



FAKULTET ZA INFORMATIKU I MENADŽMENT

Goran Vasić

Portfolio – Katalog radova

- Diplomski rad -



FAKULTET ZA INFORMATIKU I MENADŽMENT

PORTFOLIO – KATALOG RADOVA

- Diplomski rad -

Mentor:

Prof. dr Dragan Cvetković

Student:

Goran Vasić

Br. indeksa:

28/2003



FAKULTET ZA INFORMATIKU I MENADŽMENT

UNIVERZITET SINGIDUNUM
FAKULTET ZA INFORMATIKU I MENADŽMENT
Beograd, Danijelova 32

Kandidat: Goran Vasić

Broj indeksa: 28/2003

Smer: Računarska grafika i dizajn

Tema: PORTFOLIO – KATALOG RADOVA

Zadatak: Realizovati i opisati katalog radova u formi interaktivne CD-ROM prezentacije. Za rešenje zadatka, koristiti program Adobe Flash CS3 za izradu kataloga, i njegov programski jezik ActionScript 3.0 za potrebe funkcionalnosti. Omogućiti laku izmenu sadržaja i modifikovanje prezentacije.

MENTOR

Prof. dr Dragan Cvetković

DEKAN

Prof. dr Milan Milosavljević

SADRŽAJ:

1. Uvod	- 1 -
2. Rasterska i vektorska grafika	- 2 -
3. Adobe Flash	- 4 -
3.1 Radno okruženje Adobe Flash-a	- 5 -
3.2 Animacija u Flash-u	- 7 -
3.3 Formati za Flash-ove filmove	- 7 -
3.4 ActionScript 3.0	- 8 -
3.5 Dodavanje zvuka u Adobe Flash CS3	- 12 -
4. Uspešnost Flash prezentacije	- 16 -
5. Portfolio – Katalog radova	- 17 -
5.1 Početna stranica – O meni	- 20 -
5.2 Druga stranica – Galerija	- 21 -
5.3 Treća stranica – Video	- 23 -
5.4 Četvrta stranica – Usluge	- 26 -
5.5 Peta stranica – Kontakt	- 27 -
6. Zaključak	- 30 -
7. Literatura	- 31 -

1. Uvod

Sa pojavom Interneta, tačnije sa njegovom popularizacijom i prodiranjem u svest ljudi širom sveta, Web prezentacije su vizuelno bile prilično jednolične. U ogromnoj pustinji teksta sa sumornom sivom pozadinom, prezentacije sa bilo kakvom bojom ili grafičkim rešenjem su predstavljale prave osvežavajuće oaze. One su se izdvajale od ostalih prezentacija koje su imale istu ili sličnu tematiku, i samim tim, imale su i znatno veći broj posetilaca. Vizuelna komponenta prezentacije postala je presudna za njenu uspešnost, imajući u vidu da je u to vreme bilo kakav oblik interaktivnosti bio sveden na minimum zbog tehnoloških ograničenja.

Kako se svetska mreža (Internet) sve više razvijala, menjale su se i ciljne grupe na koje je ona bila fokusirana. Web prezentacije su poprimile ulogu medija za lično izražavanje, učenje ili pak trgovinu. Trebalo je na neki način učiniti prezentaciju jedinstvenom i vizuelno primamljivom za ciljnu grupu, pa se javila i potreba za dodavanjem veće količine slika. Standardni formati rasterskih slika za Web – JPEG, GIF i PNG – obezbeđuju kompresiju koja ih čini pogodnim za brzo preuzimanje, pa je i povećavanje količine grafičkog sadržaja u prezentacijama bilo ostvarljivo, bez većeg uticaja na vreme potrebno za pristupanje sadržaju. Međutim, Web dizajneri su vremenom počeli masovno da upotrebljavaju rasterske slike u svojim grafičkim rešenjima. Sa jedne strane, to je bio pun pogodak jer su posetioци umesto jednoličnih i dosadnih prezentacija dobijali živopisne i vizuelno bogate sadržaje. Ali sa povećanjem količine podataka, i vremenski period koji je bio potreban da se prezentacija prikaže korisniku je predugo trajao, pa je uglavnom dolazilo do kontra efekta, rezultirajući u osipanju broja posetilaca koji jednostavno nisu imali strpljenja da beskonačno čekaju. Web dizajneri su želeli bolji i efikasniji način da prikažu grafički sadržaj putem Interneta.

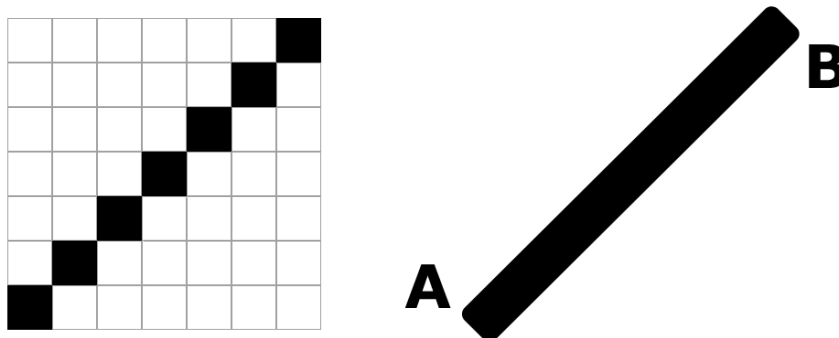
Rešenje je došlo u vidu upotrebe vektorske grafike, koja omogućuje da slike budu zanemarljivo male veličine bez gubitka u kvalitetu prikaza. Takođe, bitna prednost vektorskih u odnosu na rasterske slike je i mogućnost neometanog protoka informacija – striminga (engl. streaming). Striming dozvoljava da se neki grafički elementi odmah prikažu na ekranu, čak iako proces prenosa informacija putem Interneta nije završen, pa je i vreme potrebno da se prezentacija prikaže drastično skraćeno.

Kako je ovaj tip grafike sve više dobijao na popularnosti u oblasti Web dizajna, nastajali su i programski alati koji su podržavali rad sa vektorskom grafikom. Od mnoštva takvih alata, vremenom se izdvojio jedan program koji se specijalizovao u ove svrhe, pod imenom Adobe Flash. Zbog svojih karakteristika i mogućnosti, postao je standardni alat za dizajniranje Web prezentacija i dominantan softver u ovoj oblasti. Jedan od mnogo razloga zbog kojih je postao toliko popularan je i taj što pruža mogućnost kreiranja interaktivnih CD-ROM i Web prezentacija, što korisnicima pruža jedinstven doživljaj pri razgledanju sadržaja i neograničen broj mogućnosti dizajnerima pri osmišljavanju prezentacije. Sa razvojem Internet-a, vizuelni stil prezentacije, iako i dalje jako bitan, prestao je da bude presudan za njenu uspešnost, već je tu ulogu u velikoj meri poprimio kvalitet sadržaja prezentacije i njena interaktivnost. Prezentacije sa bogatom grafikom, ali sa malom količinom sadržaja i nedostatkom interaktivnosti vrlo brzo padaju u zaborav. Upravo za potrebe interaktivnosti je razvijen ActionScript, programski jezik koji je sastavni deo Flash-a. To je i glavni razlog zbog kog sam se ja odlučio da baš u ovom programu realizujem svoju prezentaciju, imajući u vidu interaktivnost kao važnu karakteristiku prilikom razgledanja kataloga, ali i sve ostale mogućnosti koje Adobe Flash pruža.

2. Rasterska i vektorska grafika

Podaci koji stvaraju rastersku grafiku (slika 01, levo) i podaci koji stvaraju vektorsku grafiku (slika 01, desno) imaju zajedničku osobinu da su i jedni i drugi matematičke instrukcije koje govore računaru gde i kako da stvara slike na ekranu. Ipak, rasterske slike su memorijski zahtevnije i rezultiraju manje prilagodljivom grafikom; vektorske slike su kompaktnije i potpuno matematički definisane.

Rasterska grafika – Instrukcije rasterske slike (bitmape) razlažu celu grafiku na pravougaonu mrežu sitnih, obojenih kvadrata tj. piksela (engl. *pixel*) koji se dodiruju i ne preklapaju. Vrednosti piksela su definisane za svaki piksel pojedinačno. Kada su sićušni pikseli prikazani u svojoj pravoj veličini, ljudsko oko ih teško može razlikovati, pa rasterska slika ne izgleda kao mozaik sastavljen od obojenih kvadrata, već se susedni pikseli stapaju i formiraju sliku koja je bogata bojama i detaljima. Ukupan broj piksela jedne slike predstavlja njenu rezoluciju. Ako je slika u RGB modu (*Red-Green-Blue*), tada svaki piksel zahteva tri bajta za njegovo opisivanje – po jedan bajt za svaku boju. Ako je slika crno bela, to znači da piksel zahteva samo jedan bit informacija. Dakle, od rezolucije slike i od njenog formata tj. od ukupnog broja piksela i količine informacija koju treba skladištiti u svaki piksel zavisi i veličina celog fajla. Bitmape sa većom rezolucijom uglavnom zahtevaju mnogo memorije, pa se zbog toga primenjuju razne vrste kompresije.



Slika 01 – Primer iscrtavanja linije u rasterskoj i vektorskoj grafici na ekranu

Najpopularniji format za kompresiju rasterske slike je Jpeg (jpg) format, koji posebnim algoritmima smanjuje broj informacija koje opisuju bitmapu, ali tako da se u velikoj meri očuva prvobitni kvalitet slike. Takođe se često upotrebljavaju i formati GIF (engl. *Graphics Interchange Format*) i PNG (engl. *Portable Network Graphics*). GIF podržava 8 bita informacija po pikselu, i dozvoljava paletu od 256 različitih boja u RGB modu za prikazivanje slike. Takođe podržava prikaz animacije, gde je za svaku sliku u animaciji tj. frejm (engl. *Frame*) rezervisana po jedna paleta sa 256 različitih boja u RGB modu. PNG je osmišljen tako proširi mogućnosti GIF formata i da ga uspešno zameni, pa se on upotrebljava kako bi omogućio lakši prenos slika putem Interneta. Ovaj format dozvoljava kompresiju bez gubitaka i može prikazivati slike u 24-bitnom (RGB) ili *grayscale* modu. Takođe može čuvati i podatke o stepenu providnosti pojedinačnih piksela.

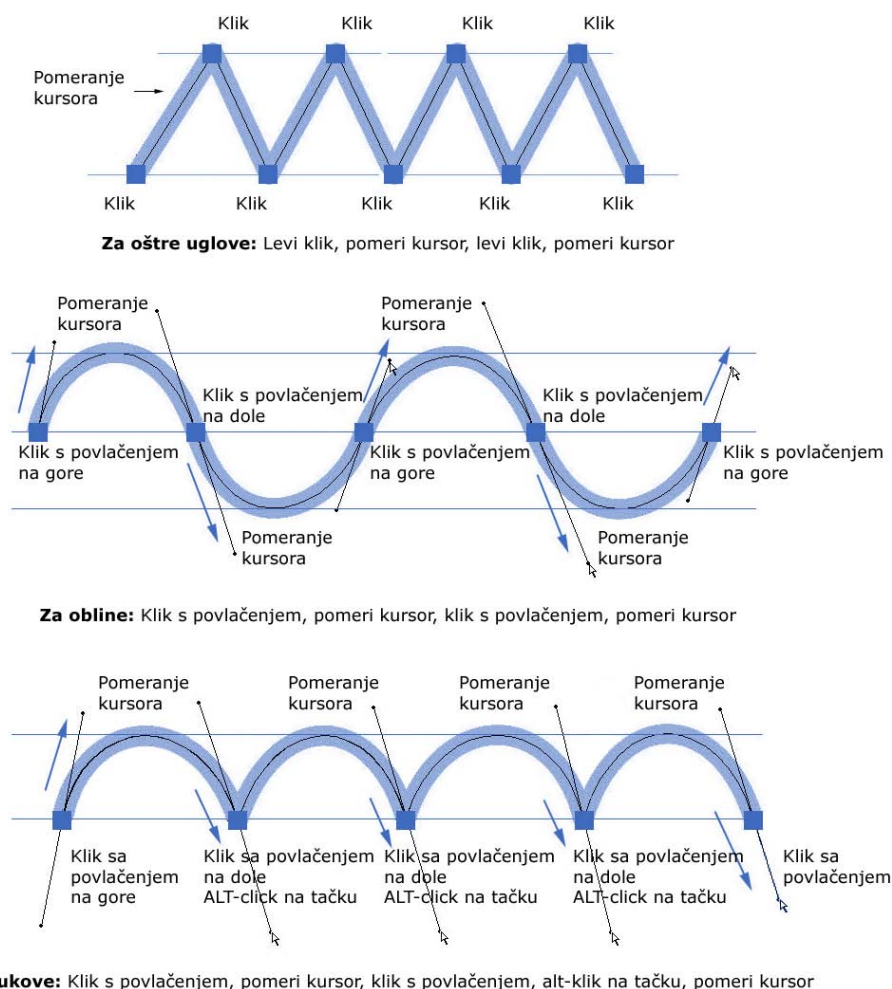
Rasterska slika se ne može povećati na veću rezoluciju bez gubitka kvaliteta prikaza. U tom slučaju, uveličana slika biva zamućena zbog primene anti-aliasing-a, odnosno tehnike „umekšavanja iskrzanih linija“ na slici koje se javljaju kao posledica menjanja njene rezolucije.

