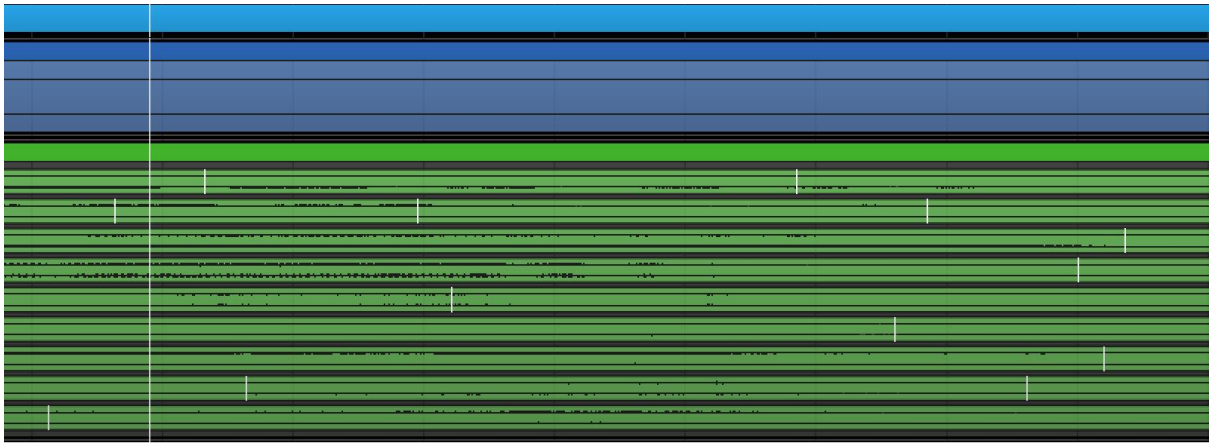


Univerzitet u Kragujevcu



PRAKTIKUM IZ PREDMETA
PRINCIPI ZVUKA I MUZIKE U
VIDEO-IGRAMA 1



Dr um. Kristina Marković

Kragujevac, 2024.

PRAKTIKUM IZ PREDMETA PRINCIPI ZVUKA I MUZIKE U VIDEO-IGRAMA 1

Autor

Dr um. Kristina Marković

Izdaje i umnožava

Univerzitet u Kragujevcu
Liceja Kneževine Srbije 1A, Kragujevac

Za izdavača

Prof. dr Nenad Filipović, rektor

Recenzenti

Mr um. Zoran Komadina, redovni profesor Filološko-umetničkog fakulteta Univerziteta u Kragujevcu

Dr um. Ljuba Brkić, redovni profesor Filološko-umetničkog fakulteta Univerziteta u Kragujevcu

Senat Univerziteta u Kragujevcu, na sednici održanoj 21.12. 2023. godine (odluka br. III-01-1251/86) prihvatio je pozitivan izveštaj recenzenata za rukopis „Praktikum iz predmeta *Principi zvuka i muzike u video-igrama I*“, i odobrio njegovu upotrebu kao univerzitetskog elektronskog pomoćnog udžbenika.

CIP - Каталогизација у публикацији
Народна библиотека Србије, Београд

004.4'277(075.8)(076)(0.034.2)
004.42PREMIERE(075.8)(076)(0.034.2)

МАРКОВИЋ, Кристина, 1985-

Praktikum iz predmeta Principi zvuka i muzike u video-igrama 1 [Електронски извор] / Kristina Marković. - Kragujevac : Univerzitet, 2024 (Kragujevac : Univerzitet). - 1 elektronski optički disk (CD-ROM) ; 12 cm

Kragujevac: Univerzitet u Kragujevcu. - Nasl. sa naslovne strane dokumenta. - Tiraž 500. - Bibliografija.

ISBN 978-86-81037-80-5

а) Дигитална обрада звука -- Вежбе б) Апликативни програм "FMOD Studio" -- Вежбе

COBISS.SR-ID 142373897

SADRŽAJ

1. UVOD	5
Pregled osnovnih funkcija u softveru	6
2. IZRADA FMOD STUDIO PROJEKTA	10
2.1 INTERAKTIVNI ZVUK	10
Ivent 1 - Varijacije zvuka	11
Randomizacija po uobičajenom metodu	12
Postavka parametara i uslova	16
Ivent 2 - Lup	21
Randomizacija početka audio-fajla	24
Ivent 3 - Tranzicija ambijentalnog zvuka	28
Destinacioni markeri	30
Postavka parametara na tranzicionu regiju	31
Tranzicioni prostor	35
Asinhroni instrument	39
Postavka parametra na asinhroni instrument	42
2.2 INTERAKTIVNA MUZIKA	45
Popis audio-fajlova i njihovih namena	47
Izrada muzičkog iventa – postavka takta i tempa	49
Postavka audio-fajlova – istraživačka muzika	51
Postavka audio-fajlova – akciona muzika	53
Probability – postavka verovatnoće emitovanja audio-fajlova	55
Postavka parametra – postepeno uvođenje audio-fajlova	57
Postavka asinhronog instrumenta – momentalno uvođenje audio-fajlova	60
Postavka audio-fajlova – muzika stanja smrti	63
Postavka horizontalnih tranzicija – smena stanja	63
Postavka tranziconih prostora	66
3. LITERATURA	74

1. UVOD

Ovaj praktikum je namenjen studentima interdisciplinarnih studija iz oblasti razvoja video-igara. Praktikum treba da pomogne studentima u sticanju veština pri korišćenju programa *FMOD Studio*. Na vežbama studenti se upoznaju sa osnovama korišćenja već pomenutog programa kroz praktičan rad, tj. izradu zadatka u vidu projekta iz interaktivnog zvuka i muzike.

Kao što je to najčešće slučaj, rad u bilo kojem softveru podrazumeva obilato korišćenje engleskog jezika. Stoga su se mnogi termini kod nas ustalili u anglikovanoj verziji (npr. *lup*, *ivent*, *fajl*, itd.). Ovakav način imenovanja je postao opšte prihvaćen među stručnjacima u okviru industrije video-igara, i ja sam ga uvela i u ovaj praktikum. Ostale manje korišćene i poznate termine sam uglavnom prevodila na takav način da, bar po mom mišljenju, najbolje odgovaraju značenju na engleskom, a opet odgovaraju duhu srpskog jezika. Pritom sam uvek ukazivala na izvorni naziv termina, a pri navođenju konkretnog izbora opcija unutar softvera, uvek navodila termine u svom originalnom nazivu na engleskom jeziku.

S obzirom da interdisciplinarnе studije iz oblasti razvoja video-igara pohađaju studenti različitog obrazovnog profila, vežbe su koncipirane i prilagođene tako da gradivo bude lako razumljivo svima. Praktikum je dizajniran tako da studentu što bliže predstavi proces praktične izrade zadatka i da kroz rad stekne i ovlada veštinama upotrebe gore spomenutog softvera. Ovakva koncepcija se zasniva na činjenici da su moderni programski paketi višenamenski, te svi pribegavamo taktici da izučimo one opcije nekog programa koje su nam potrebne, i kada su nam potrebne.

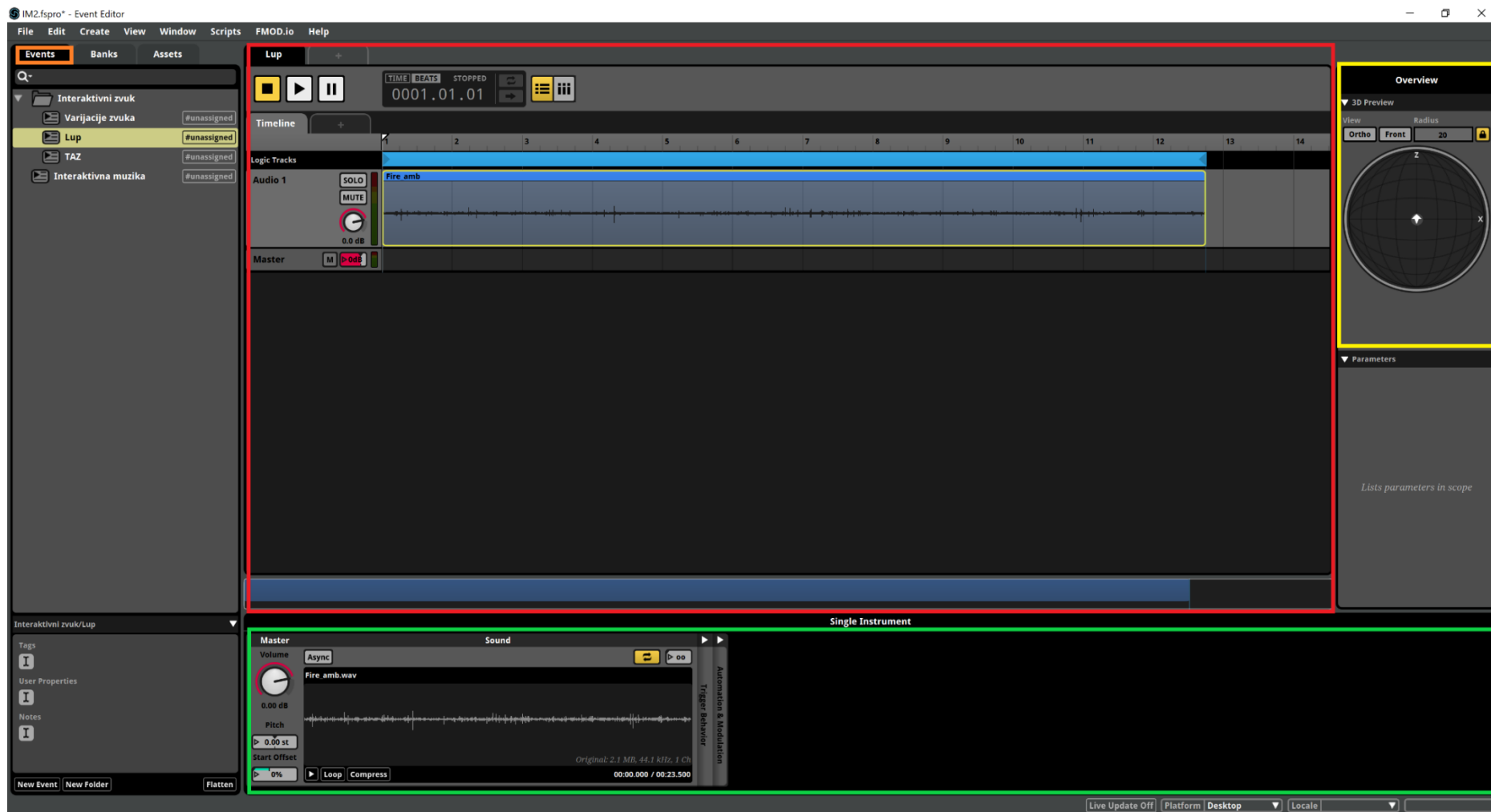
Vežbe su nastajale i kristalisale se tokom teorijskog i praktičnog rada sa studentima u okviru predmeta *Principi zvuka i muzike u video-igrama 1*. Različiti problemi koji se postavljaju pred jednog studenta pri upotrebi programa *FMOD Studio* su takođe deo ovog praktikuma i oslikavaju realne situacije u kojima su studenti imali prilike da se nađu.

Smatram da ovakav izbor vežbi obezbeđuje studentima veštine koje će im biti potrebne pri radu u industriji video-igara i pomaže im da kroz praktičan rad steknu bolji uvid o teorijskom delu gradiva, a samim tim o tome kako zapravo zvuk i muzika funkcionišu u video-igrama. Takođe, ovakav praktičan rad čini dobru osnovu da, kada i ako bude potrebno, studenti prošire svoje znanje u željenom smeru.

FMOD Studio pripada specifičnoj vrsti softvera koja služi za implementaciju zvuka i muzike u video-igrama. Ovakav tip softvera se još naziva audio-midlver (engl. audio middleware) upravo zato što u najvećem broju slučajeva služi kao posrednik između softvera za audio-produkciju (softver u kojem se obrađuje i finalno renderuje sav zvučni materijal; npr. *Cubase, Logic, Pro Tools*, itd.) i gejmn-endžina (engl. game engine) gde se sve komponente video-igre, uključujući i audio-sadržaj spajaju u jedno (primeri gejmn-endžina bi bili *Unity, Unreal*, itd). Audio-sadržaj obuhvata dijaloge, zvučne efekte i muziku. Kako bi zadobio upotrebnu vrednost u video-igri, mora da bude implementiran u gejmn-endžin. Svaki gejmn-endžin u sebi već sadrži određene alate za implementaciju audio-sadržaja, ali oni svakako nisu ni približno sofisticirani kao što je to slučaj kod audio-midlvera. On nam nudi mogućnosti napredne postavke logike promena koje se dešavaju po pitanju audio-sadržaja a koje su direktno uslovljene dešavanjima u video-igri. Dešavanje poput npr. susreta igrača sa neprijateljem će izazvati promenu stanja igrača. U žanru akcionih video-igara se ustalio standard da igrač ima dva glavna stanja – jedno se odnosi na stanje istraživanja teritorije bez preterane pretnje od strane neprijatelja i zove se *explore*; dok je drugo namenjeno stanju kada je igrač napadnut od strane neprijatelja i zove se *action* [1]. Dakle, susret sa neprijateljem će usloviti promenu u muzici, tj. prestaćemo da čujemo muziku koja je bila prisutna dok je igrač bio u stanju istraživanja i počecemo da čujemo muziku koja je namenjena stanju akcije.

Pregled osnovnih funkcija u softveru

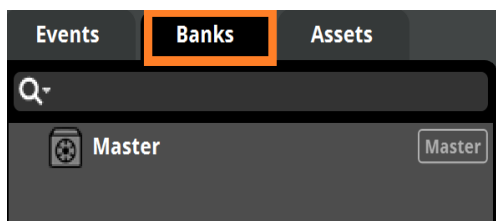
Interfejs programa *FMOD Studio* se sastoji iz par glavnih prozora koji će se najčešće koristiti pri izradi projekata interaktivnog zvuka i muzike. Za potrebe demonstracije služićemo se primerima koji su rađeni u verziji softvera 2.02.11. Može da se preuzme direktno sa sajta kompanije *Firelight Technologies* koja ga je razvijala. Pri pristupu linku <https://www.fmod.com/download> neophodno je da se napravi sopstveni nalog kako bi se softver koristio. Sledi pregled osnovnih pojmova unutar softvera. Radi lakšeg snalaženja, svaki prozor unutar *Event Editor* prozora je označen drugačijom bojom. *Banks* i *Assets* su prikazani u zasebnim slikama.



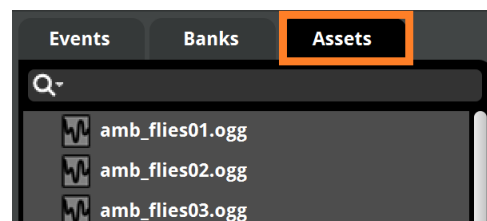
Slika 1.1. Event Editor prozor

- *Edit* (uokviren crvenim markerom) – predstavlja glavni radni prostor u kojem postavljamo audio-trake, *loop/transition regions* (ubuduće lup i tranzicione regije osim ako je u pitanju direktna upotreba funkcije unutar softvera); *tempo/destination markers* (ubuduće tempo i destinacioni markeri osim ako je u pitanju direktna upotreba funkcije unutar softvera); *transition timeline* (ubuduće tranzicioni prostor osim ako je u pitanju direktna upotreba funkcije unutar softvera); itd. O svemu ovome će da bude više reči u opisu postupka izrade zadatka. Važno je da se napomene da je *edit* prostor prazan sve dok se ne napravi prvi *event* (ubuduće ivalent, osim ako je u pitanju direktna upotreba funkcije unutar softvera). Onog trenutka kada napravimo ivalent (sledi objašnjenje), automatski se u *edit* prostoru stvara audio-traka u koju možemo da postavljamo audio-sadržaj. Kao što može da se vidi na slici 1.1, ovaj prostor je ispunjen audio-trakama na sličan način kao što je to u programima za audio-produkciju. Ono što je specifično i što nije svojstveno programima za audio-produkciju jeste svojevrsan način obrade audio-sadržaja u cilju stvaranja promenljivog, interaktivnog zvučnog okruženja.
- *Events* (uokviren narandžastim markerom) – u doslovnom prevodu znači događaji, što zapravo vrlo blisko i opisuje to što predstavlja u video- igri. Dakle, svaki tip akcije/dešavanja u video-igri jeste praćen određenim ivalentom. I najsitnije dešavanje kao što je običan klik u korisničkom interfejsu (engl. user interface, tj. UI click), će takođe da ima svoj ivalent. U *Events* brauzeru pravimo iste, a za potrebe našeg projekta koristićemo tzv. *2D* i *3D timeline events* o kojima će kasnije da bude reči. Na slici 1.1 može da se vidi da imamo jedan muzički ivalent i tri ivalenta zvučnih efekata koji su, pošto ih ima više, smešteni u folder *Interaktivni zvuk*.
- *Banks* (uokviren narandžastim markerom) – *bank* predstavlja format fajla koji određeni gejmn-endžin može da pročita. To su zapravo podaci o našem projektu, tj. svi ivalenti koje samo napravili sa svojim parametrima, podešavanjima, itd. Po pravljenju projekta se automatski kreira banka koja se zove *Master Bank* (slika 1.2). Njih svakako može da bude više od jedne, što i jeste poželjno kada su u pitanju veliki projekti. Za potrebe našeg projekta, mi ćemo da koristimo samo *Master Bank*.
- *Assets* (uokviren narandžastim markerom) – predstavlja mesto gde se nalaze svi audio-fajlovi koje smo ikad koristili u projektu (slika 1.3). Kako koji fajl ubacimo u projekat, u *Assets* brauzeru se automatski ažurira spisak fajlova. Kada bi otišli u projektni folder, videli bismo da postoji podfolder koji se zove *Assets* a on je zapravo isto što i *Assets* brauzer u programu, i u istom možemo da pronađemo identičan spisak audio-fajlova.

- *Deck* (uokviren zelenim markerom) – ovde možemo da postavljamo parametre i uslove (engl. parameters/conditions); kvantizaciju (engl. quantization), koja se strogo vezuje za izradu muzičkog dela zadatka; randomizujemo jačinu (engl. volume), visinu (engl. pitch) i početak audio-fajla (engl. start offset). O svemu ovome će takođe kasnije da bude reči.
- *3D Preview* (uokviren žutim markerom) – ovde možemo da testiramo kako će zvuk da se ponaša shodno promeni položaja izvora zvuka u 3D-prostoru pre nego što ivernt prolsedimo dalje u gejm-endžin. Krug predstavlja minimalnu i maksimalnu distancu izvora zvuka u odnosu na igrača. Strelica koja se nalazi u centru kruga predstavlja izvor zvuka. Kako pomeramo strelicu po teritoriji kruga tako se i zvuk menja. Kada je strelica u centru, tada čujemo zvuk u svom izvornom stanju, dok kada je pomeramo ka krajnjim obodima kruga, tako zvuk postaje sve manje čujan, sve dok ne dođe do krajnje tačke udaljensti gde postaje nečujan. Strelicu pomeramo tako što držimo levi klik miša i pomeramo ga u željenom smeru. Takođe može da se menja smer u kojem se zvuk emituje za 360°. To činimo tako što mišem pređemo preko strelice i pomeramo točkić miša. Pozicija igrača (slušaoca) je uvek u centru kruga i to se ne menja – ono što se menja jeste pozicija izvora zvuka u odnosu na slušaoca. Ova vrsta testiranja je svakako jednako korisna kad je u pitanju testiranje kako se zvuk menja u obrnutoj situaciji, tj. kada se položaj igrača menja u odnosu na statičan izvor zvuka.



Slika 1.2. Master Bank brauzer



Slika 1.3. Assets brauzer

2. IZRADA FMOD STUDIO PROJEKTA

U ovom poglavlju ćemo pokriti celokupni postupak izrade zadatka u programu *FMOD Studio*. Zadatak se sastoji iz dva dela od kojih se jedan bavi radom sa interaktivnim zvukom a drugi interaktivnom muzikom. Prvo ćemo se fokusirati na deo koji se odnosi na rad sa interaktivnim zvukom zato što nam je svima zvuk, tj. zvučni efekti, znatno razumljiviji tip audio-sadržaja nego što je to muzika. Samim tim, kroz rad sa zvukom bi trebalo da steknemo inicijalni uvid u to kako generalno interaktivni audio-sadržaj funkcioniše. Kroz praktikum će vrlo često da bude nadovezivanja i vraćanja na postupke kroz koje smo već prošli, kao i pravljenja paralela i uviđanja sličnosti između načina rada sa zvukom i muzikom. U velikoj meri može da se uoči podudarnost između tretmana zvuka i muzike, ali svaki od njih ima svoje osobenosti koje ćemo imati prilike da uočimo u narednim potpoglavljima.

2.1 INTERAKTIVNI ZVUK

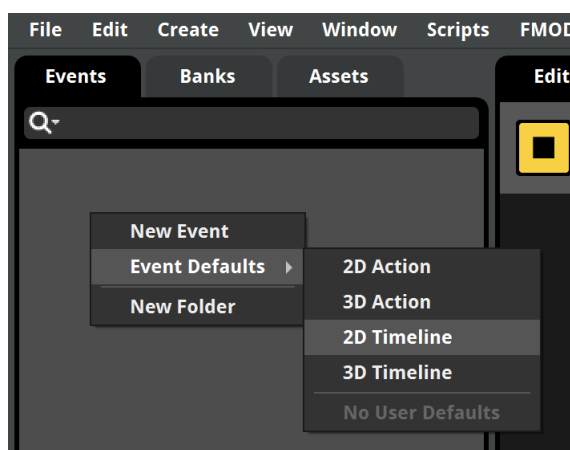
U delu zadatka koji se tiče rada sa interaktivnim zvukom treba da se kreiraju tri različita iventa. Dole su popisana sva tri iventa sa kratkim navođenjem šta je sve od gradiva predviđeno da se obradi u okviru dotičnog iventa.

1. Ivent u kojem pokrивamo sledeće pojmove: *multi instrument* (ubuduće multi-instrument osim ako je u pitanju direktna upotreba funkcije unutar softvera), postavku parametra i uslova, radnomizaciju zvuka po uobičajenom metodu, itd.
2. Ivent u kojem pokrивamo izradu lupa (engl. loop) kao i njegovu randomizaciju.
3. Ivent u kojem pokrивamo tranziciju između dva različita lupa kao i asinhrono isntrumente.

Ivent 1 - Varijacije zvuka

Zbog nepredvidivosti toka i trajanja dešavanja u video-igrama, randomizacija audio-sadržaja igra značajnu ulogu u postizanju smanjenja efekta repetitivnosti. Vrlo popularan primer randomizacije audio-sadržaja su zvukovi koraka u video-igri. Suština izrade i implemntacije zvuka u video-igri jeste da se postigne što veći nivo realističnosti, tj. da se teži oponašanju zvuka onakvog kakav jeste u realnom svetu. Stoga, vratimo se na malopređašnji primer koraka. Da li ste ikada obratili pažnju kako zvuče vaši koraci dok se krećete? Pod pretpostavkom da se krećete po jednoličnom terenu i ujednačenim tempom, primećuje se da su zvukovi pojedinačnih koraka međusobno vrlo slični ali opet različiti. Kada su u pitanju ovakve vrste zvučnih pojava, suština uspešne implmentacije jeste upravo minimalna sličnost i razlika.

Ivent se pravi tako što u *Events* brauzeru desnim klikom otvorimo padajući meni iz kojeg biramo *Event defaults – 2D Timeline event* (slika 2.1). Biramo *2D Timeline event* iz razloga što se on vezuje isključivo za igrača. To je zato što igrač, koji predstavlja oči i uši svih dešavanja u video-igri tj. u 3D-prostoru koji ga okružuje, ne može sopstveni zvuk koji emituje da doživi u istom [2]. To dakle znači da će igrač svoje korake uvek da čuje istim intenzitetom. U slučaju kada bi postojali i drugi 3D-objekti koji se kreću, tj. koračaju oko igrača, umesto *2D* bi izabrali *3D Timeline event*. U ovakvoj situaciji, igrač i drugi 3D-objekti bi delili ivent. U slučaju kada ivent koristi igrač, zvučni rezultat bi bio identičan kao kada koristimo 2D-ivent, dok bi u slučaju primene na druge 3D-objekte, pri promenama pozicija u odnosu na igrača, čuli i promene u karakteristikama zvuka. Ivent ćemo da nazovemo *Varijacije zvuka*.



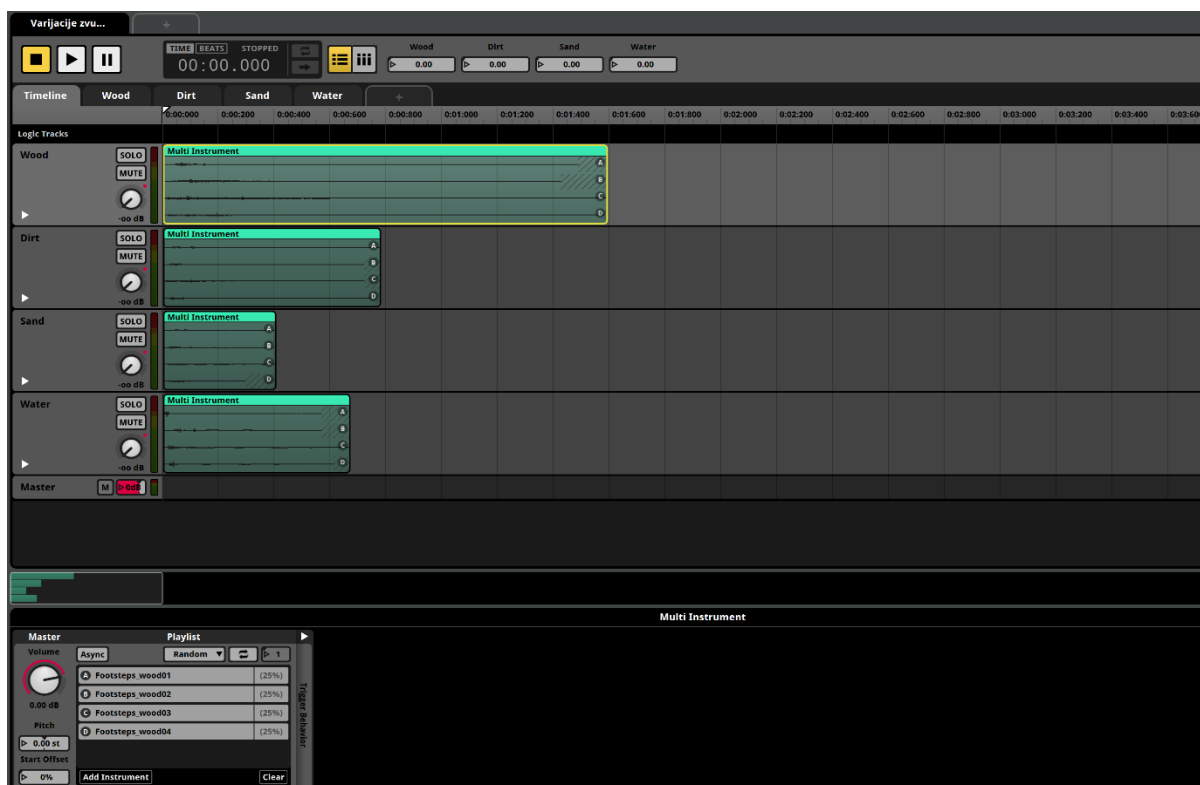
Slika 2.1. Dodavanje ivena (*2D Timeline event*)

Randomizacija po uobičajenom metodu

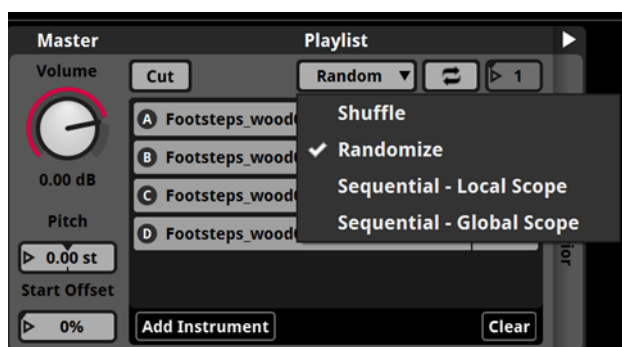
Ovakav tip randomizacije podrazumeva upotrebu multi-instrumenta sa specifičnom postavkom tipa emitovanja audio-fajlova. Multi-instrument se automatski kreira ako istovremeno pokušamo da iz foldera odgovarajućeg operativnog sistema prevučemo dva ili više fajlova u audio-traku – u suprotnom, ako prevučemo samo jedan fajl, automatski se stvara *single instrument*, što ćemo imati prilike da vidimo u narednom iventu. Dakle, multi-instrument služi za ubacivanje dva i više različitih audio-fajlova na jednom mestu, praveći na taj način ‘plejlistu’. To samo po sebi objašnjava da multi-instrument ne služi tome kako bi istovremeno čuli više različitih fajlova, već da prema potrebi čujemo po jedan fajl iz plejliste. Levim klikom na multi-instrument se otvara pomenuta plejlita u *deck* prozoru. Na slici 2.2 u donjem levom uglu se vidi da se multi-instrument sastoji iz fajlova koji se zovu: *Footsteps_wood01*, *Footsteps_wood02*, *Footsteps_wood03* i *Footsteps_wood04*. Ovakav način imenovanja fajlova predstavlja konvenciju kada su u pitanju varijacije jednog te istog zvuka [3, 4]. To znači da svi izlistani fajlovi zvuče vrlo slično ali opet različito, što je jako bitno kod ovakovog tipa zvuka koji je namenjen da se javlja učestalo i to jedan za drugim. To bi praktično zadržalo da, kako se igrač kreće u video-igri, tj. kako unosi kontrole, tako se okida ivent koraka. Pri svakom novom iniciranju iventa iz date plejliste ćemo čuti jedan od ponuđenih zvukova koraka. Redosled kojim ih čujemo zavisi od tipa emitovanja koji biramo u padajućem meniju a koji se izlistava levim klikom miša (slika 2.3). Od ponuđenih opcija imamo sledeće:

- *Shuffle* – fajlovi u plejlisti se puštaju nasumično, poštujući pritom pravilo da poslednji emitovan fajl ne sme da se ponovi. Primer redosleda puštanja bi bio: 1,3,4,1,5,3, itd.
- *Sequential* – fajlovi u plejlisti se puštaju redom, tj. od prvog do poslednjeg i ta sekvenca će tako da se ponavlja sve dok je aktivan ivent.
- *Randomize* – fajlovi u plejlisti se puštaju nasumično, međutim, ovde za razliku od *shuffle*, može da se desi da se poslednji emitovan fajl ponovi. Dakle, razlika između *shuffle* i *randomize* je praktično u tome što kod jednog ne sme, a kod drugog sme da se ponovi poslednji emitovan fajl u plejlisti. Primer ovakvog redosleda puštanja bi bio: 1,3,4,4,5,3, itd.

