

ANIMIRANI IZREZKI IN GUMBI

V tem članku si bomo pogledali dva zelo pomembna elementa v Flashu, to sta animirani izrezek in gumb. Brez njih praktično ne moremo izdelati zanimive in učinkovite animacije. Pomen animiranih izrezkov je, da najprej naredimo animacijo samih v sebi in jih šele potem uporabimo za animacijo na glavni časovni osi. Največja prednost animiranih izrezkov je večkratni učinek animacije, katerega bi na glavni časovni osi zelo težko naredili realistično. Druga velika prednost pa je modularna zgradba elementov v Flashu, ki pa jo že poznamo iz članka o uporabi knjižnice. Bralcu bolj poznani element, obravnavan v tem članku, pa je gumb. Poznamo jih praktično vsi, saj se z njimi srečujemo, če ne drugje, na spletnih straneh. Gumb v Flashu naredimo podobno kot na spletnih straneh in imajo tudi podobno vlogo.

Izdelava in uporaba animiranih izrezkov

Izdelavo in uporabo animiranih izrezkov si bomo pogledali na primeru vozečega se avta. Avto se vozi čez prizorišče, hkrati pa se mu tudi vrtijo kolesa. Če bi želeli vse to animirati na glavni časovni osi, bi imeli veliko težav prav s sinhronizacijo vseh teh animacij. Zato najprej naredimo animirane izrezke koles, jih postavimo na prizorišče, uporabimo kot dele avtomobila in vse skupaj animiramo kot vožnjo avtomobila. Postopek izdelave animiranih koles je naslednji:

✗ najprej narišemo kolo kot grafični objekt in ga pretvorimo v simbol *kolo*;

✗ isti primerek na prizorišču še enkrat pretvorimo v simbol *vrteče se*

kolo (imena so zgolj simbolična in ne vplivajo na rezultat). Tako imamo sedaj v simbolu *vrteče se kolo* ugnezen simbol *kolo*, ali drugače, simbol *kolo* je sestavni del simbola *vrteče se kolo*;

✗ v simbolu *vrteče se kolo* naredimo animacijo Motion, da se zavrti za obrat od prve do, recimo, 30. celice;

✗ vrnemo se v sceno, postavimo še en primerek simbola *vrteče se kolo* iz knjižnice na prizorišče, ga poravna-

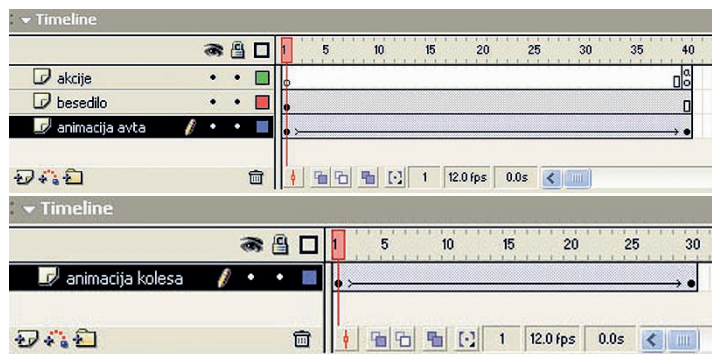
mo s prejšnjim (v isti vodoravni liniji) in narišemo obliko avta;

✗ vse skupaj označimo in ponovno pretvorimo v simbol *avto*. Pri vseh teh pretvorbah elementov v simbole nastavimo vrsto elementa Movie Clip (to smo obravnavali v članku o knjižnici). Primer naslovne vrstice tega primera prikazuje slika 1;

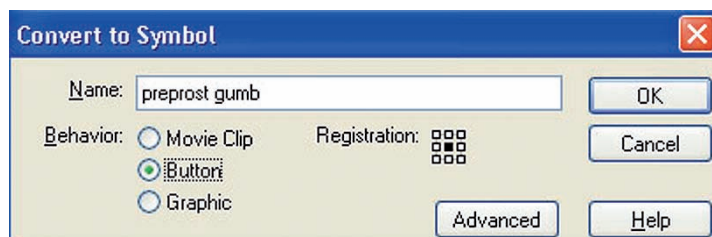
✗ ker smo zdaj na glavni sceni, lahko na časovnem traku naredimo animacijo avta, ki se premika iz levega proti desnemu delu prizorišča.



Slika 1. Prikaz ugnezenih simbolov v naslovni vrstici.



Slika 2. Animacija avta in animacija vrtečega se kolesa.



Slika 3. Izdelava gumba.



Slika 4. Pogled urejanja gumba.

Predogled animacije nam pokaže, da se avto premika z leve proti desni in se mu hkrati še vrtijo kolesa (kot smo te animirali same v sebi). Slika 2 prikazuje dve animaciji na dveh časovnih trakvih: animacijo *avta* na glavnem časovnem traku in animacijo *vrtečega se kolesa*, ki je animirano samo v sebi. Primer je prikazan na spletni strani Grafičarja.

Animirani izrezki so primerni predvsem tam, kjer želimo animacijo, ki se stalno izvaja in ni odvisna od raznih drugih prikazov, izbir in delovanja uporabnika (razni animirani logotipi, ki so stalno prikazani).

Pri animiranih izrezkih je slabost, da njihovih animacij v predogledu animacij znotraj Flasha ne moremo videti. Prednost pa je, da jih lahko poimenujemo, tako da se lahko naprej tudi sklicujemo nanje (v programski kodi actionscripta). To zadnje je bolj zahtevno, zato tega v naši seriji člankov ne bomo obravnavali.