Vektorska grafika – za iscrtavanje slike koristi geometrijske oblike kao što su tačke, linije, krive i poligoni, a koji se stvaraju na osnovu matematičkih jednačina.

Vektorski oblici se mnogo lakše pamte u memoriji nego zahtevne rasterske slike. Skoro svi današnji računarski grafički prikazi prevode vektorsku sliku u rasterski format kako bi je prikazali na izlaznom uređaju kao što je monitor. Pojam vektorska grafika se uglavnom koristi u kontekstu dvodimenzionalne računarske grafike. Velikom većinom, svako prikazivanje 3D grafike se vrši pomoću 2D vektorske tehnike tj. iscrtavanje se vrši pomoću tačaka, linija i poligona.

Ako ponovo pogledate primere sa slike 01, videćete da je za datu liniju u vektorskoj grafici potrebno definisati samo početnu i završnu tačku. Program jednostavno povezuje tačke A i B i na ekranu se iscrtava linija. Podaci koji opisuju ovaj vektor su dužina, smer vektora i boja linije. Iz ovog primera se vidi da je menjanje dimenzija vektorske slike moguće bez ikakvog gubitka u kvalitetu prikaza, i to do beskonačnosti, jer će se linije i polilinije uvek iscrtavati isto, na osnovu matematičke jednačine.

Standardni programi koji su specijalizovani za rad sa vektorskom grafikom koriste kontrolna temena, odnosno tačke (engl. *node*) za oblikovanje vektorskih objekata i Bezjeove krive (slika 02) za crtanje krivudavih linija (engl. *Bézier curve* – po francuskom inženjeru Pierre Bézier-u, koji ih je koristio pri dizajniranju automobilskih školjki).



Slika 02 – Crtanje linija u Adobe Illustrator-u CS3 pomoću Bezjeovih krivih

3. Adobe Flash



Flash je započeo kao Future Splash Animator, izvanredan mali program kompanije Future Wave, za stvaranje i animaciju vektorske grafike. Softverska kompanija Macromedia je uvidela njegov ogroman potencijal i 1996. godine kupila Future Splash Animator, promenila njegovo ime u Macromedia Flash i promovisala ga kao sredstvo za stvaranje grafičkog sadržaja za World Wide Web.

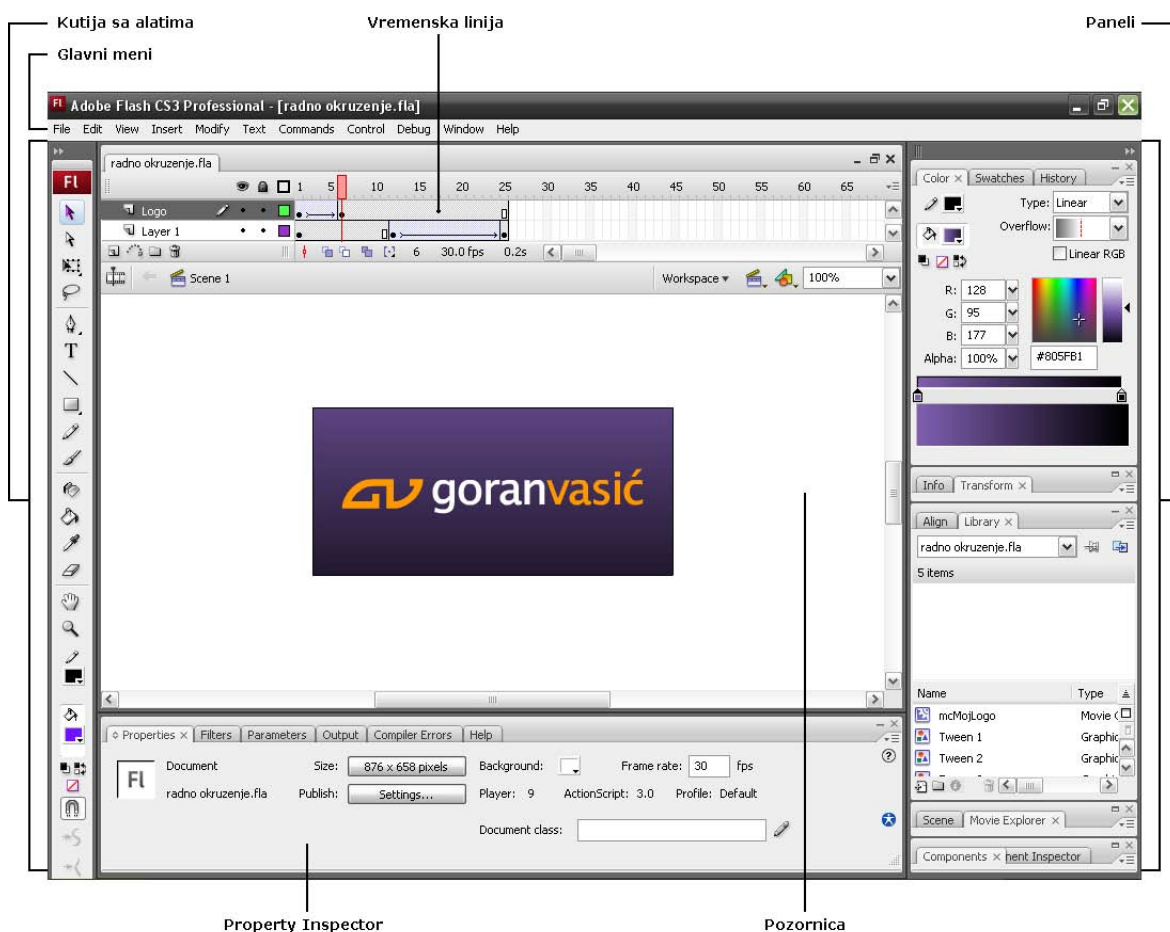
Budući da je Flash-ov .swf format (čita se svif) po svojoj prirodi *open-source*, i da je za prikazivanje Flash filmova potreban samo Flash Player (dodatak za Web čitač koji se razvija za sve vodeće platforme i besplatno distribuirati), Flash je za kratko vreme stekao ogromnu popularnost i postao dominantan među alatima za stvaranje Web prezentacija. Sa svakom novom generacijom Flash-a, Macromedia je implementirala nove funkcije i karakteristike. One su proširivale mogućnosti programa, u isto vreme zadržavajući lakoću upotrebe alata za crtanje i animaciju, kao i mogućnosti programiranja. Na jednom mestu sakupio je sve potrebne alatke: alatke za stvaranje grafike; alatke za animaciju te grafike; alatke za stvaranje elemenata interfejsa i interaktivnosti, kao i alatke za stvaranje HTML-a neophodnog za prikazivanje grafike, animacija i elemenata interfejsa u vidu Web strane u čitaču.

2005. godine, softverska kompanija Adobe Systems kupuje firmu Macromedia i dodaje Flash u kolekciju svojih programa za dizajn i obradu grafike. Kompanija Adobe je nastavila da unapređuje Flash, i on je postao neizostavan deo paketa Adobe Creative Suite, kao i jedan od tri najprodavanija programa kompanije Adobe – pored programa Adobe Photoshop i Adobe Dreamweaver. 2008. godine, sa pojavljivanjem paketa Adobe Creative Suite 4, izašao je i Adobe Flash CS4 Professional u pratnji dodatka Flash Player 10 za Web čitače. Flash od tada pruža mogućnost manipulisanja 3D objektima, objektno orijentisane animacije, upotrebu inverzne kinematike u animaciji, zatim unapređen ActionScript itd.

Standardni programi za ilustrovanje, kao što su Corel Draw i Adobe Illustrator, oslanjaju se na Bezejeve krive za stvaranje vektorskih oblika. Flash nudi slične alatke, ali takođe obezbeđuje i prirodne alatke za crtanje. One nude mogućnost rada s vektorima na neposredniji način, bez manipulisanja tangentama na krivoj ili specijalnim tačkama u liniji. Upravo ove prirodne alatke za crtanje u Flash-u obezbeđuju spontanost u radu, što privlači mnoge umetnike, kao i one koji to nisu. Flash pomaže početnicima da stvaraju jednostavnu grafiku, ali svako ko je upoznat sa animacijom može da koristi Flash-ove alatke u stvaranju veoma kompleksnih animacija. Njegov programski jezik, ActionScript, omogućava početnicima da dodaju interaktivne elemente na svoje prezentacije. ActionScript u verziji 3.0 je postao dovoljno moćan da ozbiljni korisnici mogu da stvaraju veoma sofisticirane interfejse i elemente aplikacije. Flash može da inkorporira zvuk i video, što može da unese još više živosti u prezentacije. Animacija u programu Flash se ne ograničava samo na stvaranje crtanih filmova, već obuhvata i elemente za navigaciju, kao što su dugmad i meniji. Dizajner nije ograničen samo na stvaranje prezentacija za Web, jer Flash upotrebom samostalnih projektoru dozvoljava da se filmovi distribuiraju i putem CD-ROM uređaja ili elektronske pošte. Moguće je prebaciti Flash format u druge formate, kao što su QuickTime ili Windows AVI filmovi. Ipak, stvaranje i poboljšavanje Web prezentacije su primarni ciljevi Flash-a.

3.1 Radno okruženje Adobe Flash-a

Flash-ovo radno okruženje se razlikuje od okruženja drugih programa na koje je većina korisnika navikla (slika 03).



Slika 03 – Radno okruženje u programu Adobe Flash CS3

Na centralnom delu ekrana nalazi se pozornica, koja predstavlja dvodimenzionalni koordinatni sistem sa početkom u gornjem levom uglu. Dimenzije pozornice se određuju kao globalne za ceo film i izražene su u pikselima. Pri samom vrhu ekrana nalazi se glavni meni programa, a odmah ispod je vremenska linija. Vremenska linija deli film na pojedinačne slike tj. frejmove (engl. *frame*), gde svaki frejm predstavlja po jedan trenutak u vremenu – recimo 1/30 sekunde. To znači da kada bi se film pustio, prikazivalo bi se po 30 slika u svakoj sekundi, i to jedna za drugom.

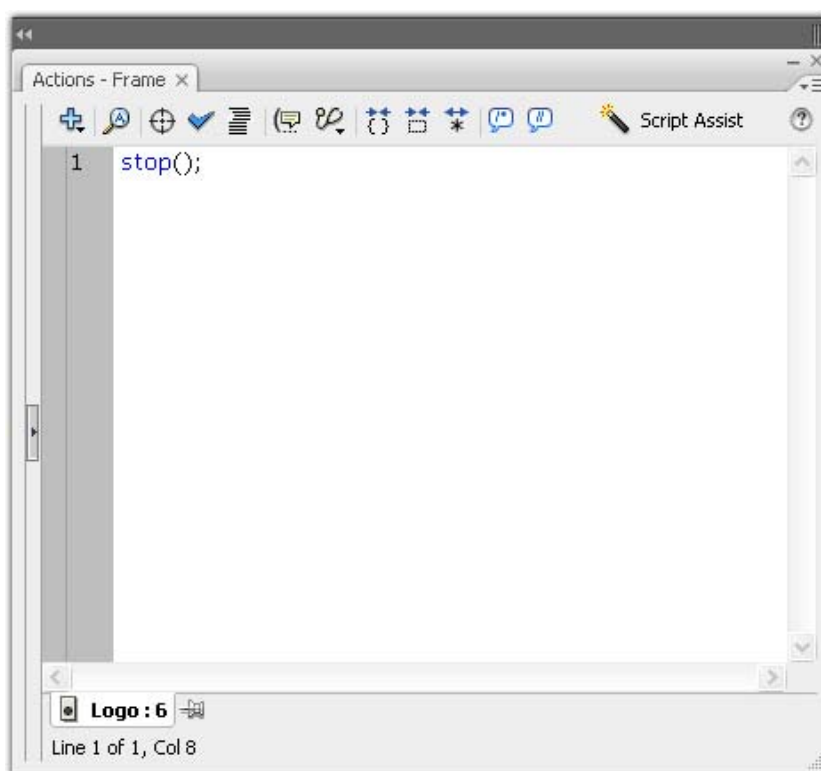
Ako vam ovo nije baš najjasnije, zamislite da je pozornica na ekranu u stvari knjiga, a da je bela površina pozornice u stvari prva stranica te knjige. Tada bi prvi frejm odgovarao prvoj strani, drugi drugoj itd. Kada u vremenskoj liniji kliknete na drugi frejm, to znači da ste okrenuli jedan list te knjige i gledate u sadržaj druge stranice. Pokretanjem filma, Flash automatski „lista tu knjigu“ određenom brzinom. Pošto bi za potrebe animacije trebalo na svakoj strani knjige nacrtati neku sliku, kako bi se prilikom „listanja“ tih stranica stvorila iluzija pokreta, u Flash-u prilikom animiranja svaki frejm u vremenskoj liniji treba da ima neki sadržaj koji se malo razlikuje od sadržaja prethodnog frejma.

Vremenska linija pruža mogućnost upotrebe slojeva tj. lejera (engl. *layers*) za potrebe lakšeg stvaranja grafike. Slojeve možemo predstaviti kao providne folije po kojima se može crtati, pisati tekst itd. Ako su dva objekta u Flash-u vidljiva na pozornici, pri čemu je svaki na zasebnom sloju i njihovi sadržaji se međusobno preklapaju, tada će na mestu gde dolazi do preklapanja biti vidljiv sadržaj gornjeg sloja jer se nalazi „iznad“ donjeg.

