP, jedan od prvih računarskih sistema za automatizovano komponovanje, implementiran u kasnim 1950.-tim i ranim 1960.-tim godinama od strane Hillera i *Roberta Bejkera* (*Robert Baker*). Ta ideja pravljenja malih, dobro definisanih kompozicionalnih funkcija i njihovo spajanje mogu da se pokažu efikasnim i da dozvole sistemu stepen fleksibilnosti i generalnosti, koji čine ovaj pristup popularnim u mnogim algoritamskim muzičkim kompozicijama čak i danas [17,19].

Još jedna od pionirskih tehnika u algoritamskoj kompoziciji potiče od *Ianisa Ksenakisa* (*Iannis Xenakis*), koji je napravio program koji kreira podatke za njegove “stohastičke” kompozicije, koje je u velikoj meri opisao u knjizi “*Formalized Music*” (1963. godina). Ksenakis je koristio računar da bi izračunao verovatnoće koje bi pomogle prilikom komponovanja. Primeri takvih kompozicija su *Morsima-Amorsima* (1962. godina) i *Atrees* (1962. godina). Program bi “dedukovao” note iz “liste gustine nota i težine verovatnoća, ostavljajući specifične odluke generatoru slučajnih brojeva”. Kao i u primeru *Illiad Suite*, kompozicije dobijene ovim metodom su izvođene uz pomoć živih izvođača na tradicionalnim instrumentima. Rad Hillera i Isaksona su pokušale da simuliraju proces komponovanja u potpunosti, prepuštajući kreativne odluke računaru [17,19].

Metodologije koje postoje u računarski generisanoj algoritamskoj muzičkoj kompoziciji su:

- 1) *Stohastički sistemi*
- 2) *Sistemi bazirani na pravilima*
- 3) *Sistemi bazirani na veštačkoj inteligenciji*

1) Stohastički sistemi

Stohastički pristupi su najjednostavniji. Oni uključuju slučajnost i mogu da budu prosti kao što je generisanje slučajnog niza nota (kao što je to u slučaju Mocartove igre sa kockom i u radovima Džona Kejdža). Takođe mogu biti kompleksniji kroz veliku količinu konceptualnih kompleksnosti koje mogu da se uvedu u računicu uz pomoć računara, statističke teorije i/ili Markovljevih lanaca. Drugačije gledište na korišćenje neočekivanosti je primena teorije haosa na algoritamsku kompoziciju. Ove aplikacije nameću razne nelinearne dinamičke jednačine kao što su fraktali koji prenose različitu muzičku informaciju.

Tokom sedamdesetih i osamdesetih godina, ponašanje sistema nelinearnih dinamičkih jednačina iterativnim pristupom je izazvalo interesovanje za njihovu primenu u algoritmima za generisanje nota. Uz pomoć ovih sistema jednačina mogu se opisati razne pojave u prirodi. Te pojave su vreme, mešanje tečnosti, fenomen turbulencije, ciklovi populacije, kucanje ljudskog srca, i vremenski raspon između kišnih kapi iz česme i tako dalje.

Ovo veliko i matematički kompleksno polje algoritamske kompozicije je dobra osnova za detaljniju analizu.

2) Sistemi bazirani na pravilima

Drugi pristup algoritamske kompozicije je korišćenje računarskih programa koji koriste sisteme bazirane na pravilima. Ta pravila su koraci koji se obično konstruišu na takav način da rezultat tih koraka uvek vodi ka sledećem koraku. Ne-računarske paralele algoritamskim kompozicijama baziranim na pravilima su već bile spomenute sa primerom kanona, metodom 12 tonova i integralnog serijalizma. Pravila konstruišu "gramatiku" koju kompozicioni proces mora da poštuje. Kao što je Hillerov *MUSICOMP*, gramatika ima formu računarskog programa u tom sistemu podrutina, i često uključuje baze podataka različitih pravila koja su ili prikupljena iz kompozicionih tehnika iz prošlosti ili tek otkrivena. Jedan primer algoritamskog komponovanja baziranog na sistemu pravila je *Vilijam Šotstaetov (William Shottstaedt)* program automatskih kontrapunktova koji piše muziku baziranu na pravilima *Johan Jozef Fuks (Johann Joseph Fux)* koja se nalaze u njegovoj knjizi uputstava kontrapunkta iz ranog osamnaestog veka namenjena mladim kompozitorima da bi kreirali striktno kontrolisani polifonski stil Palestine (1525.-1594. godina) [17,19].

Program je napravljen sa skoro 75 pravila, kao što su "Paralelne kvinte nisu dozvoljene" ili "izbeći sve tritone blizu kadence u lidijanskoj skali". Šotstaed je dodelio niz kaznenih poena koje će da opovrgnu određena pravila. Te kazne su težinske i bazirane su na činjenici da Fuks napominje da postoje neka pravila koja nikad ne smeju da se prekrše, dok se drugih ne mora pridržavati slepo. Kako se kazne akumuliraju, program napušta njegovu trenutnu granu pravila i uz pomoć povratne sprege traži novo rešenje.

Drugi primer je *Kemal Ebcioğluov (Kemal Ebcioğlu)* automatski sistem nazvan CHORAL koji generiše četvorodelne horove u stilu Johan Sebastian Baha i sadrži preko 350 različitih pravila [19].