Svaki od ovih tipova emitovanja ima svoju upotrebnu vrednost u zavisnosti od situacije. Za potrebe iventa kao što je ovaj, koristićemo *Randomize* tip emitovanja (slika 2.3).



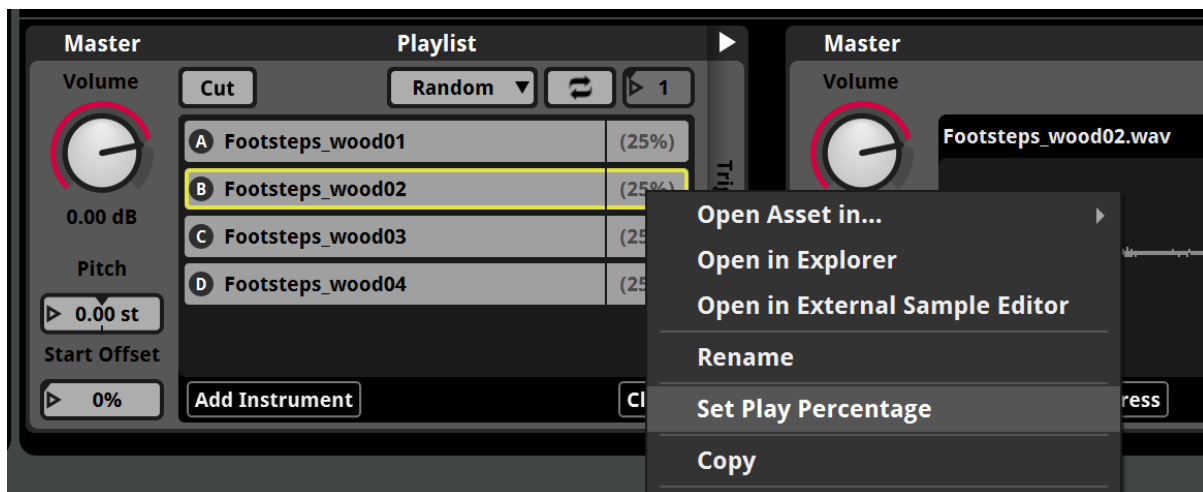
Slika 2.2. Plejlita multi-instrumenta u *deck* prozoru



Slika 2.3. Odabir tipa emitovanja (*Randomize*)

Već je gore nevedeno da naša plejlita fajlova predstavlja varijacije jednog te istog zvuka. Po konvenciji treba da postoji oko pet varijacija [3, 4]. U navedenom primeru postoji četiri. Takođe je već nevedeno da je suština kvalitetne izrade i implemntacije zvuka u video-igri da se postigne što veći nivo realizma. Samim tim, kada na našu plejlistu primenimo *randomize* tip emitovanja, utičemo na to da koračanje igrača u igri zvuči realističnije. Da smo u suprotnom koristili *sequential* tip emitovanja, posle određenog vremena bi se vrlo verovatno primetilo da se određeni patern/sekvenca ponavlja. Takođe može da se utiče i na verovatnoću javljanja pojedinih fajlova iz plejlite. U slučaju da postoji neki fajl ili više njih koje bi hteli češće da čujemo u odnosu na druge, možemo da preraspodelimo njihovu procentualnu verovatnoću emitovanja. Po automatizmu kada napravimo multi-instrument, verovatnoća

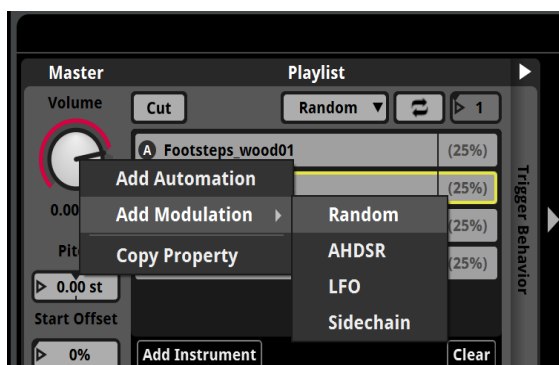
emitovanja je raspodeljena ravnopravno. U našem primeru, s obzirom da imamo četiri fajla, svaki od njih ima po 25% verovatnoće da se emituje. U slučaju da želimo da jednom od njih dodelimo nešto veću, ili pak nižu procentualnu vrednost, to postizemo tako što desnim klikom na polje sa procentom određenog fajla otvorimo padajući meni iz kojeg biramo *Set Play Percentage* (slika 2.4). Ovim postupkom nam ovo polje postaje dostupno za izmene i tu brojevima unosimo željeni procenat.



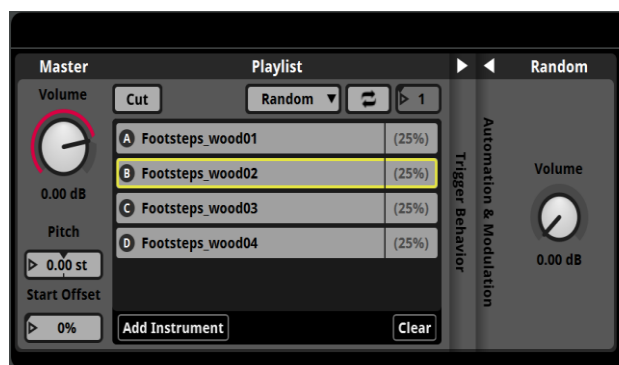
Slika 2.4. Dodavanje procentualne vrednosti verovatnoće emitovanja audio-fajla

Može da se postavi pitanje zašto biramo *randomize* a ne *shuffle*. Razlog je taj što se uz ovakav tip inventa pored randomizacije redosleda puštanja fajlova, takođe randomizuje jačina (volume) i visina (pitch) audio-fajla. Uz ovakav tretman, čak i da se određeni audio-fajl ponovi više puta zaredom, uvek će sledeće ponavljanje da zvuči drugačije u odnosu na prethodno. Na slici 2.5 se vidi postupak na koji način se postavlja randomizacija jačine [2, 5]. U *deck* prozoru, desnim klikom miša na *volume* regler multi-instrumenta otvaramo padajući meni iz kojeg biramo *Add Modulation – Random*. Ovaj postupak otvara prozorčić sa desne strane koji se zove *Random* i u kojem se nalazi novi *volume* regler (slika 2.6). Sada smo u mogućnosti da postavljamo opseg variranja vrednosti jačine audio-fajla. To činimo tako što levim klikom pomeramo kazaljku na regleru. Takođe možemo da obavimo precizniji unos vrednosti a to činimo tako što duplim levim klikom na 0.00 db¹ aktiviramo numerički unos. Opseg koji postavimo se odnosi, tj. deluje na čitavu plejlistu našeg multi-instrumenta. To znači da je svaki audio-fajl u plejlisti podložan promenama jačine u zadanom opsegu.

¹ Decibel (db) je logaritamska mera za jačinu signala, tj. jedinica za merenje jačine zvuka.



Slika 2.5. Randomizacija jačine



Slika 2.6. Random prozorčić

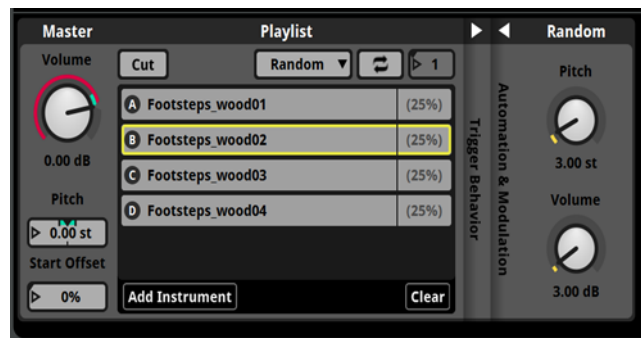
Na isti način se vrši postupak randomizacije visine. Dakle, u *deck* prozoru, desnim klikom miša na *pitch* multi-instrumenta (ovog puta slajder) otvaramo padajući meni iz kojeg biramo identična podešavanja kao za jačinu. I ovde se u okviru istog *Random* prozorčića otvara novi *pitch* (ovog puta regler). Sada smo u mogućnosti da postavljamo opseg variranja vrednosti visine audio-fajla a to činimo na isti način kao i za jačinu.

Kao što je to već bilo navedeno, suština uspešne implmentacije ovakvog tipa iventa jeste minimalna sličnost i razlika. Ova pravilo se preslikava i na postvaku opsega variranja vrednsoti jačine i visine. Stoga pri postavci maksimalnih i minimalnih vrednosti treba da se vodi računa da opseg ne bude previše širok. Kada bi npr. postavili da je maksimalna vrednost jačine 10 db a minimalna -50 db, zvučni rezultat bi bio takav da bi postojala velika verovatnoća da jedan fajl zvuči preglasno, dok bi pak neki drugi bio jedva čujan. Ovo svakako nije dobar pristup i stoga je bitno da ovaj opseg ne prelazi više od oko 3 db. Preporučuje se postavka od 0 do 3 db, kao što je prikazano na slici 2.7.

Ista pravila važe i pri postavci maksimalnih i minimalnih vrednosti visine. Kada bi npr. postavili da je maksimalna vrednost 10 st² a minimalna -10 st, zvučni rezutat bi bio takav da bi postojala velika verovatnoća da većina audio-fajlova bude izobličena usled ekstremene promene visine. Audio-fajlovi kojima bi visina bila znatno podignuta bi zvučali tanko, kratko i pisakvo, dok bi oni kojima je visina znatno spuštена zvučali nisko, tromo i razvučeno. Stoga se preporučuje da i ovde opseg ne prelazi više od 3 st. Preporučuje se postavka od 0 do 3 st, kao što je prikazano na slici 2.7. Ovakav vid randomizacije koji obuhvata sve gore navedeno, dovodi do toga da broj kombinacija postaje praktično beskonačan. S obzirom da je izuzetno nepredvidivo koliko će koraka igrač da napravi u toku igranja, jasno je zašto je ovakav vid

² Semitone (st) u prevodu znači polustepen i predstavlja mernu jedinicu tonske visine.

randomizacije toliko dragocen i dodatno se naglašava bitnost pametne implementacije audio-sadržaja. To u praktičnom smislu znači da smo putem randomizacije od samo četiri različita audio-fajla napravili praktično beskonačan broj varijacija. Kroz istoriju audio-sadržaja u video-igrama, stalno se vodila bitka između kapaciteta prostora koji igra može da zauzme i broja detalja koje autori žele da prikažu u video-igri. Ovakav pristup svakako ide u prilog rešenju takvog problema i ostavlja prostor za ubacivanje što više drugih detalja koji će uticati na to da video-igru učine što kvalitetnijim multimedijalnim delom.

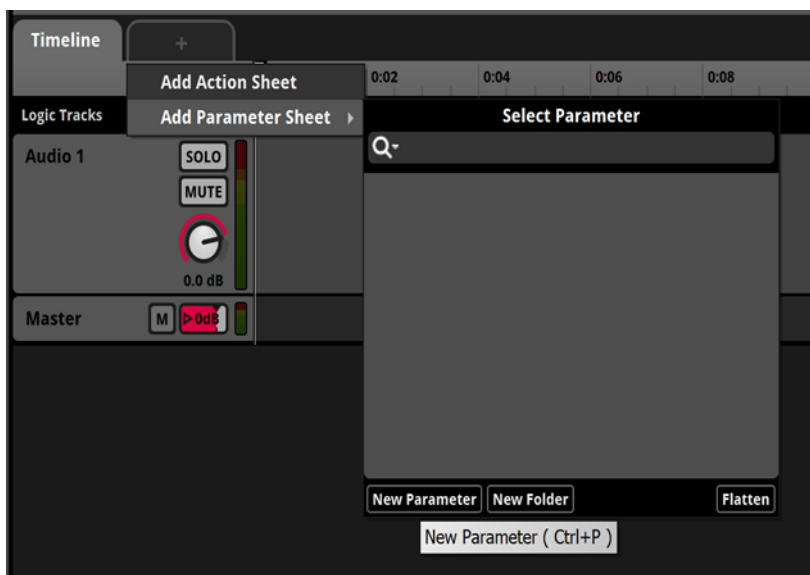


Slika 2.7. Postavka opsega variranja vrednosti jačine i visine

Postavka parametara i uslova

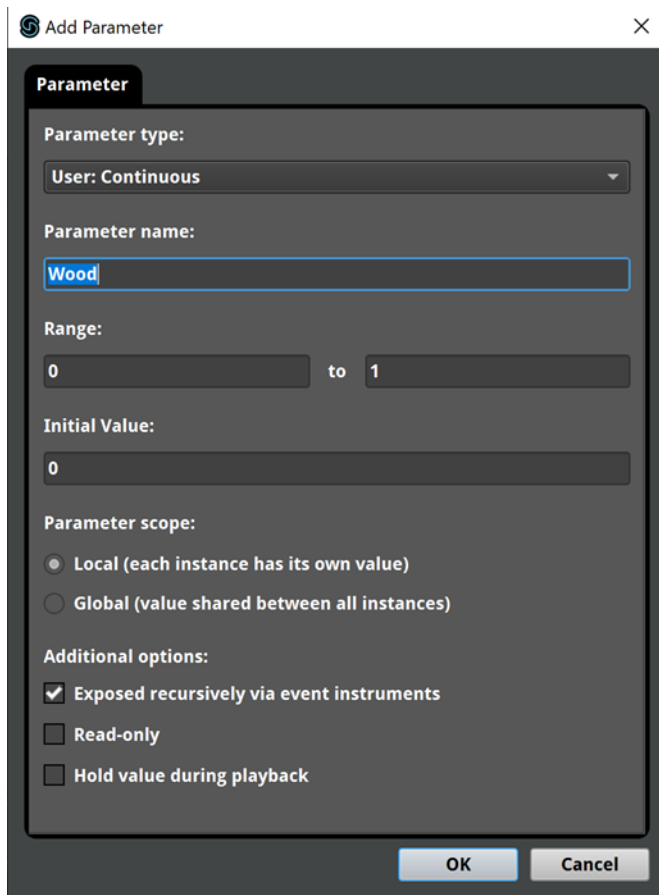
Više puta smo pomenuli da video-igre teže ka tome da igraču pruže što realističniji doživljaj. Stoga, pri izradi našeg iventa koraka, treba da uzmemo u obzir sve moguće terene po kojima igrač može da se kreće u video-igri i da shodno različitim tipovima terena napravimo odgovarajuće plejliste. Kao što smo napravili plejlistu za kretanje po drvenom terenu, tako ćemo na isti način da napravimo za ostale terene. Svaki multi-instrument je smešten u zasebnu audio-traku u okviru iventa, kao što može da se vidi na slici 2.2. Kada bi pritisli razmak na tastaturi, aktivirali bi plejbek i program bi počeo da emituje audio-sadržaj selektovanog iventa. U trenutnoj postavci bi čuli nasumice po jedan audio-fajl iz svakog multi-instrumenta, dakle, istovremeno bi čuli po jedan zvuk od svakog tipa terena, što svakako nije to što želimo. Cilj jeste da se čuje zvuk kretanja po samo jednom ili u kombinaciji dva različita tipa terena. Igrač npr. može da se kreće po drvenim daskama i onda bi se čuo samo zvuk drveta, dok bi u situaciji da se kreće po mešovitom terenu (npr. kombinacija peska i barica) istovremeno čuli oba tipa terena.

Dakle, zvuk se menja shodno terenu po kojem se igrač kreće, tj. zvuk će imati različite karakteristike kada se igrač kreće po drvetu, zemlji, pesku, itd. Kako bi se zvuk uspešno menjao, treba da se postavi po parametar za svaki tip terena [2]. S obzirom da imamo četiri različita tipa terena, treba da napravimo četiri parametra od kojih će svaki da se vezuje za određeni tip terena i shodno njemu da bude i imenovan. Dakle, pravimo parametre koji se zovu: *wood*, *dirt*, *sand* i *water*. Postoji više načina kako može da se doda parametar. U ovom iventu dodaćemo tako što u *edit* prozoru levim klikom miša na znak plus (slika 2.8) otvorimo padajući meni u kojem biramo *Add Parameter Sheet*. Dalje, kliknemo na dugme *New Parameter* što će automatski da otvori *Add Parameter* prozor. Na istoj slici se takođe vidi da postoji prečica za ovaj postupak i glasi *ctrl + P*. U polju *Parameter name* upisujemo ime parametra i za potrebe ovog iventa sva ostala polja ostavljamo onakvim kakva jesu (slika 2.9).

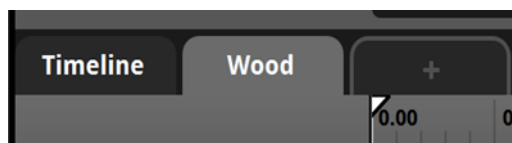


Slika 2.8. Dodavanje parametra iz *edit* prozora

Kao što može da se vidi na istoj slici, parametar koji smo postavili ima svoju maksimalnu i minimalnu vrednost. Odabrali smo da minimalna bude nula a maksimalna jedan. Klikom na dugme *OK* se izlazi iz *Add Parameter* i vraća nazad u *edit* prozor.



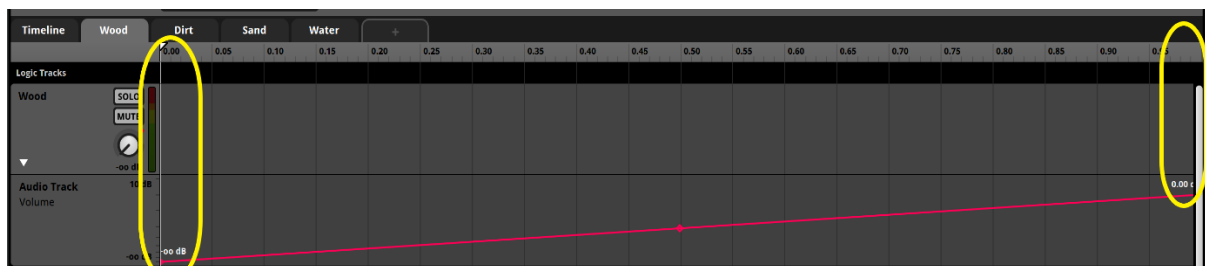
Slika 2.9. Add Parameter prozor



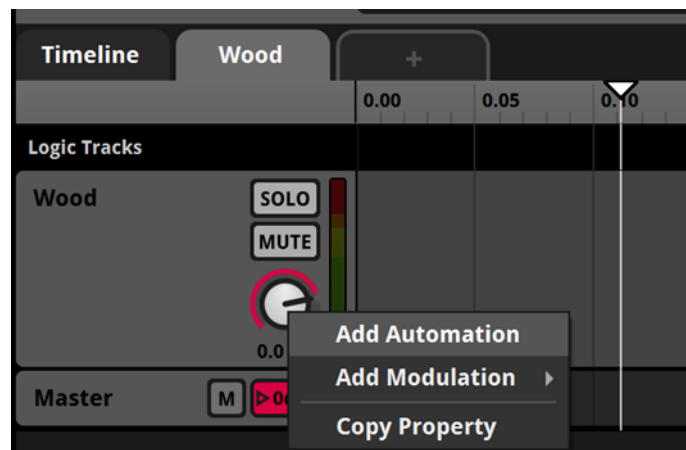
Slika 2.10. Tab parametra

Kreiranje parametra na ovakav način otvara novi tab odmah pored predefinisanoog taba *Timeline* (slika 2.10). On predstavlja tab samog parametra (slika 2.11 ilustruje otvoreni tab parametra). Primećuje se da više nema multi-instrumenta, kao i da postoji merna skala koja broji od nula do jedan. To je zapravo opseg vrednosti koje smo postavili za naš parametar. Dalje, primećuje se kosa crvena linija koja se kreće naviše sa leva na desno. Treba da se ima u vidu da je na slici prikazan gotov unos vrednosti, i da, kada inicijalno otvorimo tab parametra, crvena linija ne postoji zato što tek treba da je dodamo. Ta crvena linija, tj. kriva, se zove *volume automation*. Nju aktiviramo tako što desnim klikom otvorimo padajući meni iz kojeg biramo *Add Automation* (slika 2.12). Otvara nam se nova traka odmah ispod audio-trake koja nam služi za postavku vrednosti jačine. Inicijalno je vrednost jačine prikazana isprekidanom crvenom linijom u nivou vrednosti od 0 db (slika 2.13). To znači da je u pitanju optimalna vrednost jačine, tj. da audio-fajlove čujemo u njihovom izvornom stanju, odnosno onakve kakve smo ih

i uneli u projekat. Kada mišem pređemo preko isprekidane linije pojavljuje se kursor u obliku krstića. Na mestu krstića levim klikom miša pravimo kružić (slika 2.14). Dalje, kada opet mišem pređemo preko kružića pojavljuje se kursor u obliku krstića sa strelicama. Ovaj tip kursora nam omogućava da pomeramo kružić u sva četiri smera (gore, dole, levo i desno) i samim tim da podešavamo vrednosti jačine u odnosu na vrednosti parametra. Pošto nam je namera da kada je parametar jednak nuli ne čujemo korake, i obrnuto, kada je jednak jedinici da ih čujemo, napravićemo dva kružića – jedan skroz levo u ravni minimalne a drugi skroz desno u ravni maksimalne vrednosti parametra.



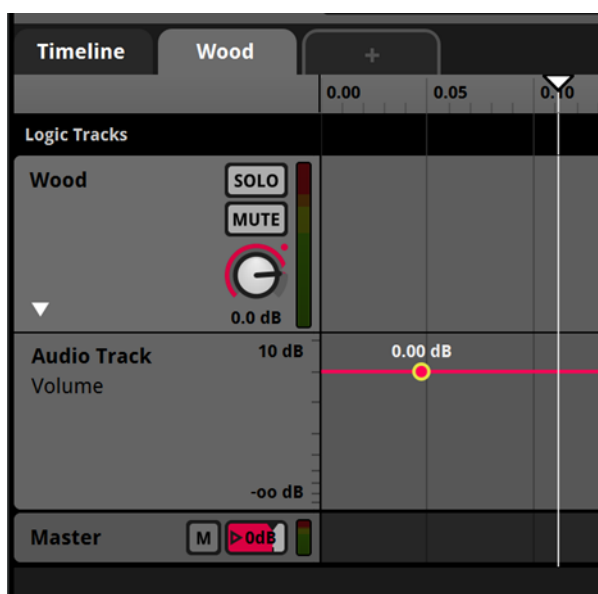
Slika 2.11. Otvoreni tab parametra



Slika 2.12. Dodavanje krive *volume automation*



Slika 2.13. Inicijalna vrednost jačine



Slika 2.14. Dodavanje vrednosti jačine

Ovakvim postupkom ćemo da postavimo uslov da kada je parametar jednak nuli, jačina bude na svojoj minimalnoj vrednosti, tj. $-\infty$ db (slika 2.11). To znači da će koraci će da budu apsolutno nečujni. U suprotnom, kada je parametar jednak jedinici jačina će da bude jednaka 0 db (ista slika) tj., čućemo korake u onom nivou jačine u kojem smo ih originalno uneli u projekat. Treba da se vodi računa da se ne pređe vrednost od 0 db. To je zato što svaki decibel iznad nule zapravo povišava originalnu vrednost jačine audio-fajla. Vrlo česta greška u zadacima jeste da se vrednosti jačine podese na minimalnu i maksimalnu vrednost ponuđenu u opsegu njene merne skale. Ovo jeste u redu kada je parametar jednak nuli, ali ako se vrednost jačine kada je parametar jednak jedinici takođe postavi na maksimalnu vrednost, što jeste 10 db, to znači da bi korake čuli za 10 db jače nego što bi trebalo. Shodno tome, ovakve greške treba da se izbegnu.

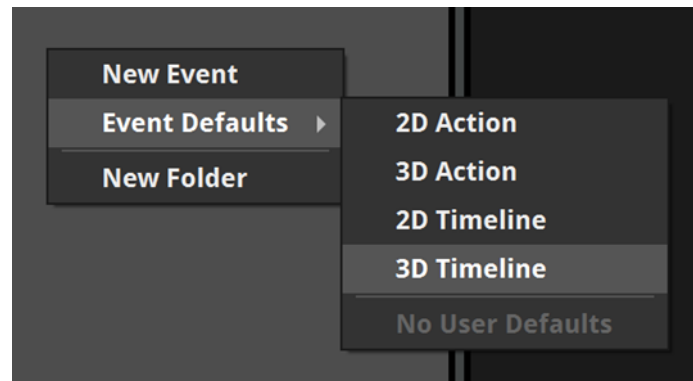
Ovaj postupak treba na isti način da se sprovede na sve ostale grupe zvukova namenjene drugim tipovima terena. Dakle, svaka grupa zvukova treba da bude smeštena u svoju audio-traku. Novu audio-traku dodajemo komandom *ctrl + T*. Slika 2.2 ilustruje kako treba da izgleda gotov ivent koraka. Nakon ove postavke vreme je da se ivent testira. Jako bitan deo implementacije jeste testiranje svega što smo odradili jer je to jedini način da uvidimo da li smo zaista dobro odradili zadatak. Treba da proverimo da li nam se odgovarajući zvukovi puštaju pod dejstvom određene vrednosti parametra. Na slici 2.2 se primećuje da svaki od unetih parametara ima svoj slajder. Nalaze se odmah iznad tabova parametara. Kada mišem pređemo preko strelice smeštene na levoj strani slajdera, kursor se pretvara u ručicu. Levim klikom nam se otvara numerički unos vrednosti parametra. Na ovaj način možemo jako lako da menjamo vrednosti i ustanovimo da li nam svi parametri rade kako treba. Isto tako, pri uzastopnom ponavljanju zvukova iz istog multi-instrumenta, tj. zvukova koraka jednog te istog terena, treba da se zapitamo da li koraci zvuče realistično. Shodno tome treba da uvidimo da li eventualno treba da se izvrše dodatne korekcije u vezi sa randomizacijom, da li pojedinim fajlovima treba da se smanji mogućnost za emitovanje i sl.

