

SIN – APLIKACIJA ZA POUČEVANJE GASILCEV

The SIN Learning Application for Firefighters

Simon Kolmanič*, Nikola Guid**, Andrej Nerat*** UDK 614.841.3:37.091.64

| Povzetek | Abstract |
|---|--|
| <p>Poučevanje različnih postopkov je pogosto zelo težavno, še posebno, če so ti kompleksni. V takem primeru si pomagamo z zaporedjem ustreznih slik, s katerim ponazorimo postopek. Opisano nedvomno velja tudi za poučevanje postopkov ukrepanja ob požarih in drugih takih nesrečah. Pri tem se pojavi težava, saj ponavadi takega slikovnega gradiva ni, ker gre za redke dogodke. Da bi rešili ta problem, predstavljamo aplikacijo SIN, ki inštruktorjem omogoča hitro pripravo zaporedja realnih slik, ki prikazujejo postopke posredovanja v posameznih fazah.</p> | <p>The teaching of handling proceedings is often very difficult especially if these are complex. In such cases the sequence of images representing the proceeding can be of great help. This is also true for the teaching of the handling procedures in the event of fire and similar disasters. The main problem in this case is a lack of appropriate images, since these situations are relatively rare. To solve this problem, the SIN application is presented, enabling the instructor to generate the sequence of realistic images which demonstrate the handling procedures in different stages in a fast and easy way.</p> |

Uvod

Dobra izurjenost gasilskih enot je bistvena za uspešno posredovanje, ki minimizira škodo in prepreči dodatno nevarnost tako za ljudi kot imetje. Za dobro izurjenost sta potrebni redno izobraževanje in pridobivanje praktičnih veščin. Pri tem je treba omeniti, da je izobraževanje zaradi svojevrstne organiziranosti gasilske službe, ki temelji večinoma na prostovoljni podlagi, posebno. Zaradi svoje redne službe se pripadniki prostovoljnih gasilskih enot namreč ne morejo udeleževati dolgih izobraževanj. Ta so tako razbita na posamezne, razmeroma kratke tečaje, na katerih slušatelji pridobijo ustrezno znanje, ki ga poglobljajo s pomočjo fizikalnih simulatorjev. Učinkovitost izobraževanja lahko povečamo tudi z uporabo abstraktnih, to je programskih simulatorjev, s katerimi lahko uporabnika vnaprej pripravimo na nalogo, ki jo mora opraviti, s čimer lahko nalogo opravi hitreje in bolje [Tate in sod., 1997, Querrec in sod., 2003, Lucas in Thabet, 2008, Backlund in sod., 2009]. Dodatna prednost programskih simulatorjev je tudi v tem, da uporabnik ni v neposredni nevarnosti, da se poškoduje, zato lahko vadi brez nadzora inštruktorja, vajo pa lahko neomejenokrat ponovi [Lucas in sod., 2008]. Poleg tega je uporaba pro-

gramskih simulatorjev tudi cenejša od uporabe fizikalnih. Kljub temu programski simulatorji ne morejo nadomestiti ne inštruktorja ne realnega treninga. Večina teh simulatorjev namreč temelji na grafičnih pogonih, razvitih za igre, ki dajejo odlične vizualne učinke, kar lahko ustvari močan občutek realnosti [Sherman in sod., 2007]. To lahko ob ponavljanju se uporabi simulatorja povzroči nepazljivost pri realnih nalogah, saj se zmanjša dovtetnost za nevarnost. Obvladovanje nalog v simulatorju še ne pomeni, da bo tečajnik nalogo opravil tudi v realnem svetu, saj ga ne pripravi na taka osnovna dejstva, da je ogenj vroč in da se mu ne moremo brez vsakih posledic približati ali da udarec povzroči bolečino. Programski simulatorji, ki temeljijo na pogonih iger, še vedno delujejo precej kot igre, kar pomeni predvsem poudarek na prvoosebni akciji, ki je primerna za vadbo že znanih postopkov, manj primerna pa je za učenje novih.