3) Sistemi bazirani na veštačkoj inteligenciji

Poslednji jedinstveni prilaz algoritamskoj kompoziciji korišćenjem računara i on se zasniva na primeni tehnika veštačke inteligencije. Takvi sistemi su slični sistemima baziranim na pravilima u tom smislu da su to programi, ili sistemi programa, koji se baziraju na predefinisanoj gramatici. Međutim, razlika u tome da sistemi veštačke inteligencije imaju mogućnost definisanja sopstvene gramatike ili prostije rečeno, imaju kapacitet da "nauče". Primer je *Dejvid Koupov (David Cope)* sistem nazvan *Experiments in Musical Intelligence (EMI)*. Kao i prethodni primer CHORAL, EMI je baziran na velikoj bazi podataka opisa stila, ili pravila, različitih kompozicionih strategija. Sa tim što, EMI ima svoj kapacitet da kreira njegovu gramatiku i bazu podataka pravila, koji sam računar zaključuje na osnovu nekoliko nota od specifičnog dela kompozitora koji su ulaz za njega. EMI se koristio za automatsko komponovanje

muzike koje uključuje već donekle uspešne stilove Baha, Mocarta, Bartoka, Bramsa, Džoplina i mnogih drugih [19].

Još jedna zanimljiva grana tehnika veštačke inteligencije je *genetsko programiranje*. Pored baziranja gramatike na ulazu u vidu nota kao što je to spomenuto u slučaju EMI, genetsko programiranje generiše svoj sopstveni muzički materijal takođe u formi sopstvene gramatike. Kompozitor pored toga mora da isprogramira *funkciju prilagođavanja* i onda preslušava razne automatski generisane izlaze na različitim nivoima procesiranja da bi odlučio koji su prikladni za konačni izlaz. Funkcija prilagođavanja predstavlja meru na osnovu koje se biraju najbolje ocenjeni izlazi. U takvom procesu kompozitor ima poslednju reč, tj. koji izlaz da odbaci a koji da sačuva.

Genetsko programiranje je metod koji u suštini koristi proces veštačkog simuliranja procesa prirodne selekcije sa namerom da se evoluiraju prosti računarski programi. Da bi se izveo ovaj proces, koristi se mali niz funkcija i terminala, ili konstanti, da bi mogao da se opiše domen u kome želimo da evoluirani program radi. U prvim iteracijama se postavljaju slučajno generisani podaci definisanog domena. Onda se broj mera pogodnosti odlučuje pri svakoj iteraciji. Ovaj broj opisuje koliko se dobro program pokazuje u datom domenu problema. Pošto su prve iteracije slučajne, njihove performanse su veoma slabe ali, posle svake iteracije se dobija malo bolji rezultat nego prethodni. Ti rezultati se uzimaju u parovima, proporcionalno njihovim merama pogodnosti, i onda se kreira nova populacija programa od tih jedinki. Operacije koje se mogu primeniti u generisanju novih populacija uključuju reprodukciju, ukrštanje, mutaciju, permutaciju i druge.

Kompozitor daje sistem sa bibliotekom funkcija ili podrutina, koje mogu da rade razne stvari na generisanom muzičkom materijalu, sa time da kompozitor nema definisan način na koji će te funkcije biti korišćene. On samo definiše šta je poželjno da bude izlaz za računar i za uzvrat pokušava da automatski postigne te rezultate koristeći snabdevene podrutine. Time, na ovu formu "algoritamskog komponovanja" korišćenjem veštačke inteligencije ili genetskog programiranja se može gledati kao na ekstreman slučaj.

Pored ova tri različita metoda algoritamskog komponovanja muzike uz pomoć računara takođe postoji dodatna razlika u tome kakav izlaz proizvode sami algoritamski kompozitori muzike. Neki sistemi daju samo notne informacije koje treba da budu realizovane od strane bilo kog akustičnog ili elektronskog instrumenta, kao što je već viđeno u ranijim slučajevima Hillera i Ksenakisa a takođe važi i u slučaju EMI kompozitora i u većini drugih spomenutih do sada. Drugi sistemi, ne kreiraju notni zapis i fokusiraju se na sintezi elektronskog zvuka i manipulaciji sa snimljenim zvucima, ili možda kao kombinacija ovih aktivnosti.

5 Predloženi algoritam i implementacija

U daljem radu je predlog jednog algoritma za generisanje melodije sa akcentom na melodije popularne moderne muzike dvadesetog veka.

Osobine ovog algoritma su:

1. Prema podeli algoritamskog komponovanja muzike ovaj algoritam spada u sisteme bazirane na pravilima.
2. Prema konstrukciji bazira se na vodopad modelu dizajna
3. Koristi generatore slučajnih brojeva
4. Pravila su konstruisana prema osnovama muzičke teorije.

Svaki put prilikom prolaska kroz algoritam pravi se niz odgovarajućih koraka. Svaki korak u algoritmu ima mogućnost grananja na jedan od više različitih koraka. Koji će korak biti sledeći prepuštamo generatoru slučajnih brojeva. Svakom mogućem koraku koji sledi dodelimo celobrojnu vrednosti od kojih će se slučajnim odabirom odabrati sledeći korak. Generatori slučajnih brojeva se takođe koriste i prilikom odabira nekog elementa iz liste tonova i dužina trajanja nota. Ovakva procedura obezbeđuje da se prilikom svakog prolaza kroz algoritam generiše jedinstvena slušljiva melodija.

Sam algoritam i sva pravila u njemu su kreirani prema autorovom subjektivnom osećaju za muziku i njegovom viđenju "dobre" slušljive melodije. Ograničenja algoritma su ta što se svaki takt generiše nezavisno jedan od drugog i potreban je dodatan trud da bi se osigurao koncept muzičke teme.