Ivent 2 - Lup

Nepredvidivost toka i trajanja dešavanja je vrlo svojstvena video-igramama. Baš iz tog razloga je lup izuzetno bitno sredstvo i način izražavanja. Ovakav tip audio-fajla predstavlja kraću ili dužu audio-sekvencu koja je napravljena tako da može da se ponavlja beskonačno, a da to ljudsko uvo ne primeti. Treba da se ima u vidu da se ovakav tip audio-fajla jednako koristi kod zvučnih efekata, kao i kod muzike. Samo po sebi je jasno da upotreba ovakvog tipa audio-fajla vrlo uspešno premošćuje nepredvidiva trajanja i dešavanja u video-igri [4]. Vrlo jednostavan primer bi bio neki 3D-objekat u video-igri, npr., baklja koja neprekidno gori. Svakako se ne očekuje, pogotovo za ovako jednostavan tip dešavanja, da imamo audio-fajl koji traje u nedogled kako bi ispratio animaciju gorenja baklje, koja i sama predstavlja kratak video-lup. Stoga se koristi kraći audio-fajl koji traje oko 20 s, a koji je izrađen u formi lupa i koji samim tim može da zameni sate i sate emitovanja zvuka gorenja baklje.

Za razliku od prethodnog iverenta koji je bio 2D, ovaj će da bude 3D. To je iz razloga što je ovaj ivent namenjen zamišljenim 3D-objektima rasprostranjenim po 3D-prostoru po kojem se igrač kreće. To u praktičnom smislu znači da kako igrač bude menjao svoju poziciju u

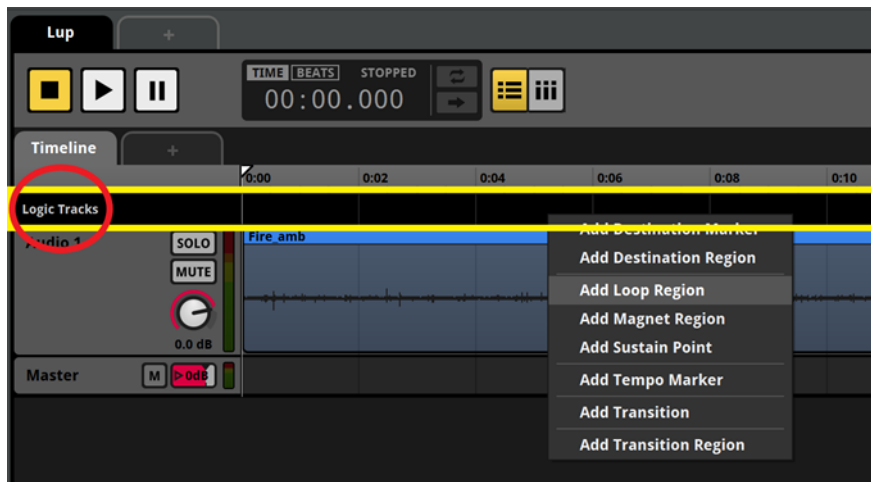
odnosu na baklje koje su statične, tj. kako im se bude približavao, odnosno odaljavao, tako će da ih čuje glasnije, odnosno tiše. Ovu pojavu svakako možemo da testiramo u okviru odeljka *3D Preview* koji smo pomenuli pri predstavljanju osnovnih pojmova u softveru.



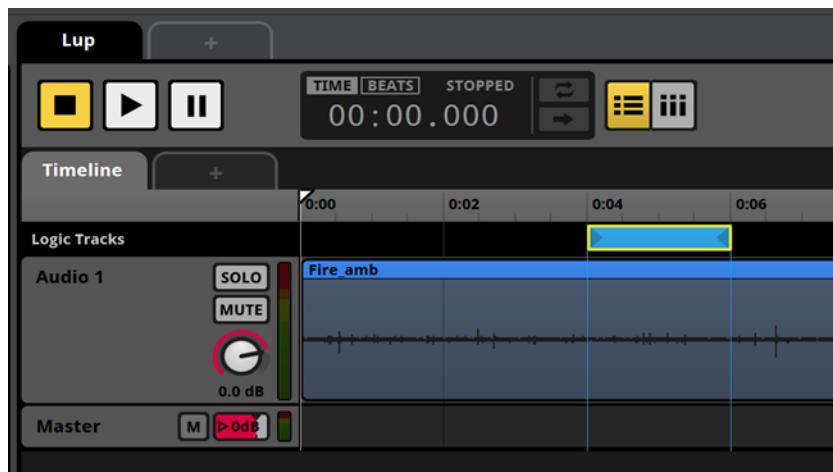
Slika 2.15. Dodavanje iventa (*3D Timeline event*)

Ovaj ivalent postavljamo tako što u ivalent brauzeru desnim klikom otvorimo padajući meni iz kojeg biramo: *Event defaults - 3D Timeline event* (slika 2.15). Nazvaćemo ga *Lup*. U audio-traku koja se automatski kreira po kreiranju iventa prevlačimo odgovarajući audio-fajl. *Lup* se postavlja tako što na audio-fajl koji smo upravo prevukli postavimo tzv. *lup-regiju*. Nju možemo da postavimo na dva načina:

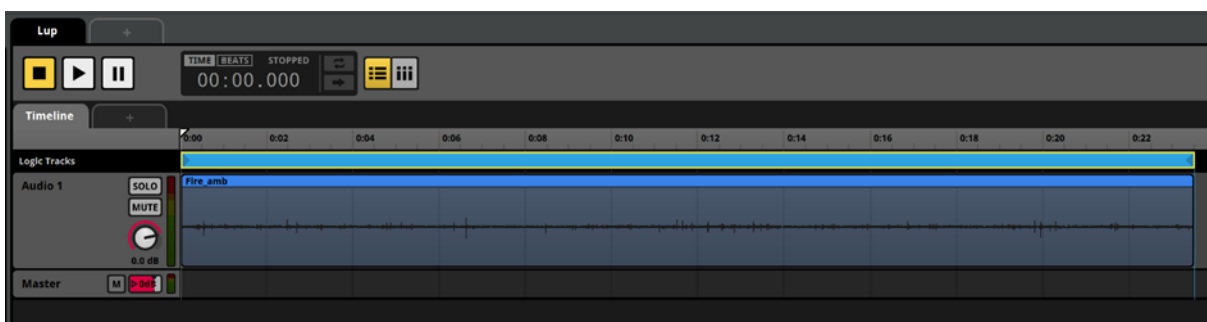
- 1) Desnim klikom na ‘crni prostor’ koji se zove *Logic Tracks* (slika 2.16) otvaramo padajući meni iz kojeg biramo *Add Loop Region*. Ovo će napraviti plavu traku, tj. *lup-regiju* (slika 2.17) koju potom levim klikom prevlačimo na levu, odnosno desnu stranu, čime regulišemo njeno trajanje. Pre prevlačenja neophodno je da se mišem pređe preko njenih krajnjih granica, sve dok kursor ne zadobije oblik horizontalne crtice sa strelicom na levoj i desnoj strani. Nakon toga, traka je spremna za prevlačenje i trebalo bi da joj podesimo trajanje prema trajanju samog audio-fajla (slika 2.18).



Slika 2.16. Dodavanje lup-regije



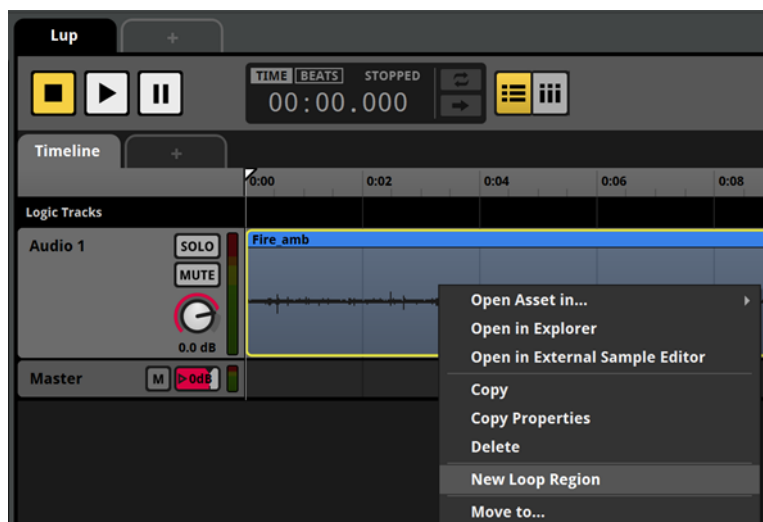
Slika 2.17. Izgled lup-regije



Slika 2.18. Podešavanje trajanja lup-regije

- 2) Jednostavniji način od prethodnog jeste da desnim klikom na sam audio-fajl otvorimo padajući meni iz kojeg biramo opciju *New Loop Region* (slika 2.19). Na ovaj način se

takođe kreira plava traka ali se njeno trajanje automatski podešava prema trajanju selektovanog audio-fajla, dakle, ovakvim postupkom bi odmah dobili rezultat kao sa slike 2.18. Preporuka je da se praktikuje ovaj način postavke, ali svakako treba da budemo svesni da je ovakav tip trake fleksibilan i da može da se produžava odnosno skraćuje prema potrebi.



Slika 2.19. Jednostavniji način postavke lup-regije

Randomizacija početka audio-fajla

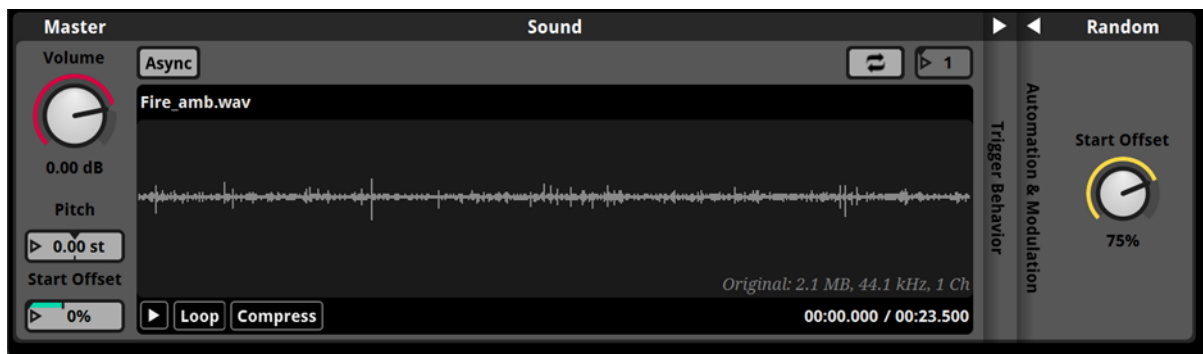
Ovaj vid randomizacije zvuka se uveliko razlikuje od randomizacije po uobičajenom metodu zato što lup ne možemo da tretiramo na isti način kao multi-instrument. Lup je ukontinuirani zvuk čija svrha i jeste da se ponavlja beskonačno i bez prekida. Svaki vid randomizacije po uobičajenom metodu bi remetio njen tok. Ovde uopšte i ne postoji opcija randomizacije redosleda emitovanja zvukova iz prostog razloga što ne postoji plejlista već samo jedan fajl koji se beskonačno ponavlja. Ako bi randomizovali jačinu, pri ponavljanju ova promena bi remetila tok lupa i zvučala kao greška. Na isti način i još primetnije bi remetila tok randomizacija visine. Ono što možemo da randomizujemo jeste početak lupa. Ovo je naročito korisno u situacijama kada 3D-objekti dele jedan te isti ivent lupa. To bi predstavljalo situaciju kada imamo 3D-objekte koji se često ponavljaju u 3D-prostoru – npr. baklje. U slučaju da se dva ista 3D-objekta nađu vrlo blizu jedan drugog, čućemo udvostručeno emitovanje audio-fajla, što proizvodi loš zvučni rezultat. Ako zamislimo da je lup krug, shvatićemo da iz bilo koje tačke da počnemo, na kraju ćemo se vratiti na istu tačku iz koje smo počeli. Ako bi jedan

audio-fajl počeo iz jedne tačke a drugi iz neke druge, stvorila bi se iluzija da čujemo dva slična ali opet različita zvuka. I van ovog primera, randomizacija početka audio-fajla je vrlo korisna jer u slučaju da je u pitanju neki zvuk koji se često javlja, vrlo je verovatno da bi u jednom trenutku primetili da zvuk uvek počinje na isti način [3, 4].

Randomizacija početka audio-fajla se postavlja tako što selektujemo audio-fajl. Ovo nam otvara *deck* prozor u kojem desnim klikom na slajder *Start Offset* u donjem levom uglu, otvorimo padajući meni iz kojeg biramo: *Add Modulation – Random* (slika 2.20). Ovo će na desnoj strani *deck* prozora kreirati dodatni prozorčić koji se zove *Random* u kojem se nalazi *Start Offset* regler. Ovde možemo dalje da podešavamo u procentima u kojoj meri je randomizovan početak audio-fajla. To činimo tako što držimo levi klik i pomeramo kazaljku u levu i desnu stranu. Preporuka je da se ovaj procenat kreće između 70 i 80% (slika 2.21).



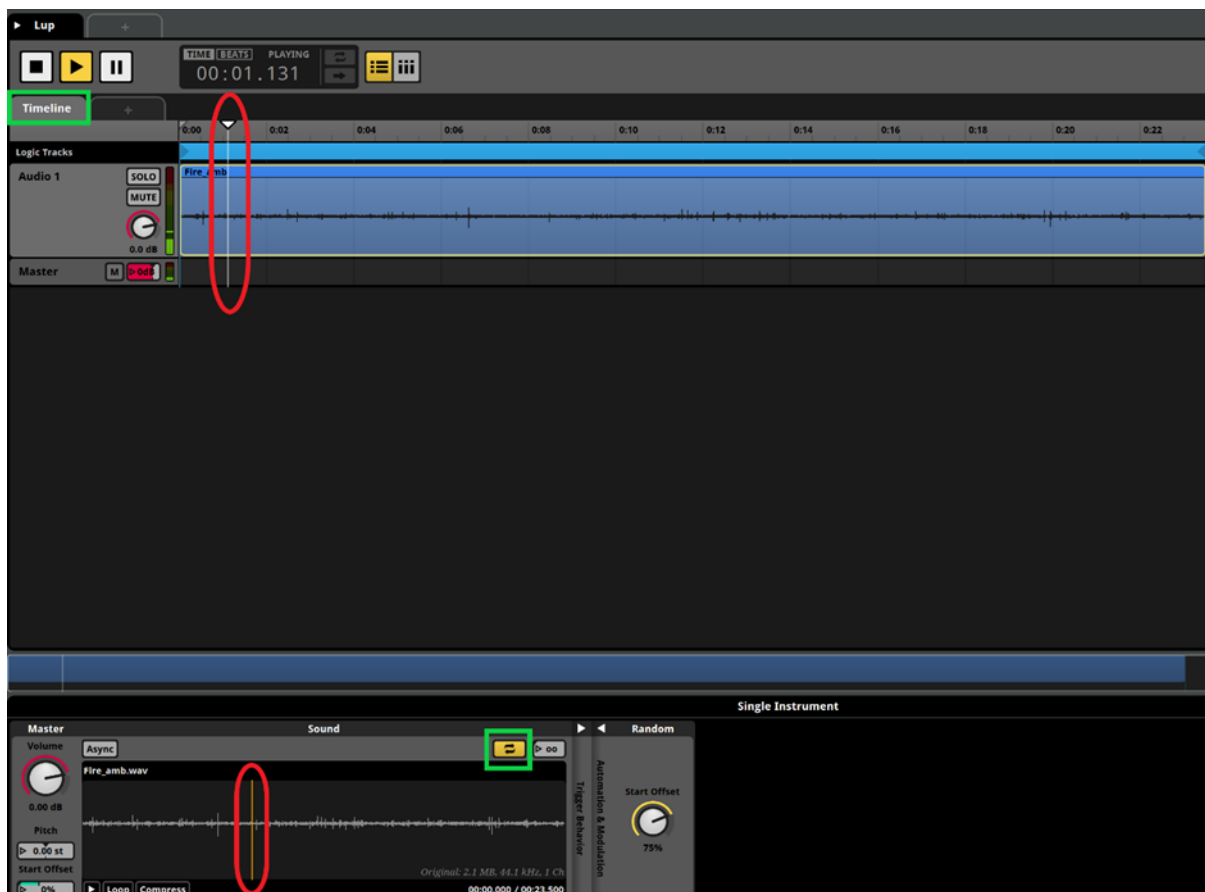
Slika 2.20. Postavka randomizacije početka lupa



Slika 2.21. Preporučena vrednost randomizacije

Treba da se napravi razlika između opcija i funkcija koje su nam dostupne u različitim prozorima. Ukratko, može da se kaže da u *edit* prozoru manipulišemo audio-sadržajem na nivou audio-traka dok u *deck* prozoru manipulišemo na nivou audio-fajlova. Npr., u *edit* prozoru možemo da pojačamo, odnosno smanjimo jačinu audio-fajla povlačeći *volume* regler na desnu ili levu stranu, ali ne možemo da je randomizujemo kao što smo to činili u *iventu* koraka. Ovakve vrste radnji obavljamo u *deck* prozoru. Na sličan način možemo da poredimo i postavku randomizacije početka lupa. U *edit* prozoru smo postavili lup regiju koja kontroliše kretanje kursora – drugim rečima, kada pokrenemo plejbek (što činimo pritiskom na taster razmak) kursor se kreće sa leva na desno i u trenutku kada dođe do kraja lupa vrati se na početak istog. Dakle, ovde besprekidno ponavljanje funkcioniše po principu regije zahvaćene plavom trakom.

Randomizacija početka audio-fajla utiče na ovaj neprekidni tok tako što fajl svaki put kreće iz drugačije tačke. U praktičnom smislu to znači da u slučaju da lup krene npr. od svoje šeste sekunde, u *edit* prozoru će kursor najnormalnije da krene od početka lupa dok ćemo u *deck* prozoru da vidimo realno vreme otpočinjanja, tj. da je lup krenuo od svoje šeste sekunde. Problem će da nastane onog trenutka kada lup dođe do kraja svog trajanja u *deck* prozoru dok u *edit* prozoru još uvek traje. Ovaj diskurs će da proizvede efekat ‘praznog prostora’ tj., imaćemo prekid u emitovanju audio-sadržaja. Ovo se izbegava tako što lupu i u *deck* prozoru zadamo da se ponavlja beskonačno. To činimo klikom na dugme sa znakom sa dve strelice usmerene levo i desno u okviru prozorčića *Sound* (znak uokviren zelenim markerom na slici 2.22). Kada dugme zasvetli žuto znamo da smo uspešno uključili ovu funkciju i sada se na dugmetu sa desne strane pored, u kojem je dotad stajao broj jedan (što označava broj ponavljanja), pojavio znak beskonačnosti što označava beskonačan broj ponavljanja.



Slika 2.22. Odnos realnog i randomizovanog protoka vremena

Dakle, na ovaj način smo obezbedili beskonačno ponavljanje audio-fajla na nivou 'realnog vremena'. Ono se odnosi na *edit* prozor – sama reč *Timeline* (znak uokviren zelenim markerom na slici 2.22) nam govori da je reč o regularnom protoku vremena. Pored toga, takođe smo obezbedili beskonačno ponavljanje na nivou nepredvidivog randomizovanog toka, što se odnosi na *deck* prozor. Na slici 2.22 takođe može da se uoči razlika (označeno crvenim markerom) između pozicije kursora u *edit* i *deck* prozoru. Ovde se vidi da je kursor u *edit* prozoru negde na početku lupa dok se u *deck* prozoru nalazi negde na trećini. Suština jeste da je sve vreme u pitanju jedan te isti lup, tj. audio-fajl, što se u *edit* prozoru grafički prikazuje upravo kroz kretanje kursora u okviru lup-regije. Kao što je već rečeno, lup treba da zamislimo kao krug u kojem iz bilo koje tačke da počnemo, na kraju ćemo se vratiti na tu istu tačku, što je grafički takođe prikazano u *deck* prozoru. Promenom početne tačke lupa se ne menja njegova fizionomija u smislu trajanja audio-fajla – drugim rečima, dužina trajanja ostaje dosledna trajanju koje vidimo u *edit* prozoru, tj. trajanju lup-regije. U praktičnom smislu, odnosno u zamišljenom primeru, ovo bi značilo da kako se učita scena u video-igri koja u sebi ima npr.

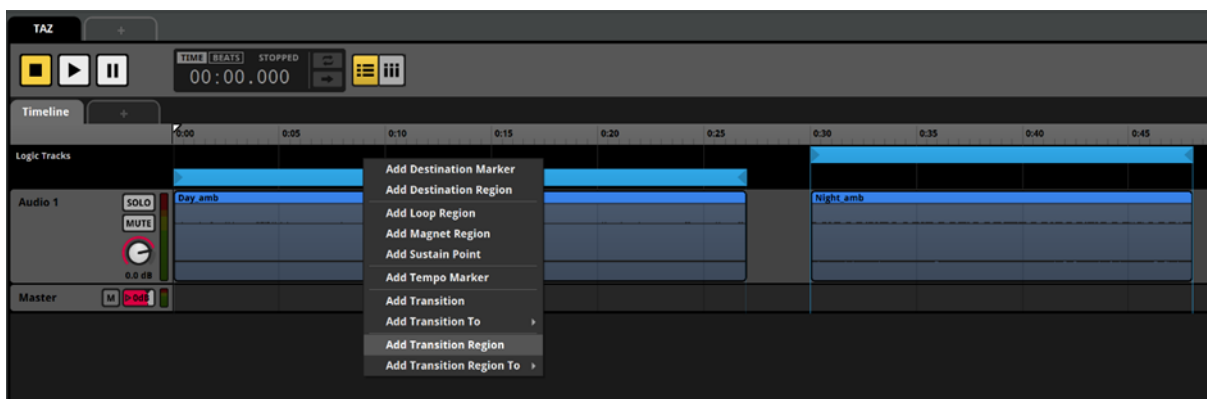
deset baklji, svaka od njih će da učita jedan te isti iverent. Ono što je različito jeste to što će svaka od tih baklji da učita iverent sa različitom početnom tačkom lupa, stvarajući na taj način iluziju da svaka baklja koristi nešto različiti audio-fajl [6]. Ovakav postupak randomizacije u video-igramama je jako koristan jer istovremeno stvara utisak zvučnog varijeteta a štedi prostor [4].

Iverent 3 - Tranzicija ambijentalnog zvuka

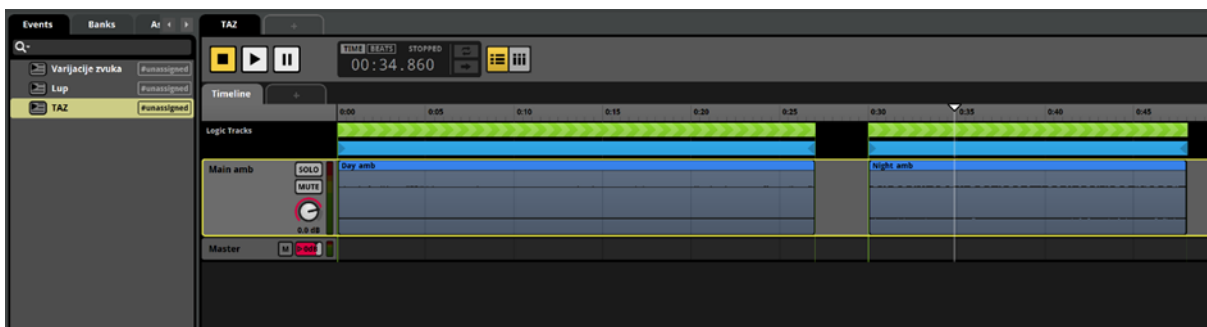
U ovom iverentu pored već poznatih pojmova kao što su lup-regija i multi-instrument uvodimo i tranzicionu regiju, destinacioni marker i asinhroni tip instrumenta. Zadatak je da se audio-sadržaj implementira na takav način da se pod dejstvom određenih vrednosti parametra smenjuju audio-fajlovi. Konkretno na našem primeru imaćemo smenu ambijentalnog zvuka dnevne i noćne atmosfere. Dakle, zadatak je da napravimo tranziciju iz dnevnog u noćni ambijent i obrnuto. Imamo dva glavna ambijentalna zvuka i to su audio-fajlovi *Day_amb* i *Night_amb*. Oba predstavljaju lup i oba su praćena sa po jednom grupom sporednih ambijentalnih zvukova. *Day_amb* prate audio-fajlovi: *amb_flies01*, *amb_flies02* i *amb_flies03*; dok *Night_amb* prate sledeći: *amb_magic_wind01*, *amb_magic_wind02* i *amb_magic_wind03*.

Za početak svakako treba da napravimo novi iverent. U pitanju je *2D Timeline event*, a to je zato što se ovakav tip iverenta opet vezuje za samog igrača. Već možemo da povučemo paralelu između iverenta koraka i tranzicije ambijentalnih zvukova – dakle, ono što ih spaja jeste direktno vezivanje za igrača. Sva objašnjenja u vezi sa tim šta se dešava sa zvukom po pitanju promene pozicije u 3D-prostoru važe i ovde. U audio-traku koja nam se postavkom novog iverenta automatski kreirala, postavljamo oba glavna ambijentalna zvuka. Dakle, imamo audio-fajl namenjen za glavni dnevni ambijent i zove se *Day_amb* (ubuduće dnevni lup) i fajl *Night_amb* (ubuduće noćni lup) namenjen za glavni noćni ambijent. Treba da se vodi računa o tome da se svaki fajl zasebno prevuče u projekat, zato što će u suprotnom da nastane multi-instrument. Na svaki od ovih audio-fajlova postavljamo lup-regiju kako bi obezbedili neprekidan tok. Dalje, površ svake lup-regije treba da se postavi i već pomenuta tranziciona regija. Kako bi napravili prostor da možemo komotno da dodamo tranzicionu regiju, treba da se ima u vidu da svi tipovi traka u okviru *logic tracks* mogu da se pomeraju gore-dole i da im se podešavaju pozicije međusobno. To činimo tako što držimo levi klik na željenoj traci i pomeramo je u željenom smeru. Konkretno u našem slučaju držaćemo levi klik na lup-regiji i pomerićemo je naniže. Ovo će da stvori slobodan prostor za tranzicionu regiju. Treba da se ima

na umu da nju ne možemo, kao što je to slučaj sa lup-regijom, da napravimo tako što desnim klikom na audio-fajl otvorimo padajući meni i izaberemo *New Loop Region* (v. način br. 2 na str. 17). Stoga, desnim klikom na *logic tracks* otvaramo padajući meni iz kojeg biramo opciju *Add Transition Region* (slika 2.23). Ovo će napraviti žuto-zelenu traku koju potom, na isti način kao lup-regiju, levim klikom prevlačimo na levu, odnosno desnu stranu. Ovim regulišemo njeno trajanje koje ćemo isto kao i kod lup-regije da podesimo prema trajanju samog audio-fajla. Ovaj postupak treba da se primeni na oba audio-fajla, dakle *Day_amb* i *Night_amb*. S obzirom da ćemo u ovom iventu da kombinujemo glavne i sporedne ambijentalne zvukove, to znači da ćemo da imamo više od jedne audio-trake, tačnije dve. Shodno tome bi bilo dobro da ih imenujemo. Audio-traku sa glavnim ambijentalnim zvukovima ćemo da nazovemo *Main amb*. Dosadašnji postupak bi trebalo bi da izgleda kao na slici 2.24.