Na levoj strani je kutija sa alatima, a desno su paneli koji služe za pregled, organizaciju i modifikaciju elemenata Flash filma. Važan deo panela predstavlja biblioteka simbola, u koju se automatski smeštaju svi objekti koji su kreirani na pozornici, ali je u nju moguće i uvoziti eksterne fajlove poput zvukova, rasterskih slika itd.

Donji deo ekrana je rezervisan za kontekstni panel Property Inspector, koji prikazuje različite informacije na osnovu odabranog elementa ili alatke. Ovde je moguće podesiti dimenzije pozornice, zatim brzinu kojom se frejmovi prikazuju u sekundi, dodati filtere za selektovani element pozornice (senka, zamućenje itd.), kao i podesiti parametre za trenutno izabranu alatku. U Property Inspector-u se takođe mogu prikazati informacije ili greške prilikom testiranja filma u jezičku Output.

Važan deo radnog okruženja je i panel Actions, koji se, putem prečice na tastaturi, prikazuje pritiskom na funkcijski taster F9 (Slika 04). U njegovom prozoru se piše ActionScript kôd za prezentaciju, ali pod uslovom da je korisnik pre toga kliknuo na frejm u kom će se skripta izvršavati. U prethodnim verzijama Flash-a, na primer, bilo je moguće upisivati kôd direktno u selektovano dugme na pozornici i na taj način mu dodati funkcionalnost. ActionScript 3.0 ne dozvoljava upisivanje kôda direktno u objekat, nego se on vezuje isključivo za određeni frejm na vremenskoj liniji.



Slika 04 – Panel Actions

3.2 Animacija u Flash-u

Flash koristi standardne tehnike za animaciju da bi stvorio iluziju pokreta. Standardna animacija podrazumeva stvaranje serije nepokretnih slika, gde je svaka malo drugačija od prethodne. Kada se te slike brzo prikazuju jedna za drugom, stvara se iluzija pokreta. Flash nudi mogućnost da animator napravi samo nekoliko ključnih kadrova (*engl. keyframe*), na primer sa početnom i završnom pozicijom nekog objekta, a da program popuni praznine između njih. U tom slučaju, sličice (*engl. frame*) koje nedostaju između tih ključnih kadrova program kreira proračunavajući poziciju objekta u datom trenutku, u zavisnosti od dužine trajanja animacije.

Flash animirane filmove prikazuje brzinom koju korisnik određuje prilikom početnog podešavanja scene. U zavisnosti od zamućenosti slike (*engl. blur, motion blur*), kao i od broja prikazanih sličica u jednoj sekundi, ljudsko oko može razlikovati „tečnu“ animaciju koja stvara iluziju prirodnog pokreta od loše animacije, za koju je karakteristično „seckanje“ (*engl. flickering*) u kretanju objekta prilikom filmske projekcije. Za Flash prezentacije dovoljno je 30 sličica tj. frejmova u sekundi (*engl. fps – frames per second*) kako bi se stvorila iluzija prirodnog kretanja objekata na sceni.

3.3 Formati za Flash-ove filmove

Pod operativnim sistemom Windows, ekstenzija datoteke koja je rezervisana za Adobe Flash je *.fla* i takav fajl predstavlja radni fajl u kom korisnik kreira prezentaciju. Da bi se animacija koju je korisnik kreirao prikazala kao film, potrebno je kompajlirati sadržaj *.fla* datoteke u format koji podržava Adobe Flash Player, a to je *.swf* format (SWF - *engl. Shockwave Flash; izgovara se suif*). Konvertovanje u SWF format se obavlja tako što korisnik, dok je u radnom okruženju Flash-a, klikne na opciju *Control* koja se nalazi na glavnom meniju u gornjem delu ekrana, i zatim iz padajućeg menija izabere opciju *Test Movie*. Takođe postoji i prečica kojoj se pristupa istovremenim pritiskom tastera *Ctrl* i *Enter* na tastaturi, i ova kombinacija je u okviru Flash-a rezervisana upravo za testiranje filma, pa se na taj način skraćuje vreme rada zbog česte potrebe za testiranjem. SWF fajl se nakon kreiranja smešta u isti direktorijum gde se nalazi i radni fajl prezentacije. Konvertovani film se može puštati samo uz podršku Flash Player-a, koji se danas razvija za sve vodeće Web čitače i distribuira se besplatno. U slučaju da iz nekog razloga korisnik koji želi da pogleda prezentaciju nema Flash Player na svom računaru, moguće je iz Flash-a opcijom *File > Publish* kreirati projektor u vidu *.exe* fajla, koji će prikazivati film bez obzira na to da li je Flash Player instaliran ili ne.

Kako je rasla popularnost ovog formata zbog njegove male memorijske zahtevnosti i mogućnosti da prikazuje animiranu vektorsku i rastersku grafiku (što samo po sebi premašuje W3C standard), pojavila su se i programska rešenja koja mogu SWF format konvertovati natrag u *.fla* fajl. Korišćenjem ovih programa Web dizajneri mogu detaljno proučavati prezentacije koje su na njih ostavile jak utisak ili poseduju neku vrstu interaktivnosti za koju nisu sigurni na koji način je realizovana. Jedan od takvih programa koji daje dosta dobre rezultate je *SWF Decompiler* firme Sothink, koji trenutno pruža podršku i za SWF fajlove kreirane u Adobe Flash-u CS3.

3.4 ActionScript 3.0

ActionScript 3.0 predstavlja moćan, objektno-orijentisani programski jezik zasnovan na *ECMAScript*-u, i sastavni je deo programa Adobe Flash. *ECMAScript* je standardizovan od strane *Ecma International*, i ovaj programski jezik je u širokoj upotrebi na Web-u, naročito u formi dva njegova najpoznatija dijalekta: *Javascript* i *Jscript*.

ActionScript je stvoren za potrebe razvijanja Web prezentacija i softverskih aplikacija korišćenjem platforme Adobe Flash Player. Pokreće ga virtuelna mašina AVM (*ActionScript Virtual Machine*) koja je sastavni deo Flash Player-a. Programski kôd je sadržan u samom *.swf* fajlu, mada je podržano i korišćenje eksternih datoteka sa klasama. ActionScript je razvila kompanija Macromedia za potrebe kontrolisanja vektorskih 2D animacija i implementiranja interaktivnosti u Web prezentacije, a od 2005. godine je u vlasništvu kompanije Adobe.

Od svoje prve verzije, koja je obuhvatala najjednostavnije komande koje su služile za kontrolisanje načina na koji se prikazuje Flash film (kao što su „play“, „stop“, „gotoAndPlay“ itd.) pa do danas, ActionScript je prešao dugačak razvojni put. U verziju Flash 4 koja se pojavila 1999. godine, implementirane su nove mogućnosti za ActionScript. Uvedene su promenljive, izrazi, operatori, uslovna grananja i programske petlje. U to vreme se u priručniku za Flash 4 ovaj programski jezik nazivao imenom „Actions“. Tek od pete verzije Flash-a je uvedeno ime „ActionScript“, i tada je pružio mogućnost proceduralnog i objektno-orijentisanog programiranja. ActionScript se od tog trenutka zasniva na *ECMAScript*-u. U septembru 2003. godine, kada se pojavio Flash Player 7, predstavljen je i ActionScript 2.0, koji je bio kompatibilan sa prethodnom verzijom. Između ostalog, uvedene su klase, što je dalje proširilo mogućnosti ovog programskog jezika. Ova verzija je bila odgovor na zahteve Flash programera koji su želeli programski jezik koji će lakše moći da izlazi na kraj sa glomaznim i kompleksnim aplikacijama. Od tog trenutka je ActionScript 2.0 počeo da liči na programske jezike kao što su *Java* i *C++*. Sredinom 2006. godine, ActionScript 3.0 je debitovao na platformi Flash Player 9.0 u pratnji odgovarajuće tehnologije pod imenom Adobe Flex 2.0. ActionScript 3.0 je fundamentalno preuređen i optimizovan, pa je za njegove potrebe stvorena i nova virtuelna mašina – AVM2. Zbog ovakvog poteza, ActionScript 3.0 je pružao bolje performanse prilikom izvršavanja, po cenu nekompatibilnosti sa prethodnim verzijama ovog programskog jezika. Jednostavne ActionScript 3.0 aplikacije sa minimalnim brojem linija kôda su možda po veličini veće od ActionScript 2.0 aplikacija koje obavljaju istu funkciju, ali prilikom realizacije kompleksnih aplikacija sa stotinama linija kôda, ActionScript 3.0 ima višestruko veću prednost upravo zbog bolje optimizovanosti kôda. Sledeći primer ukazuje na razlike u pisanju kôda za poslednje dve verzije ActionScript-a:

ActionScript 2.0:

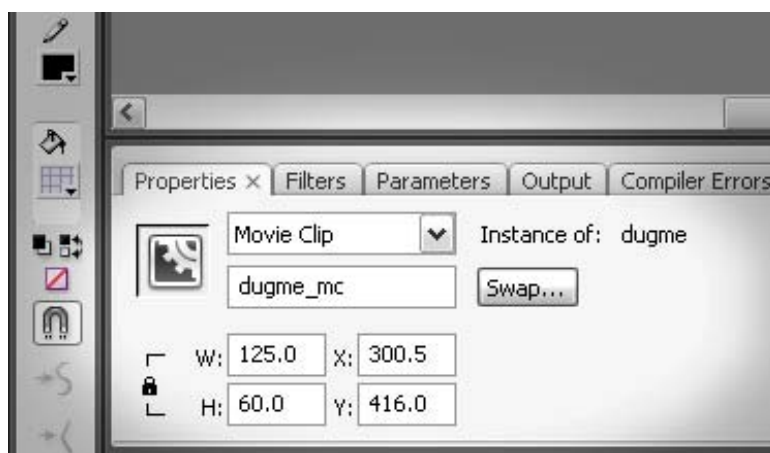
```
createTextField("tema", 0, 0, 0, 100, 100);  
tema.text = "Katalog radova";
```

ActionScript 3.0:

```
var tema:TextField = new TextField();  
tema.text = "Katalog radova";  
this.addChild(tema);
```

U prethodnom primeru se može videti da obe verzije ActionScript-a obavljavaju istu radnju, ali razlika je u tome što ActionScript 2.0 ima dve linije koda: prvo dodaje prazno tekstualno polje na scenu prezentacije, pa tek posle toga upisuje tekst. ActionScript 3.0 prvo kreira prazno tekstualno polje u memoriji računara, zatim u njega unosi tekst, pa se na scenu dodaje tekstualno polje koje, u trenutku kada se prvi put pojavi na ekranu, već ima neki sadržaj.

ActionScript 3.0 objektima na sceni pristupa preko instanci (slika 05). Pristupanje ugnježdenim instancama se vrši navođenjem imena instanci po redu, pri čemu se imena razdvajaju tačkama. Na ovaj način je moguće pristupiti čitavom sadržaju prezentacije i kontrolisati na koji način će se ponašati na sceni ili menjati njihove parametre kao što su na primer dimenzije, trenutna pozicija, transparentnost itd.



Slika 05 – Instanca objekta „dugme“ sa imenom „dugme_mc“

Menjanje parametra se obavlja tako što se iza imena instance dodaje tačka i ime željenog parametra, zatim, odvojeno znakom jednakosti, vrednost parametra. Svaka linija kôda se završava znakom „;“. Vrednosti parametara mogu biti različite, i zavise od tipa parametra. Osnovni tipovi podataka su: Boolean, int, Null, Number, String, uint i void.

Ako u biblioteci prezentacije imamo *Movie clip* sa imenom *dugme*, a na sceni njegovu instancu sa imenom *dugme_mc*, tada vrlo lako možemo pristupiti ovom objektu. Sledeći primer menja transparentnost instance *dugme_mc* na vrednost od 50%, i dodaje ovom *Movie clip*-u kursor miša koji je karakterističan za dugme:

```
dugme_mc.alpha = 0.5;  
dugme_mc.buttonMode = true;
```

Na ovaj način je moguće pretvoriti bilo koji deo prezentacije u dugme ili u kompleksne elemente interfejsa koji mogu u sebi sadržati čitavu animaciju ili čak uticati na druge objekte na sceni. To pruža dizajneru veliku slobodu pri osmišljavanju prezentacije.