5.1. Vodopad model

Za ovaj algoritam najbolje se u praksi pokazao vodopad model dizajna. Postoje razni pristupi razvoju softvera, prikladno definisani i dizajnirani, koji se primenjuju prilikom razvoja samog programa. Ovi pristupi su poznati kao „*Model razvojnog procesa*“. Svaki model procesa prati određeni životni ciklus da bi osigurao uspeh procesa prilikom razvoja softvera.

Jedan od poznatijih pristupa koji se koristi prilikom razvoja softvera je „*Vodopad model*“. Vodopad model je među prvim modelima koji su uvedeni i široko prihvaćeni u softverskoj tehnici. U vodopad modelu, čitav proces je podeljen u odvojene faze. Ove faze se izvršavaju kaskadno jedna za drugom. Svaka faza mora da se završi pre nego što se pristupi sledećoj fazi. Zbog takvog postupka i dizajna model podseća na vodopad i sa time i nosi takvo ime.

5.2. Potrebne informacije za generisanje melodije

Da bi se opisala neka melodija potrebno je dosta informacija. Struktura je šira nego samo slučajno generisanje nota iz nekog skupa nota.

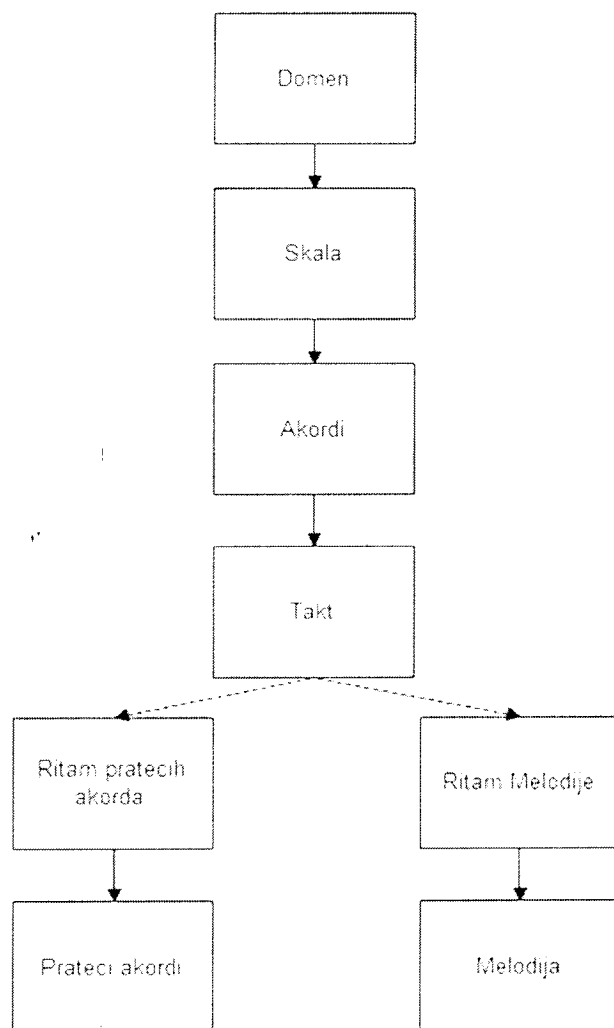
Potrebne informacije za generisanje melodije:

- 1) *Domen* – Ovo predstavlja skup svih tonova koji mogu da se izvedu.
- 2) *Skala* – Na osnovu domena kreiramo podskup domena koji nam predstavlja samu skalu. Elementi iz skupa skale se koriste da bi se kreirala sama melodija.
- 3) *Akordi*– Kada je poznata skala mogu da se kreiraju akordi koji su u okviru skale.
- 4) *Takt*– Da bi mogao da se generiše ritam neke melodije, potrebno je znati vremensko trajanje takta.
- 5) *Tempo* – Potrebno je da znamo brzinu izvođenja kompozicije.
- 6) *Broj taktova* – Kada generišemo neku kompoziciju potrebno je da znamo koliko različitih taktova treba da se generiše.
- 7) *Ritam* – Na osnovu ukupnog trajanja takta može se napraviti lista dužina trajanja nota koje učestvuju u melodiji tako da ta suma trajanja tih nota odgovara ukupnom trajanju takta.
- 8) *Melodija*– Od ritma može da se dobije informacija o tome koliko tačno nota je potrebno da se generiše i da se odrede iz domena odabrane skale.

Na osnovu ovih podataka lako se uočava da potrebne informacije mogu lako da se implementiraju u okviru vodopad modela.

5.3. Predloženi algoritam

Na slici 5.1. predstavljen je vodopad model samog algoritma. Malo je modifikovan i proširen u smislu da se grana ispod stavke *Takt*, ali prolaz kroz stablo je u skladu sa vodopad algoritmom.