Slika 2.23. Postavka tranzicione regije



Slika 2.24. Izgled tranzicionih regija

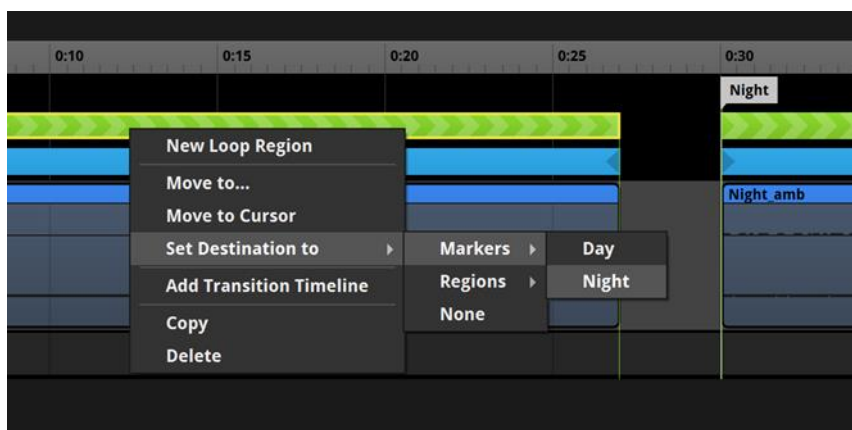
Destinacioni markeri

Tranziciona regija na sličan način kao što je to slučaj sa lup-regijom, označava opseg delovanja. U praktičnom smislu ovo znači da tranzicija može da se desi u bilo kom trenutku naznačene regije. Ona, kao što i sam naziv nagoveštava, služi tome da se izvrši tranzicija iz jedne tačke u drugu – konkretno u našem slučaju između ova dva ambijentalna zvuka. Kako bi tranziciona regija znala gde treba da izvrši tranziciju, moramo da je pravilno uputimo. Stoga, treba da napravimo posebnu vrstu markera koji se zove destinacioni marker. On služi kao neka vrsta putokaza u iventu. Na sam početak dnevnog lupa ćemo da postavimo marker koji ćemo da nazovemo *Day*. To činimo na isti način kao što postavljamo lup i tranzicionu regiju – prvo napravimo prostor u *logic tracks* tako što naniže pomerimo obe trake; zatim desnim klikom na *logic tracks* otvorimo padajući meni iz kojeg biramo *Add Destination Marker* (slika 2.25).



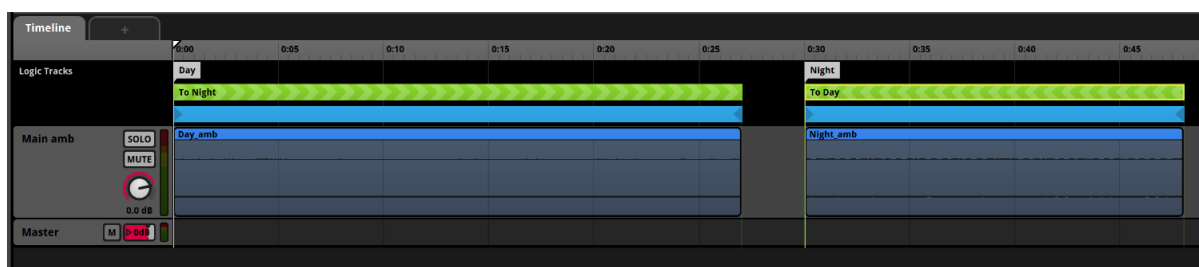
Slika 2.25. Postavka destinacionog markera

Isti postupak ćemo da sprovedemo i za noćni lup, a marker će da se zove *Night*. Treba strogo da se vodi računa o tome da se markeri postave tačno na početak lupova. Sada je vreme da usmerimo tranzicije – dnevni lup ćemo da usmerimo na marker *Night*, a noćni lup na marker *Day*. To radimo tako što desnim klikom na tranzicionu regiju otvorimo meni iz kojeg biramo: *Set Destination To – Markers*, i onda shodno lupu koju smo selektovali biramo *Day*, odnosno *Night* (slika 2.26).



Slika 2.26. Odabir destinacije

Dakle, ako selektujemo tranzicionu regiju koju smo postavili na dnevni lup, biramo marker *Night* i obrnuto, u slučaju da selektujemo tranzicionu regiju postavljenu na noćni lup, biramo *Day*. Kada sve ovo odradimo, na tranzicionim regijama bi trebalo da nam se ispišu zadate destinacije. Krajnji rezultat bi trebalo da izgleda kao na slici 2.27.

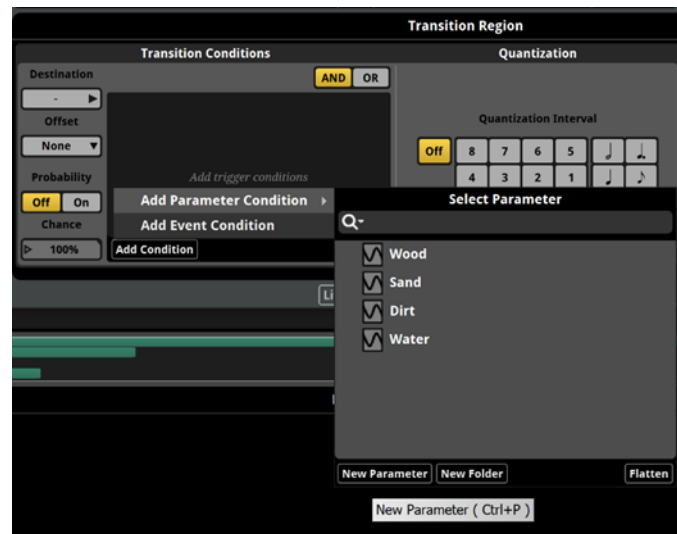


Slika 2.27. Izgled postavljenih tranzicija

Postavka parametara na tranzicionu regiju

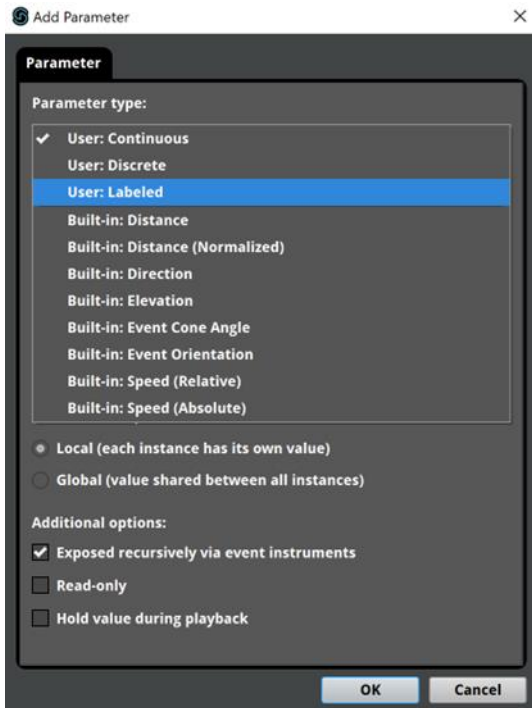
Kada bi probali da pustimo dnevni lup iz bilo koje tačke u okviru lup-regije, kursor bi uprono i nekontrolisano odlazio na marker *Night*. Isto bi se desilo i u slučaju da pustimo noćni lup. To je zato što još nismo postavili nikakav parametar i uslov koji bi kontrolisao delovanje tranzicije, i samim tim tranzicije se dešavaju bezuslovno. Ovde ćemo i parametar i uslov da postavimo na nešto drugačiji način, tj. za razliku od situacije sa inventom koraka, postaviti ćemo ih u samom *deck* prozoru. Kada selektujemo tranzicionu regiju, automatski nam se otvara *deck* prozor. U njemu sada možemo da postavimo parametar i uslov na samu traku. To činimo tako

što u donjem levom uglu prozorčica *Transition Conditions* kliknemo na *Add Condition* a zatim pređemo mišem preko *Add Parameter Condition*. Ovaj postupak otvara listu svih parametara koje smo već napravili. Svakako možemo da koristimo sve parametre koji su nam već na raspolaganju, ali s obzirom da nam je potreban novi parametar za potrebe ovog inventa, biramo *New Parameter* (slika 2.28). Ovo dalje otvara prozor *Add Parameter* na isti način kao što se to dogodilo kada smo pravili parametre za invent koraka.

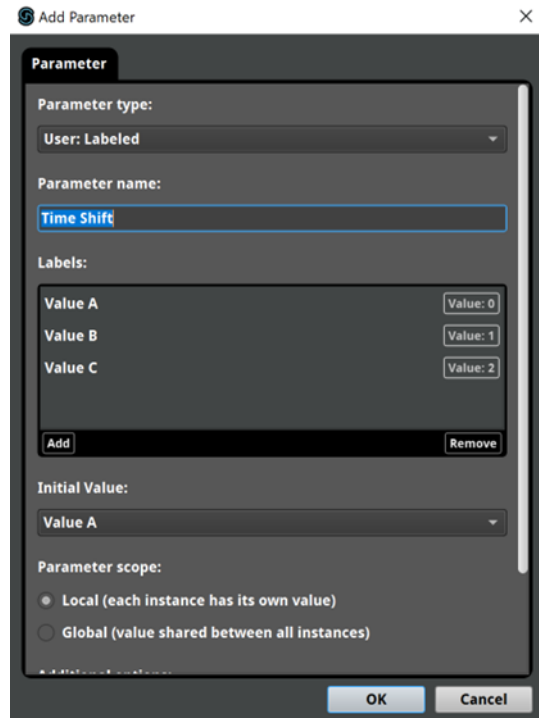


Slika 2.28. Dodavanje parametra iz *deck* prozora

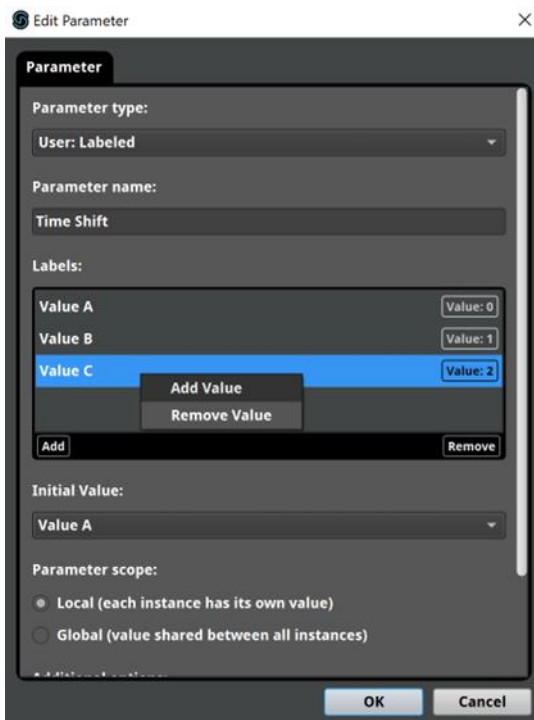
Ovaj parametar će imati nešto drugačiju postavku. Dosad smo u padajućem meniju *Parameter type* birali *User: Continuous*. Sada klikom otvaramo padajući meni i biramo *User: Labeled* (slika 2.29). Parametar ćemo da nazovemo *Time Shift*. Kada izaberemo ovakav tip parametra, u prozorčicu *Labels* po automatizmu imamo tri različite vrednosti: *Value A*, *B* i *C* (slika 2.30). S obzirom da su nam potrebne samo dve vrednosti, tj. *Day* i *Night*, izbrisaćemo *Value C*, a to činimo tako što desnim klikom na *Value C* otvorimo padajući meni i biramo *Remove Value* (slika 2.31). Dvostrukim klikom na *Value A* i *B* možemo da menjamo njihova imena – *Value A* ćemo da nazovemo *Day* a *Value B* *Night* (slika 2.32).



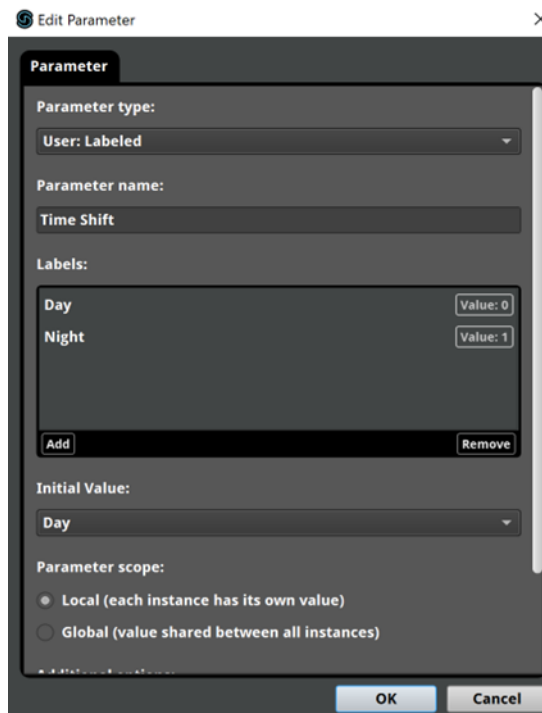
Slika 2.29. Odabir tipa parametra



Slika 2.30. Predefinisane vrednosti parametra

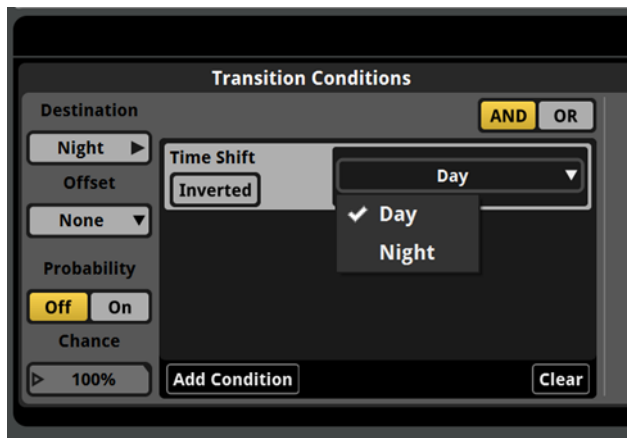


Slika 2.31. Brisanje vrednosti parametra



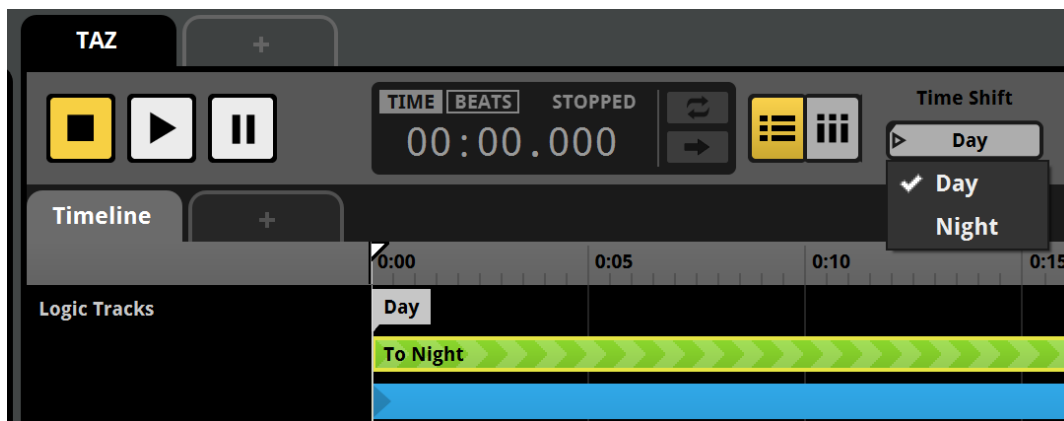
Slika 2.32. Imenovanje vrednosti parametra

Kada sve ovo odradimo, kliknemo na dugme *OK* i sada je parametar spreman za postavku koja je vrlo jednostavna. U zavisnosti od toga koju tranzicionu regiju selektujemo, kao uslov postavljamo vrednost *Day*, odnosno *Night*. Važno je da se obrati pažnja na to da smo u demonstraciji prvo selektovali tranzicionu regiju koja vodi na marker *Night*, dakle, u pitanju je regija na kojoj smo i napravili sam parametar *Time Shift*. Kada postavljamo parametar na tranzicionu regiju koja vodi u destinacioni marker *Day*, svakako nećemo ponovo da pravimo isti parametar, već ćemo iz padajuće liste parametara da biramo već postojeći *Time Shift*, zato što on kontroliše obe tranzicije. Dakle, na obe tranzicione regije postavljamo isti parametar ali biramo različite vrednosti – ako tranziciona regija vodi u destinacioni marker *Night* onda biramo vrednost *Night*, a ako vodi u *Day*, onda biramo vrednost *Day* (slika 2.33).



Slika 2.33. Postavka uslova

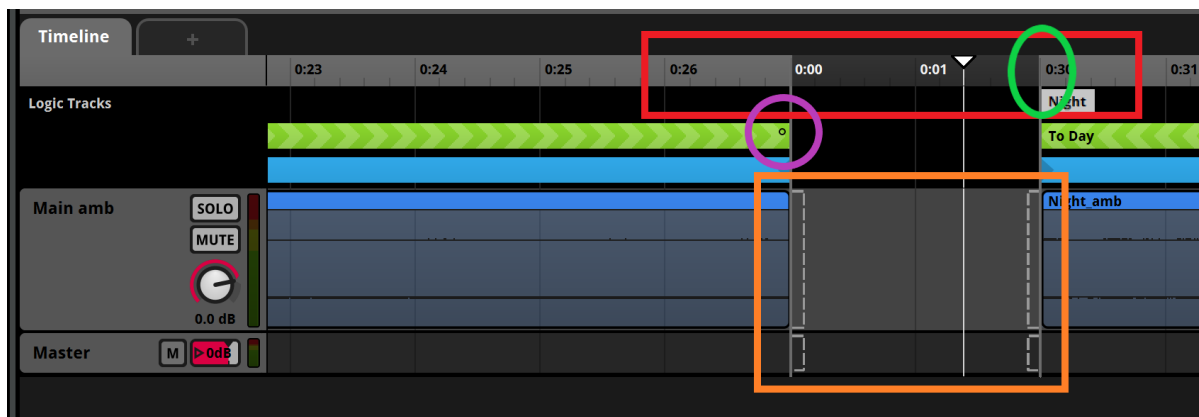
Primećuje se da nam se iznad *edit* prozora napravilo dugme parametra *Time Shift* (slika 2.34). Sada na njemu možemo da menjamo vrednosti parametra kako bi testirali da li nam tranzicije funkcionisu kako treba. Dakle, kao što smo pre puštali lup-regije i primetili da nam se tranzicije dešavaju bezuslovno, sada ćemo primetiti da se tranzicije dešavaju samo kada se ispune određeni uslovi. Ako trenutno emitujemo dnevni lup a pritom je aktivna vrednost *Day*, ništa neće da se promeni. Lup će nastaviti da se emituje sve dok se vrednost parametra na promeni u *Night*, kada će se kursor pomeriti na istoimeni destinacioni marker. Sve ovo isto važi i za tranziciju u suprotnom smeru.



Slika 2.34. Testiranje rada parametra i tranzicija

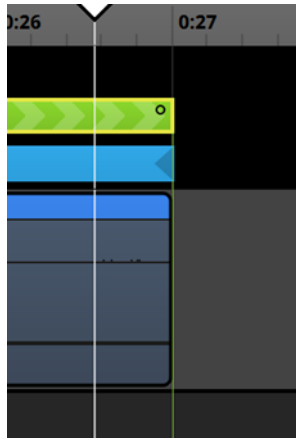
Tranzicioni prostor

Sledeće što je neophodno da se postavi po pitanju tranzicija jeste tranzicioni prostor. Ovo je nov pojam i predstavlja tzv. imaginarni prostor koji služi da podesimo način na koji se tranzicija dešava. Testirajući tranzicije primetili smo da se one dešavaju u skladu sa vrednostima parametra ali ishitreno – drugim rečima, kada se desi tranzicija kursor jednostavno ‘skoči’ na željenu destinaciju bez ikakvog finog prelaza. Upravo zbog već mnogo puta pomenutog nepredvidivog toka dešavanja u video-igri, uvek treba da se teži ka tome da se promene u audio-sadržaju koje su uslovljene promenama u video-igri, dešavaju što logičnije i neprimetnije [4]. Stoga ćemo da napravimo tranzicioni prostor na obe tranzicione regije. On se otvara dvostrukim klikom na tranzicionu regiju (slika 2.35). Dosad smo u više navrata primetili da u svakom iventu u gornjem levom uglu imamo tab koji se zove *Timeline*, što označava regularan protok vremena. Kao i u svakom softveru za audio-produkciju, vreme se broji od nulte sekunde pa nadalje. Možemo da kažemo da ovo predstavlja ‘realno vreme’. Kada otvorimo tranzicioni prostor vidmo da vreme ponovo počinje da se meri od nulte sekunde, iako se poziciono nalazi na samom kraju lupa, što je zapravo 0'27" – dakle, primećuje se da tranzicioni prostor ima sopstveno merenje protoka vremena (uokvireno crvenim markerom na slici 2.35).



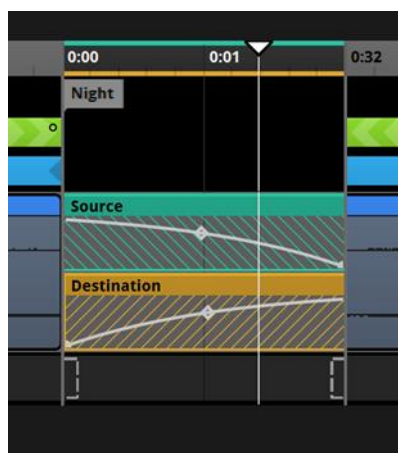
Slika 2.35. Tranzicioni prostor

Ova pozicija nema nikakve veze sa realnim vremenom zato što tranzicioni prostor meri vreme koje je potrebno da bi se tranzicija izvršila. Kada kursor miša postavimo na skroz desnu stranu merne skale (zaokruženo zelenim markerom na slici 2.35), on promeni oblik i dobija levu i desnu strelicu. To znači da sada možemo da podešavamo trajanje izvršenja tranzicije. Ako povlačimo na desno produžavamo trajanje i suprotno, ako povlačimo na levo smanjujemo. Hajde da postavimo da nam trajanje bude dve sekunde (slika 2.35). Ovo je zgodan trenutak za digresiju, tj. da napomenemo jedan koristan savet koji može da spreči potencijalne greške pri izradi zadatka. Kada napravimo tranzicioni prostor, sa leve strane se prikaže kružić (zaokružen ljubičastim markerom na slici 2.35). Treba da se ima u vidu da nam je tranzicioni prostor aktivan dokle god stoji ovaj znak. To je bitno iz razloga što može da se desi da slučajno na tranzicionoj regiji napravimo tranzicioni prostor i da onda ne možemo da shvatimo zašto odjednom nastupi tišina. To je upravo zato što je kursor ušao u tranzicioni prostor koji kao što već znamo, ima svoj sopstveni protok vremena i trajanje. Samim tim, mi zapravo čujemo prazan tranzicioni prostor. Dvostrukim klikom na tranzicionu regiju mi samo zatvaramo tranzicioni prostor ali se primećuje da je znak kružića i dalje prisutan (slika 2.36). Ovo znači da smo samo zatvorili tranzicioni prostor ali da je on i dalje prisutan u iventu. U slučaju da smo greškom napravili tranzicioni prostor, brišemo ga tako što dok je otvoren, postavimo kursor miša na skroz desnu stranu merne skale (dakle identično kao i kada postavljamo dužinu trajanja tranzicije) i onda povučemo sa leva na desno sve dok se tranzicioni prostor u potpunosti ne zatvori. Nakon ovog postupka bi trebalo da bude izbrisan i da je kružić samim tim nestao.



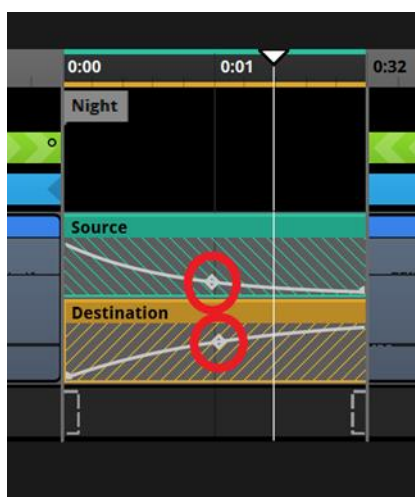
Slika 2.36. Zatvaranje i brisanje tranzicionog prostora

Vratimo se malopredašnjoj diskusiji. Na slici 2.35 vidimo narandžastim markerom uokvirene isprekidane linije audio-fajlova koje se nalaze na obodima tranzicionog prostora. Kada mišem pređemo preko njih kursor promeni oblik i počinje da liči na ‘vratanca sa bravom’. To znači da sada možemo da ih pomeramo sa leva na desno i obrnuto. Povlačenjem isprekidane linije koja pripada dnevnom lupu otvara se ‘prostor’ tirkizne boje na kojem piše *Source* (ubuduće izvor osim ako je u pitanju direktna upotreba funkcije unutar softvera, slika 2.37). Dakle, ovo je polazište iz kojeg se vrši tranzicija pa se zato tako i zove. Međutim, povlačenjem isprekidane linije koja pripada noćnom lupu sa desna na levo se otvara ‘prostor’ narandžaste boje na kojem piše *Destination* (ubuduće destinacija osim ako je u pitanju direktan direktna upotreba funkcije unutar softvera, slika 2.37). I same reči objašnjavaju značenje ovog konteksta, dakle, pomeranjem isprekidanih linija podešavamo odnos između sadržaja iz kojeg i u koji se vrši tranzicija.



Slika 2.37. Izvor i destinacija

Tranzicioni prostor traje ukupno dve sekunde, tako da ćemo trajanje izvora i destinacije da postavimo u toj srazmeri, drugim rečima, izvor i destinaciju treba da prevučemo do krajnjih granica tranzicionog prostora. Primećujemo da kako prevlačimo jedno naspram drugog, da nam se tako prave krive – kako u izvoru tako i u destinaciji. Ovo se zove krosfejđ (engl. crossfade) i znači da se fajlovi međusobno stišavaju, odnosno pojačavaju. Izvor ima silaznu krivu što znači da se fajl stišava, što i jeste logično, s obzirom da se vrši tranzicija u novi ambijentalni zvuk pa samim tim više i ne treba da čujemo onaj koji je prethodio. Suprotno tome, destinacija ima uzlaznu krivu što znači da se fajl pojačava, što opet jeste logično, s obzirom da treba da se lagano uvede nov ambijentalni zvuk. Opše pravilo pri tranzicijama putem krosfejda jeste da je ono što je prethodilo manje bitno od onog što sledi [4]. Samim tim, kriva od izvora treba da bude zaoštrenija a od destinacije zaobljenija. To znači da će audio-sadržaj iz izvora da se brže stiša a iz destinacije brže pojača. Podešavanje oblika krive se radi tako što držimo levi klik miša na ‘kvačici u obliku dragulja’ (uokvireno crvenim markerom na slici 2.38) i pomeramo je gore-dole. Kada kvačicu pomeramo naviše, tako kriva postaje zaobljenija i obrnuto. S obzirom da se krosfejđ automatski postavlja tako da su mu obe krive zaobljene, jedino što treba da uradimo jeste da zaoštrimo krivu izvora (slika 2.38). Na kraju postavke odnos između izvora i destinacije treba da izgleda približno kao na već pomenutoj slici. Ovaj postupak treba da se primeni na obe tranzicione regije.