Prethodni primer možemo dodatno proširiti kako bi kreirali malo kompleksnije dugme. ActionScript 3.0 pruža mogućnost da se određeni deo kôda koji bismo napisali za potrebe ovog dugmeta iskoristi i za bilo koji drugi element prezentacije, pa se na taj način smanjuje i ukupan broj linija u kôdu tj. povećava se fleksibilnost kôda. U tom slučaju, potrebno je dodati liniju kôda koja će novom elementu dodeliti istu interaktivnost. Evo i primera realizacije kompleksnijeg dugmeta, sa opisom tj. komentarima pojedinačnih linija:

```
//stopira trenutni Movie clip
stop();

//u flash se uvoze prethodno definisane klase ili paketi za potrebe ActionScript-a
import fl.transitions.Tween;
import fl.transitions.easing.*;

//dugme je 20% providno i dodeljuje mu se kursor koji je karakteristican za dugme
dugme_mc.alpha = 0.2;
dugme_mc.buttonMode = true;

//inicijalizacija promenljive koja ce kontrolisati transparentnost instance dugme_mc
var alfa:Tween;

//dugmetu se dodeljuju funkcije koje ce mu dati interaktivnost
dugme_mc.addEventListener(MouseEvent.ROLL_OVER, fadeIn);
dugme_mc.addEventListener(MouseEvent.ROLL_OUT, fadeOut);
dugme_mc.addEventListener(MouseEvent.CLICK, naKlik);

//funkcija se izvrsava kada korisnik predje misem preko dugmeta, i ono je neprovidno
function fadeIn(e:MouseEvent):void {
    alfa = new Tween(e.target,"alpha",Regular.easeOut,e.target.alpha,1,6,false);
}

//funkcija se izvrsava kada korisnik skloni mis sa dugmeta, dugme postaje providno
function fadeOut(e:MouseEvent):void {
    alfa = new Tween(e.target,"alpha",Regular.easeOut,e.target.alpha,0.2,6,false);
}

//funkcija se izvrsava kada korisnik klikne na dugme i tada se prikazuje poruka
function naKlik(e:MouseEvent):void {
    trace("Kliknuli ste na " + e.target.name);
}
```

Interaktivnost dugmeta *dugme_mc*, odnosno funkcije *fadeIn*, *fadeOut* i *naKlik* se mogu dodeliti novom elementu sa imenom *noviElement_mc* ubacivanjem sledećih linija u kôd iz prethodnog primera, bez bilo kakve druge prepravke:

```
noviElement_mc.addEventListener(MouseEvent.ROLL_OVER, fadeIn);
noviElement_mc.addEventListener(MouseEvent.ROLL_OUT, fadeOut);
noviElement_mc.addEventListener(MouseEvent.CLICK, naKlik);
```

Programski jezik ActionScript 3.0 je osmišljen tako da pruži unapređen i konzistentan model programiranja; u skladu sa industrijskim standardima i sa performansama koje daju bolje rezultate nego u prethodnim verzijama. Iako je u trenutku kada se pojavio predstavljao novi model programiranja, Web developer-i sa osnovnim nivoom znanja o objektno-orijentisanom programiranju su za kratko vreme mogli da nauče kako da ActionScript 3.0 primene u razvoju svojih aplikacija.

ActionScript 3.0 je ispunio sledeće zahteve:

- **Sigurnost:** Ovaj jezik ima sintaksu koja je nedvosmislena, pa je i kôd lako održiv.
- **Jednostavnost:** Dovoljno je intuitivan tako da programeri mogu lako da čitaju i pišu kôd, pa nema potrebe za stalnim konsultovanjem priručnika.
- **Performanse:** Pruža mogućnost pisanja kompleksnih programa koji će raditi efikasno i bez grešaka.
- **Kompatibilnost:** Pruža delimičnu kompatibilnost sa prethodnim verzijama i kompatibilnost za naredne verzije po industrijskim standardima. ActionScript 3.0 je dijalekt ECMAScript-a koji formuliše naredbe i mogućnosti ActionScript-a 2.0, dodaje mogućnosti ECMAScript-a za rad sa XML-om (E4X), i ujedinjuje jezik u koherentnu celinu.

Sastoji se od dva dela: osnovnog programskog jezika i Flash Player API-ja (*Application Programming Interface*). Jezik je definisan osnovnim elementima čijom upotrebom se ispisuju programski kôdovi aplikacija. Primer ovih elemenata su naredbe, uslovi, programske petlje, tipovi podataka i mnogi drugi. Flash Player API čine klase i funkcije koje predstavljaju i omogućavaju pristup specifičnim funkcionalnostima Flash Player-a.

Ovaj kratak pregled ActionScript-a 3.0 ocrtava samo neke od mnogih mogućnosti ovog programskog jezika. Od kada se pojavio u kombinaciji sa desetoj verzijom Flash Player platforme, ActionScript 3.0 pruža punu podršku za rad sa objektima u 3D prostoru, animiranje upotrebom inverzne kinematike, objektno orijentisane animacije, uvodi nove mogućnosti API-ja i obezbeđuje programerima i dizajnerima izuzetno moćan alat prilikom izrade interaktivnih prezentacija ili kompleksnih softverskih aplikacija.

3.5 Dodavanje zvuka u Adobe Flash CS3

Adobe Flash je robustan program koji podržava rad sa najpopularnijim formatima zvučnog zapisa, u koje spadaju:

- **MP3 (Moving Pictures Expert Group Level-1, Audio Layer-3):** Ovo je najrasprostranjeniji format za zapisivanje zvučnog signala. Takođe predstavlja standard za digitalnu audio kompresiju sa gubitkom, posebno za potrebe transfera muzike na audio player i njenu reprodukciju. Svoju popularnost je stekao upotrebom Internet servisa kao što su Napster i razni P2P programi (engl. *peer-to-peer*), jer je muzika zapisana u MP3 formatu lako mogla da se prenosi putem Mreže. MP3 je format zatvorenog tipa (engl. *closed-source*) i zaštićen je patentom.
- **WAV (od engl. reči Waveform):** Audio format za skladištenje zvučnog zapisa na PC računarima. WAV spada u grupu RIFF formata pa se njegov zapis sastoji iz „parčića“ (engl. *chunks*). Ovo je standardni format operativnog sistema Microsoft Windows za zapis audio signala i uglavnom čuva nekompresovani audio signal.
- **QuickTime:** Ovaj format ima ekstenziju *.qta* ili *.mov* i može u sebi čuvati audio signal zapisan u različitim formatima kao što su Apple Lossless, AIFF, MP3, WAV, CDDA itd. Koristi se na operativnim sistemima Mac OS X i Microsoft Windows.
- **AIFF (Audio Interchange File Format):** Koristi se pre svega na Apple Macintosh računarskim sistemima za potrebe čuvanja nekompresovanog audio signala, dok se za kompresovani signal koristi AIFF-C format. Razvijen je u saradnji sa Apple Computers 1988. godine, a zasnovan je na IFF formatu (engl. *Electronic Arts' Interchange File Format*) koji se koristio na Amiga računarima. Pruža iste frekvencije sempliranja audio signala kao i WAV format i upotrebljava se u profesionalnim audio i video aplikacijama.

Flash zvučne zapise prilikom njihovog ubacivanja u film smešta u biblioteku simbola. Ubacivanje audio fajlova se vrši tako što korisnik bira *File > Import > Import to Library...* preko glavnog menija programa (slika 06).

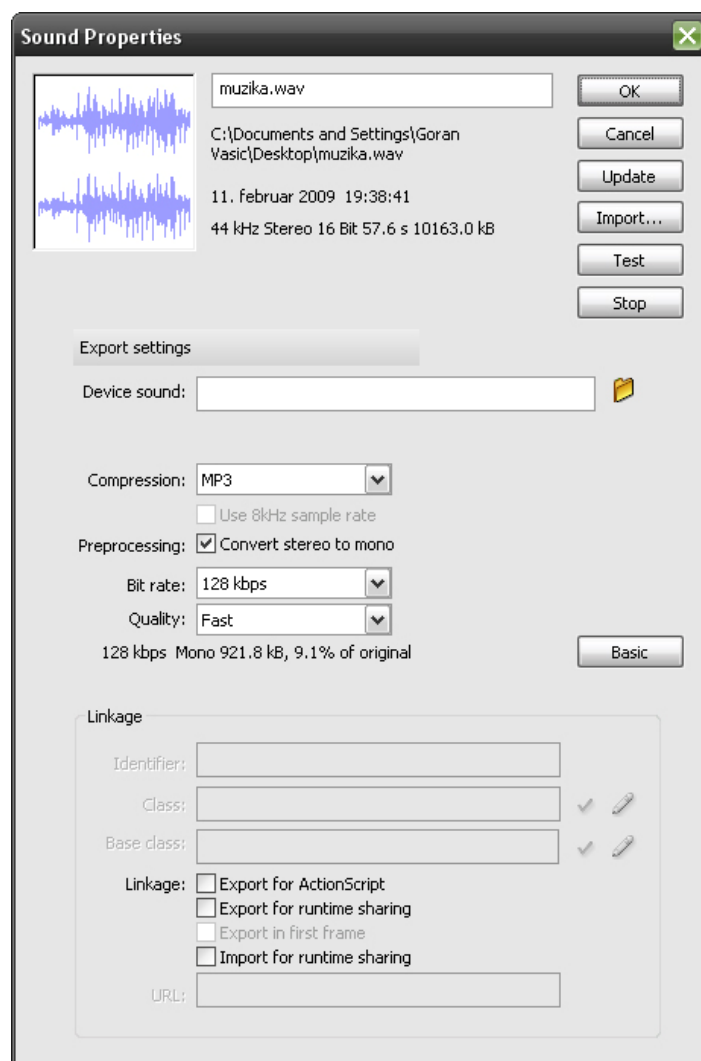
Ovde je moguće podešavati razne parametre audio fajla: recimo, preimenovati ga za potrebe boljeg organizovanja biblioteke, ili dodeliti klasu koja će omogućiti upotrebu zvuka u ActionScript-u 3.0 i koja će biti eksportovana pri kompajliranju prezentacije u *.swf* fajl. Ovom podešavanju se pristupa desnim klikom na simbol biblioteke koji reprezentuje audio fajl, nakon čega je potrebno izabrati opciju *Linkage...* iz ponuđenog menija.

Slika 06 – Zvuk koji je uvezen u biblioteku simbola



Pre samog importovanja zvuka u biblioteku prezentacije, dizajner se mora zapitati da li će veličina tog audio fajla u većoj meri uticati na vreme potrebno da se prezentacija učita. Ne znači da je upotreba boljeg kvaliteta audio signala uvek i najbolji izbor prilikom realizacije prezentacije. Recimo, zvuk koji se čuje u trenutku klika mišem na neko dugme traje jako kratko, pa je velika verovatnoća da korisnik neće ni primetiti lošiji kvalitet tog zvuka. Ali će zato dizajner upotrebom audio zapisa sa manjom frekvencijom sempliranja osigurati korisniku brže učitavanje cele prezentacije. Iz tog razloga je dovoljno za potrebe samog interfejsa koristiti 8-bitni audio zapis sa frekvencijom sempliranja od 22 KHz. Takođe, ako korisnik poseduje zvučnike lošijeg kvaliteta, muzika u pozadini sa ovom frekvencijom sempliranja se neće po kvalitetu bitnije razlikovati od muzike CD kvaliteta.

Ako u biblioteku simbola uvezete audio zapis koji nije kompresovan, dobro je znati da Flash nudi mogućnost njegove naknadne kompresije prilikom kompajliranja. Na ovaj način sami određujete kvalitet i stepen kompresije zvuka za potrebe prezentacije (slika 07).



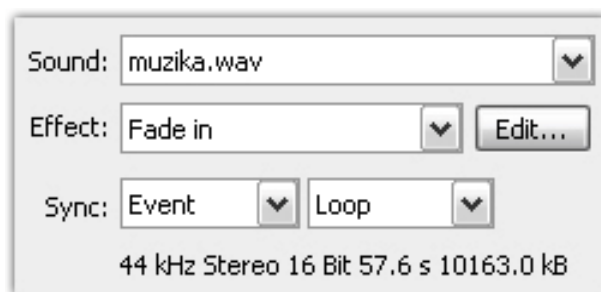
Slika 07 – Određivanje konačnog kvaliteta audio zapisa u Adobe Flash-u

Treba naglasiti da se prilikom ovakvog načina rada dodatno produžava vreme testiranja prezentacije, jer se svaki put iznova vrši kompresija audio zapisa.

Kada se zvučni zapis uveze tj. kopira u biblioteku simbola, on više ne zavisi od originalnog audio fajla na hard disku. Ovo povećava veličinu radnog *.fla* fajla, ali prilikom njegovog prenosa na drugi računar nije potrebno kopirati i ostale fajlove koji se koriste u filmu. Zbog toga je jako korisna opcija za osvežavanje audio zapisa u biblioteci simbola (engl. *update*). Ovom opcijom moguće je lako zameniti zvuk koji prati neku animaciju, bez izmene same prezentacije ili ActionScript 3.0 kôda. Pre svega, potrebno je sačuvati novi audio zapis pod istim imenom i na istom mestu hard diska sa kog je bio ubačen u biblioteku, a zatim i osvežiti audio zapis u biblioteci.

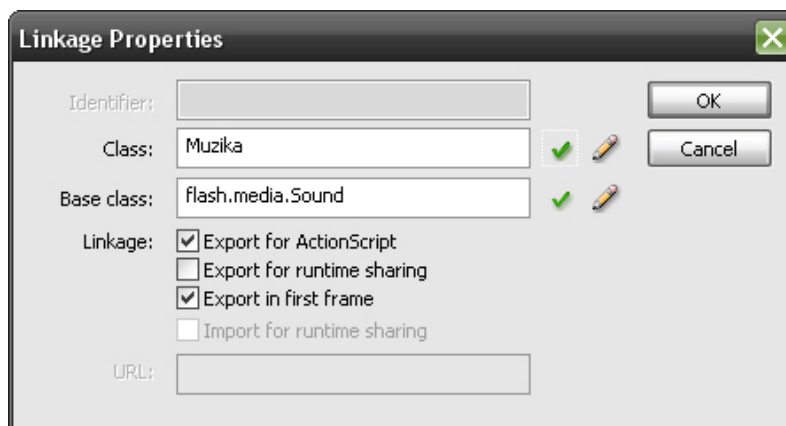
Dodavanje zvuka prezentaciji ili nekom elementu interfejsa se može obaviti na dva načina:

Prvi način je dodavanjem zvuka na određeni ključni frejm (engl. *keyframe*) vremenske linije. Kada je taj frejm izabran, u Property Inspector-u treba iz padajućeg menija *Sound* izabrati željeni zvuk iz biblioteke (slika 08). Ovde je moguće podesiti na koji način će se zvuk reprodukovati i koliko puta za redom.