Slika 5.1. Vodopad model predloženog algoritma

5.3.1. Ulazni podaci

Ulazni podaci koji su potrebni od korisnika su:

1. Skala koju želi da koristi. Bira se tip skale (durska ili molska) i bira se početni ton skale (kao što je C, C#, D, ...)
2. Tempo ili broj otkucaja po minuti (bpm). Ovo nam govori koliko se brzo ili sporo izvodi neka melodija
3. Takt melodije. Ova vrednost se predstavlja u obliku dve celobrojne vrednosti koje predstavljaju imenilac i delilac vremenskog trajanja takta
4. Broj različitih taktova koji želi da se generiše

Primer ulaznih podataka

1. Skala: C dur
2. Tempo: 120 otkucaja po minuti
3. Takt: 4 / 4
4. Broj taktova 4

5.3.2. Izlazni podaci

Kao rezultat dobija se slučajno generisana melodija. Ona se formira uz pomoć dva niza. Jedan niz predstavlja ritam same melodije dok drugi predstavlja tonove koji treba da se izvedu. Samo izvođenje melodije dobija se spajanjem informacija iz modula ritma i iz modula melodije. Modul ritma za melodiju govori kako se melodija izvodi ritmički, dok modul melodije govori koji se tonovi izvode.

Modul ritma				
Generisana melodija	$\frac{1\ 1\ 1\ 1\ 1}{8'4'4'4'8}$	$\frac{1\ 1\ 1\ 1\ 1\ 1}{8'4'8'8'8'4}$	$\frac{1\ 1\ 1\ 1\ 1}{4'4'4'8'8}$	$\frac{1\ 1\ 1\ 1\ 1}{8'4'4'8'4}$
Generisani akordi	4/4	4/4	4/4	4/4
Modul melodije				
Generisana melodija	C1,D1,E1, C1,C1	C1,D1,C1,H1,C1,E1	F1,G1,G1,F1,G1	C1,D1,C1,H0,C1
Generisani akordi	C dur	A mol	F dur	C dur

Slika 5.2. – Primer izlaza iz algoritma

5.3.3. Domen

Domen je predstavljen listom svih tonova koji mogu da se izvedu. Preporučeni broj oktava je 5 ili više iz razloga da se tonovi melodije i akorda koji se izvode ne bi preklapali. Tonovi akorda mogu ali ne moraju da budu u nižoj oktavi u odnosu na melodiju. Isto važi i u suprotnom slučaju.

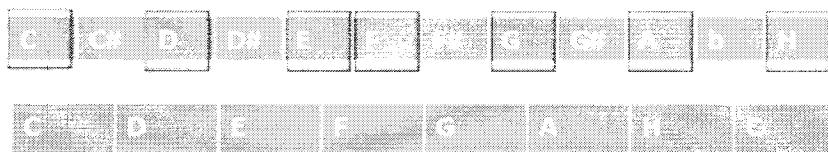
Pored toga, sama klasa domena sadrži metode koje na osnovu lista pozicija vraća tonove koji odgovaraju njima. Na slici 5.3. prikazan je domen.



Slika 5.3. – Domen skale

5.3.4. Skala

Kada je odabran početni ton i tip skale možemo kreirati skalu iz domena. Poznato je da se skala kreira na osnovu pravila razmaka stepena i polustepena između tonova. Na osnovu tih pravila jednostavno je kreirati skalu. Skala se kreira kao lista pozicija u okviru liste domena. Nije nam potrebna informacija o visini tona ili slično.

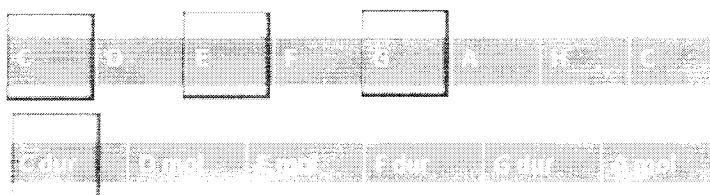


Slika 5.4. – Kreiranje Cdur skale

5.3.5. Akordi

Na osnovu poznate skale mogu da se kreiraju akordi koji su u okviru te same skale. Iz svakog tona skale se pravi akord koji se u većini slučajeva sastoji od 3 tona koji se izvode ili sviraju istovremeno. Ti tonovi su uvek u razmaku 2 elementa u odnosu na prethodni ton skale.

Konačno na osnovu lista pozicije tonova skale kreira se lista akorda.



Slika 5.5. – Kreiranje Cdur akorda

5.3.6. Takt

Takt čuva celobrojne vrednosti imenioca i delioca vremenskog trajanja takta, realnu vrednost koja se dobije na osnovu deljenja imenioca i delioca i tempu melodije koja treba da se generiše.

5.3.7. Ritam melodije

Prilikom generisanja ritma neke melodije potrebna je samo informacija koliko traje neki takt. Pošto je takt predstavljen kao razlomak, može se izračunati njegova realna vrednost koja će predstavljati ukupno vremensko trajanje takta. U suštini, problem ritma melodije se može svesti na obično ispitivanje sume realnih brojeva i njihovog ukupnog zbira. Realna vrednost ukupnog vremenskog trajanja takta predstavlja gornju granicu ukupnog zbira.

Potrebna je lista jedinstvenih sabiraka koji mogu da učestvuju u sumi. Na ovu listu se može gledati kao na domen mogućih realnih brojeva. Klasa ritma melodije pored te liste mora da sadrži i listu slučajno odabranih sabiraka iz tog domena. Suma svih elemenata u listi slučajno odabranih sabiraka moraju da odgovaraju trajanju takta. Pošto se melodija generiše takt po takt svaki takt će imati svoju listu slučajno odabranih sabiraka.

Neka je trajanje neke note predstavljeno realnim brojem, na primer ako se koristi nota trajanja $\frac{1}{4}$ to se može predstaviti sa vrednošću 0,25. Slična procedura se primeni i na samo vremensko trajanje takta, na primer takt vremenskog trajanja $\frac{4}{4}$ odgovara realna vrednost 1,0.