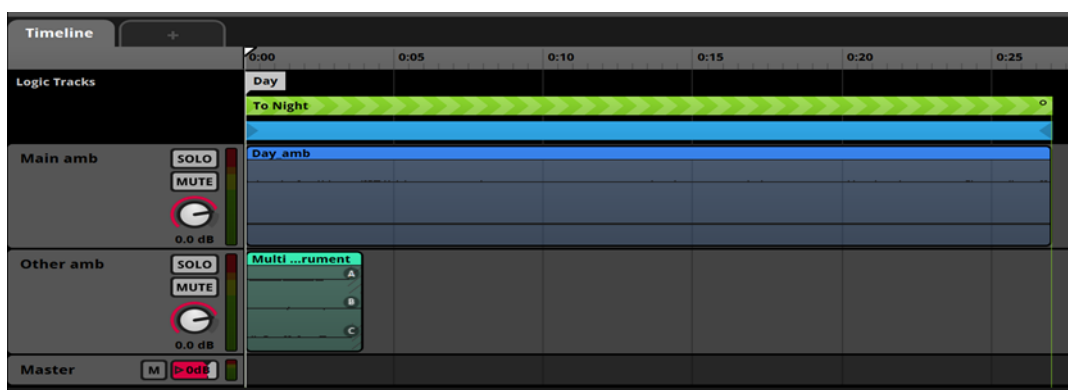
Slika 2.38. Podešavanje krosfejda između izvora i destinacije

Nakon ovih postavki koje se vezuju za glavne ambijentalne, treba da se pozabavimo i pratećim ambijentalnim zvukovima. Zovu se prateći iz razloga što predstavljaju sastavni ali ne obavezni deo lupa, tj. oni se javljaju samo kada se ispune određeni uslovi. Shodno tome, postojaće

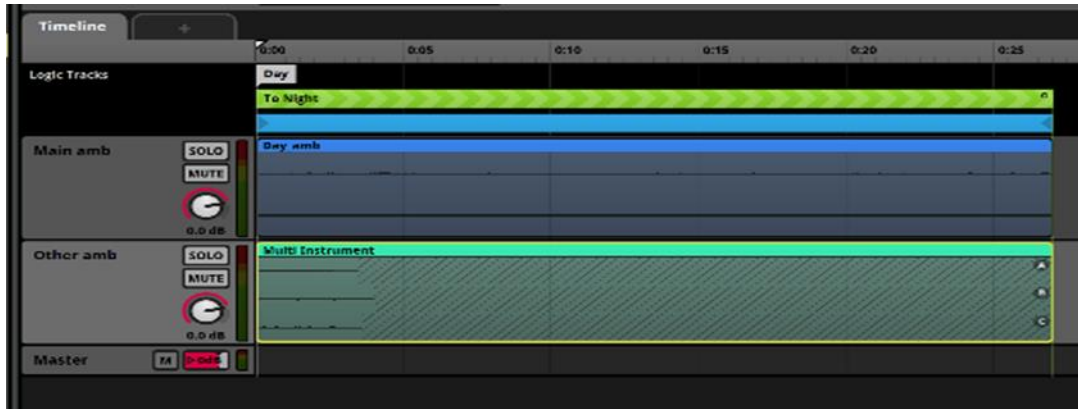
parametri čije će vrednosti da regulišu njihovo prisustvo, odnosno odsustvo. Kao što već znamo, prateće ambijentalne zvukove treba da smestimo u zasebnu audio-traku koju ćemo da nazovemo *Other amb*. S obzirom da imamo grupe zvukova koje predstavljaju varijacije jednog te istog zvuka, postavilićemo ih u multi-instrument. Svaki multi-instrument postavljamo u okviru iste audio-trake na sam početak odgovarajuće lup-regije. Dakle, u ovom iventu kombinujemo dva različita tipa zvuka – lup i varijacije.

Asinhroni instrument

Za početak, hajde da ubacimo grupu sporednih ambijentalnih zvukova koja prati dnevni lup. Da se podsetimo, nju čine sledeći audio-fajlovi: *amb_flies01*, *amb_flies02* i *amb_flies03*. Na slici 2.39 se primećuje da trajanje multi-instrumenta nije u srazmeri sa trajanjem lup-regije. Verovatno se pitamo na koji će to način multi-instrument da se usinhronizuje sa lup-regijom. Ako probamo da razvučemo multi-instrument sa leva na desno kako bi pokrili trajanje čitave lup-regije (što činimo na isti način kao kada podešavamo trajanje lup i tranzicionih regija) videćemo da u jednom trenutku, tj. po završetku trajanja multi-instrumenta, počinju da se javljaju kose linije (slika 2.40). To označava tišinu. Ako bi pustili audio-sadržaj od početka lupa, na samom početku bi čuli nasumice izabran fajl iz plejliste, a potom ništa više do narednog ponavljanja lupa.

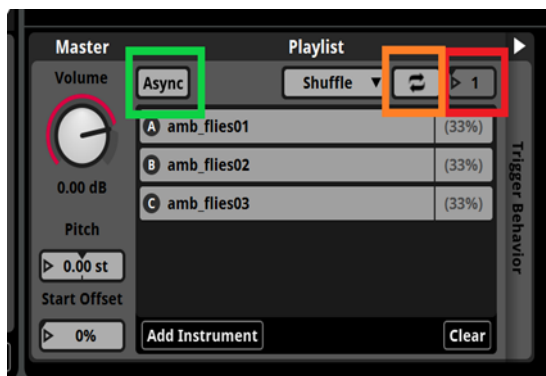


Slika 2.39. Multi-instrument



Slika 2.40. Adaptiranje trajanja multi-instrumenta prema lup-regiji

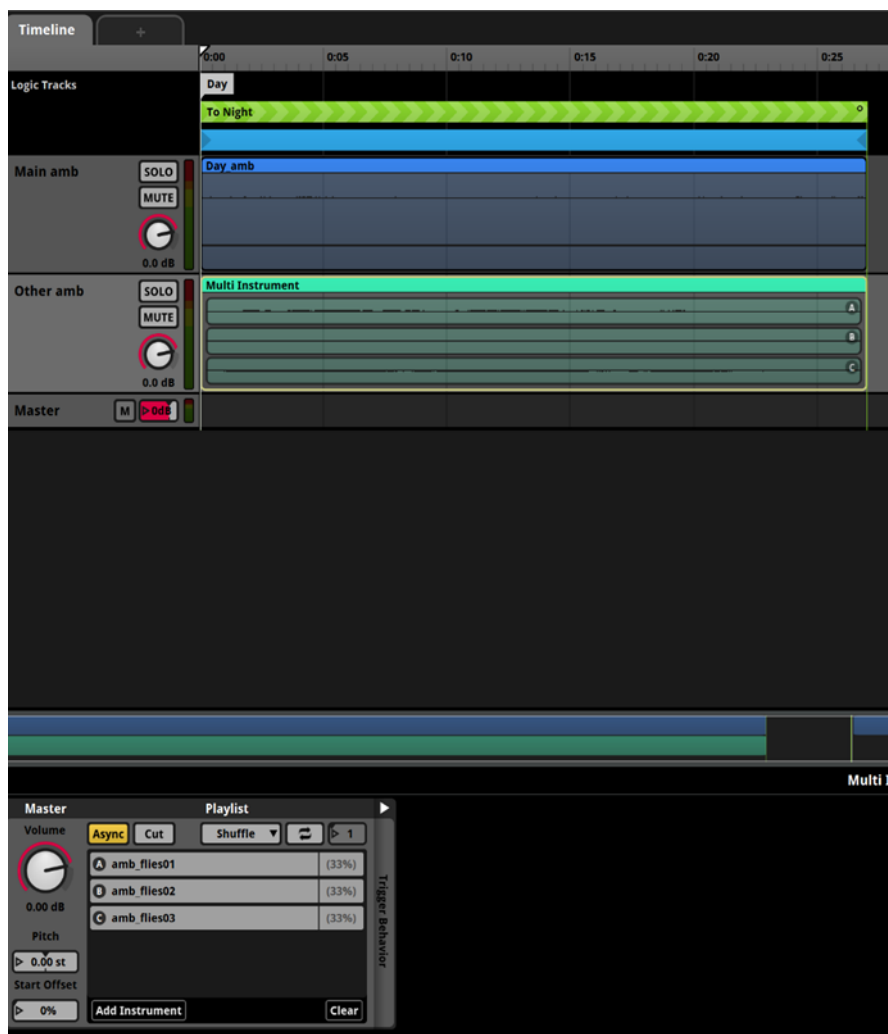
Kada selektujemo multi-instrument i otvori nam se njegov sadržaj u *deck* prozoru, u gornjem levom uglu prozorčića *Playlist* se primećuje dugme *Async* (uokvireno zelenim markerom na slici 2.41). Kada kliknemo, tj. aktiviramo to dugme, primećuje se da ono zasvetli žuto i da se onaj prazan prostor naizgled ispunio audio-sadržajem (slika 2.42). Naš instrument je upravo postao asinhron što znači da on koegzistira sa lup-regijom u kojoj se nalazi, a njegov izgled predstavlja samo vizuelni prikaz delovnja u okviru iste. Dakle, njegovo delovanje se prostire duž cele lup-regije. Ako bi pustili lup od početka, čuli bi jedan nasumice izabran fajl iz plejliste. Do novog emitovanja bi morali da čekamo da lup završi ceo krug i krene od početka.



Slika 2.41. Aktivacija/postavka asinhronog instrumenta

Isprva deluje kao da ne postoji razlika između asinhronog i običnog multi-instrumenta. Ako pogledamo bolje u prozorčić *Playlist* videćemo da nam je broj ponavljanja postavljen na jedinicu (uokvireno crvenim markerom na slici 2.41). Iz tog razloga se i jeste emitovao samo jedan fajl. Ako bi postavili da je broj ponavljanja veći od jedan, čuli bi onaj broj emitovanja

koliko smo i uneli. Ako postavimo da je broj ponavljanja beskonačan, onda će fajlovi da se emituju jedan za drugim sve dok se nalazimo u okviru lup-regije. To činimo tako što kliknemo na dugme pored onog u koje unosimo broj ponavljanja (uokvireno narandžastim markerom na slici 2.41). Kada kliknemo na njega ono će zasvetleti žuto, a dugme u koje unosimo broj ponavljanja će da dobije znak beskonačnosti. Dakle, u pitanju je ono isto dugme koje smo uključili u iventu lupa kako bi obezbedili neprekidan tok audio-fajla usled primene randomizacije početka lupa.



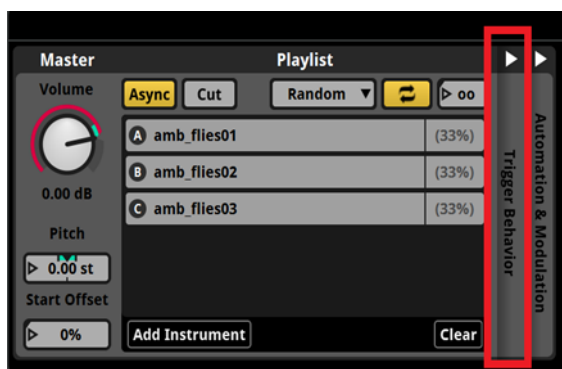
Slika 2.42. Izgled asinhronog instrumenta

Fajlovi se emituju jedan za drugim, dakle, kako jedan završi sa emitovanjem, tako drugi preuzima, i tako u nedogled. Ovakava vrsta ponašanja je tipična za asinhroni instrument i ne može da se postigne kod običnog instrumenta. Sve što smo od randomizacije primenili u iventu

koraka možemo i ovde da primenimo – dakle, randomizacija same plejliste, tj. redosled emitovanja fajlova, kao i randomizacija po uobičajenom metodu. Dalje, ceo postupak koji je demonstriran kroz primer na grupi sporednih ambijentalnih zvukova koja prati dnevni lup treba da se primeni i na drugu grupu koja prati noćni lup. Da se podsetimo, nju čine: *amb_magic_wind01*, *amb_magic_wind02* i *amb_magic_wind03*.

Postavka parametra na asinhroni instrument

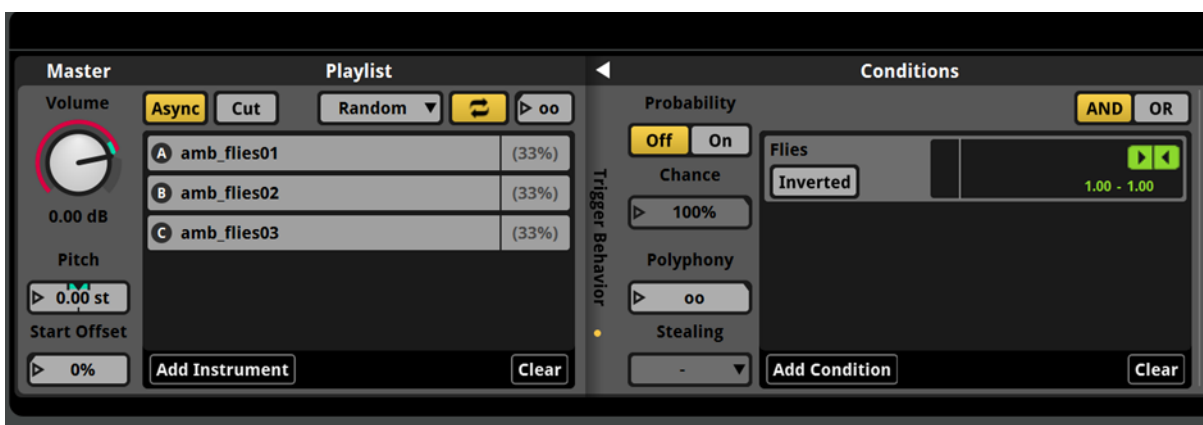
Zamislamo sada scenario u video-igri gde hoćemo da se zvuk insekata javlja samo onda kada se insekti pojave na sceni. Shodno tome, sledeće što treba da postavimo su parametri koji će da regulišu prisustvo, odnosno odsustvo pratećih zvukova. Dakle, za grupu zvukova *amb_flies* postavljamo parametar koji se zove *Flies*, a za grupu *amb_magic_wind* parametar *Magic Wind*. Oba parametra se postavljaju u *deck* prozoru – dakle, na isti način kao kada smo postavljali parametar na tranzicione regije. Jedina je razlika što ovde parametar postavljamo direktno na multi-instrument – samim tim, jako je bitno da nam u trenutku pravljenja parametra multi-instrument bude selektovan. Međutim, kada selektujemo multi-instrument, situacija nije identična kao kada smo parametar postavljali na tranzicionu regiju. Ovde nam se prozorčić *Conditions* ne pojavljuje po automatizmu, stoga treba da ga otvorimo. To činimo tako što kliknemo na strelicu koja se nalazi na vrhu i u ravni sa natpisom *Trigger Behavior* (uokvireno crvenim merkerom na slici 2.43).



Slika 2.43. Postavka uslova na multi-instrument

Ovo nam otvara dva prozorčića: *Conditions* i *Delay & Quantization* – nas trenutno zanima samo *Conditions*. Sada je postupak postavke parametra identičan kao u iventu koraka.

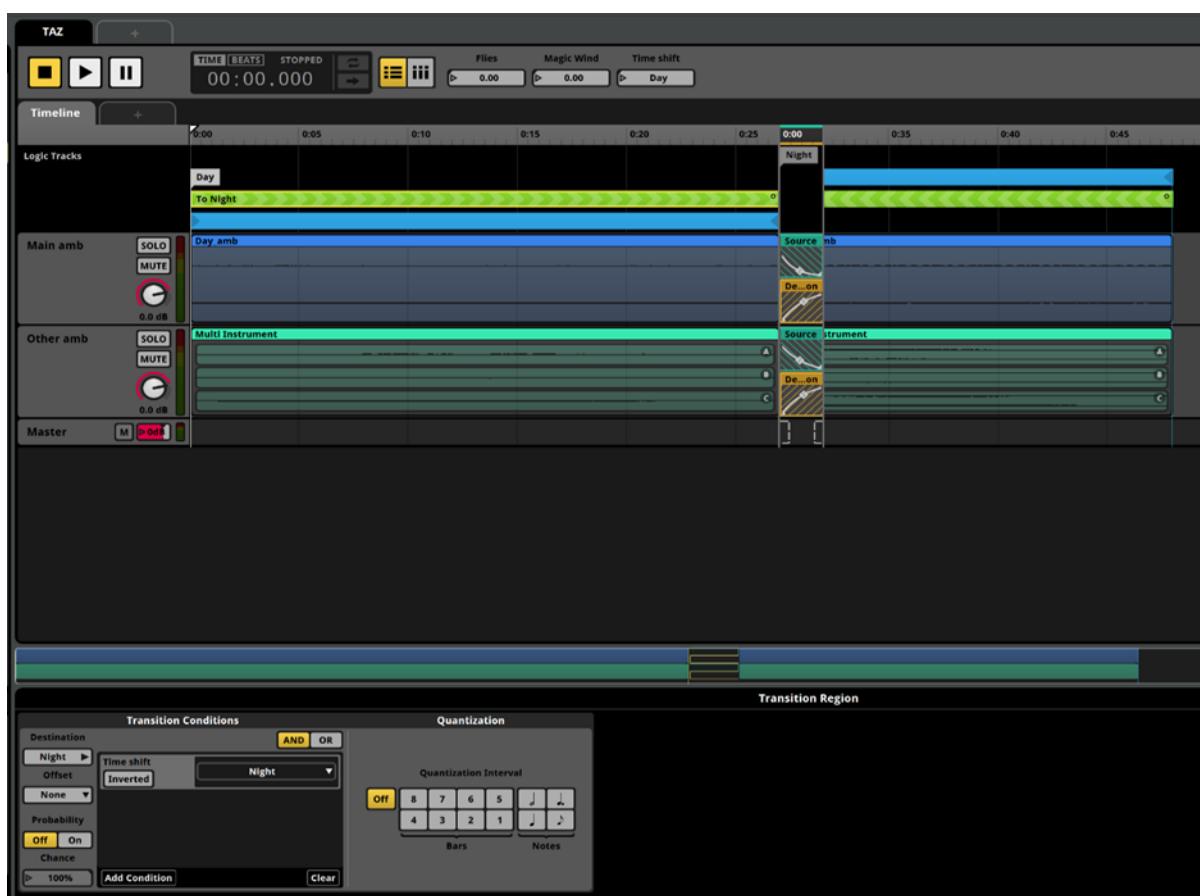
Nazvaćemo ga *Flies* a tip koji biramo jeste *User Continuous*. Vrednosti treba da se postave tako da minimalna bude nula a maksimalna jedan, gde nula predstavlja odsustvo a jedan prisustvo zvukova. Pošto smo upravo postavili parametar na multi-instrument, on nam se automatski pojavljuje u prozorčiću *Conditions*. Ono što treba da uradimo jeste postavka uslova. Za razliku od situacije sa glavnim ambijentalnim zvukovima gde smo parametar postavljali na tranzicionu regiju i služili se tipom parametra *User: Labeled*, ovde postavka uslova deluje malo drugačije. Primećujemo da ovde umesto padajućeg menija imamo slajder koji možemo da pomeramo levo-desno (slika 2.44) i koji sadrži čitav spektar vrednosti parametra. Mi želimo da se naš multi-instrument javlja samo kada je parametar *Flies* jednak jedinici. Shodno tome, desni slajder će da nam ostane tu gde jeste – na maksimalnoj vrednosti, tj. jedinici; dok ćemo levi da povučemo na desno i postavimo ga takođe tačno na jedinicu. Na ovaj način obezbeđujemo to da se eimovanje fajlova iz grupe *amb_flies* dešava isključivo kada je parametar *Flies* jednak jedinici.



Slika 2.44. Slajder parametra *Flies*

Ceo ovaj postupak se primnjeuje i na drugu grupu pratećih zvukova *amb_magic_wind*. Parametar ćemo da nazovemo *Magic Wind*. Kao što smo to radili sa glavnim ambijentalnim zvukovima, i prateće možemo da testiramo u okviru iventa. I ovde su nam se povrh *edit* prozora automatski pojavili slajderi parametara pomoću kojih lako možemo da menjamo njihove vrednosti i samim tim uvidimo da li nam uslovi zaista funkcionišu kako treba. Za razliku od parametra *Time Shift*, ovde ne postoji padajući meni sa vrednostima parametara. Parametre *Flies* i *Magic Wind* testiramo na isti način kao što smo to činili u iventu koraka. Ovo je i logično, zato što nam je u oba slučaja ista maksimalna i minimalna vrednost parametra.

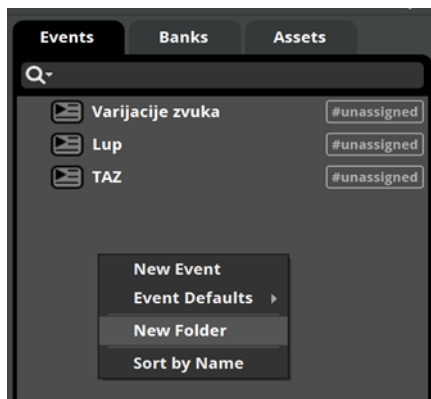
Poslednje što treba da uradimo jeste da i prateće ambijentalne zvukove uključimo u tranzicioni prostor. Iako oni ne predstavljaju obavezni deo lupa, u slučaju da su aktivni u toku same tranzicije, svakako i oni treba pravilno da se uključe i isključe, tj. da se krosfejduju. U suprotnom, ako ih ne bi uključili u tranziciju zvuk bi se samo prekinuo pri istoj i to bi zvučalo kao greška. Stoga treba da se postavi krosfejd između pratećih ambijentalnih zvukova identičan onome koji smo postavili na glavne, što podrazumeva postavku u oba smera tranzicije. Na kraju bio ceo iverent trebalo da izgleda kao na slici 2.45.



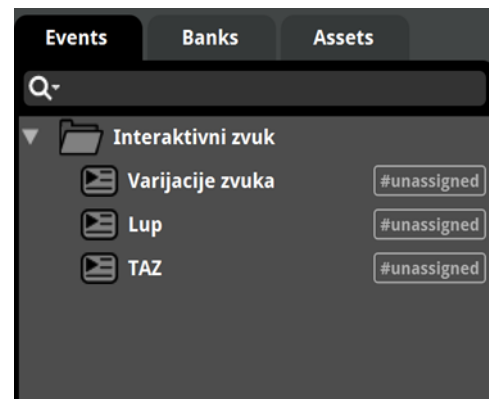
Slika 2.45. Izgled iverenta tranzicije ambijentalnih zvukova

Ovim iverentom smo završili deo zadatka koji se tiče interaktivnog zvuka. Radi preglednosti i organizacije unutar projekta, na kraju ćemo sva tri iverenta da prevučemo u folder koji ćemo da nazovemo *Interaktivni zvuk*. Folder se pravi tako što desnim klikom miša u *Events* brauzeru otvorimo padajući meni iz kojeg biramo *New Folder* (slika 2.46), a zatim upišemo naziv. Kako bi prevukli iverente, treba da držimo taster *shift* i selektujemo sva tri iverenta i tako selektovane ih prevučemo u folder koji smo upravo napravili. To bi na kraju trebalo da izgleda

kao na slici 2.47. Sada možemo da podižemo i spuštamo folder klikom na strelicu pored imena foldera. Organizovanje ivenata po folderima možda deluje besmisleno na našem primeru, ali je svakako dobra praksa, te je zato i pominjemo. U slučaju nekog većeg projekta gde bi postojalo mnoštvo ivenata različitih tipova, namena itd., pokazala bi se kao jako korisna [6, 7].



Slika 2.46. Dodavanje foldera



Slika 2.47. Izgled foldera sa iventima

2.2 INTERAKTIVNA MUZIKA

Uloga muzike u video-igrama je slična kao u filmskoj umetnosti [7]. Ona služi da podrži i osnaži celokupni vizuelni, narativni i dramski sadržaj. To znači da treba na pravi način da dočara mesto i vreme u kojem se radnja dešava; da nam predstavi likove iz video-igre (najčešće protagonistu), što znači da muzika vrlo često pored dočaravanja nekog ambijenta treba da predstavi i nečije unutrašnje, psihičko stanje. Dakle, muzika je uglavnom apstraktna i opisna, tj. nedijegetska – upravo takvom muzikom ćemo se mi baviti u našem zadatku [4].

Za razliku od dela zadatka koji se tiče interaktivnog zvuka gde smo imali tri ivenata, po pitanju muzike ćemo da imamo samo jedan, ali će on da bude znatno sadržajniji nego što su to bili iveni zvučnih efekata. Zadatak je da kroz ivent prenesemo, tj. sprovedemo zamišljeni scenario u isto tako zamišljenoj akcionoj video-igri. Zamislimo da naš protagonist prolazi kroz teritorije različitog nivoa pretnje – u pojedinim se kreće krajnje opušteno, jednostavno istražujući novu teritoriju bez ili sa vrlo malo pretnje od strane neprijatelja. Samim tim se

ovakav tip stanja u video-igri popularno zove *explore* (već smo pominjali na samom početku u uvodu) i predstavlja stanje niskog stepena pretnje, odnosno tenzije. Samim tim je i muzika koja prati ovakav tip stanja prilagođena njemu. Nadalje ćemo ovo stanje, kao i muziku koja ga prati da zovemo stanje istraživanja, odnosno istraživačka muzika. Ona je uglavnom relativno mirnog karaktera, repetitivna i sve vreme srednje tihe dinamike, što odgovara ulozi tzv. ‘pozadinske muzike’. Sa druge strane, još jedan razlog zašto je ona takvog karaktera jeste i to što treba da pruži izvestan kontrast u odnosu na akcionu muziku. Kako smo ovo pomenuli, stižemo i do stanja suprotnom istraživanju. Za razliku od njega, ovo stanje se vezuje za trenutak kada igrač prestaje da istražuje teritoriju i kada je skoncentrisan na sopstveno preživljavanje usled napada od strane neprijatelja. Stoga se ovakav tip stanja u video-igri popularno zove *action* ili *combat* (i ovo smo pominjali u uvodu) i predstavlja stanje visokog stepena pretnje, tj. tenzije. Muzika koja prati ovakav tip stanja, kao što je već rečeno, predstavlja kontrast u odnosu na istraživačku. Ona je znatno intenzivnija, glasnija, vrlo često bržeg tempa, prožeta napetim ritmom, dramatičnom melodijom, itd. Nadalje ćemo i ovo stanje, kao i muziku koja ga prati da zovemo stanje akcije, odnosno akciona muzika. Možemo da zaključimo da je uloga muzike u video-igri višestruka – pored toga što treba pravilno da odgovori na dešavanje u video-igri koje se tiče same estetike, dramaturgije, scenografije, psiho-emotivnog stanja protagoniste, itd.; ona takođe treba da prenese jasnu poruku igraču u smislu upozorenja i obaveštenja na novonastalu promenu i da ga pripremi za nadolazeći izazov. Poslednje stanje koje treba da se pokrije je stanje nakon akcije/borbe a u ređim slučajevima i nakon istraživanja. Postoje dva ishoda:

- 1) Igrač uspešno savladava neprijatelje i vraća se nazad u stanje istraživanja.
- 2) Igrač ne uspeva da savlada neprijatelje tj., ishod se završava smrću igrača. Upravo ovo je to poslednje stanje i zove se smrt (popularan naziv *death*, u svetu video-igara). Ono dakle prekida ustaljeni tok smene stanja istraživanja i akcije, i s obzirom da je igrač ‘izgubio igru’, sada ima priliku da se odluči da li želi da proba ponovo. Postoji i ređi slučaj, kada igrač iz stanja istraživanja naglo pređe u stanje smrti. Ovo se dešava kada igrač biva iznenada napadnut od strane neprijatelja sa momentalnim smrtnim ishodom.

Veoma bitan deo muzičkog zadatka je da predstavimo različite tehnike tranzicija/promena. Prema mišljenju Majkla Svita (*Michael Sweet*) postoje tri glavne tehnike pisanja interaktivne muzike za video-igre [4], a način pisanja se direktno odnosi i na način same implementacije. Dosta smo o njima pričali u okviru teorijskih predavanja ali da se podestimo:

- 1) Već dosta pominjani lup – s obzirom na svoje karakteristike, moglo bi se reći čak fundament audio-sadržaja u video-igramama.

- 2) Horizontalno resekveciranje (engl. horizontal re-sequencing) se odnosi na tranziciju/promenu u smislu prelaska iz jednog muzičkog segmenta u drugi – dakle, ovde se radi o horizontalnom tipu tranzicije. Već postojeći ivalent tranzicija ambijentalnih zvukova bi mogao da posluži kao primer – dakle, ovde se vrši prelazak iz jednog ambijentalnog zvuka u drugi. Identična situacija je i sa muzikom samo što ona svakako ima svoje specifičnosti o kojima će kasnije da bude reči.
- 3) Vertikalno remikosvanje (engl. vertical re-mixing) se se odnosi na tranziciju/promenu u smislu dodavanja, odnosno oduzimanja određenih slojeva. To bi u praktičnom smislu značilo da možemo da imamo određene okidače (engl. triggers) u video-igri koji iniciraju dodavanje određenog sloja ili više njih. Npr., ulazak na neki specifičan deo teritorije može da inicira dodavanje još jedne instrumentalne deonice, tj. sloja i obrnuto; napuštanje tog dela teritorije može da inicira oduzimanje tog istog sloja, odnosno povratak na malopređašnje stanje [2]. Opet, već postojeći ivalent tranzicija ambijentalnih zvukova bi mogao da posluži kao primer, gde bi sporedni ambijentalni zvukovi igrali ulogu dodatnog sloja – dakle, oni su prisutni samo pod dejstvom određenih parametara i pridružuju se glavnom ambijentalnom zvuku kao dodatni sloj. Onog trenutka kada parametar više nije aktivan, inicira se povlačenje sporednih ambijentalnih zvukova, tj. povratak na inicijalno stanje.