Slika 08 – Podešavanja za zvuk na vremenskoj liniji

Drugi način je upotrebom ActionScript-a 3.0 čime je osigurana znatno veća kontrola nad reprodukovanjem zvučnog zapisa. Pre svega, potrebno je izabrati opciju *Linkage* i u ponuđenom prozoru izabrati opciju *Export for ActionScript*. Unošenjem imena u polje *Class* i potvrđivanjem Flash-ovog upozorenja, simbolu iz biblioteke se dodeljuje klasa preko koje ActionScript može pristupati audio fajlu (slika 09). Ime klase bi trebalo da započinje velikim slovom i ne sme sadržati nedozvoljene karaktere.



Slika 09 – Dodela klase zvučnom zapisu putem opcije *Linkage*

ActionScript 3.0 može reprodukovati zvuk bez upotrebe zvučnog kanala (engl. *sound channel*) ili kroz zvučni kanal. Takođe, osim zvukova iz biblioteke simbola, ActionScript 3.0 može učitavati i reprodukovati eksterne audio fajlove. Sledeći primer prikazuje prvi način: U biblioteci simbola se nalazi audio fajl *muzika.wav* kome je preko opcije *Linkage* dodeljena klasa *Muzika*. ActionScript se izvršava na prvom frejmu, pa se zvuk čuje odmah po pokretanju filma:

```
//povezivanje promenljive zvuk sa fajlom muzika.wav preko klase Muzika
var zvuk:Muzika = new Muzika();
//izvršavanjem sledece linije ActionScript 3.0 zapocinje reprodukciju zvuka
zvuk.play();
```

Prethodni primer daje isti rezultat kao kada bi ubacili zvuk u prvi frejm vremenske linije.

Prednost učitavanja audio fajlova sa eksterne lokacije je u tome što se veličina *.swf* fajla uopšte ne povećava, pa se znatno skraćuje vreme potrebno da se prezentacija prikaže posetiocu. Ovo se obavlja preko *URLRequest*-a tako što ActionScript 3.0 govori Flash-u da učita audio fajl čije ime i lokacija su definisani znacima navoda. Za potrebe sledećeg primera, audio fajl sa imenom *muzika.mp3* je smešten u direktorijum gde se nalazi radni *.fla* fajl:

```
//Flash preko promenljive ucitajZvuk pristupa eksternom fajlu muzika.mp3
var ucitajZvuk:URLRequest = new URLRequest("muzika.mp3");
//kreira se promenljiva zvuk koja ce prihvatiti eksterni fajl
var zvuk:Sound = new Sound();
//u promenljivu zvuk se ucitava eksterni fajl preko definisane putanje
zvuk.load(ucitajZvuk);

//formira se promenljiva zvucniKanal koja ce reprodukovati zvuk
var zvucniKanal:SoundChannel;
//ActionScript 3.0 zapocinje reprodukciju zvuka kroz zvucniKanal i pusta ga 3 puta
zvucniKanal = zvuk.play(0,3);
```

Kompajlirani SWF fajl iz ovog primera (zauzima oko 4 KB memorije) manje je veličine od fajla iz prethodnog primera (oko 110 KB), jer on ne sadrži u sebi čitav zvuk već samo pet linija ActionScript kôda. Takođe, ako fajl *muzika.mp3* zamenimo nekom drugom muzikom tj. audio fajlom sa istim imenom, sa sledećim pokretanjem SWF fajla promeniće se i muzika u pozadini prezentacije bez potrebe za ponovnim kompajliranjem u Flash-u.

Preko zvučnog kanala kontroliše se način na koji se reprodukuje zvučni zapis, sa mogućnošću kontrolisanja levog i desnog kanala stereo zapisa odvojeno. Sledeći primer učitava audio fajl *muzika.mp3* koji se reprodukuje sa 50% originalne jačine signala (engl. *volume*) i to samo na levom zvučniku (engl. *panning*):

```
//Flash ucitava eksterni fajl muzika.mp3 u promenljivu zvuk
var zvuk:Sound = new Sound(new URLRequest("muzika.mp3"));
//definise se kontroler za zvucniKanal i odredjuju se njegovi parametri
var kontroler:SoundTransform = new SoundTransform(0.5, -1);
//zvucniKanal reprodukuje zvuk sa parametrima kontrolera
var zvucniKanal:SoundChannel = zvuk.play(0, 1, kontroler);
```

4. Uspešnost Flash prezentacije

Pre nego što započne sa realizacijom neke Web prezentacije, dizajner treba imati u vidu određene faktore koji mogu uticati na prijem njegove prezentacije kod ciljne grupe korisnika. Drugim rečima, uspešna prezentacija je ona koja neće posetioca ostaviti ravnodušnim na vizuelni sadržaj, ili pak isfrustriranim zbog loše funkcionalnosti i/ili predugog čekanja da se prezentacija učita. Ne postoji recept za realizaciju uspešne Flash prezentacije, ali svakako bi bilo dobro, između ostalog, obratiti pažnju i na sledeće:

- Da prezentacija bude vizuelno dopadljiva ciljnoj grupi.
- Da sadržaj prezentacije uvek bude u prvom planu tj. da prezentacija prati sadržaj, a ne obratno. Ovo se odnosi na prezentacije koje sadrže dosta teksta.
- Da sadrži interaktivni deo koji mora biti u funkciji prezentacije, ali koji bi trebalo i da zadrži posetioce što duže na stranici (neka vrsta mini igre, ili dinamične animacije elemenata, opet u skladu sa sadržajem te stranice).
- Da omogući posetiocu da brzo shvati kako da se kreće kroz stranice prezentacije, tj. da može brzo sa trenutne stranice da se prebaci na sledeću. Idealno bi bilo da se glavni meni ni u jednom trenutku ne izgubi, i da vreme prelaska na sledeću stranicu traje što kraće. Čest je slučaj kod prezentacija koje imaju ugnježdene stranice da dizajner ne obrati pažnju na funkcionalnost, pa korisnik uglavnom mora da se vrati na početnu stranu prezentacije kako bi pogledao sledeću, ili da iznova gleda dugačku animaciju kada klikne na neko dugme.
- Da muzikom, zvukovima, animacijom ili grafikom stvori određenu atmosferu
- Da ne nameće posetiocu nešto što mu se možda neće svideti. Na primer, korisnik treba da ima mogućnost da isključi muziku koja prati prezentaciju, ili da preskoči intro kada se prezentacija prvi put pokrene.

Na Web strani „Favourite Website Awards“, koja je i sama realizovana u Flash-u, može se pronaći mnoštvo prezentacija koje su ispunile neke ili sve od ovih uslova. Kriterijumi po kojima se na ovom sajtu ocenjuju prezentacije su sledeći:

1. Dizajn (40%)
2. Navigacija (25%)
3. Grafika (15%)
4. Sadržaj (15%)
5. Atmosfera (5%)

Ako vas interesuje koje su sve mogućnosti Adobe Flash-a, ili želite da vidite primere uspešno realizovanih prezentacija, predlažem da posetite neki od sledećih linkova:

www.thefwa.com / *Favourite Website Awards* / Katalog najboljih Flash prezentacija
www.a-to-s.co.uk / *Mercedes A-to-S* / Interaktivna prezentacija kompanije Mercedes
www.enigmind.com / *Enigmind* / Interaktivni katalog agencije Enigmind
soytuaire.labuat.com / *Herraiz Soto & Co.* / Interaktivni muzički video spot
www.progressivered.com/portfolio.html / *Progressive Red* / Stefan Kovač – Portfolio
www.thibaud.be / *Thibaud's Portfolio* / Interaktivni katalog radova

5. Portfolio – Katalog radova

Katalog radova je realizovan u vidu interaktivne Flash prezentacije. Cela prezentacija je osmišljena tako da njen sadržaj bude odmah i u svakom trenutku dostupan korisniku. Svaka stranica je na određeni način interaktivna, a prilikom prelaska sa jedne stranice na drugu, korisniku se prikazuju kratke među-animacije. Takođe, ostavljena je mogućnost isključivanja prateće muzike.

Sve stranice, kao i sama prezentacija, se velikim delom oslanjaju na ActionScript 3.0. Glavni interfejs je osmišljen tako da pruža mogućnost lake izmene sadržaja, ili čak ubacivanje nove stranice. Ovo je realizovano pomoću programskog kôda, tako što se svako dugme na sceni povezuje preko prethodno definisanog pokazivača sa odgovarajućom stranicom koja je smeštena u biblioteci. Drugim rečima, stranice su smeštene u biblioteci prezentacije i pojavljuju se tek kada se klikne na odgovarajuće dugme. Pokušaću da pojasnim ovo sledećim primerom, ali imajte na umu da je to samo mali deo programskog kôda koji ne bi funkcionisao bez njegovog ostatka:

```
//funkcija koja se izvršava samo jednom, i to kada korisnik klikne na dugme
function naKlik(e:MouseEvent):void {
    //sledecaStranica se povezuje sa odgovarajućom stranicom iz biblioteke
    sledecaStranica = e.currentTarget.pokazivac;
    //trenutnaStranica pokrene sopstvenu outro animaciju
    trenutnaStranica.gotoAndPlay("Outro");
    //dodaje se funkcija koja će izvršiti zamenu stranice
    this.addEventListener(Event.ENTER_FRAME, zameniStranicu);
}

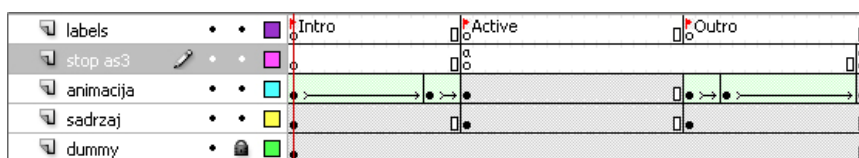
//funkcija koja menja stranicu i čeka na izvršenje sve dok se ne ispuni zadati uslov
function zameniStranicu(e:Event):void {
    //ova funkcija se izvršava tek kada trenutna stranica završi svoju animaciju
    if (trenutnaStranica.currentFrame == trenutnaStranica.totalFrames) {
        //uklanja se trenutna stranica
        this.removeChild(trenutnaStranica);
        //dodaje se stranica iz biblioteke koja odgovara pritisnutom dugmetu
        this.addChild(sledecaStranica);
        //sada ova nova stranica postaje trenutnaStranica
        trenutnaStranica = sledecaStranica;
        //trenutnaStranica pokrene svoju intro animaciju
        trenutnaStranica.gotoAndPlay("Intro");
        //funkcija se prekida
        stage.removeEventListener(Event.ENTER_FRAME, zameniStranicu);
    }
}
```

Iz ovog primera se može videti velika fleksibilnost kôda. Zbog ovakvog načina realizovanja prezentacije, moguće je dodati novu stranicu, bez izmene ili dodatnog animiranja ostatka prezentacije. Takođe, ovim je omogućena sloboda kreiranja i testiranja nove stranice u zasebnom fajlu, jer ona ne zavisi od ostatka prezentacije. Prednost rada u zasebnom fajlu je u tome što se znatno skraćuje vreme potrebno za realizaciju, jer je dizajner fokusiran samo na tu stranicu, ima veću preglednost unutar Flash-a i brži odziv programa prilikom testiranja. Po završetku, sve što je potrebno, jeste da se *Movie clip* koji sadrži celu stranicu kopira u biblioteku finalne prezentacije, a na scenu jednostavno treba dodati dugme koje će ukazivati na taj *Movie clip* preko pokazivača.

U finalnoj prezentaciji, postoje i dodatne linije kôda koje u prethodnom primeru nisu prikazane zbog bolje preglednosti, a čitav proces izgleda ovako:

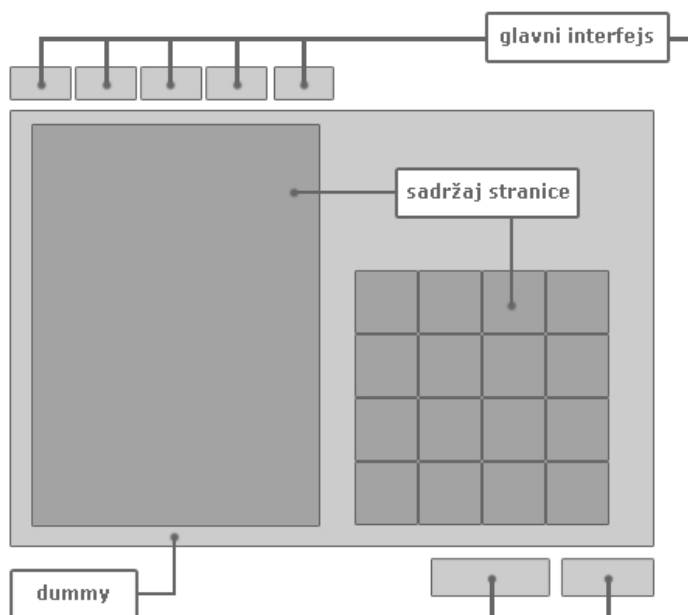
Kada korisnik klikne na neko dugme, glavni meni prestaje da bude aktivan. Trenutna stranica prikazuje animaciju koja se nalazi u njenom delu „Intro“, zatim se ona uklanja sa scene i umesto nje se postavlja stranica koja odgovara izabranom dugmetu. Nova stranica prikazuje sopstveni „Intro“, a animacija se zaustavlja u trenutku kada dodje do dela „Active“. Tada glavni meni ponovo postaje aktivan.