Generatorom slučajnih celobrojnih vrednosti biraju se indeksi iz niza domena sabiraka, smeštaju se u listu trenutno korišćenih sabiraka i računa se trenutna suma. Ta vrednost se upoređuje sa ukupnom sumom takta i ako je vrednost manja od sume bira se novi element iz niza domena i ponavlja se postupak.

Da bi se osigurao uvek dobar izbor sabiraka koji učestvuju prilikom kreiranja sume mora se uvesti mehanizam kontrole trenutne sume, tj. kontrole odabira elemenata iz liste sabiraka.

Postoji par scenarija koje treba izbeći kao što su:

- 1) Ukupna suma za takt je manja od moguće dužine trajanja note u listi sabiraka
- 2) Razlika između trenutne sume i ukupnog zbira ne može da ima u sebi notu dužeg trajanja

Zbog toga lista samih sabiraka u domenu mora da bude uređena u rastućem ili opadajućem poretku prema realnoj vrednosti. Čuvaju se podaci o trenutno minimalnom i maksimalnom indeksu u listi domena tj. o noti minimalnog i maksimalnog trajanja.

U prvom koraku se izvrši prvo filtriranje niza sabiraka u domenu prema ukupnoj sumi takta i ako je ukupna suma manja od note nekog određenog trajanja izbaciti tu notu iz daljeg razmatranja. Neka se uzme na primer takt koji ima vremensko trajanje $\frac{3}{4}$. Takav takt ne može da sadrži notu trajanja $\frac{4}{4}$ tako da se mora izbaciti iz domena.

Prilikom dodavanja svakog novog sabiraka, proverava se rastojanje trenutne sume i ukupne sume takta i shodno prema tome se koriguju indeksi niza domena sabiraka. Nije potrebno prolaziti kroz celu listu domena svaki put, dovoljno je smanjiti/povećati maksimalni ili minimalni indeks za 1 poziciju i proveriti da li ti elementi na tim pozicijama odgovaraju prethodnim kriterijumima, ako ne ponoviti postupak smanjivanja/povećanja indeksa, sve dok se ne postignu uslovi.

Pošto se radi sa sabiranjem realnih vrednosti, može doći do greške sabiranja. Dobra strana je ta da je sam skup sabiraka ograničen i uglavnom mali. Predlog je da se ne koriste note koje su manjeg trajanja od $\frac{1}{32}$ jer sve vrednosti manje od toga su isuviše kratkog trajanja za ljudsko uvo i zbog toga nemaju mnogo estetskog smisla. Kada se dužine trajanja nota u takvom skupu predstavljaju realnim

vrednostima dobijaju se brojevi koji su u opsegu brojeva 10^3 . Sabiranjem brojeva te preciznosti i testiranjem epsilon okolinē oko ukupne sume možemo uvek obezbediti korektno generisan ritam melodije.

Takođe, korisno je i ograničiti skup sabiraka domena u smislu da se ne koriste note svih mogućih trajanja. Posmatranjem muzičkih kompozicija može se uočiti da su vrednosti trajanja note u nekom taktu uglavnom bliske i nema mnogo oscilacija u smislu da posle note dugog trajanja sledi nota najkraćeg mogućeg trajanja.

Konačno, iz liste slučajno odabranih sabiraka može da se dobije informacija o dužinama nota koje učestvuju u melodiji i broj nota koje je potrebno generisati. Na slici 5.2. se može videti primer izlaza ritma melodija iz algoritma.

5.3.8. Ritam pratećih akorda

U većini slučajeva ritam pratećeg akorda zauzima celo trajanje takta. Trajanje pratećeg akorda dobija se iz vrednosti ukupne sume trajanja takta. Takođe, ritam akorda se generiše nezavisno od ritma melodije.

5.3.9. Melodija

Pošto je kreirana struktura ritma same melodije, potrebno je izgenerisati note melodije. U modulu melodije jedina informacija koja je potrebna je broj nota koji treba da se generiše. Ovaj modul nema predstavu o trajanju nota, skali kojoj pripada ili tempu. Ova klasa radi samo sa indeksima brojeva u opsegu od 0 do 15 tj. u okviru broja tonova koji su potrebni za dve oktave ili 0 do 7 u slučaju da se koristi jedna oktava. Ti indeksi se koriste kao redni broj pozicije tona u listi tonova same skale.

Posmatranjem melodija moderne popularne muzike, može se doći do zaključka da postoje neki šabloni prilikom izvođenja samih melodija. U većini različitih melodija, može se primetiti da većina tonova u njoj konvergira ka tonovima akorda predstavnika same skale. *Akord predstavnik skale* je akord koji se izvodi iz prvog tona skale kao što je na primer Cdur akord (C E G) pod Cdur skalom. U okviru samog modula Melodije to su indeksi na pozicijama 1,3, 5, 8, 10, 12 u okviru dve oktave za melodiju. Ako bi ovo bilo jedino pravilo melodija bi samo varirala oko tonova akorda predstavnika, a pritom je poznato da tonovi melodije mogu da imaju veće razmake između njih. Zbog toga potrebno je uvesti „skok“ sa trenutne pozicije na neku novu poziciju. Skokovi po bilo kojoj poziciji u nizu daju loše rezultate dok ograničavanje tih skokova na pozicije u okviru akorda predstavnika daju dobre rezultate.

