

Adaugarea sunetelor

Pentru a face jocul mai frumos, vom adauga si sunete (care sa apara, de exemplu in urma unui eveniment).

In fisierul de configurare pentru resourceManager vom avea si elementul <sounds>.

```
<sounds>
  <sound id="1">Resources/Sounds/hapciu.wav</sound>
  <!--...etc... -->
</sounds>
```

In cadrul acestui element vom defini elementele <sound> fiecare continand o cale catre un sunet.

De unde puteti downloada sunete:

<https://www.freesound.org/browse/>
<http://soundbible.com/tags-alert.html>
<http://soundbible.com/tags-game.html>
<http://www.wavsource.com/>

Pentru a introduce sunete in proiect vom folosi biblioteca FMOD.

Vom instala API-ul FMOD, care se gaseste la adresa:

Atentie de acolo nu luati FMOD Studio ci doar API-ul (unde scrie FMOD Studio Programmer's API and Low Level Programmer API). E posibil sa va ceara sa va faceti cont pentru a downloada API-ul). Veti selecta de acolo platforma Windows. Installerul prespune optiuni destul de simple; puteti sa le lasati pe cele default, deci faceti click pe "Next" pana la sfarsit.

Veti avea acum nevoie de headerele si bibliotecile cu care a venit API-ul.

Va duceti in folderul in care s-a instalat api-ul (de exemplu, la mine este: C:\Program Files (x86)\FMOD SoundSystem\FMOD Studio API Windows\). Intrati in folderele api → lowlevel I observati doua foldere: *inc* si *lib*. Din folderul *inc* copiat headerele (h si hpp) si apoi le includeti in proiect (nu e nevoie si de alte fisiere). Din folderul *lib* copiti fisierele cu extensia lib in folderul Lib din GameEngine. Iar dll-urile din folderul *lib* le veti pune in BIN-Nvidia sau BIN-ATI in functie de ce configuratie de build ati folosit pana acum.

Va trebui acum sa facem niste mici configurari pentru proiectul NewTrainingFramework, ca sa poata folosi bibliotecile pe care le-am adus.

In SolutionExplorer faceti click-dreapta pe numele proiectului si apoi mergeti la *Properties*. De acolo mergeti in tabul Linker si verificati ca in dreptul proprietatii Additional Library Directories, avem folderul Lib (daca nu este, il adaugati; atentie puneti cale relativa, nu absoluta, folosind un macro). Apoi intrati pe tabul *Input* si la Additional Dependencies, adaugati lib-urile copiate in proiect.

In ResourceManager vom crea un pointer pentru sistemul FMOD:

```
FMOD::System *fmodSystem;
```

In Init-ul lui ResourceManager vom obtine referinta la sistemul FMOD si apoi vom apela metoda sa init()

```
if (FMOD::System_Create(&fmodSystem) != FMOD_OK){  
    //nu avem sunete, vom afisa un mesaj de eroare  
}  
else  
    fmodSystem->init(32, FMOD_INIT_NORMAL, NULL); //initializat la 32 de canale
```

Incarcarea unui sunet se face prin instructiunea:

```
fmodSystem->createSound(cale_fisier, FMOD_LOOP_OFF, 0, &sound); //FMOD_LOOP_OFF inseamna  
ca nu se va relua sunetul dupa terminarea acestuia
```

Variabila sound este de tipul:

```
FMOD::Sound* sound;
```

Incarcarea sunetelor va fi tot de tipul load on demand. O idee de implementare este un enum care sa cuprinda un nume sugestiv si id-ul asociat(din fisierul de configurare). Pentru diverse evenimente se poate declansa acelasi sunet.

Ca sa redati sunetul, folositi instructiunea playSound:

```
fmodSystem->playSound(sound, 0, false, 0);
```

Pentru a sterge sunetul din memorie se foloseste:

```
sound->release();
```

Task1: se vor defini in enum: MOVE_UP, MOVE_DOWN, MOVE_LEFT, MOVE_RIGHT, care vor avea asociat sunetul toc.wav. La fiecare 100 de pasi pe x sau pe z in loc de toc.wav se va auzi sunetul haptic.wav.

Task2: luati niste fisiere-sunet de pe internet si asociati-le unor actiuni/evenimente in cadrul jocului.