Izdelava gumbov in nastavitve njihovih lastnosti

Za izvajanje interaktivnosti (sodelovanje uporabnika v animaciji) v Flashu so nujno potrebni element gumbi. Največkrat nas popeljejo na kako drugo vsebino, včasih pa tudi zaganjajo animacije, jih ustavljamo, potrjujemo izbire itn. Flash omogoča preprosto izdelavo precej zapletenih gumbov z uporabo različnih oblik, animacij in tudi zvoka (tega bomo obravnavali v enem prihodnjih člankov). Sama izdelava gumbov je precej preprosta, kot je tudi preprosta nastavitve njihovih lastnosti. Gumb izdelamo tako, da element na prizorišču označimo in ga pretvorimo v simbol. Gumb poimenujemo in pri vrsti simbola, v

Samo za vaše oči.

Da bi odkrili skrivnosti papirjev Hello, ne potrebujete vohunov. Rezultati naše do sedaj največje raziskave med potrošniki so vsem na ogled. Brez cenzure na www.hellopaper.com.



HELLO

The answer.

www.hellopaper.com

nasprotju z dosedanjim delom, nastavimo lastnost Button. Primer prikazuje slika 3.

V tem primeru se gumb pojavi kot simbol v knjižnici. Urejamo ga tako kot vsak simbol v knjižnici ali njegov primer na prizorišču (najpogostejše dvojni klik na primer na prizorišču). V tem primeru se nam pokaže v oknu s časovnim trakom malce drugačen pogled. Prikazuje ga slika 4.

Gumbu lahko nastavljamo štiri osnovna stanja:

✕ Up (normalno stanje) je prikaz gumba v normalnem stanju, ko kazalec miške ni nad njim in gumb ni pritisnjen;

✕ Over (stanje prehoda kazalca) je prikaz gumba, ko uporabnik nanj zapelje kazalec miške;

✕ Down (stanje pritisnjenega gumba) je podoba gumba, ko uporabnik nanj klikne z miško;

✕ Hit (območje občutljivosti gumba). Z njim pa določamo območje, kjer se bo gumb vedel kot gumb. To pride v poštev pri »votlih« gumbih in gumbih v obliki besedila.

Izdelovalcem spletnih strani so ta stanja gumbov verjetno znana, saj je v programih za izdelavo spletnih strani (npr. Dreamweaver) njihova definicija enaka. Prva tri stanja so lažje razumljiva, zato malce bolj podrobno razložimo stanje Hit. Kadar za gumb uporabimo grafično obliko, ki ima znotraj razne prazne površine (»votel« gumb), ali kadar za gumb uporabimo besedilo, bi bil ta zelo neuporaben, saj bi bil aktiven le tam, kjer je grafika narisana oziroma besedilo napisano. Takrat v stanju Hit z grafično obliko določimo, kje je gumb aktiven. Pri besedilnem gumbu ponavadi kar čez besedilo narišemo kvadrat. Barva grafike v stanju Hit ni pomembna, saj le določa področje občutljivosti in je tako le indikator področja (podobno kot pri izdelavi maske iz prejšnjega članka). Primeri pomena uporabe stanja Hit so prikazani na spletni strani Grafičarja. Slika 5 prikazuje vsa štiri stanja besedilnega gumba na omenjeni spletni strani.

Doslej smo imeli v posameznih stanjih gumba navadne grafike. Najpogostejše te grafike tudi pretvo-

rilo v simbole v knjižnici, predvsem zato, da jim lahko nastavljamo vse lastnosti, ki jih lahko spreminjamo, in še bolj pomembno, da jih lahko animiramo in tako dobimo animirane gumbe.

Animirani in nevidni gumbi

S kombinacijo doslej osvojenega znanja Flasha in malce dela lahko naredimo že zelo privlačne animirane gumbe. Tipičen primer so naslov in povezave na posamezne članke na glavni strani predstavitve Flash na spletni strani Grafičarja. Ko se z miško zapeljemo na te gumbe, se začne izvajati animacija. Trik je le v tem, da v ustrezno stanje gumba vstavimo simbol, ki se vede kot animiran izrezek, in ga predhodno ustrezno animiramo. Načelo izdelave animiranega gumba je naslednje:

✕ napisano besedilo pretvorimo v gumb;

✕ v stanju Up to besedilo pretvorimo v navaden grafični simbol;

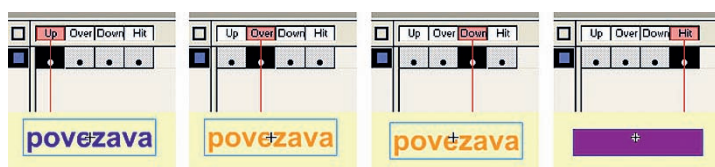
✕ v stanju Over to isto besedilo pretvorimo v simbol animiranega izreza;

✕ znotraj animiranega izreza naredimo animacijo (to je razloženo na začetku tega članka);

✕ po potrebi določimo še stanje občutljivosti.

Z uporabo animiranih gumbov lahko naredimo zanimive menije, ki so tipični navigacijski meniji na spletnih straneh. Nekaj primerov je prikazanih na spletni strani Grafičarja.

Drugi pomemben element so nevidni gumbi. Njihov pomen je, da v animaciji niso vidni, so pa kljub temu delujoči in imajo svojo funkcijo (te bomo spoznali v prihodnjem članku). Uporabljamo jih, kadar želimo na poljubnih slikah ali grafikah določiti občutljiva območja, pri čemer teh slik in grafik ne bi radi »pokvarili« s slikami gumbov. Najpogostejši primer uporabe nevidnega gumba so zemljevidi, na katerih želimo, da so občutljiva mesta, ki nas recimo ob pritisku popeljejo na kako drugo vsebino (spletna povezava). Sama izdelava je zelo preprosta:



Slika 5. Prikaz štirih stanj besedilnega gumba.



Slika 6. Prikaz nevidnih gumbov.

Nadaljevanje
na strani 26.