Najveći problem predstavlja ton melodije koji se nalazi u okviru 7 ili 14 pozicije niza, na primeru Cdur skale bi to bio ton H. Ton na toj poziciji je problematičan zato što može da napravi kakofoniju u melodiji/akordu ili prostije rečeno zvuk koji ne prija ljudskom uvu. Ali opet ne treba ga u potpunosti odbaciti. Kada se posle tona H odmah predje na ton C ili obrnuto dobijaju se melodije koje su slušljive i imaju smisla.

Opseg indeksa koji se koristi prilikom generisanja melodija su podeljeni u dva niza celobrojnih vrednosti. Jedan predstavlja indekse koji predstavljaju tonove u akord predstavniku skale, dok drugi predstavlja sve ostale indekse. Pravi se niz logičkih konstanti koji predstavljaju *opis melodije* melodije

koja se generiše. On sadrži onoliko elemenata koliko nota treba da se generiše za melodiju. Kada je vrednost elementa niza *tačno(T)* onda to označava da se bira ton iz akorda predstavnika *a* u suprotnom znači da se biraju tonovi koji nisu iz tog akorda. Na slici 5.2. se može videti izlaz generisane melodije iz algoritma.

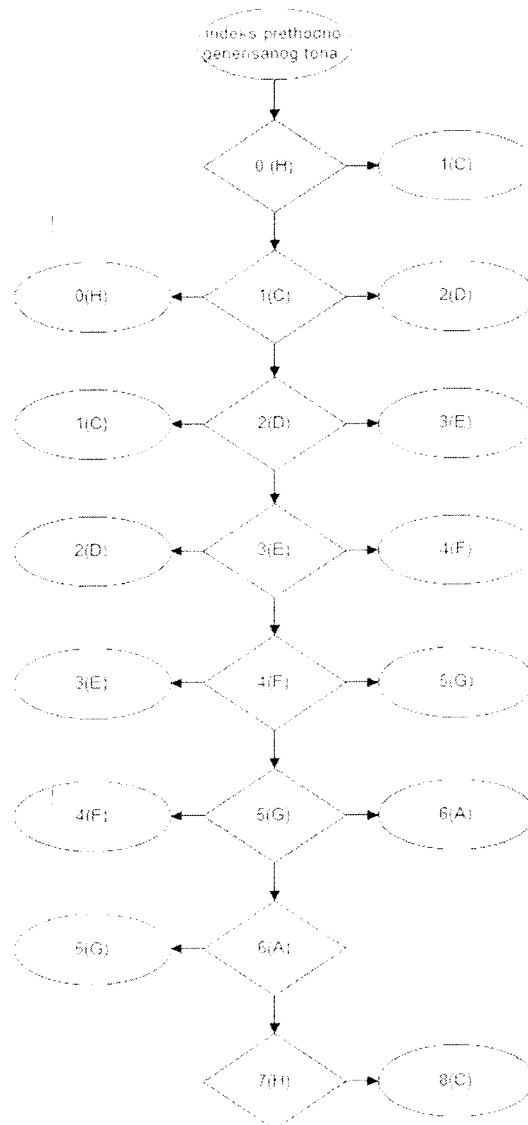
5.3.10. Opis melodije

Da bi se obezbedilo da melodija uvek teži ka notama akorda predstavnika skale uvodi se niz binarnih vrednosti koji će opisivati da li je ton u akordu ili van njega. Neka je na primer potrebno opisati strukturu melodije koja se sastoji od četiri tona. Od svih mogućih kombinacija interesuju nas samo varijante niža koje nemaju dva tona van akorda predstavnika zaredom. Generisanje svih mogućih kombinacija svaki put bi bio veliki i nepotreban posao, pogotovu zato što u jednom taktu broj tonova može biti veliki.

```
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
```

Slika 5.6. – Sve kombinacije opisa melodije za melodiju od 4 tona

Kada smo opisali tonove melodije koja se generiše potrebno je odabrati tonove koji je čine. Prema opisu melodije generatorom slučajnih brojeva odabiramo tonove iz odgovarajućeg skupa. Mada i u ovakvom slučaju potrebna su dodatna ograničenja. Kada se slučajno biraju tonovi iz akorda predstavnika uzimamo bilo koju vrednost. Ovo omogućuje da sami tonovi melodije ne budu previše blizu jedan drugom. Kada se dešavaju prelazi sa tona iz akorda predstavnika na ton van akorda primenjuju se pravila definisana na slici 5.7.

