

***Tematska izhodišča za izdelavo samostojnih nalog pri predmetu DPS  
(samostojno delo)***

V nadaljevanju ponujamo nekaj tematskih izhodišč za izdelavo samostojnih nalog (mini-seminarjev) pri predmetu DPS. Izhodišča so v pomoč, sicer z njimi niste omejeni – tematiko svoje samostojne naloge si lahko izberete sami. Naj vas le opozorimo, da korektno in dosledno navedete že narejena dela in vire, ki ste jih uporabili pri svojem delu in da se držite pravil. Z oddajo naloge jamčite svojo avtorstvo oziroma korektno navedbo virov in dela, ki ga niste opravili sami.

Ne obremenjujte se s tem, da morate izumiti nekaj novega in najboljšega. Dovolj je že, da na nekem problemu preizkusite kakšno svojo idejo, ki morebiti ni najbolj uspešna. Pomembno je, da njeno uspešnost znate ovrednotiti, jo izboljšati, ali pa vsaj analizirati njene slabosti.

Priporočamo tudi, da si navzkrižno s prijatelji in kolegi izmenjate osnutke in opravite vsaj enkrat tudi vlogo recenzenta. Z nasveti tako pomagajte kolegu do bolj razumljive in jasne vsebine, sebi pa do boljšega razumevanja tematike in dela v skupini.

## 1. Področje sinteze zvoka

Samostojna naloga s tega področja predstavlja kakršnokoli samostojno delo na področju, ki smo ga obdelali v dosedanjih vajah in praktičnih primerih. Namenjena je predvsem tistim, ki jih področje in tovrstno delo bolj zanima.

Svoje delo poskušajte čimbolj sistematično opisati: najprej sam problem, kratka umestitev problema v neko področje, potem obstoječe možnosti reševanja in morebitno lastno idejo, razmišljanje ali realizacijo oziroma rešitev. Komentirajte dobljene rezultate in ocenite možnosti nadaljnjih izboljšav. Navedite vse uporabljene vire. Naloga naj bo zasnovana tako, da se lahko samostojno objavi na spletu in predstavlja zaključeno celoto.

Če potrebujete kakršnokoli pomoč, se oglasite. Za lažje razmišljanje naj vas spomnim na že objavljene seminarje in zanimive rešitve vaših predhodnikov (<http://laps.fri.uni-lj.si/dps/dps-seminarji.htm>) in naštejemo nekaj izhodišč, izzivov oziroma idej:

- raziščite in implementirajte sintezo zvoka izbranega instrumenta ali kakršnegakoli drugega zvoka
- simulacija igranja akustične kitare (seminar kolega Kopina Aljoše)
- implementacija bralnika midi datotek, ki bi prebral midi datoteko in jo prepisal v seznam v zgornji obliki (primer implementacije: <http://www.kenschutte.com/midi/>, Midi2Scilab <http://sourceforge.net/projects/midi2scilab/> (seminar kolega Jekovca))
- izboljšanje sinteze obstoječih instrumentov :
  - Karplus Strong:
    - npr. manjše variranje frekvence osnovnega tona
    - implementacija neceloštevilčne zakasnitve
    - izboljšave modela
      - <http://www.music.mcgill.ca/~amburns/physique/introduction.html>
  - aditivni model klarineta:
    - podrobnejša analiza posnetkov klarinetov in dopolnitev osnovnega nabora komponent
    - iskanje drugih možnosti izboljšave aditivnega modela ali uporabe druge metode.
  - uporaba FM sinteze za druge harmonske in neharmonske zvoke (zvon, tolkala, ...)
  - preučitev uporabe drugih programskih orodij za delo z zvokom (npr. Ptolemy, PureData, Faust, SPEAR in podobni, zajemanje in obdelava zvoka s programskimi orodji npr. Java, C++ in ostali) in izdelava preprostih primerov

Ostali viri :

- Harmony Central:
  - [http://www.harmony-central.com/Synth/Articles/Physical\\_Modeling/](http://www.harmony-central.com/Synth/Articles/Physical_Modeling/)
- Analiza in izboljšave Karplus-Strong modela:
  - <http://www.music.mcgill.ca/~amburns/physique/introduction.html>
- Obširnejši gradivi na temo generiranja zvokov in ostalih zvočnih efektov:

- <http://ccrma.stanford.edu/~jos/Mohonk05/Mohonk05.html>
- <http://www-ccrma.stanford.edu/~jos/waveguide/>

## 1.1 Primer enotne samostojne naloge, ki smo jo reševali prejšnja leta:

### Implementacija sintetizatorja zvoka različnih instrumentov v Scilabu - Navodilo

Realizirati želimo funkcijo v Scilabu, ki bo glede na vhodne zapise posameznih melodij, sintetizirala končni zvok izbranega instrumenta. Pri tem lahko uporabite podatke v spodnjem okvirju.

Razlaga podatkov :

- music\_notes : vektor z zapisi not, kot znakovnimi nizi
- music\_notes\_freq : vektor z ustreznimi frekvencami osnovnih tonov not v music\_notes. Primer :
  - želimo dobiti frekvenco osnovnega tona A4 (rezultat je 440) :  
`note='A4';`  
`note_freq=music_notes_freq(find(music_notes == note));`
- solo: seznam not prvih treh taktov solo melodije »yesterday«.
- bas: seznam not prvih treh taktov spremljave melodije »yesterday«.
- podatki v seznamih solo in bas:
  - za vsako noto imamo podane 3 podatke:
    - oznaka note – znakovni niz ('A4'), ki je uporaben za določanje osnovne frekvence tona
    - začetek – relativna oznaka začetka posamezne note, kjer vrednosti  $0 \leq x \leq 1$  pomeni 1. takt,  $1 \leq x \leq 2$  drugi itd... (npr. 1.25 pomeni začetke note ob prvi četrtini 2. takta)
    - trajanje – relativna oznaka trajanja note (npr.  $\frac{1}{4}$  pomeni četrtno,  $\frac{1}{2}$  polovino itd...)
- določitev »tempa«: tempo oziroma trajanje taktov lahko določite po svojem občutku. Najenostavneje to naredimo tako, da posameznemu taktu določimo trajanje v številu vzorcev in s tem potem množimo relativne oznake začetka in trajanja posameznih not in na ta način dobimo začetek in konec posamezne note v številu vzorcev.

Napotki za implementacijo:

Na začetku določimo prazen vektor za izhodni signal  $y(n)$  in nato v for zanki pregledujete sezname not, sintetizirate zvoke za posamezne note in jih prištevate  $y(n)$ .

Posamezne elemente seznamov dobite enostavno: npr. solo(2)(1) je oznaka ["(1)"] 2.note v seznamu solo ["(2)"]....