Iz svega navedenog, jasno je da je svaka stranica u stvari *Movie clip*, koji u sebi sadrži tri dela: „Intro“, „Active“ i „Outro“ (slika 10). Ova tri dela su realizovana kao delovi vremenske linije unutar stranice (*engl. labels*). Prednost ovog metoda je u tome što glavni ActionScript kôd nije vezan za frejmove unutar stranice, pa se animacije mogu naknadno menjati, kao i dužina trajanja animacije.



Slika 10 – Delovi nove stranice i njen sadržaj

Sloj *dummy*, koji se nalazi na samom dnu, sadrži transparentni (providni) pravougaonik, čija je jedina uloga da vizuelno olakša dizajneru pozicioniranje sadržaja stranice prilikom njene izrade u zasebnom fajlu. Dimenzije pravougaonika odgovaraju dimenzijama prostora koji je rezervisan za tu stranicu u završnoj prezentaciji (slika 11). Na taj način dizajner može lako da rasporedi sadržaj stranice dok radi u zasebnom fajlu i da bude siguran da neće doći do preklapanja njenog sadržaja sa ostatkom glavnog interfejsa.



Slika 11 – Pravougaonik *dummy* određuje dimenzije stranice

Prezentacija je realizovana tako da koristi stranice koje su smeštene u njenoj biblioteci, pa glavna vremenska linija sadrži samo dva frejma: na prvom je početni ekran, a drugi je rezervisan za samu prezentaciju.

Prvi frejm sadrži grafiku koja se odmah prikazuje posetiocu i ima funkciju da obavesti korisnika o vremenu koje je potrebno da se prezentacija učita u vidu *preloader*-a. Ovaj frejm se prikazuje samo ako se prezentaciji pristupa putem Interneta. U slučaju da se prezentacija pokreće sa CD-ROM uređaja, korisnik neće videti ovu grafiku već će odmah videti sadržaj prezentacije.

Drugi frejm sadrži glavni meni, dugmad za kontrolisanje muzike u pozadini i za izlazak iz kataloga. Ovde se izvršava ActionScript 3.0 kôd koji kontroliše ove elemente i dodaje stranice iz biblioteke simbola. Čitava prezentacija se izvršava samo na ovom frejmu, pa je glavni interfejs uvek prikazan i aktivan. Ovim je korisniku obezbeđeno lako i neometano kretanje kroz stranice prezentacije.

Prezentacija sadrži pet stranica:

O meni – informacije o autoru,

Galerija – katalog radova iz oblasti vektorske i rasterske grafike,

Video – katalog radova iz oblasti video animacije i compositing-a,

Usluge – tri oblasti dizajna kojima se profesionalno bavim, i

Kontakt – formular putem kog korisnik može stupiti u kontakt sa mnom, ili predložiti saradnju na izradi nekog projekta.

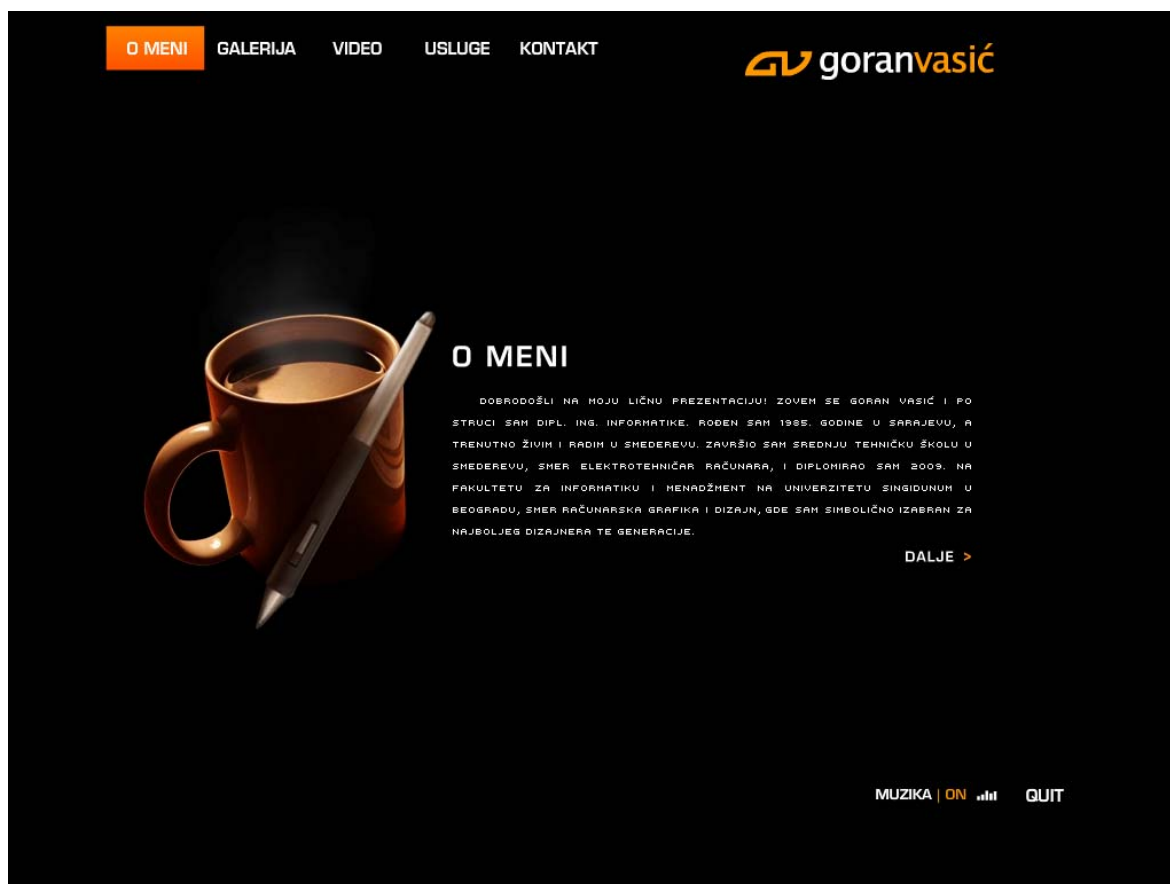
Svaka stranica u delu „Active“ sadrži ActionScript 3.0 kôd, koji animira njen sadržaj i omogućuje interaktivnost. Ovaj kôd se odnosi isključivo na tu stranicu, i ne utiče na funkcionalnost ostatka prezentacije. Interaktivnost je u ovom slučaju upotrebljena u funkciji što lakšeg pregledanja sadržaja prezentacije, pa je pažnja posetioca usmerena pre svega na izložene radove.

Stranice su testirane i optimizovane i za ekstremne situacije, pa je otklonjena mogućnost da korisnik iz nekih nepredviđenih razloga mora da napusti prezentaciju ili da mu se onemogući pregledanje kataloga. Jedna od tih situacija je i slučaj sa video player-om, koji preuzima čitav ekran da bi prikazao video fajl, pa kada korisnik pokuša da se vrati na normalnu veličinu ekrana, prezentacija napušta *fullscreen* mod i prikazuje se radna površina računara.

„Portfolio – Katalog radova“ je CD-ROM prezentacija, ali je ostavljena i mogućnost postavljanja na Internet. Prezentaciji se pristupa pokretanjem izvršnog fajla sa CD-a.

5.1 Početna stranica – O meni

U gornjem levom uglu nalazi se glavni meni, koji vodi ka ostalim stranicama kataloga (slika 12). Ovaj meni je u svakom trenutku dostupan i pruža mogućnost da korisnik pogleda bilo koju stranicu bez nepotrebnog vraćanja na početnu stranicu. Takođe, kada je određena stranica aktivna, odgovarajuće dugme u meniju je obeleženo i isključeno. Svaki put kada korisnik klikne mišem na neku stranicu, trenutna stranica nestaje u vidu animacije, a izabrana stranica se odmah nakon toga formira, takođe kroz animaciju. Ovo vreme traje veoma kratko, pa korisnik ne mora bez potrebe da čeka, a naglašava se osećaj interaktivnosti.



Slika 12 – Početna stranica „O meni“

Na sredini ekrana, sa leve strane nalazi se animirana slika šolje sa tek pripremljenom kafom, na koju je naslonjena Wacom olovka. Ovo simbolično prikazuje način na koji radim, tačnije, poručuje da sam freelancer, profesionalac koji svoje vreme maksimalno usmerava na trenutni projekat. Sa desne strane, prikazan je tekst sa informacijama o meni. Tekst je interaktivan, i klikom na dugme „dalje“ (ili „nazad“) korisnik može da pregleda tekst uz propratnu animaciju sa odgovarajućim zvukovima.

5.2 Druga stranica – Galerija

Stranica „Galerija“ predstavlja interaktivni katalog mojih radova (slika 13). Osmišljena je kao intuitivan interfejs, koji ni u jednom trenutku ne stvara konfuziju kod korisnika. Sa desne strane se nalazi matrica od 16 sličica, koje reprezentuju radove. Sa leve strane, nalazi se veliki prozor u kom je prikazana prva slika.

Klikom miša na jednu od petnaest preostalih sličica, izabrana sličica postaje neaktivna a prethodna ponovo postaje interaktivna. U tom trenutku, u velikom prozoru sa leve strane se menja trenutna i prikazuje odgovarajuća slika u punoj veličini. Ako korisnik sačeka nekoliko trenutaka, u velikom prozoru će početi da se prikazuju uveličani detalji te slike, koje korisnik inače ne bi mogao da vidi golim okom.



Slika 13 – Katalog vektorskih i rasterskih radova, stranica „Galerija“

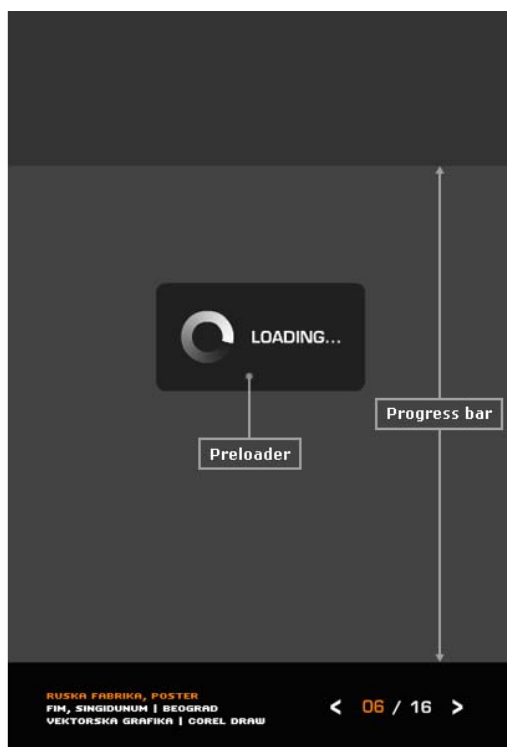
Svaki deo ovog interfejsa je praćen odgovarajućim zvukovima ili animacijom, pa korisnik lako može da shvati na koji način interfejs funkcioniše.

Ako korisnik mišem pređe preko velike slike, u dnu prozora se pojavljuju informacije o toj slici (slika 14). Pored informacija, ovde korisnik dobija mogućnost i da se kreće kroz ceo katalog pomoću opcionog interfejsa, bez potrebe za prelaskom miša sa velikog prozora na sličice sa desne strane. Ako korisnik ipak poželi da skloni miš sa velikog prozora, informacije i opcioni interfejs nestaju.