Popis audio-fajlova i njihovih namena

Svako od pomenutih stanja nosi sa sobom određene audio-fajlove. Unutar stanja postoji još mnoštvo dešavanja i svaki audio-fajl ima specifičnu namenu. Važno je da su svi fajlovi, osim onog koji se vezuje za stanje smrti, pravljeni u formi lupa, tako da se to u buduće podrazumeva. Sledi popis stanja, audio-fajlova i njihovih namena:

1) Stanje istraživanja:

- *Explore_Loop_4-4_90bpm* – kao što smo u ivalentu tranzicija ambijentalnih zvukova imali glavne i sporedne audio-fajlove, tako i ovde imamo glavni i sporedni sloj. Ovaj fajl predstavlja glavni, tj. obavezni sloj.
- *Explore_AddLayer_Loop1_4-4_90bpm* i *Explore_AddLayer_Loop2_4-4_90bpm* – ova dva fajla predstavljaju dodatne slojeve koje ćemo da postavimo u multi-instrument. Njihova uloga nije da se aktiviraju pod dejstvom određenih parametara, već da obogate celokupan muzički sadržaj i ublaže efekat repetitivnosti u video-igramama koji smo dosta

pominjali. Dataljnije o ovome će da bude reči kasnije, ali je suština da se obavezni sloj *Explore_Loop_4-4_90bpm*, svaki put ponovi u nešto drugačijem, tj. variranom izdanju. Dok smo u stanju istraživanja on se uvek ponavlja nepromenjeno. Dodatni slojevi deluju na obavezni upravo tako što nešto što se ponavlja iznova i iznova obagate dodavanjem novih linija, detalja, itd.

2) Stanje akcije:

- *Action_Loop_4-4_90bpm* – predstavlja glavni, tj. obavezni sloj na koji će se ovog puta pod dejstvom određenih parametara dodavati/oduzimati preostala dva sloja, a to su *Action_Melody_Loop_4-4_90bpm* i *Action_AddLayer_Loop_4-4_90bpm*.
- *Action_Melody_Loop_4-4_90bpm* – predstavlja dodatni i neobavezni sloj koji ćemo u zadatku, iz čisto edukativnih razloga, da obradimo na dva načina:

1) priključuje se pod dejstvom tzv. *Probability* funkcije [8]. Ovo nam omogućava da na osnovu postavljene verovatnoće (u procentima) emitovanja audio-fajla randomizujemo nastup dotičnog sloja.

2) priključuje se pod dejstvom određenog parametra. Možemo sada da zamislimo da je stupanje na neki specifičan deo teritorije okidač za priključenje sloja *Action_Melody_Loop_4-4_90bpm*. O ovome će da bude reči kasnije.

- *Action_AddLayer_Loop_4-4_90bpm* – i ovo predstavlja dodatni i neobavezni sloj koji se priključuje pod dejstvom određenog parametra. Ovde možemo da zamislimo da je okidač npr. ekstremno teška situacija unutar borbe, tj. momenat kada je igrač jako blizu stanja smrti [9]. Ono što ovaj sloj čini drugačijim jeste to što će on da se javi u formi asinhronog instrumenta, tj. imaće sopstveni muzički tok unutar već postojećeg.

3) Stanje smrti:

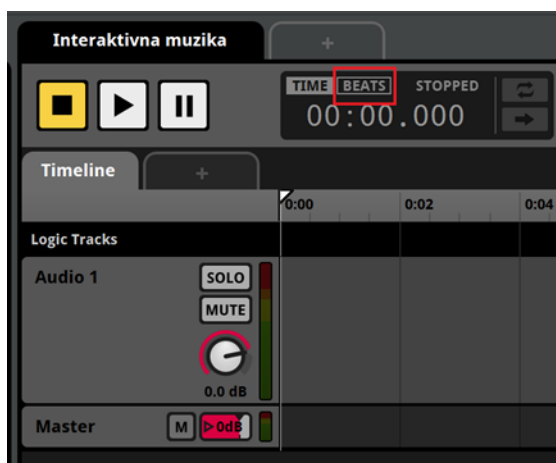
- *Death_4-4_90bpm* – kao što je već rečeno, ovo je jedini fajl koji nije pisan u formi lupa. To je iz razloga što ovaj fajl predstavlja neku vrstu bar trenutnog završetka neprekidnog toka smene muzike istraživanja i akcije [9]. Nakon njega će uslediti tišina koja označava vreme kada igrač odlučuje svoj dalji potez. Ako reši da proba ponovo da igra, vraća se u stanje istraživanja, dok u suprotnom, ostaje u tišini. Što se tiče tranzicije u stanje smrti, ovde opet imamo primer primene tehnike horizontalnoog resekveniranja. Tranzicija može da se desi prvenstveno iz akcionog ali i iz istraživačkog muzičkog segmenta.

Izrada muzičkog iventa – postavka takta i tempa

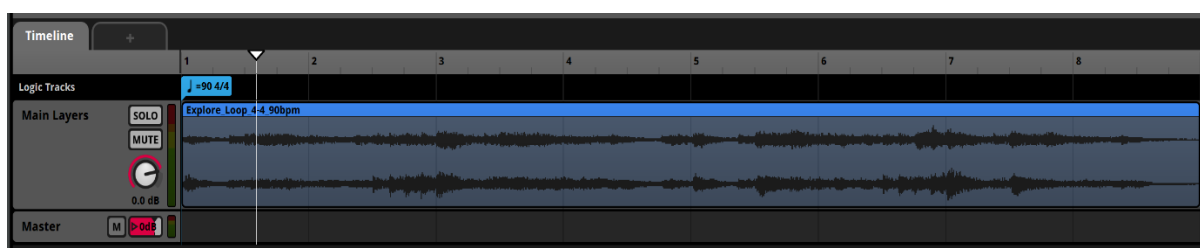
Kao što smo to dosad i činili, za početak treba da napravimo novi ivent. Za potrebe muzike biramo *2D Timeline Event*. Ovo jeste iz razloga što se i ovaj ivent, kao i iventi koraka i ambijenta, vezuje za samog igrača. Dakle, i muziku ćemo uvek da čujemo nepromenjenim intenzitetom bez obzira na promene položaja igrača unutar 3D-prostora. Ono što će aktivno da se menja unutar muzike su stanja, a samim tim i audio-fajlovi koje čujemo. Ivent ćemo da nazovemo *Interaktivna muzika*.

Kada napravimo ivent vreme ja da lagano prevlačimo fajlove, ali pre toga je jako bitno da postavimo odgovarajući takt i tempo. Dosad smo se što se tiče merenja vremena u projektu služili sekundom kao mernom jedinicom. Kada je muzika u pitanju tu je situacija nešto drugačija. Ona je uglavnom pisana u određenom taktu i tempu. Takt bi najbliže mogao da se objasni kao otkucaj, tj. puls muzike koji jeste u skladu sa ritmom i fraziranjem unutar njenog sadržaja. Tempo označava brzinu otkucaja. Određivanje takta i tempa svakako ne predstavlja deo zadatka, zato što je ta informacija uvek dostupna. Konkretno, ove vrednosti mogu da se nađu u samom nazivu fajla, npr. u nazivu *Action_Loop_4-4_90bpm* je jasno da je muzika pisana u taktu od 4/4 sa otkucajem brzine 90 bpm (engl. beats per minute, tj. broj otkucaja u minutu). Imenovanje fajlova na ovakav način predstavlja konvenciju zato što nije uvek slučaj (pogotovo u većim timovima i projektima) da se kompozitor pored pisanja muzike bavi i njenom implementacijom [7]. Postoje slučajevi u kojima postoji osoba u timu koja je isključivo zadužena za implementaciju audio-sadržaja. Jasno je da su ovakve informacije unutar naziva fajlova dragocene, zato što ubrzavaju proces implementacije i samim tim čitav radni tok čine efikasnijim. Ono što je naš zadatak jeste da ove informacije pravilno unesemo u projekat.

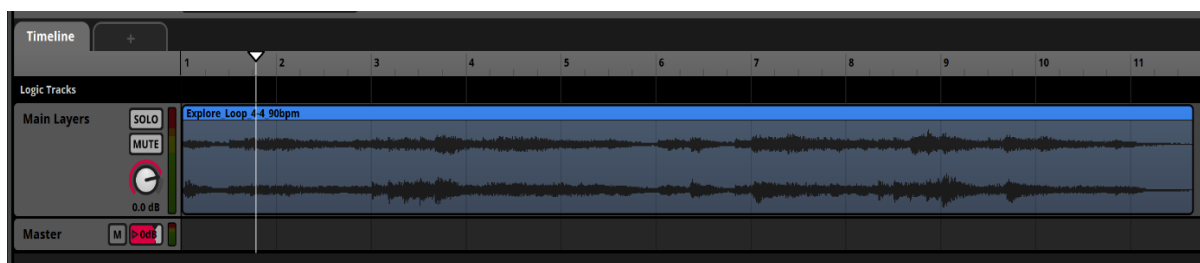
Za početak, treba da prebacimo mernu skalu sa odbrojavanja sekundi na odbrojavanje taktova. Na slici 2.48 se jasno vidi uokvireno crvenim markerom polje *Beats*, smešteno odmah pored polja *Time*. Kao što može da se pretpostavi, *Beats* se odnosi na odbrojavanje muzike, što i jeste logično, s obzirom da nam je merna jedinica *bpm*. *Time* se naravno odnosi na odbrojavanje zvuka, što nam je uveliko poznato jer smo dosad samo sekunde i koristili. Kada kliknemo na polje *Beats*, primetićemo da nam se merna skala promenila. Kao što je već rečeno, jako je bitno da se takt i tempo pravilno unesu u projekat jer u suprotnom, postalo bi izuzetno problematično da se prati muzički tok, zato što bi fajlovi stalno ispadali van svojih propisanih granica.



Slika 2.48. Promena mernih jedinica



Slika 2.49a. ISPRAVNO

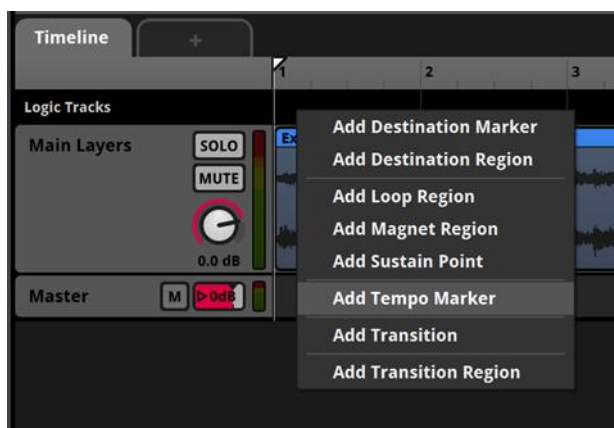


Slika 2.49b. NEISPRAVNO

Na sledećem primeru može da se uporedi kako fajl izgleda u odnosu na mernu skalu kada su takt i tempo pravilno uneseni a kako kada nisu (slike 2.49a i b). Kao što može da se vidi na slici 2.49a, takt i tempo su pravilno postavljeni. To je razlog zašto nam se kraj fajla tačno poklapa sa krajem osmog tatka. U proseku se muzički segmenti uglavnom završavaju punim taktom, tako da u slučaju da se spazi da postoji neka nepravilnost u poklapanju taktova, to može da nam bude jasan pokazatelj da nešto nije u redu. Baš u vezi sa tim, na slici 2.49b se vidi da se kraj fajla ne poklapa sa krajem takta. Naravno, ovo je zato što nismo postavili ispravne vrednosti takta i tempa. U ovom slučaju smo samo prevukli audio-fajl u projekat i

ništa dalje nismo intervenisali. Predefinisani tempo unutar bilo kojeg programa za audio-produkciju, pa tako i unutar programa *FMOD Studio*, jeste 120 bpm. Stoga i jeste došlo do nepoklapanja u taktovima, te nam se završetak fajla našao negde između 11. i 12. takta.

Konačno, takt i tempo se postavljaju tako što desnim klikom na *logic tracks* otvorimo padajući meni iz kojeg biramo *Add Tempo Marker* (slika 2.50). Jako je bitno da marker postavimo tačno na početak audio-trake a to činimo tako što držimo marker levim klikom i prevlačimo ga sa desna na levo. Nakon toga, dvostrukim klikom na marker nam se otvara numerčki unos. S obzirom da naš takt već jeste 4/4 tu nećemo ništa da menjamo ali ćemo zato 120 da prepravimo u 90 bpm.

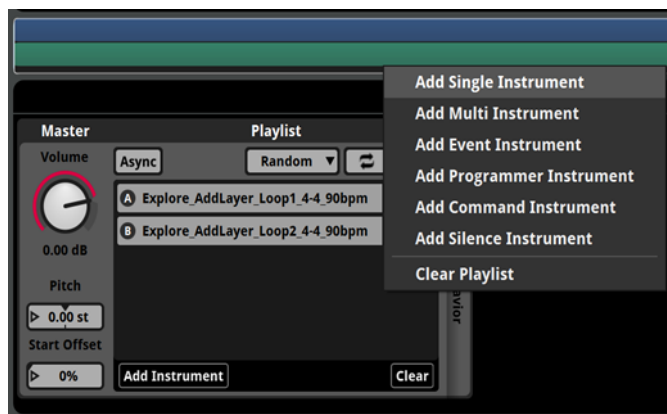


Slika 2.50. Dodavanje tempo markera

Postavka audio-fajlova – istraživačka muzika

Pošto segment istraživačke muzike treba da se ponavlja sve dok ne nastupi neka promena, postavimo lup-regiju na fajl koji vidimo na slici 2.49a, tj. na *Explore_Loop_4-4_90bpm*. Sada treba da dodamo i preostale fajlove. S obzirom da je reč o zasebnom sloju, napravićemo novu audio-traku u koju ćemo da prevučemo *Explore_AddLayer_Loop1_4-4_90bpm* i *Explore_AddLayer_Loop2_4-4_90bpm* i time kreiramo multi-instrument. Pošto sad već imamo dve različite audio-trake a biće ih i više, zbog dobre organizacije projekta trebalo bi da im damo imena. Audio-traku u koju smo smestili *Explore_Loop_4-4_90bpm* ćemo da nazovemo *Main Layers*; a onu u koju smo smestili *Explore_AddLayer_Loop1_4-4_90bpm* i *Explore_AddLayer_Loop2_4-4_90bpm*, *Add. Layers* (skr. od additional).

Naš multi-instrument trenutno ima samo dve varijacije. Ono što možemo da učinimo da naš segment istraživačke muzike deluje još manje repetitivno jeste da 'dodamo tišinu'. To činimo tako što, dok nam je još multi-instrument selektovan i samim tim otvoren u *deck* prozoru, desnim klikom otvorimo padajući meni iz kojeg biramo *Add Single Instrument* (slika 2.51). Ovo će dodati prazan instrument u našu plejlistu. Njegova uloga je, dakle, da napravi varijantu ponavljanja u kojoj čujemo samo glavni, tj. obavezni sloj.



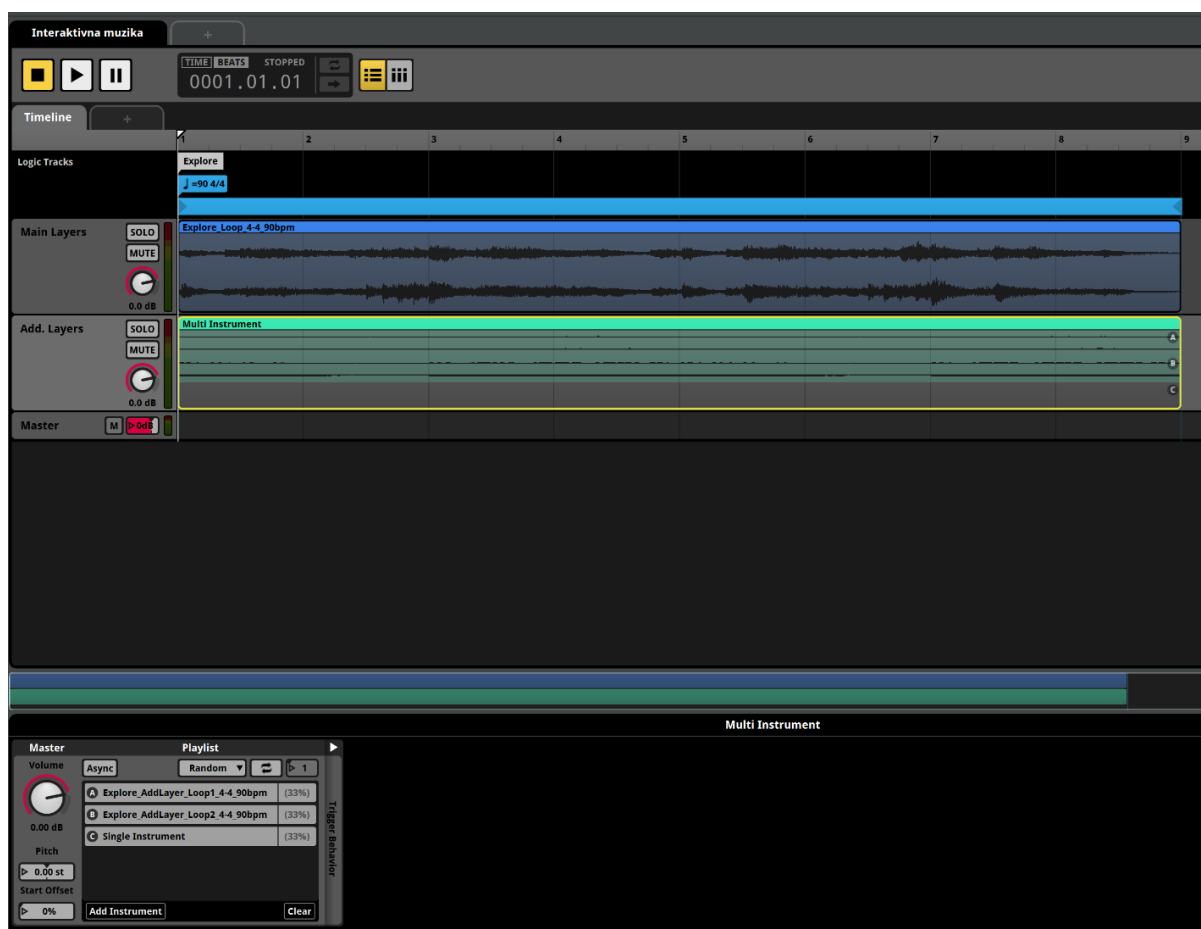
Slika 2.51. Dodavanje praznog instrumenta u plejlistu

Trenutno je verovatnoća emitovanja audio-fajlova jednako raspodeljena, tj. svi imaju po 33% verovatnoće. Hajde da kažemo da nam je jedan od fajlova prioritet, tj. da želimo da se emituje češće od drugih. Neka to bude *Explore_AddLayer_Loop1_4-4_90bpm*. U tom slučaju ćemo da preraspodelimo verovatnoću, tj. njemu ćemo da postavimo da verovatnoća bude jednaka 50% (za postavku procenata v. odeljak **Randomizacija po uobičajenom metodu**). Ovo će automatski da preraspodeli ostatak procenata jednako na preostala dva činioca plejliste, tj. *Explore_AddLayer_Loop2_4-4_90bpm* i prazan instrument će imati po 25% verovatnoće da se emituju. Takođe, ne zaboravimo da našoj plejlisti randomizujemo redosled puštanja fajlova (dugme *random*). Primećuje se da ovde ne postoji randomizacija po uobičajenom metodu. Razlog je dvostruk:

- 1) U pitanju su lupovi pa samim tim sva pravila o kojima je bilo reči u vezi sa randomizacijom po uobičajenom metodu u iventu lupa važe i ovde.
- 2) Muziku i zvuk ne možemo da randomizujemo na identičan način upravo zato što se muzika bavi taktom, tempom, a na sve to i tonalitetom. Stoga bi vid randomizacije po uobičajenom metodu davao krajnje čudne rezultate. Ovo je u redu

u slučaju da se baš želi postići takav efekat, ali ovakve situacije su retke i svakako ih nećemo primenjivati u našem zadatku.

Pošto smo upravo postavili sve audio-fajlove vezane za istraživačku muziku i pošto ćemo imati tranzicije u različite segmente, sada je vreme da postavimo i destinacioni marker (za postavku v. odeljak **Destinacioni markeri**). Nazvaćemo ga *Explore*. Naš muzički iverent bi zasad trebalo da izgleda kao na slici 2.52.

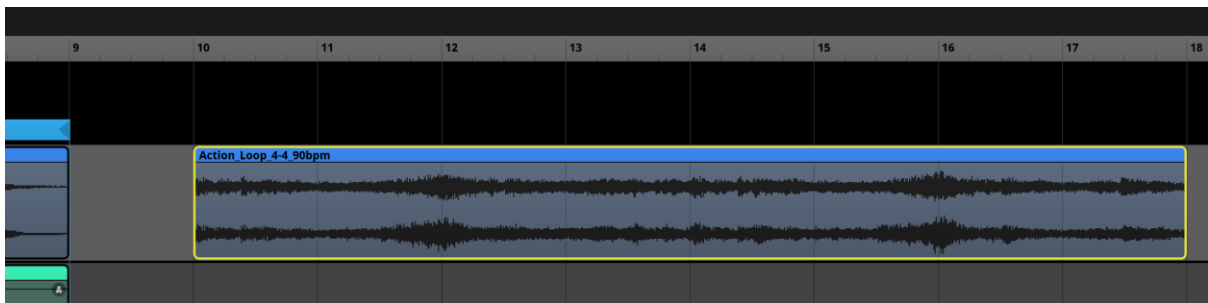


Slika 2.52. Izgled muzičkog iverenta sa postavljenom istraživačkom muzikom

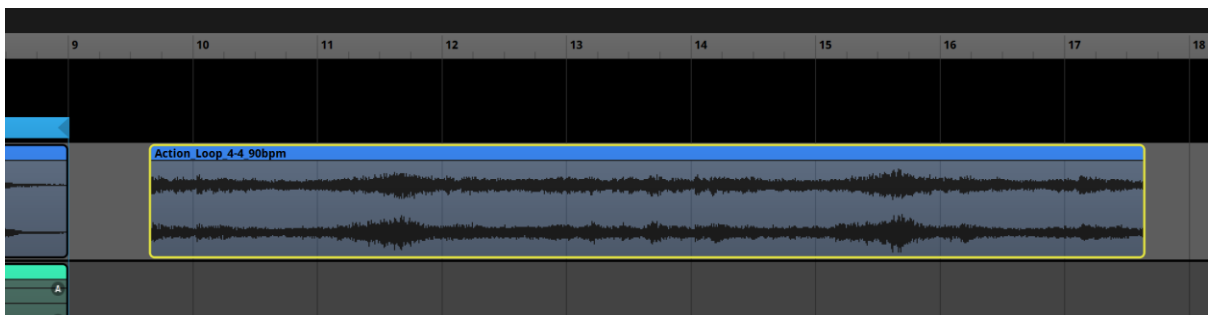
Postavka audio-fajlova – akciona muzika

Na sličan način kao što smo postavljali glavne ambijentalne zvukove u trećem iverentu interaktivnog zvuka, tako ćemo i ovde da postavimo glavne muzičke slojeve. Dakle, glavni akcioni sloj, tj. *Action_Loop_4-4_90bpm* bi onda trebalo da postavimo u istu audio-traku gde

smo smestili glavni sloj istraživačke muzike. Postavka audio-fajla na sam početak, odnosno prvi takt, je bila slučaj samo sa istraživačkom muzikom. Ostale fajlove koji se vezuju za akciju i smrt ćemo da postavljamo negde drugde u projektu, a tada će da bude važno da početak fajla dođe na sam početak takta zbog lakše i pravilnije organizacije i orijentacije. Opet, i ovde ćemo da demonstriramo ispravnu i pogrešnu postavku (slika 2.53a i b). Na slici 2.53a se vidi, kao i u slučaju istraživačke muzike, da audio-fajl kreće od početka takta (u primeru je to 10. takt, ali ovo ne mora nužno da bude tako zato što zavisi od toga koliko taktova nam traje lup) i završava se na kraju takta (u primeru 17. takt). Dakle, i akciona muzika se, baš kao istraživačka, sastoji iz osam taktova. Na slici 2.53b se vidi da audio-fajl ne kreće od početka takta što za posledicu ima to da se i ne završi na kraju takta. Ovakva postavka fajlova na prvom mestu stvara zabunu, zato što je logično i prirodno da muzički segment krene na početku takta. Druga jako bitna stvar, o kojoj će da bude više reči kasnije, jesu nepravilnosti koje bi usledile uz primenu kvantizacije (više o njoj kasnije). To je zato što se ona vodi jedinicama takta, što samo po sebi objašnjava da moramo vrlo pažljivo i smisleno da postavljamo audio-fajlove.



Slika 2.53a. **ISPRAVNO**



Slika 2.53b. **NEISPRAVNO**

Sada je vreme da fajlu *Action_Loop_4-4_90bpm* dodelimo sve što smo i *Explore_Loop_4-4_90bpm* – dakle, treba da postavimo lup-regiju i destinacioni marker koji ćemo da nazovemo *Action*. Tempo marker ne postavljamo ponovo zato što onaj koji smo postavili na početku važi za ceo iverent sve dok ne postavimo novi tempo marker sa novim vrednostima.

Probability – postavka verovatnoće emitovanja audio-fajlova

Nakon postavke glavnih i obaveznih slojeva treba da se pozabavimo postavkom neobaveznih. Kao što je već rečeno, *Action_Melody_Loop_4-4_90bpm* (koji ćemo ubuduće da zovemo ‘melodija’) predstavlja dodatni i neobavezni sloj koji ćemo u zadatku, iz čisto edukativnih razloga da obradimo na dva načina. Sada ćemo konkretno da se bavimo priključenjem pod dejstvom *Probability* funkcije. Za početak, hajde da prevučemo fajl *Action_Melody_Loop_4-4_90bpm* u projekat. Smestićemo ga u audio-traku *Add. Layers*. Kada kliknemo na fajl, kao što već znamo, otvara nam se *deck* prozor. U istom otvaramo polje *Trigger Behavior* (slika 2.43) gde nam se nalazi funkcija *Probability*.



Slika 2.54. *Probability* funkcija

Na slici 2.54. se vidi da ispod natpisa *Probability* postoje dva dugmeta – *Off* i *On*. Po automatizmu, funkcija *Probability* je postavljena na *Off*, na šta upućuje žuta boja dugmeta. Ispod ovih dugmadi vidimo natpis *Chance* sa slajderom. Dok nam je selektovano dugme *Off*, ovaj slajder nam je nedostupan – stoga je i zasenčen i vrednost je strogo postavljena na 100% i ne može da se menja. Ovo važi za svaki audio-fajl koji postavimo i znači da je trenutno sve što smo postavili od audio-fajlova podložno bezuslovnom emitovanju (stoga je vrednost na

slajderu *Chance* nepromenljivo 100%), osim ako nije uslovljeno nekim parametrom i tome sličnim. Onog trenutka kada selektujemo dugme *On* slajder postaje aktivan, na šta upućuje žuta boja dugmeta, i možemo da mu menjamo vrednosti (slika 2.55). Numerički unos vršimo na isti način kao i kod svih drugih slajdera.