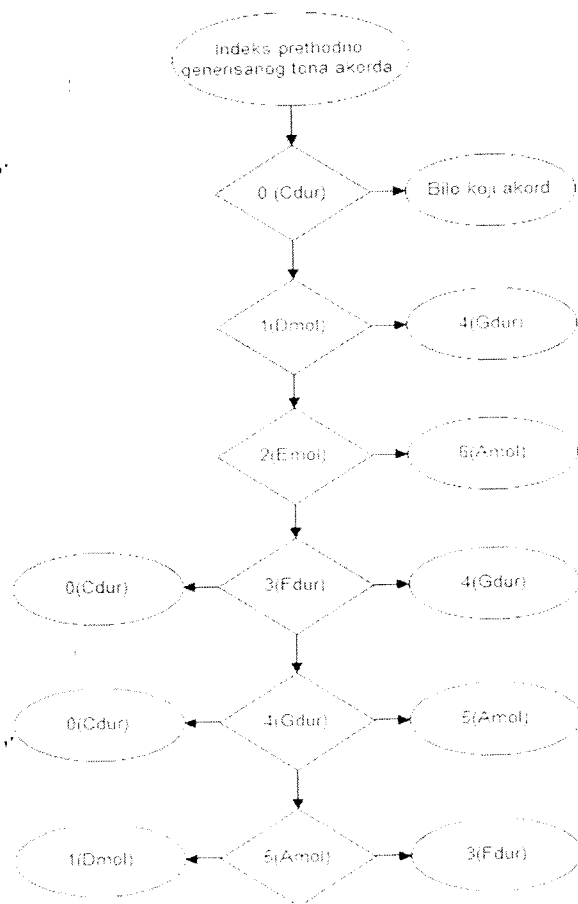


Slika 5.7.-Stablo pravila za generisanje sledećeg tona melodije

5.3.11. Akordi

Modul Akorda prilikom generisanja melodije je takođe nezavisan od skale, tempa i dužine trajanja nota. Generisanje akorda se vrši prema trenutnom taktu koji se obrađuje. Generiše se samo jedan akord po taktu na osnovu pravila diatoničke progresije akorda. Potrebno je samo znati koji je indeks prvog tona prethodno odabranog akorda i na osnovu jednostavnih pravila odabrati koji će akord biti sledeći uz pomoć slučajno odabrane celobrojne vrednosti.

Bitno je napomenuti da se sedmi akord u skali (poznatiji kao prekomerni akord) gotovo nikada ne koristi prilikom kreiranja neke kompozicije popularne muzike. On je neharmoničan akord ali ima i on svoje primene u apstraktnijoj vrsti muzike kao što je npr. džez muzika.



Slika 5.8. -Pravila diatoničke progresije akorda

5.3.12. Rezultat

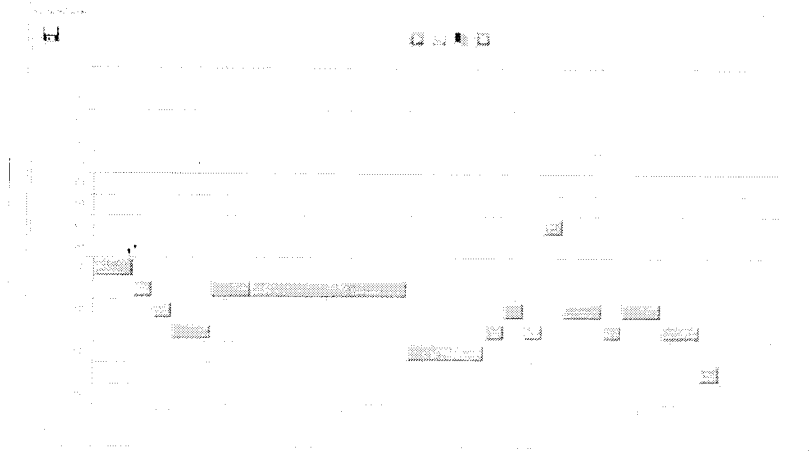
Konačno kada je dobijena finalna melodija tj. lista indeksa niza skale, treba da se identifikuju pozicije tonova u domenu i kao rezultat imamo melodiju. Takođe čuva se informacija o strukturi ritma i prilikom prikaza/izvođenja prate se obe liste istovremeno. Korišćenjem ove dve liste sa informacijama na ulazu može se konstruisati ulaz za MIDI sekvencer i time dobiti izvođenje generisane melodije.

5.3.13. Vremenska složenost algoritma

Pošto algoritam u svakom koraku ili modulu radi sa samo jednom listom njegova vremenska složenost nije velika. Najzahtevnija operacija u algoritmu je sortiranje liste koja se dešava prilikom izračunavanja ritma a poznato je da se sortiranje prosečno izvršava u vremenskoj složenosti $O(n \log n)$, gde je n broj elemenata niza. Mada, i sortiranje kao operacija se može izbeći ako su nam ulazni podaci uređeni u odgovarajućem poretku.

Što se tiče prostorne složenosti ona je malo zahtevnija, zato što svaki modul kao rezultat kreira sopstvenu listu elemenata sa kojom sledeći modul radi. Treba čuvati svaku listu dok traje proces kreiranja melodije.

Na samom prikazu melodije postoje dugmići koji služe za izvođenje i čuvanje same melodije. Čuvanje kreira datoteku koja je u MusicXML formatu. Pored toga, postoje opcije za kontrolu izvođenja generisane melodije i to su izvođenje melodije, ponavljanje melodije, izvođenje iste melodije sa drugačijom progresijom akorda i zaustavljanje izvođenja.



Slika 5.10. – Primer generisane melodije

5.5. Primeri rada programa i generisanih melodija

1. Primer kompozicije koja je sklopljena uz pomoć generisanih melodija se može čuti na sledećoj internet adresi: <http://soundcloud.com/markomatf/majormelodyrhythmtest2>
2. Video snimak okruženja i rada aplikacije razvijene u programskom jeziku C# se može videti na sledećoj internet adresi: http://youtu.be/kkfUXMZas_k
3. Testiranje generisanja melodija u aplikaciji sa fiksnim parametrima se mogu videti na sledećim internet adresama:
 - a. <http://youtu.be/xhxldP52HAg> - primer molske skale sa notama kraćeg trajanja
 - b. <http://youtu.be/OLpo1LgGl-o> – primer molske skale sa notama srednje dužine trajanja
 - c. <http://youtu.be/pwoZoRk7vQQ> - durska skala sa notama dužeg trajanja

6 Zaključak

Pre svega upoznali smo se sa osnovama muzičke teorije. To je bilo potrebno da bi se objasnila struktura zapisa same muzike i da bi se objasnila većina muzičkih pojmova. Da sama pravila zapisa nota i melodija nisu striktna možda ne bi mogla tako lako da se opišu sama pravila nekim algoritmom.