Slika 14 – Informacije o slici i opcioni interfejs

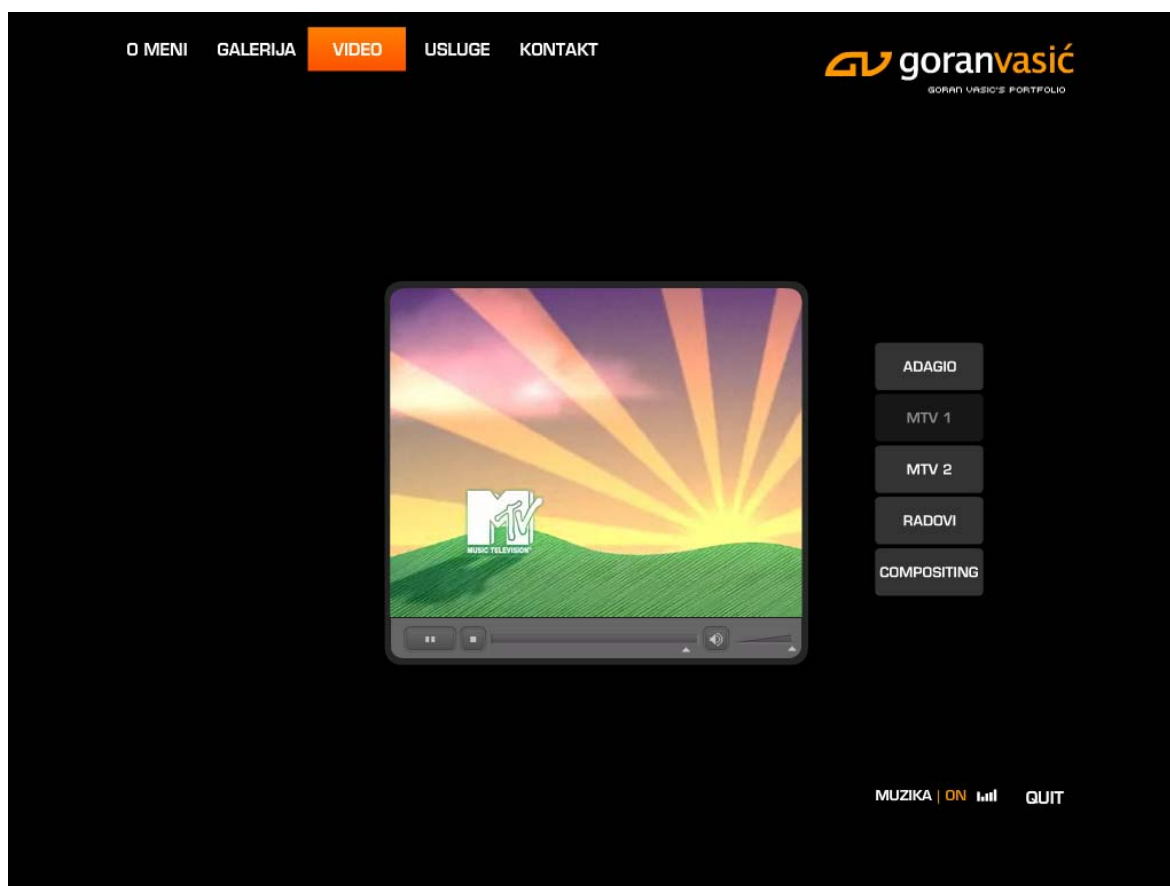
Budući da je ceo katalog napisan u programskom jeziku ActionScript 3.0, animacije su dinamične, a ako bi se prezentacija postavila na Internet, svaki put kada bi korisnik poželeo da pogleda neki drugi rad, pojavio bi se *preloader* u pratnji *progress bar*-a, koji bi ispunjavao veliki prozor po y-osi i na taj način obavestio korisnika koliko procenata slike je učitano (slika 15). Korisniku je omogućeno da izbegne nepotrebno čekanje, jer može lako da prekine učitavanje ako ono dugo traje i da izabere bilo koji drugi rad iz kataloga.



Slika 15 – Preloader i progress bar

5.3 Treća stranica – Video

Stranica „Video“ sadrži interaktivni video player koji se formira prilikom ulaska na stranicu i koji prikazuje moje radove iz oblasti animacije, 3D grafike i video montaže (slika 16). Interfejs player-a je funkcionalan i jednostavan. Pre svega, sadrži dva dugmeta koji kontrolišu reprodukciju video snimka – „pusti“ i „stopiraj“ tj. *play* i *stop*, pri čemu dugme *play* postaje „pauza“ odnosno *pause* kada film počne da se prikazuje. Desno od ova dva dugmeta je traka čija dužina predstavlja vreme trajanja filma, a ispod nje se nalazi klizač koji se pomera od leve ka desnoj strani trake tokom prikazivanja filma. Povlačenjem klizača, korisniku je pružena mogućnost premotavanja ili proizvoljnog preskakanja određenih delova filma. Desno od trake se nalaze još dva elementa interfejsa – prekidač za isključivanje zvuka i klizač za regulisanje jačine zvuka.



Slika 16 – Interaktivni video player, stranica „Video“

Kako bi video player bio u potpunosti funkcionalan, na ovoj stranici se nalazi i pomoćni meni, u kom svako dugme služi za puštanje jednog od pet filmova u player-u.

Ova stranica je osmišljena sa ciljem da korisniku obezbedi razgledanje video radova bez ikakvog ometanja. Prilikom realizacije ove stranice, posebnu pažnju sam posvetio testiranju i optimizaciji kôda, kako bih eliminisao sve greške i nepredviđene situacije koje se mogu javiti kao posledica aktivnosti korisnika prilikom razgledanja kataloga.

Mogući problemi su nastajali zbog same prirode *FLVPlayback* komponente koja prikazuje film. Kada korisnik klikne na stranicu „Video“, umesto da se prikaže njena uvodna animacija (deo „Intro“ ovog Movie clip-a), iznenada bi počeo da se prikazuje prvi video fajl koji se nalazi na ovoj stranici i to u punoj veličini ekrana tj. *FLVPlayback* komponenta bi naprasno preuzela čitav ekran prezentacije za potrebe puštanja filma (engl. *fullscreen mode*). Ako bi nakon toga korisnik poželeo da napusti fullscreen mod, tada bi i čitava prezentacija izašla iz fullscreen moda. Drugim rečima, korisniku bi se prikazala radna površina Windows-a. Ovaj problem je moguće rešiti na dva načina. Kod prvog – potrebno je ubaciti ActionScript 3.0 kôd u frejm prezentacije gde se prvi put pojavljuje *FLVPlayback* komponenta. Ovaj kôd naređuje Flash-u da ne dozvoli komponenti da preotme fullscreen mod koji je dodeljen prezentaciji. Takođe video player od trenutka pojavljivanja na pozornici mora biti sve vreme prisutan na vremenskoj liniji, do poslednjeg frejma. Ako dizajner ne želi da player bude prisutan na pozornici sve vreme, onda se, u trenucima kada on nije potreban, može stopirati tj. premotati film i ukloniti sa pozornice određivanjem novih koordianta za, recimo, y-osu. Sledi primer za kôd upotrebljen na instanci *FLVPlayback* komponente sa imenom *videoPlayer* koja se pojavljuje na početnom frejmu prezentacije:

```
//ova linija sprecava instancu videoPlayer da preuzme fullscreen mod
videoPlayer.fullScreenTakeOver = false;
//fullscreen mod se dodeljuje prezentaciji
fscommand("fullscreen", "true");
fscommand("allowscale", "false");
```

Drugi način se nadovezuje na prvi – Nakon ubacivanja kôda iz prethodnog primera u prvi frejm, moguće je video player i potpuno ukloniti sa vremenske linije u određenom delu prezentacije. U tom slučaju, potrebno je dodati *EventListener* koji će neprekidno izvršavati funkciju proveravanja svakog objekta koji se učitava sve dok korisnik ne napusti prezentaciju. Ako je učitani objekat *FLVPlayback* komponenta, tada će se kroz izvršavanje funkcije sprečiti preuzimanje fullscreen moda:

```
//ova linija dodaje EventListener
stage.addEventListener(Event.ADDED, proveriObjekat, true);

//funkcija koja se neprekidno izvrsava tokom trajanja prezentacije
function proveriObjekat(e:Event):void {
    //ispituje se da li je ucitani objekat FLVPlayback komponenta
    if (e.target is FLVPlayback) {
        //ako jeste, Flash joj nece dozvoliti da preuzme fullscreen mod
        e.target.fullScreenTakeOver = false;
    }
}
```

Međutim, kada se prezentacija distribuira putem CD-ROM uređaja, što je i ovde slučaj, ostaje problem ubacivanja dugmeta za fullscreen mod na interfejs video player-a, jer bi sada ono vraćalo prezentaciju na radnu površinu Windows-a. Potrebno je izbeći i

ovaj problem, a tehnika koju treba primeniti u tom slučaju podrazumeva prethodno definisanje dimenzija filma za potrebe prikazivanja na celom ekranu. Umesto da *FLVPlayback* preuzme fullscreen mod od prezentacije, film se jednostavno uveličava kako bi bio prikazan na čitavom ekranu. Ovim „varanjem“ se zaobilazi cela zbrka oko fullscreen moda, i eliminišu su greške koje mogu narušiti interaktivnost čitave CD-ROM prezentacije.

U ovu stranicu sam implementirao još jednu vrstu među-animacija koje u potpunosti pokreće ActionScript 3.0 i koje dodatno pojačavaju osećaj interaktivnosti celog interfejsa. Svaki put kada korisnik promeni film koji trenutno gleda, ekran video player-a se prekriva, a čitav njegov interfejs se zatim širi ili skuplja (slika 17), u zavisnosti od dimenzija filma koji treba da se prikaže. Pošto se filmovi učitavaju sa CD-ROM uređaja, tek nakon ubacivanja u prezentaciju se film ponovo otkriva i pušta. Ovo učitavanje, naravno, traje delić sekunde, ali sam ostavio otvorenu mogućnost za postavljanje prezentacije na Internet.

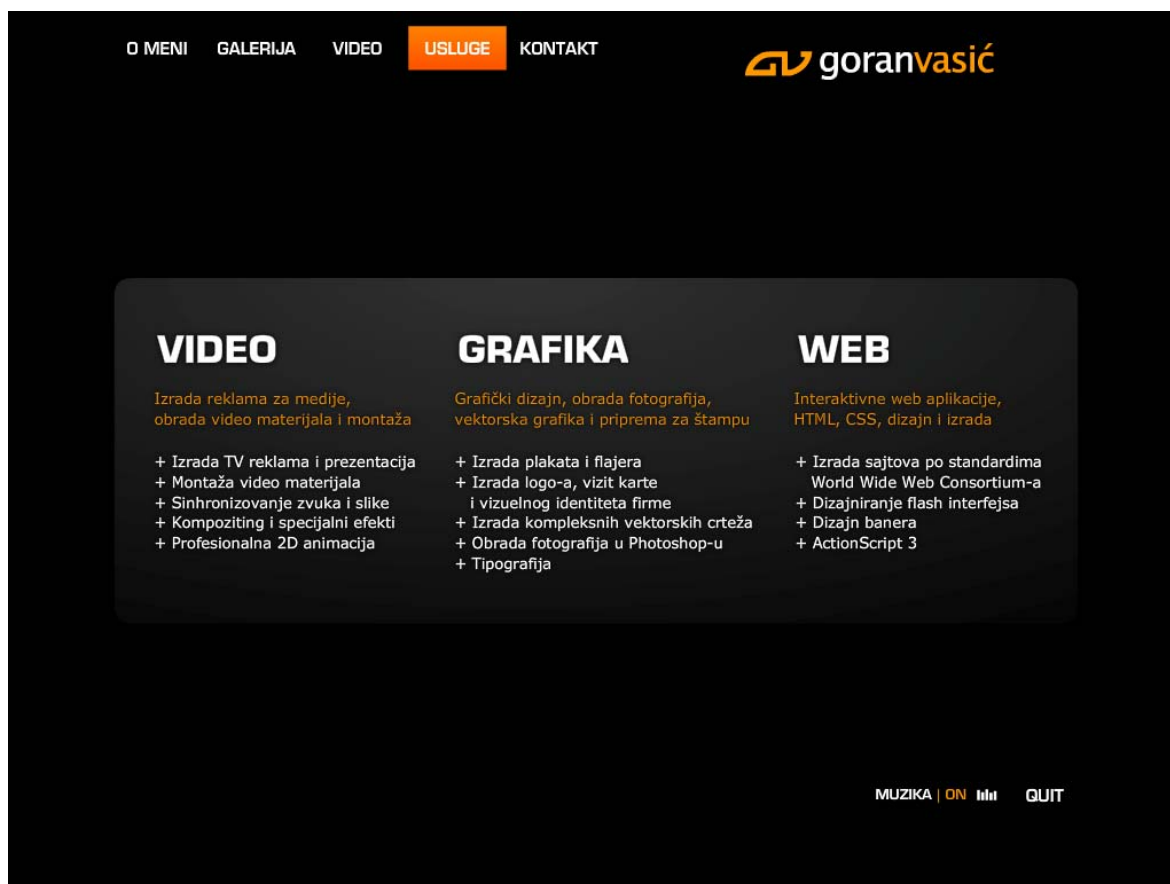


Slika 17 – Prozor video player-a može menjati dimenzije

Po završetku gledanja filmova, kada korisnik poželi da pređe sa stranice „Video“ na neku sledeću, prozor video player-a se ponovo kroz animaciju vraća na svoje početne dimenzije, a zatim, praćeno odgovarajućim zvučnim efektima, nestaje.

5.4 Četvrta stranica – Usluge

Stranica „Usluge“ sadrži interaktivni flajer na kom se mogu pročitati informacije vezane za posao kojim se bavim kao freelancer. Ovde su navedene tri oblasti dizajna: „Video“, „Grafika“ i „Web“, kao i vrste usluga koje spadaju u te tri kategorije (slika 18).