Slika 2.55. Postavka procenta verovatnoće

Vrednost ćemo da postavimo na 50% što znači da sada fajl ima jednake šanse da se emituje, odnosno ne emituje. Ovo svakako ne znači da ćemo da imamo situaciju da pri jednom emitovanju lupa čujemo melodiju, a pri narednom ne, i tako nadalje. Nasuprot, emitovanje će da se dešava nasumice, ili bolje rečeno randomizovano, ali sve u okvirima zadate vrednosti od 50%. Ako bi npr. postavili da je vrednost 10%, to bi značilo da postoji jako mala verovatnoća da se melodija uopšte emituje, i suprotno od toga, ako bi postavili da je vrednost 90%, to bi značilo da će da bude jako retka situacija da se melodija ne emituje. Svrha upotrebe funkcije *Probability* predstavlja još jedan vid randomizacije radi izbegavanja repetitivnosti audio-sadržaja i postizanja varijeteta pri svakom ponavljanju. Baš kao što smo u segmentu istraživačke muzike imali multi-instrument čija se plejlita sastoji iz tri varijante, ovde imamo jednu varijantu koja se nekad javlja a nekad ne. S obzirom da se i multi-instrument služi verovatnoćom emitovanja, možemo da povučemo paralelu između ova dva slučaja. Tako gledano, postavka funkcije *Probability* na melodiju bi zapravo bila ekvivalentna postavci multi-instrumenta gde bi činiooci plejlite bili samo *Action_Melody_Loop_4-4_90bpm* i prazan instrument, a oba činioaca bi imala jednaku verovatnoću da se emituju, tj. po 50%. Svakako, ovo napominjemo samo u teoriji, u praksi je svakako prirodno da se za ovakve situacije služimo funkcijom *Probability*.

Postavka parametra – postepeno uvođenje audio-fajlova

Sada je vreme da prođemo kroz implementaciju melodije služeći se postavkom parametra. Kao što je već gore napomenuto, ovde će melodija da se javlja isključivo pod dejstvom određenog parametra koji bi u našem zamišljenom scenariju u isto tako zamišljenoj video-igri, bio vezan npr. za deo teritorije u kojem se trenutno igrač nalazi. Dakle, kao i kod primene *Probability* funkcije, i ovde postoji određeni stepen randomizacije, samo što on ne zavisi od trenutnog proračuna od strane programa, već od interakcije samog igrača, tj. ljudskog bića. Suština je da u oba slučaja imamo nepredvidiv tok dešavanja, što i jeste jedna od ključnih osobina medija kao što je video-igra [4].

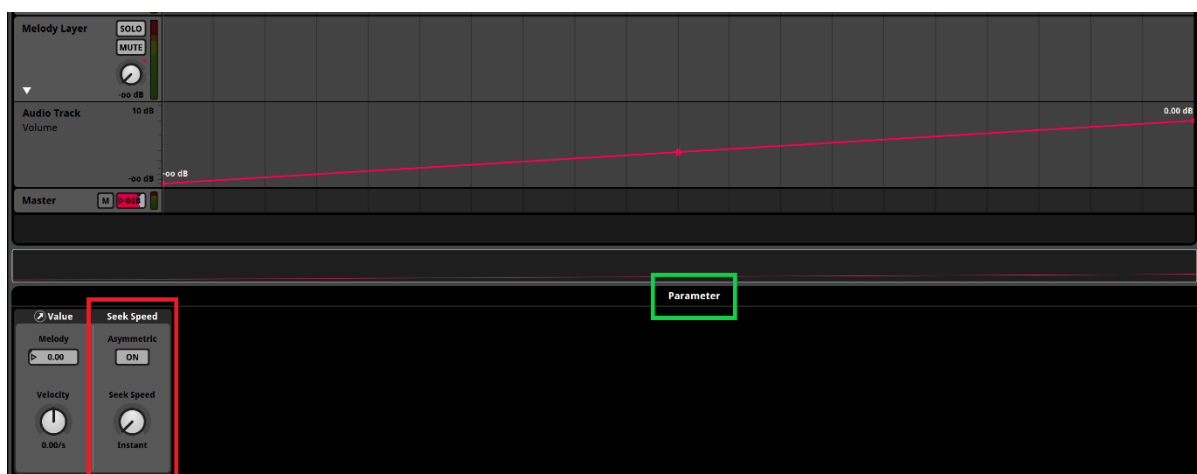
Hajde da napravimo novu audio-traku koji ćemo da nazovemo *Melody Layer* i da u nju prevučemo fajl *Action_Melody_Loop_4-4_90bpm*. Ovo možemo da uradimo na više načina:

- 1) Možemo ponovo da ga prevučemo, kao što smo to dosad radili, iz foldera odgovarajućeg operativnog sistema.
- 2) Možemo da prekopiramo audio-fajl koji smo već uneli u ivalent i zalepimo ga u audio-traku *Melody Layer*.
- 3) S obzirom da fajl *Action_Melody_Loop_4-4_90bpm* već postoji u projektnom podfolderu *Assets*, možemo da odemo na tab *Assets* (slika 1.3), pronađemo fajl i prevučemo ga u odgovarajuću audio-traku.

Kojigod metod da odaberemo, bitno je da fajl postavimo, kao i sve ostale što smo dosad postaljvali, tačno u okvir lup-regije akcione muzike.

Sada je vreme da napravimo parametar koji ćemo da nazovemo *Melody*. Setimo se kada smo pravili parametre za ivalent koraka. I sada ćemo se služiti istim tipom parametra, samo što ćemo ovde da optimizujemo vreme koje je potrebno za promenu njegove vrednosti od najviše ka najnižoj i obrnuto. Dakle, ovde parametar ne postavljamo direktno na audio-fajl, već ga dodajemo kao kod ivalenta koraka u formi tzv. *Parameter Sheet* (slika 2.8). I ovde ćemo vrednosti parametra *Melody* da uskladimo sa vrednostima jačine na isti način kao što je to bilo u ivalentu koraka, tj. vrednost parametra jednaka nuli označava odsustvo, dok vrednost jednaka jedinici označava prisustvo melodije. Stoga, hajde da napravimo odgovarajuću krivu vrednosti jačine koja korespondira sa vrednostima parametra koje smo upravo opisali (slika 2.56). Kada bi sad testirali rad parametra *Melody*, konstatovali bi da se parametar okida na identičan način kao što je to slučaj u ivalentu koraka. To znači da ćemo melodiju da čujemo u punoj jačini istog trenutka kada je parametar jednak jedinici. Ovo momentalno emitovanje audio-fajlova se

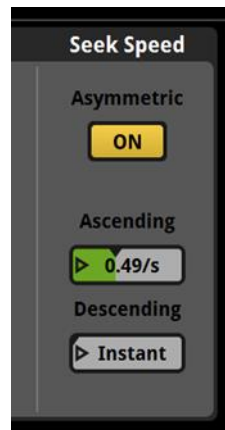
pokazalo kao jako korisno u iventu koraka, ali to je upravo zato što je u pitanju drugačiji tip, odnosno način okidanja iventa nego što to treba da bude slučaj sa melodijom. Kod koraka je baš poželjno da imamo momentalan odziv audio-fajlova na promene vrednosti parametra, zato što se ivent koraka okida pri svakom novom koraku igrača, što je neophodno kako bi koraci zvučali realistično u video-igri.



Slika 2.56. Postavka tranzicije između najviše i najniže vrednosti parametra

Kod muzike je situacija uglavnom, ali ne uvek, nešto drugačija. Bilo da su u pitanju horizontalne ili vertikalne tranzicije, gotovo uvek je dobro da budu fine i zaobljene. Stoga ovakav vid momentalnog uvođenja audio-fajlova nije dobar pristup za našu melodiju. Kako bi rešili ovaj problem, podesićemo brzinu, odnosno vreme koje je potrebno da parametar izvrši tranziciju iz vrednosti nula u jedinicu i obrnuto. To činimo tako što dvostrukim klikom na *Parameter Sheet Melody* otvorimo *deck* prozor. Na slici 2.56 se prema izgledu *deck* prozora jasno vidi da smo uspešno selektovali parametar (uokvireno zelenim markerom) i sada možemo da se bavimo finim podešavanjima istog. Na istoj slici primećujemo prozorčić uokviren crvenim markerom koji se zove *Seek Speed*. Po automatizaciji mu je istoimena funkcija postavljena na vrednost *Instant*. To znači upravo to što piše, tj. da se tranzicije između najniže i najviše vrednosti dešavaju momentalno. Vreme tranzicije menjamo tako što levim klikom pomeramo kazaljku na regleru sa leva na desno. Što je više okrenuta ka desnoj strani to je vreme tranzicije kraće i obrnuto. Stoga je u proseku najbolja opcija da nam kazaljka bude negde na sredini reglera, ali ovo svakako zavisi od same muzike i efekta koji želimo da proizvedemo. Najbolje je da sami testiramo pažljivim slušanjem i procenimo šta dobro i simisleno zvuči.

Pored funkcije *Seek Speed* takođe se primećuje funkcija *Asymmetric* koja je po automatizaciji isključena. To znači da vrednost koju smo postavili na *Seek Speed* regleru važi za tranzicije u oba smera, tj. za uključenje i isključenje parametra. Kada kliknemo na dugme *ON*, ono postaje žuto i otvore nam se dva slajdera: *Ascending* i *Descending* (slika 2.57).



Slika 2.57. Asimetrična postavka tranzicije

Slajder *Ascending* je postavljen na onu vrednost koju smo zadali putem *Seek Speed* reglera, dok je slajder *Descending* postavljen na vrednost *Instant*. Ova funkcija je korisna u slučaju da želimo da nam vremena tranzicija u različitim smerovima budu drugačija, npr., ako želimo da se melodija brže uvede u muzički aranžman, postavimo joj kraći vremenski interval na *Ascending* slajderu, dok ako npr., želimo da je postepnije izvedemo iz muzičkog aranžmana, postavimo duži vremenski interval na *Descending* slajderu. U praksi, *Asymmetric* funkciju nećemo koristiti, ali je svakako korisno da znamo čemu ona služi i u krajnjoj liniji, da je upotrebimo ako smatramo da bi takav postupak dao bolje zvučno rešenje.

Jako je bitno da se napomene da je za ovakav tip dodatnog sloja od ključnog značaja da se postavi u zasebnu audio-traku. To je zbog toga što promene u vrednostima parametra direktno utiču na vrednosti jačine celokupne audio-trake (setimo se između ostalog da je svaki multi-instrument zvučnik koraka bio smešten u zasebnu traku). U slučaju da se ovakav tip sloja nađe u istoj traci kao i neki drugi sloj koji ne treba da bude podložan ovakvom tipu promena, sva delovanja će da utiču na sav audio-sadržaj koji se nađe u dotičnoj traci. Kada bi npr. melodiju postavili u istu traku gde su i dodatni slojevi istraživačke muzike, rezultat bi bio da, kad nam je vrednost parametra *Melody* jednaka nuli, a samim tim i vrednost jačine na svom minimumu, ne bi čuli ni melodiju ni slojeve istraživačke muzike.

Postavka asinhronog instrumenta – momentalno uvođenje audio-fajlova

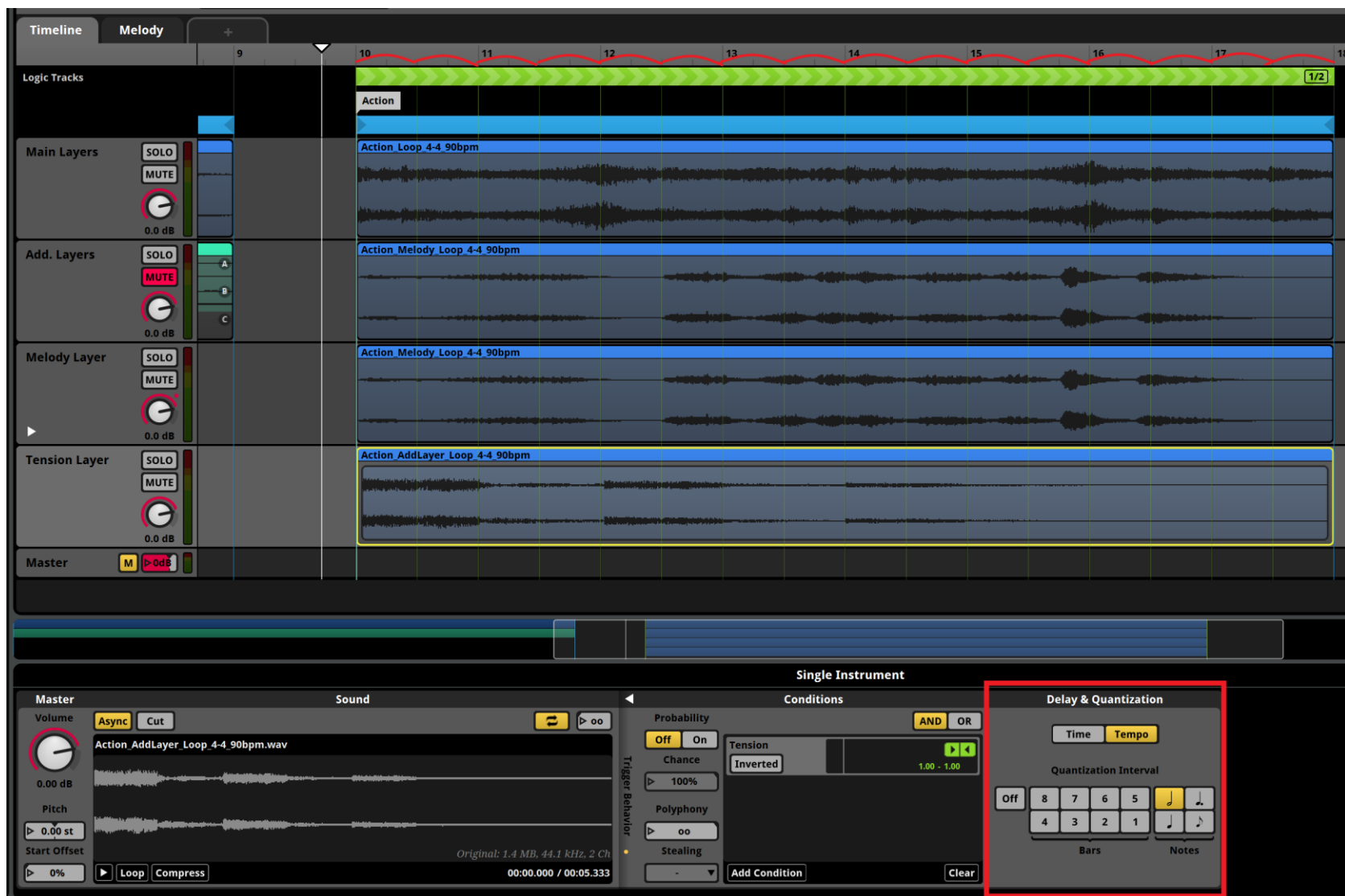
Poslednji sloj akcione muzike koji treba da postavimo je *Action_AddLayer_Loop_4-4_90bpm*. Kao što već znamo, i on predstavlja dodatni i neobavezni sloj koji se priključuje pod dejstvom određenog parametra. Kao deo našeg imaginarnog scenarija smo već naveli da možemo da zamislimo da je okidač npr. ekstremno teška situacija unutar borbe, tj. momenat kada je igrač jako blizu stanja smrti. Shodno tome, parametar koji budemo pravili za ovaj sloj će da se zove *Tension*. Pre toga, napravićemo novu audio-traku koji ćemo da nazovemo *Tension Layer* u koju ćemo da prevučemo fajl *Action_AddLayer_Loop_4-4_90bpm* (ubuduće ‘sloj tenzije’). Pošto je ideja da ovaj sloj ima sopstveni muzički tok unutar već postojećeg, pretvorićemo ga u asinhroni instrument a njegovo delovanje prilagoditi trajanju lup-regije akcione muzike. Sada je vreme da postavimo parametar *Tension*. Na isti način kao što je to bio slučaj u iventu ambijentalnih tranzicija, parametar postavljamo direktno na audio-fajl i biramo isti tip parametra, tj. *User: Countinous*. Što se tiče postavke vrednosti parametra i uslova koji se vezuju direktno za njih, i ovde se držimo standardnih vrednosti gde nula predstavlja odsustvo a jedinica prisustvo muzičkog sloja. Sve u vezi sa postavkom asinhronog tipa instrumenta kao i ovakvog vida postavke parametra može da se nađe u odeljku **Asinhroni instrument**.

Kada bi sada testirali rad parametra *Tension* uvideli bi da se on zaista ponaša onako kako smo ga uputili. Međutim, s obzirom da je u pitanju muzika, sve promene treba da se dešavanju na takav način da što manje remete muzički tok. S obzirom da želimo da se sloj tenzije priključi momentalno, tj. bez uvođenja, kao što je to bio slučaj kod melodije, može da se ispostavi kao jako komplikovano da se postigne da ovi ‘upadi’ zvuče muzički smisleno. Do ove poteškoće dolazi upravo iz razloga što muzika broji svoj vremenski tok u taktovima. Priroda sloja tenzije je takva da deluje kao neka vrsta otkucaja ili udarca što je adekvatno trenutnom dešavanju u video-igri. Dakle, trebalo bi da proizvede neku vrstu osećaja pulsiranja i odbrojavanja vremena, a istovremeno i neke vrste upozorenja. Sve ovo i jeste razlog zašto ovaj sloj želimo da plasiramo na ovakav način. No, kada se ovaj sloj javi momentalno po pozivu odgovarajuće vrednosti parametra, u najvećem broju slučajeva ćemo imati situaciju da se on dogodi u trenutku koji po pitanju muzike i taktova nema previše smisla, pa samim tim može da zvuči kao greška. Ono što možemo da učinimo jeste da ‘ukrotimo’ njegove nastupe. To postizemo korišćenjem kvantizacije.

Na slici 2.58 možemo da pogledamo kompletnu postavku sloja tenzije. Između ostalog, u *deck* prozoru, sa desne strane prozorčića *Conditions* primećujemo crvenim markerom uokviren prozorčić koji se zove *Delay & Quantization*. U njemu imamo dva dugmeta. Kao što možemo

da pretpostavimo, dugme *Time* se odnosi na rad sa zvučnim efektima, a dugme *Tempo* na rad sa muzikom. Po automatizaciji je selektovano dugme *Tempo* koje nam inače i treba. Ispod dugmadi vidimo natpis *Quantization Interval* a ispod njega vrednosti podeljene u dve frakcije: *Bars* – što na engleskom znači takt i *Notes* – što na engleskom znači note i konkretno se odnosi na jedinice manje od celog takta. Takt našeg muzičkog primera je 4/4, što znači da se on sastoji iz četiri jednaka otkucaja brzine 90 bpm. U frakciji taktova, vrednosti kojima se bavimo su isključivo celi taktovi, dok, ako želimo da nam vrednost kvantizacije bude manja od celog takta, onda u frakciji nota možemo da biramo vrednosti kao što su npr. polovina, četvrtina, itd. Kada mišem prelazimo preko nota prikazuje nam se koja je vrednost u pitanju. Elem, najbitnije od svega jeste čemu zapravo služi kvantizacija i postavljanje svih ovih vrednosti. Ona služi tome da kontrolišemo trenutak u kojem je sloj tenzije dozvoljeno da se priključi muzičkom aranžmanu. Na slici 2.58 se vidi da smo kvantizaciju postavili na polovinu notne vrednosti. To u praktičnom smislu znači da iako je parametar *Tension* jednak jedinici, tj. iako je ispunjen uslov za priključenje sloja tenzije, on se neće priključiti sve dok se ne odbroji polovina notne vrednosti, odnosno polovina takta. Na ovaj način smo postigli to da sloj tenzije može da nam se javi isključivo na svakoj odbrojanoj polovini unutar takta. Na slici 2.58 je korišćena tranziciona regija samo radi prikazivanja gde su tačno mesta mogućih priključenja. Dakle, sve iscrtane vertikalne žuto-zelene linije koje vidimo, predstavljaju zapravo mesta mogućih priključenja. Ako bolje pogledamo, videćemo da od jedne moguće tačke priključenja do druge i tako u nedogled, brojimo tačno polovinu takta (raspon označen crvenim markerom). Na ovaj način smo obezbedili muzičku smislenost koju smo pominjali i sada ovaj sloj više deluje kao integralni deo muzičkog aranžmana, a ne kao neki udarac koji se dešava nasumice i ne prati muzički tok. O kvantizaciji će da bude još reči kada se budemo bavili horizontalnim tranzicijama između segmenata.

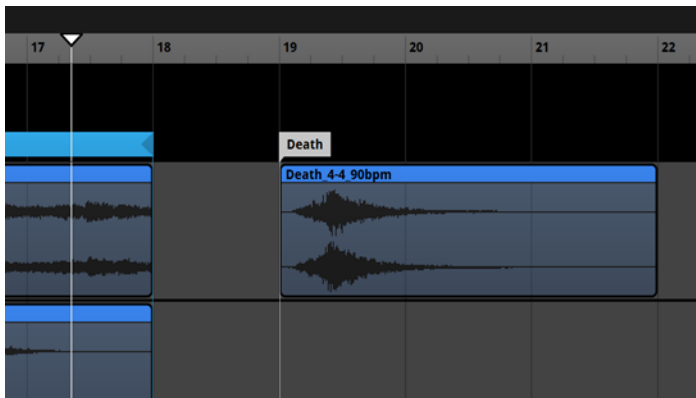
Poslednje što treba da uradimo, kao što je to bio slučaj i u iventu ambijentalnih tranzicija, jeste da obezbedimo neprestani tok asinhronog instrumenta. To činimo tako što u prozorčiću *Sound* aktiviramo dugme sa znakom beskonačnosti. U suprotnom, kada bi se asinhroni instrument aktivirao, čuli bi njegov nastup samo jednom, a to nije to što želimo. Naprotiv, želimo da se ovaj sloj ponavlja sve dok je parametar *Tension* jednak jedinici i da na taj način stvara osećaj pulsiranja i odbrojavanja vremena.



Slika 2.58. Postavka sloja tenzije

Postavka audio-fajlova – muzika stanja smrti

Jedini preostali fajl koji treba da postavimo u projekat je *Death_4-4_90bpm*. Primetili smo da smo dosad koristili samo fajlove u formi lupa – ovo je jedini fajl koji predstavlja fragment. To je iz razloga što on ima nešto drugačiju ulogu od prethodnih, tj. on služi da nas vrlo jasno obavesti o tome da je igra gotova. Za razliku od lupa koji predstavlja uokvirenu muzičku celinu, namenjenu da traje dok je određeno stanje aktivno, on je jednostavna perkusivna fraza (udarac) i služi da okonča stanje akcije, ili u ređem slučaju istraživanja. Pošto ga ne prate nikakvi propratni slojevi, može da se kaže da je ovaj fajl jedini i glavni sloj. Stoga ćemo ga smestiti u audio-traku *Main Layers*.



Slika 2.59. Postavka smrti

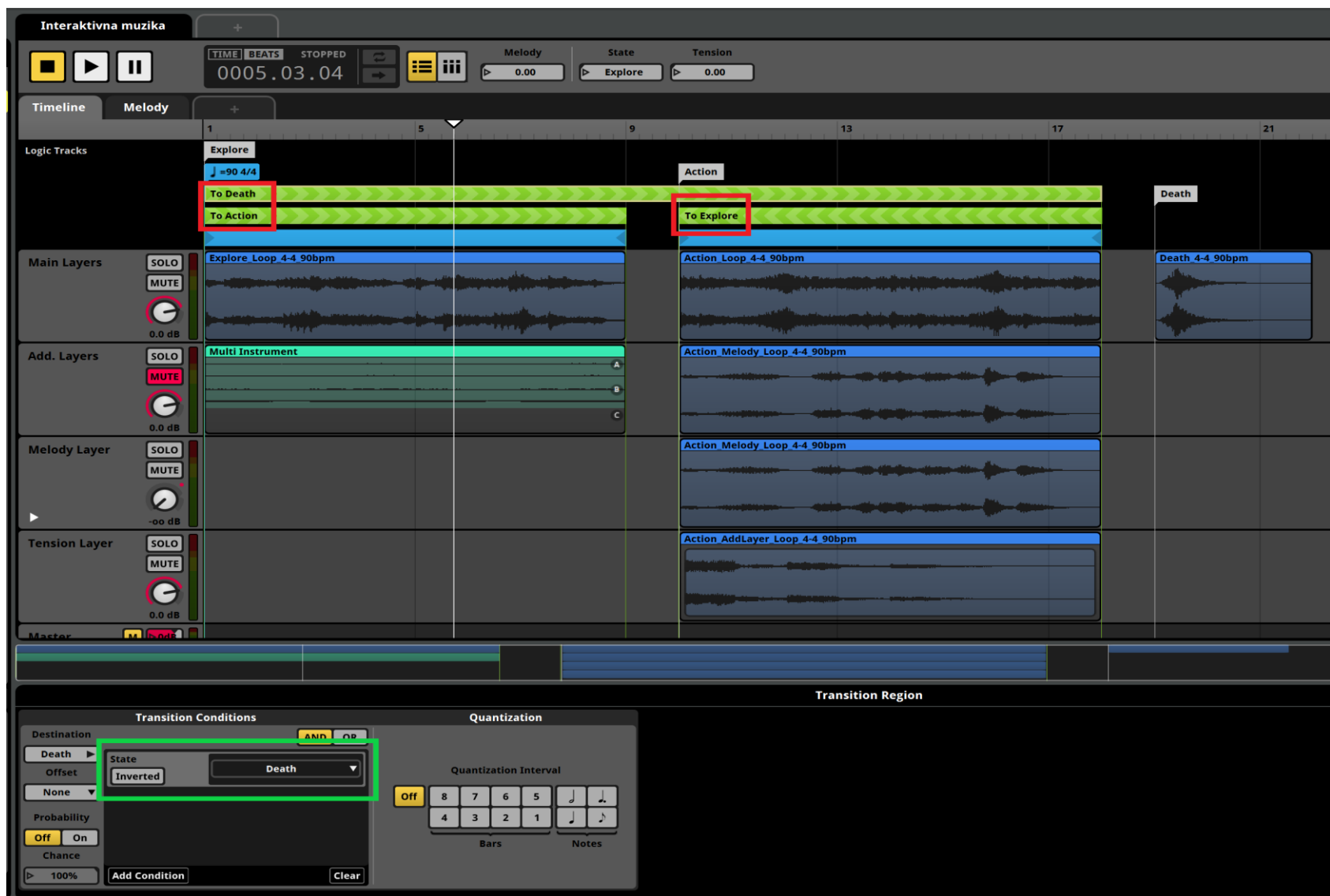
Pri postavci bi, kao što je to bio slučaj sa postavkom akcione muzike, opet trebalo da vodimo računa da fajl (koji ćemo ubuduće da zovemo ‘smrt’) postavimo na početak takta. Jedino presotaje da još postavimo destinacioni marker koji ćemo da nazovemo *Death* (slika 2.59) i onda smo spremni za konačni postupak u izradi našeg muzičkog inventa, a to je postavka horizontalnih tranzicija. One će zapravo da ožive naš invent i da sve ove promene stanja o kojima smo toliko pričali zapravo obistine.

Postavka horizontalnih tranzicija – smena stanja

Kao što je već najavljeno, jedino nam je preostalo da postavimo tranzicije između različitih segmenata, odnosno stanja. Krenimo sa tranzicijom između istraživačke i akcione muzike. Za početak, treba da se postave tranzicione regije na oba segmenta i da se pravilno

usmere – dakle, tranziciona regija postavljena na segmentu istraživačke muzike ide na destinacioni marker *Action*, a ona koja je postavljena na segmentu akcione muzike ide na destinacioni marker *Explore*. Nakon toga sledi postavka parametra koji će da kontroliše promene sva tri stanja: istraživanje, akcija i smrt. Ovde ćemo da se služimo istim tipom parametra koji smo koristili za smenu dana i noći u iventu tranzicije ambijentalnih zvukova. Parametar se isto kao u pomenutom iventu vezuje direktno za tranzicione regije. Stoga ćemo parametar kao i uslove da postavimo direktno na njih. Krenućemo od tranzicione regije smeštene na segmentu istraživačke muzike na kojoj ćemo i da kreiramo sam parametar koji ćemo da nazovemo *States*. Da se podsetimo, prvo selektujemo tranzicionu regiju a zatim u *deck* prozoru biramo: *Add Condition – Add parameter Condition – Browse – New Parameter*. Dalje, kad nam se otvori prozor parametra biramo isti tip, tj. *User: Labeled*. Dajemo naziv parametru kao i vrednostima unutar njega – dakle, *Value A* postaje *Explore*; *Value B* postaje *Action* i *Value C* postaje *Death*. Sada je vreme da pravilno postavimo uslove dešavanja promena, tj. da pravilno usmerimo tranzicije. Opet možemo da se ugledamo na ivent tranzicije ambijentalnih zvukova – kao što smo tamo dnevni lup usmeravali ka noćnom i obrnuto, tako ćemo i ovde lup istraživačke da usmerimo na lup akcione muzike i obrnuto.