V okviru razpisa Uprave Republike Slovenije za zaščito in reševanje ter v sodelovanju z Izobraževalnim centrom za zaščito in reševanje na lgu smo razvili aplikacijo za poučevanje gasilcev, imenovano simulator intervencij oziroma SIN. Zaradi prej opisanega smo izhajali iz aplikacij, kakršni sta CommandSim [splet 1, 2011] in RescueSim [splet 2, 2011]. V obeh primerih gre sicer za tridimenzionalni aplikaciji, ki pa nista prvoosebni. Prav tako se omejeta le na najnujnejše vizualne učinke, zato se zelo razlikujeta od običajnih računalniških iger. Aplikacija SIN združuje dobre lastnosti obeh omenjenih orodij za pomoč pri poučevanju gasilcev in temelji na zaporedju slik nesreče ter postopkov, ki vodijo v rešitev razmer. Inštruktor pri tem izhaja iz fotografije, znane slušateljem. S tem pritegne njihovo pozornost in tako poveča pedagoški učinek. Pri tem smo želeli postopek priprave scenarija čim bolj poe-

* dr., Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Smetanova 17, Maribor, simon.kolmanic@uni-mb.si

** dr., Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Smetanova 17, Maribor, nikola.guid@uni-mb.si

*** Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Smetanova 17, Maribor, andrej.nerat@uni-mb.si

nostaviti in tako inštruktorju omogočiti preprosto spreminjanje že obstoječih scenarijev in njihovo prilagajanje znanju slušateljev. Aplikacija SIN je zaradi realne slike ozadja dvodimenzionalna, čeprav pripravljena tako, da daje vtis tridimenzionalnosti. To dosežemo s pomočjo posebnih gradnikov. Za pripravo scenarija nesreče in reševalcev so na voljo številni gradniki, ki so lahko statični ali dinamični. Statični gradniki se s časom ne spreminjajo, to so na primer stavbe, avtomobili in osebe, dinamični pa se s časom spreminjajo, to pa so plameni in dim v različnih oblikah. Pri tem za povezovanje gradnikov z ozadjem uporabljamo preizkušeno tehniko zelenega ozadja (Chroma Key). Statični gradniki so slike virtualnih objektov 3D na zelenem ozadju, pripravljene z animacijskim paketom LightWave 3D, ki jim daje največjo mogočo stopnjo realnosti. V nasprotju s tem se dinamični gradniki izračunavajo sproti, če strojna oprema računalnika to omogoča, ali pa se preberejo iz vnaprej izračunanega zaporedja. To nam omogoča, da aplikacija teče tudi na računalnikih brez grafičnega pospeševalnika, na primer na starejših prenosnih računalnikih. Inštruktor lahko z miško vpliva na vse glavne attribute gradnikov, kot so njihova velikost, razteg v eni ali drugi smeri, njihova postavitve v sceni in vrstni red izrisa, in to ne glede na to, za kakšen gradnik gre. Aplikacija SIN je preprosta za uporabo, zato lahko inštruktor z malo truda in v kratkem času ustvari veliko podobnih scenarijev, na katerih lahko prikaže tudi posledice nepravilnih postopkov, ki jih drugače ne bi bilo mogoče prikazati. SIN je tako dobra alternativa aplikacijama ComandSim in RescueSim, priprava scenarijev pa je lažja ter hitrejša. Podobno kot omenjeni aplikaciji lahko tudi to uporabljamo za preverjanje pridobljenega znanja, saj za njeno uporabo ni potrebno dodatno usposabljanje.

V nadaljevanju bomo opisali arhitekturo aplikacije SIN, nato pa bomo podrobneje predstavili statične gradnike. Sledi opis dinamičnih gradnikov, osredotočili se bomo na pripravo dima, pri čemer smo uporabili nove pristope, ki omogočajo preprost nadzor nad tem pomembnim gradnikom, kar daje zelo dobre rezultate. Aplikacija se lahko trenutno uporablja za poučevanje postopkov pri požarih in nesrečah z nevarno snovjo.