U ovom radu takođe su analizirani pristupi algoritamskog komponovanja muzike. Kroz istoriju algoritamskog komponovanja upoznali smo se metodologijom same oblasti a i interesantnim dešavanjima koja su definisala muziku onakvu kakvom je mi danas i poznajemo.

Analizirani su metodi zapisa zvuka na računaru. Objasnjeni su neki od poznatijih formata ali sa većom pažnjom na MusicXML format. Ovaj format je omogućio da se kreirane melodije mogu prenositi i koristiti sa drugim programima za obradu zvuka, notnog zapisa, komponovanja itd.

Razvijen je originalan algoritam zasnovan na skupu muzičkih pravila prema podeli same metodologije algoritamskog komponovanja. Proučena je vodopad struktura samog algoritma i objašnjena su pravila koja se koriste prilikom generisanja melodije. Zbog problema opisivanja same muzike i zbog subjektivnog doživljaja muzike kod svakog čoveka, većina zaključaka u ovome radu se bazira na autorovom ličnom doživljaju muzike i slušljive „dobre“ melodije.

Algoritam i aplikacija daju zadovoljavajuće rezultate. Generisane melodije mogu da odstupe od harmonije i zvuče ponekad čudno, ali to se dešava u retkim slučajevima. Odstupanje od dobre melodije se najčešće može desiti prilikom generisanja melodija iz harmonijske molske skale. Razlog tome je povišeni sedmi ton skale u odnosu na prirodnu molsku skalu. Poznato je od ranije da je sedmi ton problematičan u melodiji i akordu jer često može da stvori kakofoničan zvuk koji može da zvuči lošije. Kada se poviši taj ton u harmonijskoj molskoj skali on dodatno produbljuje kakofoniju, mada opet sa druge strane ne treba izbaciti ton u potpunosti zato što u većini slučajeva zvuči dobro.

Još uvek ima puno mesta gde se algoritam može proširiti i poboljšati. Nekoliko još uvek neistraženih tema su kreiranja celih pesama, kreiranje još boljih melodija, istraživanja drugih vrsta muzika, drugih muzičkih skala, žanrova i još mnogo toga. Prednost ovog algoritma je ta što je modularan, tako da se lako mogu dodavati nova pravila, nove mogućnosti i proširivati sama struktura algoritma.

Jedan od opštijih zaključaka koji može da se izvede je taj da se muzika u nekom prostijem obliku može opisati uz pomoć algoritma. Ovaj rad je pokušaj objašnjenja i opisa delića kompleksne oblasti poznate kao muzika. Spajanjem matematike, računarstva i muzike u jednu celinu predstavljao je, barem autoru, jedan zabavan i ne tako jednostavan zadatak.

7 Reference

1. Wikipedia, "Waterfall model",
http://en.wikipedia.org/wiki/Waterfall_model,
2. Buzzle, "The Waterfall Model Explained",
<http://www.buzzle.com/editorials/1-5-2005-63768.asp>
3. Karen Cuneo Ramirez, "Determining chord progressions in a song",
<http://youtu.be/fXIFmMDwc7E>
4. Wikipedia, "The MIDI File Format",
<http://midi.mathewvp.com/aboutMidi.htm>
5. Wikipedia, "Resource Interchange File Format",
http://en.wikipedia.org/wiki/Resource_Interchange_File_Format
6. Wikipedia, "MIDI",
<http://en.wikipedia.org/wiki/MIDI>
7. Univerzitet u Beogradu, Matematički fakultet, "Predstavljanje zvuka", Uvod u organizaciju računara
8. Recordare, "MusicXML",
<http://www.makemusic.com/musicxml>
9. Wikipedia, "General MIDI",
http://en.wikipedia.org/wiki/General_MIDI,
10. Wikipedia, "Pulse-Code modulation",
http://en.wikipedia.org/wiki/Pulse-code_modulation
11. Wikipedia, "Digital audio",
http://en.wikipedia.org/wiki/Digital_audio
12. Wikipedia, "Formati datoteka",
http://bs.wikipedia.org/wiki/Formati_datoteka
13. Wikipedia, "Music",
<http://en.wikipedia.org/wiki/Music>
14. Wikipedia, "Melody",
<http://en.wikipedia.org/wiki/Melody>
15. Zanimljiva muzika, "Kadence",
<http://www.zanimljivamuzika.com/wp-content/uploads/2008/03/kadence.pdf>
16. About, "Minor scales",
<http://musiced.about.com/od/lessonsandtips/qt/minosclaes.htm>
17. Adam Alpern, Hampshire College (1995), "Techniques for Algorithmic Composition of Music"
18. Wikipedia, "Algorithmic composition",
http://en.wikipedia.org/wiki/Algorithmic_composition
19. Stanford University, John A. Maurer, "A Brief History of Algorithmic Composition",
<https://ccrma.stanford.edu/~blackrse/algorithm.html>
20. Wikipedia, "Serialism",
<http://en.wikipedia.org/wiki/Serialism>
21. Britannica, "Serialism",
<http://www.britannica.com/EBchecked/topic/535601/serialism>
22. Dnevni list „Politika“, „Roditelji su, ipak, u pravu – muzika je sve glasnija“
23. Madarasz Sz. Rosalia, „Matematika i muzika“
24. M. Tajčević, "Osnovna teorija muzike"