Ono što je ovde drugačije jeste to što imamo i treće stanje, tj. smrt. Ovakva vrsta promene je moguća da se desi iz oba stanja, samo što je mnogo veća verovatnoća da to bude iz stanja akcije. Smrt u toku stanja istraživanja bi se javila u vidu nekog iznenadnog napada sa fatalnim ishodom. Stoga je neophodno da postavimo još jednu tranzicionu regiju koja će da obuhvati oba segmenta i koju ćemo da usmerimo kao destinacionom markeru *Death*, postavljajući pritom istoimenu uslov. Naš muzički ivent bi dosad trebalo da izgleda kao na slici 2.60. Crvenim markerom su uokvirene destinacije koje smo postavili po tranzicionim regijama a zelenim je uokvirena postavka uslova na trenutno selektovanoj tranzicionoj regiji.



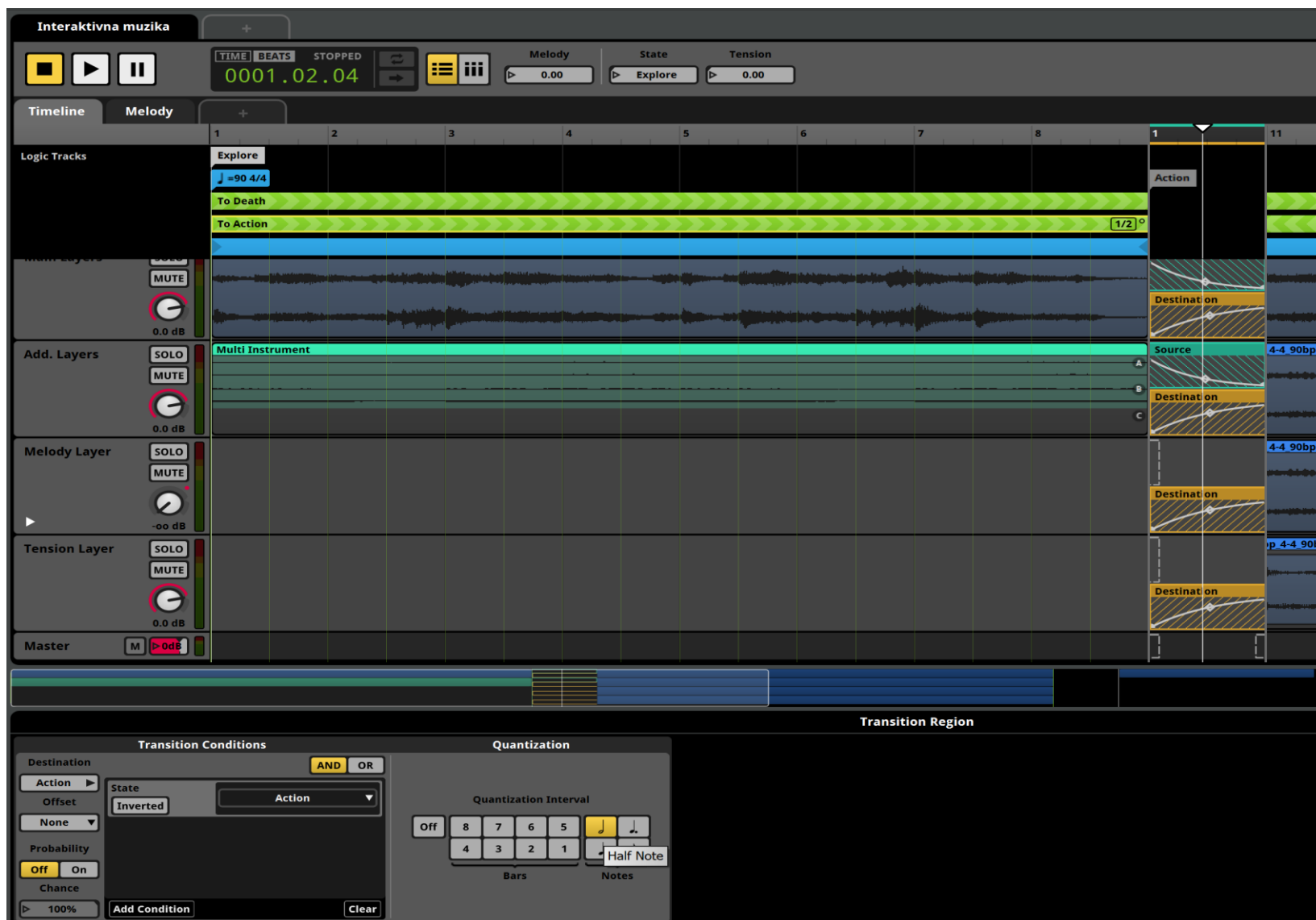
Slika 2.60. Izgled muzičkog inventa sa svim postavljenim muzičkim segmentima i tranzicijama

Postavka tranziconih prostora

Opet ćemo da postavimo paralelu sa ivalentom tranzicije ambijentalnih zvukova. Ista pravila koja su važila u pomenutom ivalentu važe i ovde. Dakle, tranzicije između segmenata ćemo da postignemo krosfejdomanjem, poštujući opet pravilo da ono što sledi, tj. destinacija, je bitnije od onog što je prethodilo i samim tim treba brže da se uvede i suprotno; ono što je prethodilo je u datom trenutku manje bitno i treba brže da se izvede. Takođe ne sme da se zaboravi na sve ostale sporedne i neobavezne slojeve jer svi oni, iako nisu stalno prisutni, sačinjavaju, odnosno pripadaju jednom segmentu. Stoga, i oni treba da se uključe u tranzicioni prostor. U malopre pomenutom ivalentu smo trajanje tranzicionog prostora određivali služeći se sekundama. U muzičkom ivalentu ćemo, shodno njegovoj prirodi, da se služimo muzičkim jedinicama, odnosno taktovima. Trajanje iznosi ukupno jedan takt.

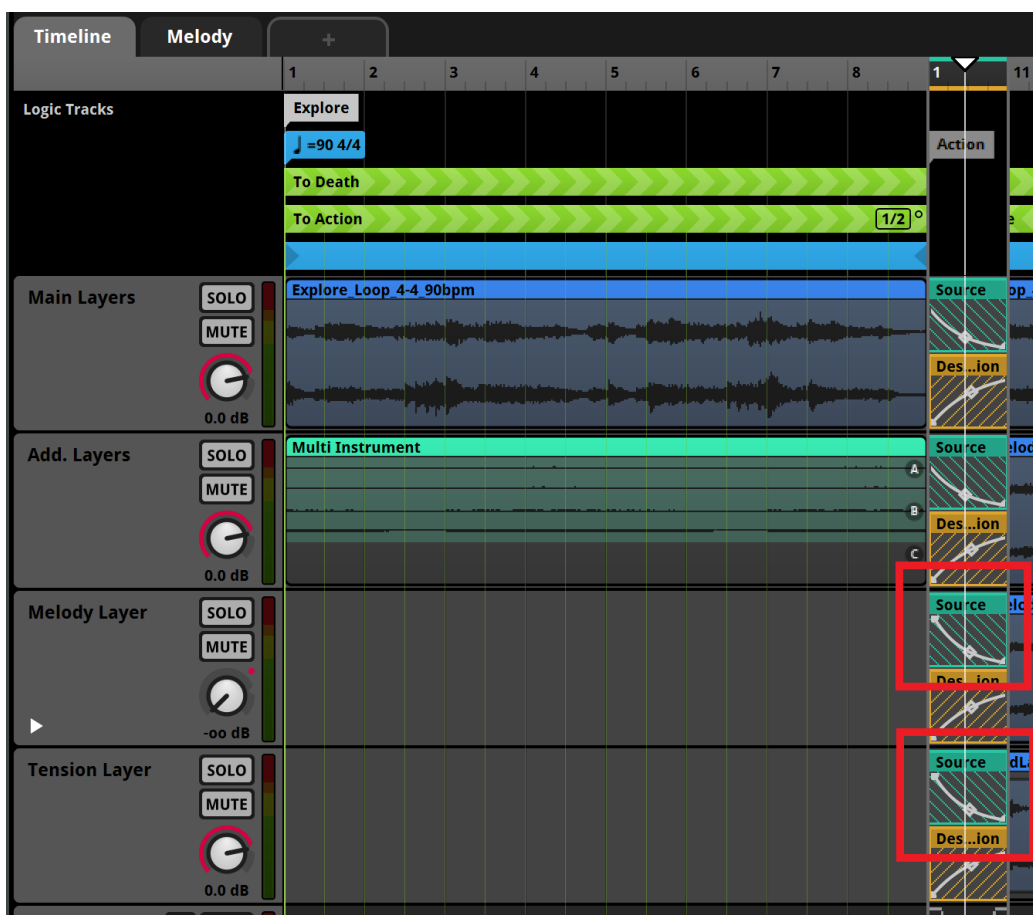
Ono što dosad nismo primenjivali u tranzicionom prostoru, a što je bilo najavljeno pri postavci asinhronog instrumenta jeste kvantizacija. Nju nismo praktikovali u ivalentu tranzicije ambijentalnih zvukova zato što se radilo sa zvučnim efektima koji ne broje taktove i ne sadrže harmoniju i melodiju. Pošto muzika sadrži sve pomenuto, moramo da obratimo pažnju na to da pri izvršenju tranzicija što manje remetimo njen tok. Dakle, jednostavan prekid muzičkog toka usled dejstva parametra neće da bude dovoljan, već ćemo morati da ga usmerimo i prilagodimo taktu. Kao što smo to činili pri postavci asinhronog instrumenta, i ovde ćemo da ograničimo trenutke kada tranzicija može da se desi. Zamislimo da se trenutno nalazimo u stanju istraživanja i samim tim čujemo odgovarajuću muziku. Kada je parametar *State* jednak vrednosti *Action* kursor će iz svoje trenutne pozicije, gde god se ona nalazila u okviru segmenta istraživačke muzike, da se prebaci na početak tranzicionog prostora. Ali, ako bi na tranzicionu regiju postavili kvantizaciju u vrednosti od polovine takta, rezultat bi bio takav da iako je uslov za tranziciju već ispunjen, mora da se sačeka da kursor stigne do naredne polovine takta. Možemo da se vratimo na sliku 2.58 kako bi se podestili i uporedili ovu situaciju sa trenutnom. Dakle, sve što smo primenili pri postavci asinhronog instrumenta važi i za horizontalnu tranziciju, samo što nisu u pitanju mesta mogućih priključenja već prekida.

Postavka se vrši tako što selektujemo željenu tranzicionu regiju i u *deck* prozoru u okviru prozorčića *Quantization* (primećujemo da za razliku od situacije kad smo se bavili asinhronim instrumentom ovde namamo prozorčić *Delay & Quantization* već samo *Quantization*) postavimo da je vrednost kvantizacije polovina takta (slika 2.61).



Slika 2.61. Postavka tranzicionog prostora usmerenog ka akcionoj muzici

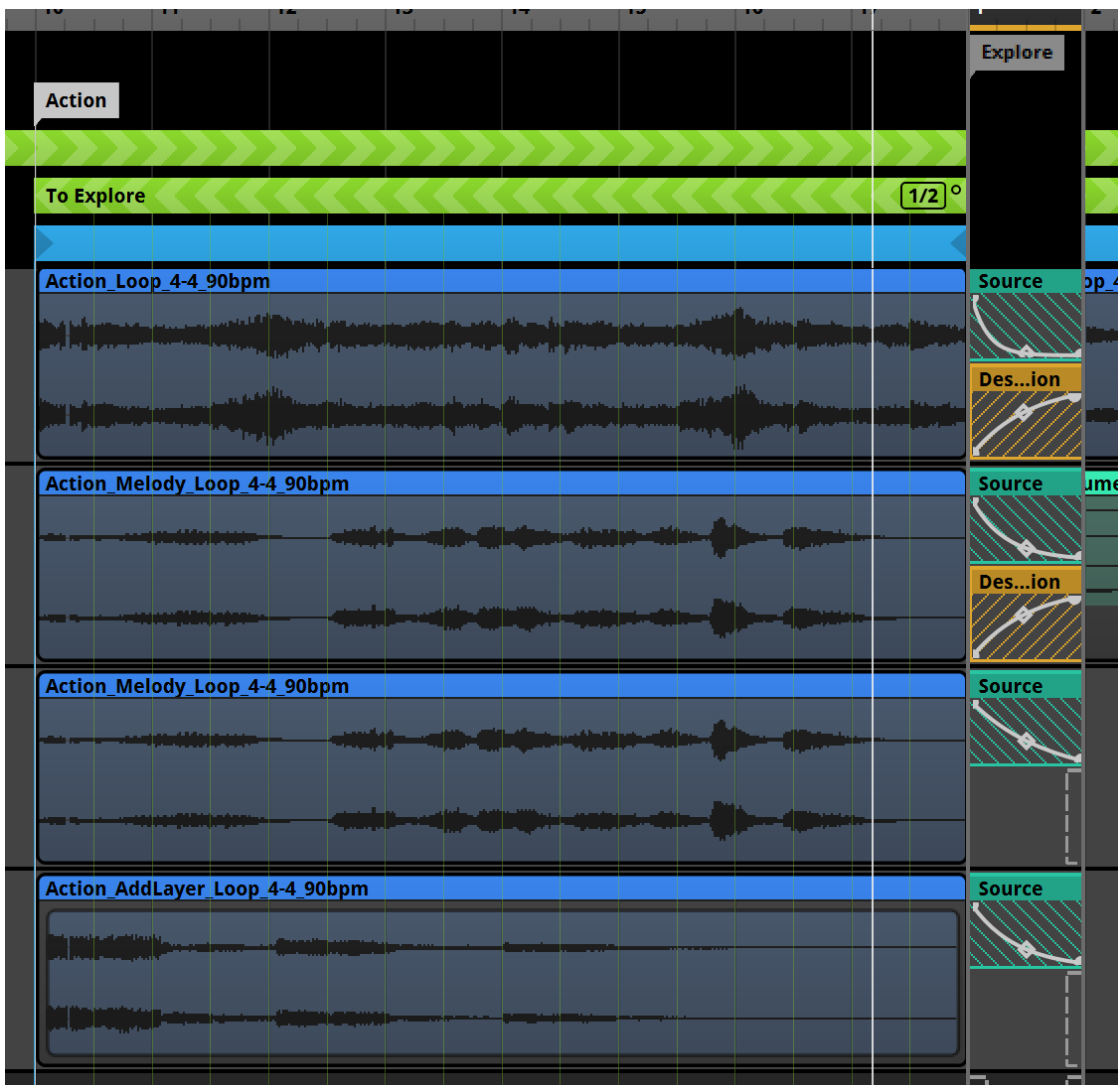
Kao što se očekivalo, i ovde su se kreirale žuto-zelene vertikalne linije koje označavaju moguća mesta prekida muzičkog toka. Dalje, na istoj slici se primećuje organizacija fajlova u tranzicionom prostoru. Sudeći po krosfejdu, tj. oblicima kriva (engl. volume curves), jasno je da se oba sloja istraživačke muzike (izvor) brže povlače iz muzičkog aranžmana, dok se nasuprot tome, slojevi akcione muzike (destinacija) brže priključuju. Jedna od vrlo čestih grešaka u zadacima jeste to da se bez obzira na sadržaj unutar audio-traka, sve prevlači u tranzicioni prostor. To podrazumeva i prevlačenje audio-sadržaja koji ne postoji, odnosno tišine. Na slici 2.62 se jasno vidi da je u tranzicioni prostor prevučen sadržaj iz audio-traka *Melody Layer* i *Tension Layer* za koje uveliko znamo da pripadaju isključivo akcionoj muzici, tj. destinaciji, što znači da one nikako ne mogu da budu deo koji prevlačimo iz izvora. Između ostalog, jasno se vidi da u okviru lup-regije istraživačke muzike, u pomenutim audio-trakama ne postoji nikakav sadržaj.



Slika 2.62. Prevlačenje nepostojećeg sadržaja u tranzicioni prostor

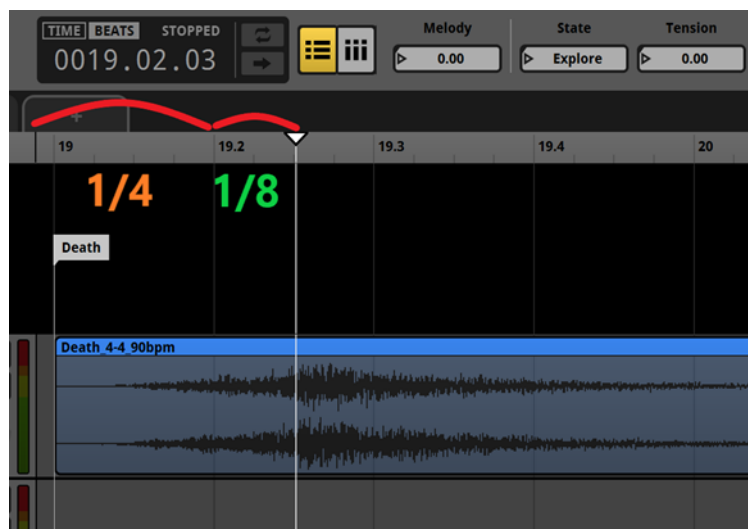
Na slici je crvenim markerom uokviren postupak koji predstavlja grešku. Ovakav postupak svakako ne bi naškodio sveukupnom zvučanju muzike, ali predstavlja tipičan primer nelogičkog posmatranja i izrade zadatka po automatizmu i bez razumevanja.

Kvantizaciju i tranzicioni prostor bi sada trebalo da postavimo i na segment akcione muzike. Sve ostaje isto osim što se krosfejd sada dešava u suprotnom smeru – dakle, ono što je prethodno bilo izvor sada postaje destinacija i obrnuto. Shodno tome, slojevi akcione muzike se sada povlače dok se slojevi istraživačke muzike priključuju (slika 2.63). I ovde nikako ne smemo da zaboravimo na neprevlačenje nepostojćeg sadržaja – stoga smo krosfejdovali samo postojeće slojeve istraživačke muzike.



Slika 2.63. Postavka tranzicionog prostora usmerenog ka istraživačkoj muzici

Pređimo sada na postavku tranzicije koja vodi u smrt. S obzirom da u ovo stanje može da se pređe bilo iz stanja istraživanja ili akcije, tranzicioni prostor ćemo da postavimo na tranzicionu regiju kojom smo obuhvatili oba segmenta i postavili odgovarajuću destinaciju. Ovde će trajanja tranzicionog prostora i kvantizacije da budu drugačija nego u prethodna dva slučaja. Setimo se da smo za audio-fajl *Death_4-4_90bpm* pričali da on jedini ima trentman fragmenta. Za razliku od lupa čija je svrha da traje, fragment predstavlja neku vrstu završetka muzičkog ishoda. Samim tim je i njegova struktura drugačija. Posmatrajmo fizionomiju zvučnog talasa (engl. waveform) na slici 2.64. Primećuje se da ovde postoji nagli rast i pad, drugim rečima, ovaj fajl ima svoj prirodni uzlazni i silazni pokret (engl. fade in/fade out). Iz ovog razloga, kada ga budemo postavljali u tranzicioni prostor, neće da bude potrebe da mu postavljamo krivu. To je zato što je ona po prirodi stvari već deo njegove fizionomije. Kao što je već rečeno, ovde postoji nagli rast, a u sredini je vrh nakon kojeg sledi nagli pad.

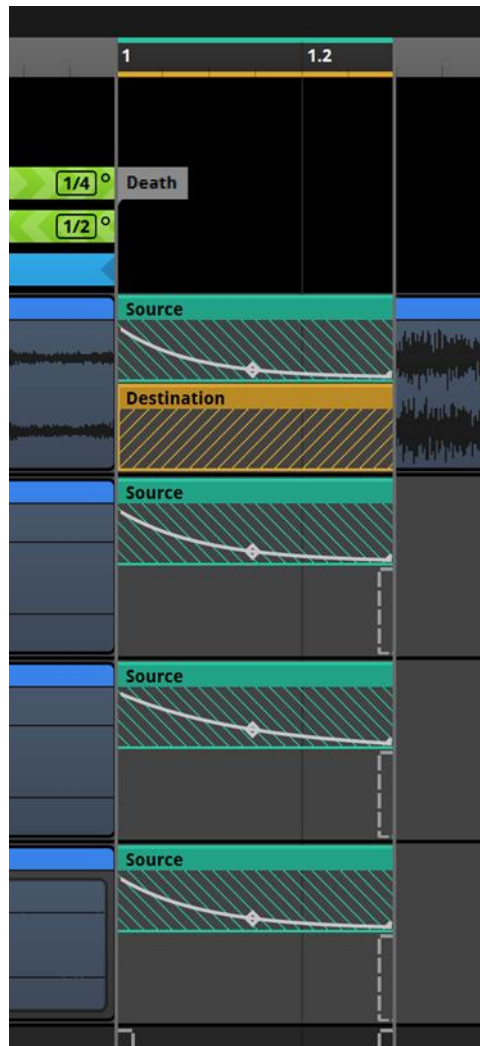


Slika 2.64. Fizionomija fragmenta smrti u odnosu na mernu skalu

Na slici 2.64 se jasno vidi mesto gde se vrh nalazi, kao i odnos njegove pozicije prema mernoj skali. On se nalazi tačno na mestu gde je i postavljena linija kursora, a njeno tačno mesto se još jasnije vidi u gornjem levom uglu gde je ispisano brojevima. Ovde se čita sledeće: 19. takt – druga četvrtina – treća šesnaestina. Dakle, ovo znači da nam se pozicija linije kursora nalazi tačno na trećoj šesnaestini (odnosno drugoj osmini) druge četvrtine 19. takta. Iz ovoga može da se zaključi da vremenski period od početka smrti do njenog vrha traje ukupno jednu četvrtinu i jednu osminu (označeno markerima na sl. 15.4). Stoga ćemo i trajanje tranzicionog

prostora da podredimo ovom merenju. Cilj nam je da se ono što je prethodilo isključi u roku trajanja rasta, odnosno dostizanja vrha.

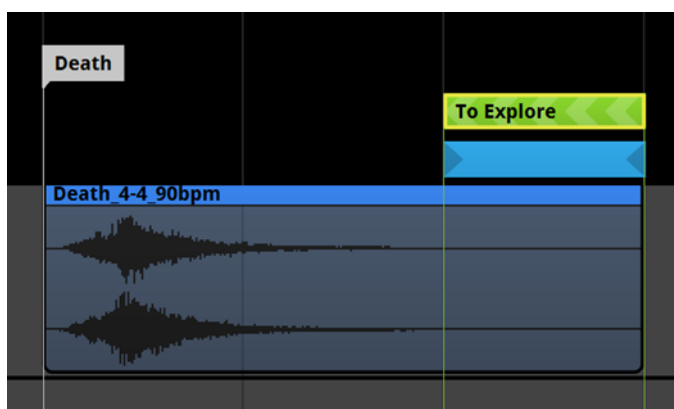
Pri postavci trajanja treba strogo da vodimo računa o brojevima koji nam se prikazuju u mernoj skali. Pogledajmo sliku 2.65. Broj 1 svakako označava početak tranzicionog prostora, pa samim tim i početak takta. Broj 1.2 označava kraj prve, odnosno početak druge četvrtine u okviru tog istog takta. Svaka četvrtina je podeljena na četiri jednaka dela, tj. četiri šesnaestine (označeno rezovima). Ostaje nam samo da ih pažljivo prebrojimo. Celokupna postavka tranzicionog prostora, uključujući njegovo trajanje kao i odnose između izvora i destinacije bi na kraju trebalo da izgleda kao na već pomenutoj slici.



Slika 2.65. Postavka tranzicionog prostora usmerenog ka muzici smrti

S obzirom da prelazak u stanje smrti treba da bude praktično momentalan, moramo da povedemo računa i o kvantizaciji koju ćemo da postavimo. Muzika treba da očuva svoju logiku, ali istovremeno i da ispoštuje tok dešavanja u video-igri. Stoga ćemo ovde da postavimo da kvantizacija bude jednaka četvrtini takta. To znači da ćemo sada, za razliku od tranzicija između istraživačke i akcione muzike da imamo dvostruko više mogućih mesta prekida muzičkog toka. Ovo će da omogući blagovremenu reakciju muzike na promenu vrednosti parametra *State*.

Setimo se da je naš imaginarni scenario video-igre osmišljen tako da nakon smrti igrač može da bira da li želi da proba ponovo. Ostalo je samo da rešimo na koji ćemo način da sprovedemo čekanje na igračevu odluku. To ćemo da učinimo tako što ćemo na sam kraj fragmenta da postavimo lup i tranzicionu regiju. S obzirom da fajl traje tri takta i da je poslednji praktično tišina, postavilićemo ih na poslednji takt (slika 2.66).



Slika 2.66. Lup tišine

Dakle, mi smo ovde zapravo postavili da nam se tišina ponavlja. To zvuči besmisleno, ali s obzirom da želimo da se muzički tok zaustavi dok igrač ne odluči šta dalje, ovo je način da praktično zarobimo muzički tok do tog trenutka. Kada igrač pređe u stanje smrti, izvršiće se tranzicija na destinacioni marker *Death* i kursor će da se premesti na fragment smrti i nastavi svojim putem. Onog trenutka kada uđe u zonu lup-regije, nastaviće da se kreće u okviru iste. Tranziciona regija je tu da odreaguje na igračevu odluku, drugim rečima, ako igrač reši da pokuša ponovo, parametar *State* će da bude jednak vrednosti *Explore*. Zato je, kao što se vidi na slici 2.66, postavljen odgovarajući uslov na tranzicionu regiju. Primećuje se da nam je ovo jedina tranziciona regija na koju nismo postavili kvantizaciju. To je iz prostog razloga što ne

postoji potreba za kvantizacijom tišine. Onog trenutka kada je parametar *State* jednak vrednosti *Explore*, kursor se vraća nazad na istoimeni destinacioni marker i ceo proces kreće iznova.

Ovim se završava izrada zadatka iz interaktivnog zvuka i muzike. Prešli smo osnovne funkcije unutar softvera *Fmod Studio* kao i načine rada i tretmana audio-sadržaja u video-igrama (kako se radi sa lupovima a kako sa varijacijama, tj. grupama audio-fajlova; kako funkcionišu horizontalne a kako vertikalne tranzicije/promene i sl.). Tokom praktikuma smo se vrlo često vraćali i nadovezivali na već pomenute procedure sprovedene po različitim ivotima i međusobno ih poredili. Ovo jasno govori da u velikoj meri može da se uoči podudarnost između tretmana zvuka i muzike. Svaki od njih ima svoje osobenosti, ali suština je specifičan tretman audio-sadržaja u video-igrama koji ih povezuje. Nakon ovog praktikuma, student bi trebalo da stekne znanja, veštine i samostalnost pri izradi sopstvenog zadatka. Takođe treba da stekne bolji uvid i razumevanje teorijskog dela predavanja iz predmeta *Principi zvuka i muzike u video-igrama 1*, kao i da se pripremi za predstojeće gradivo koje se tiče analize audio-sadržaja u video-igrama.

LITERATURA

- [1] David Sonnenschein
SOUND DESIGN: The expressive power of music, voice, and sound effects in cinema,
Studio City, CA: Michael Wiese Productions, 2001
- [2] YouTube
<https://www.youtube.com/watch?v=DI72MHPfh0k&ab>, sajt je posećen 01.09.2023
- [3] Aaron Marks
The complete guide to game audio – For composers, musicians, sound designers, and game developers
Oxford: Elsevier Inc. 2009
- [4] Michael Sweet
Writing interactive music for video games – A composer's guide
New Jersey, Pearson Education, Inc., 2015
- [5] YouTube
<https://www.youtube.com/watch?v=GSRpgLTBn9Y&ab>, sajt je posećen 01.09.2023
- [6] YouTube
<https://www.youtube.com/watch?v=eJEpNmpy7Uo&ab>, sajt je posećen 01.09.2023.
- [7] Karen Collins
Game Sound: an introduction to the history, theory and practice of video games music and sound design
Massachusetts, London, England, The MIT Press - Cambridge, 2008
- [8] YouTube
<https://www.youtube.com/watch?v=7PpSYcigCUQ&t=473s&ab>, sajt je posećen 01.09.2023.
- [9] YouTube
<https://www.youtube.com/watch?v=Dhc8qmHHU5c&ab>, sajt je posećen 01.09.2023.