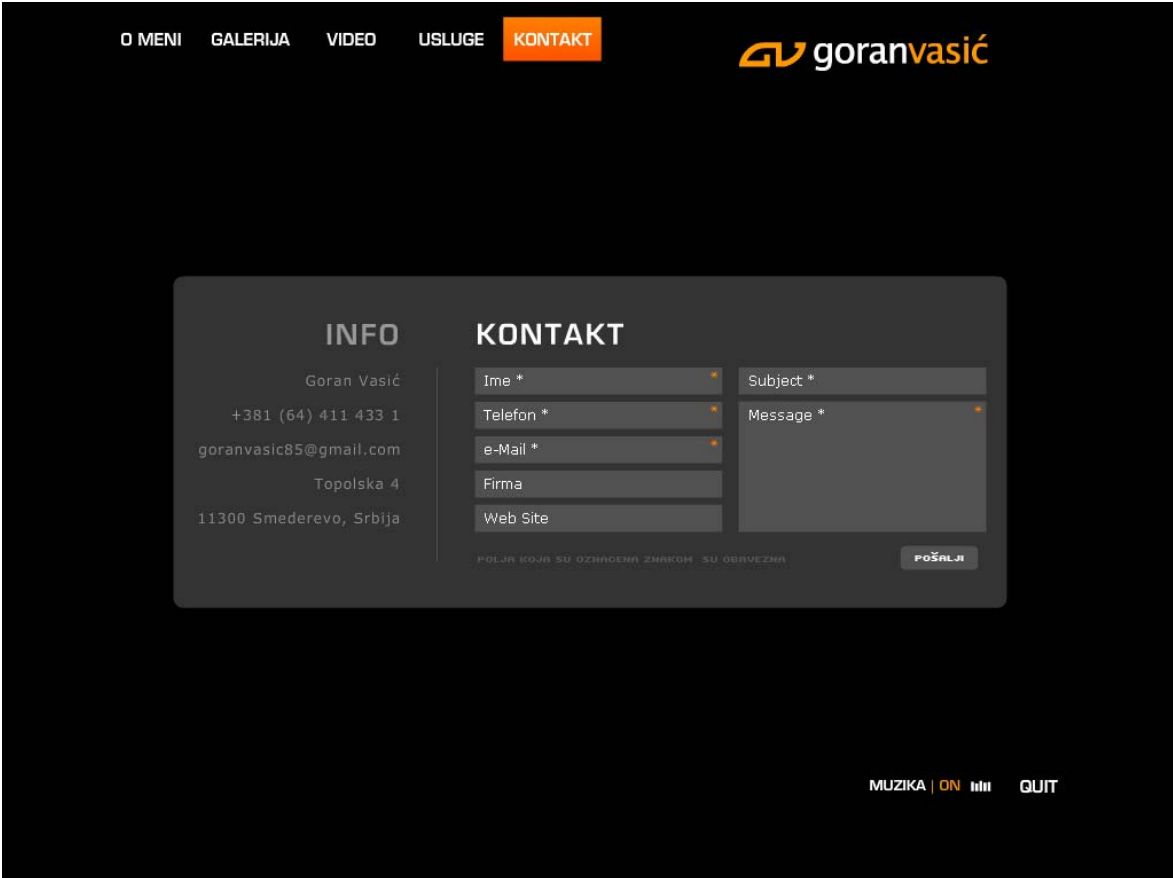
Slika 18 – Oblasti dizajna kojima se profesionalno bavim, stranica „Usluge“

Ova stranica ne poseduje interaktivnost u meri koju pružaju ostale stranice, već ima više informativnu funkciju. Jedina interaktivnost je osvetljavanje određene kategorije kada se pređe kursorom miša preko nje.

5.5 Peta stranica – Kontakt

Stranica „Kontakt“ je ujedno i poslednja stranica prezentacije. Nakon razgledanja sadržaja kataloga, tj. mojih radova i usluga, korisniku se ovde pruža mogućnost da stupi u kontakt sa mnom putem formulara koji poruku šalje na moju e-Mail adresu (slika 19).

Ova stranica se, kao i sve ostale na prezentaciji, svaki put prikazuje kroz animaciju koju prate odgovarajući zvučni efekti. Animacija je osmišljena tako da formira interaktivnu kovertu koja je podeljena na dva dela: levi deo „Info“ u kom se nalaze moji lični podaci (broj mobilnog telefona, e-Mail adresa, kao i moja adresa stanovanja), dok je na desnoj strani interaktivni formular „Kontakt“ u koji korisnik unosi svoje podatke.



The screenshot displays a website interface with a dark background. At the top, a navigation bar contains links: O MENI, GALERIJA, VIDEO, USLUGE, and KONTAKT (highlighted in orange). The logo 'goranvasić' is positioned on the right. The main content area is divided into two sections: 'INFO' and 'KONTAKT'. The 'INFO' section lists contact details for Goran Vasić, including a phone number (+381 (64) 411 433 1), email (goranvasic85@gmail.com), and address (Topolska 4, 11300 Smederevo, Srbija). The 'KONTAKT' section features a form with fields for Name, Telephone, e-Mail, Company, Web Site, Subject, and Message. A 'POŠALJI' button is located at the bottom right of the form. A small note at the bottom of the form states: 'POLJA KOJA SU OZNAČENA ZNAKOM * SU OBRUČENA'. At the bottom right of the page, there is a 'MUZIKA' toggle set to 'ON' and a 'QUIT' button.

Slika 19 – Formular za unos podataka, stranica „Kontakt“

Prelaskom kursora miša sa dela „Kontakt“ na deo „Info“, formular sa desne strane postaje u određenoj meri zatamnjen dok moji lični podaci dolaze do izražaja.

Formular se skoro u potpunosti oslanja na ActionScript 3.0 kôd, a za potrebe slanja poruke putem Internet-a koristi se eksterni PHP fajl koji po pritisku na dugme „Pošalji“ prikuplja unešene podatke iz polja formulara i šalje ih na moju e-Mail adresu. U zavisnosti od toga da li je proces slanja poruke obavljen uspešno ili ne, ispod formulara se ispisuje odgovarajuća poruka belim, odnosno crvenim slovima kako bi korisnik znao da li je potrebno ponovo pokušati sa slanjem poruke.

Formular se sastoji iz sedam polja za unos teksta: *Ime*, *Telefon*, *e-Mail*, *Firma*, *Web Site*, *Subject* i *Message*. Svako polje je dinamički kreirano pomoću ActionScript-a i dozvoljava unošenje određenog broja karaktera (npr. polje *Ime* dozvoljava unos 25 karaktera, dok polje *Message* dozvoljava maksimalno 500 karaktera). Za realizaciju svih polja koristi se samo jedna funkcija koja je napisana tako da svakom polju odmah omogući podešavanje parametara prilikom kreiranja.

Na sledećem primeru, prikazaću jedan deo ove funkcije (cela funkcija poseduje više parametara koje je moguće podešavati) i način na koji se njenom upotrebom kreira formatirano polje:

```
//funkcija kreirajPolje i parametri koje cemo kasnije podesavati prilikom pozivanja
function kreirajPolje(imePolja:String,pozX:Number,pozY:Number,duzina:int):void {
    //kreira se prazno tekstualno polje novoPolje
    var novoPolje:TextField = new TextField();
    //novoPolje je polje za unosenje teksta
    novoPolje.type = TextFieldType.INPUT;
    //promenljiva format ce sadrzati sve informacije za formatiranje teksta
    var format:TextFormat = new TextFormat();

    //formatiranje teksta
    format.font = "Verdana";
    format.color = 0xFFFFFF;
    format.size = 12;
    format.leftMargin = 6;
    //novoPolje dobija formatirani font na osnovu sadrzaja promenljive format
    novoPolje.defaultTextFormat = format;

    //formatiranje polja
    novoPolje.width = 216;
    novoPolje.height = 23.6;
    novoPolje.border = false;
    novoPolje.wordWrap = true;
    //dozvoljeni broj karaktera u polju se odredjuje pomocu parametra duzina
    novoPolje.maxChars = duzina;
    //ime instance odgovara parametru imePolja
    novoPolje.name = imePolja;
    //u novoPolje se unosi tekst koji je identican imenu polja
    novoPolje.text = imePolja;
    //odredjivanje pozicije polja na pozornici preko parametara pozX i pozY
    novoPolje.x = pozX;
    novoPolje.y = pozY;

    //polje sa tekстом se kreira na pozornici
    addChild(novoPolje);
}

//pozivanjem funkcije kreirajPolje sa podesavanjem parametara kreira se prvo polje
kreirajPolje("Prvo tekstualno polje",50,50,25);

//kreiranje drugog tekstualnog polja
kreirajPolje("Drugo tekstualno polje",100,100,25);
```

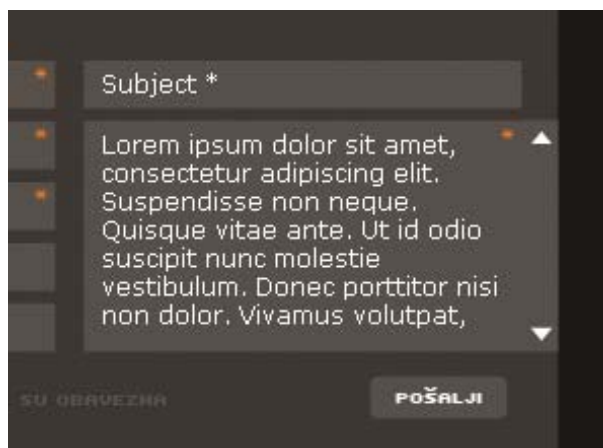
Prethodni primer prikazuje sa kakvom lakoćom se mogu napraviti desetine različitih tekstualnih polja upotrebom samo po jedne linije kôda sa odgovarajućim parametrima za svako od njih. Izmena nekog polja se u tom slučaju vrši upravo u frejmu gde se izvršava ActionScript 3.0, pa ne dolazi do nepotrebnog gubljenja vremena kako bi se to polje pronašlo u „džungli“ radnog fajla.

Naravno, ovako kreiran tekst se može i animirati. U tome je još jedna prednost ovakvog načina rada, jer biblioteka simbola ostaje kompaktna i pregledna. Kod klasičnog animiranja upotrebom vremenske linije, biblioteka simbola bi sadržala gomilu Movie clip simbola za svako polje pojedinačno, dok se upotrebom ActionScript 3.0 kôda ovo u potpunosti zaobilazi.

Da bih prikazao na koji način se može pristupati poljima za potrebe animacije, nakon kreiranja, prvo tekstualno polje iz prethodnog primera možemo pomeriti za dodatnih 200 piksela u desno dodavanjem sledeće linije na kraj kôda:

```
(getChildByName("Prvo tekstualno polje") as TextField).x += 200;
```

Formular na stranici „Kontakt“ pruža korisniku još jednu vrstu interaktivnosti. Naime, kada korisnik u polje *Message* unese više od sedam linija teksta (toliko linija se maksimalno može prikazati u ovom polju), tada se desno pojavljuje traka pomoću koje se korisnik može kretati kroz tekst gore-dole, pa po završetku unosa može pregledati ono što je napisao (slika 20).



Slika 20 – Traka za kretanje kroz tekst

6. Zaključak

Iz svega navedenog, jasno je da Adobe Flash predstavlja idealno rešenje za realizaciju Web i CD-ROM prezentacija. Ovaj projekat prikazuje samo neke od mogućnosti Flash-a, ali su u njegovoj realizaciji primenjene napredne tehnike koje se velikim delom zasnivaju na znanju stečenom tokom vežbi iz predmeta „3D grafika i animacija“ smeru Računarska grafika i dizajn. Ove napredne tehnike bi trebalo da podstaknu buduće Flash *devigner*-e (*developer* + *designer* = *devigner*) da dalje unapređuju svoje veštine i proširuju svoje znanje iz ove oblasti dizajna kako bi njihove buduće prezentacije bile što kvalitetnije. Sa velikom zajednicom korisnika Flash-a (koja se neprestano povećava), na Internetu je više nego lako pronaći dodatna uputstva za njegovu upotrebu. Svako ko poželi da na neki način iskaže svoju kreativnost (u vidu animiranog filma, interaktivne prezentacije, video igre pa čak i softverske aplikacije) može lako savladati tehnike Flash-a koje su ovde prikazane. Bilo da ste dizajner ili programer i bez obzira na nivo vaše veštine, Flash će vas za kratko vreme nagraditi pružanjem neograničenih mogućnosti za ispoljavanje vaše kreativnosti.

7. Literatura

- [1] **D. Cvetković, Z. Kostić**, *3D grafika i animacija*, Univerzitet Singidunum, FIM, Beograd, 2007.
- [2] **D. Cvetković, Z. Kostić**, *Vektorska i rasterska grafika – Praktikum*, Univerzitet Singidunum, FIM, Beograd, 2007.
- [3] **V. Mišković, R. Popović, Z. Kostić, N. Minić**, *Materijal za vežbe i predavanja iz predmeta Multimedija*, Univerzitet Singidunum, FIM, Beograd, 2007 – 2009
- [4] **J. Krstić**, *Diplomski rad – Prezentacija kozmetičke kuće*, Univerzitet Singidunum, FPI, Subotica, 2008.
- [5] **A. Tasić**, *Diplomski rad – Galerija radova studenata*, Univerzitet Singidunum, FIM, Beograd, 2008.
- [6] **Ellen Finkelstein, Gurdy Leete**, *Flash CS3 For Dummies*, Wiley Publishing, Inc., 2007.
- [7] **Tom Green, Adam Thomas**, *Foundation Flash CS3 Video*, 2008.
- [8] **Ted Padova, Kelly L. Murrdock**, *Adobe Creative Suite 3 Bible*, Wiley Publishing, Inc., 2007.
- [9] **Russell Chun**, *Macromedia Flash MX – Napredne tehnike za Windows i Macintosh*, CET Computer Equipment & Trade, 2003.
- [10] **Wikipedia**, <http://www.wikipedia.org/>
- [11] **Garry Grossman, Emmy Huang**, *ActionScript 3.0 overview*, http://www.adobe.com/devnet/actionscript/articles/actionscript3_overview.html
- [12] **Greg Lunn**, *Full-Screen Video in Flash CS3*, <http://monkeyflash.com/tutorials/full-screen-video>
- [13] **Tom Green, David Stiller**, *Adding Audio to Flash CS3*, <http://www.communitymx.com/content/article.cfm?cid=D2491>