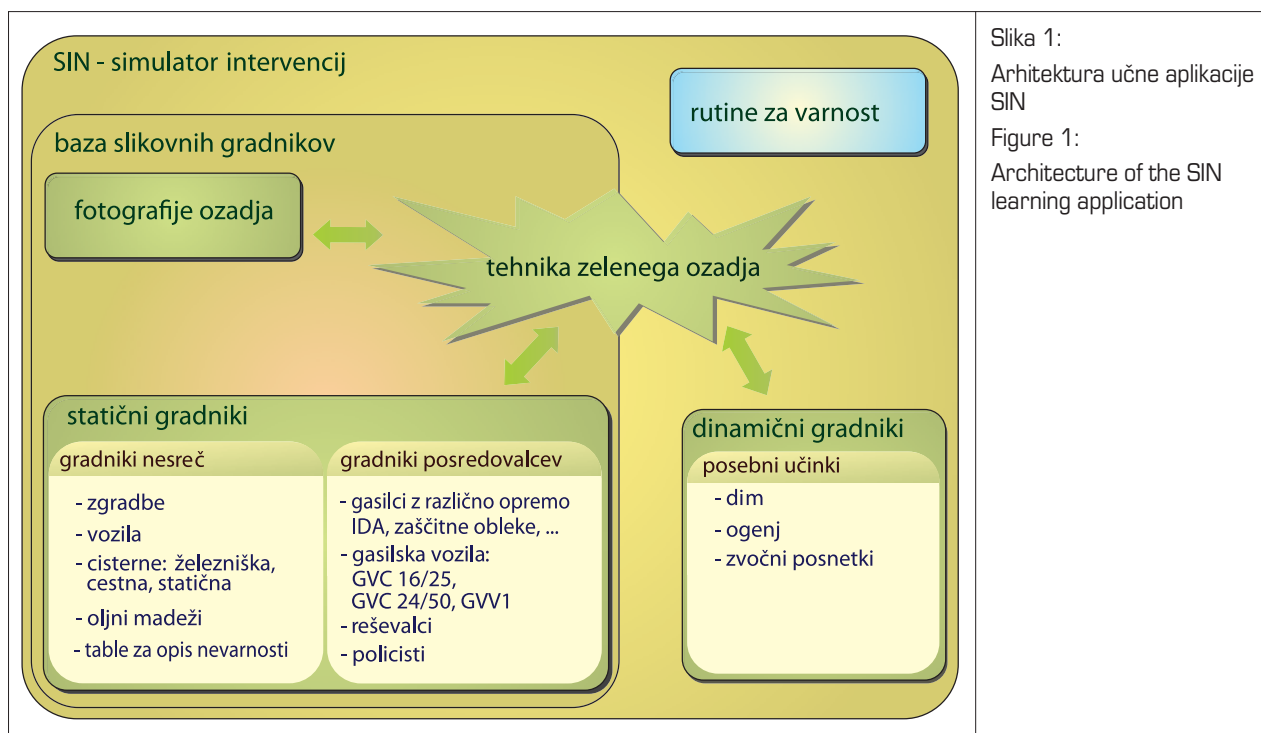
Arhitektura aplikacije SIN

Cilj aplikacije SIN je povečati pridobljeno znanje slušateljev na tečajih. Pri tem smo pregledali trenutne rešitve na trgu in kot najzanimivejša ocenili že v uvodu omenjena CommandSim in RescueSim. Skupna lastnost obeh rešitev je, da je grafika veliko manj slikovita od na primer grafike v švedskem simulatorju za poučevanje gasilcev SIDH, ki temelji na grafičnem pogonu igre Half-Life 2. V nasprotju s simulatorjem SIDH prva dva namenjata več pomena poučevanju taktičnega delovanja, pri simulatorju SIDH pa gre za učenje z igranjem igre, ki poteka v posebnem okolju in s posebno vhodno opremo, kot sta zaščitna obleka in ročnik, opremljena s senzorji. Tudi sami smo pri pripravi aplikacije SIN pred slikovitimi, toda motečimi vizualnimi učinki dali prednost okolju, ki je tečaj-

nikom blizu, in smo scenarij zasnovali okrog fotografije okolja, ki ga poznajo. Tako se poveča njihova pozornost in inštruktor ima možnost prikazati več postopkov v enakem času ali pa kompleksnejše postopke. Na sliki 1 si lahko ogledamo arhitekturo aplikacije, ki temelji na zbirki statičnih gradnikov, h katerim prištevamo tudi fotografije ozadij, ki so podlaga vsakega scenarija. Želeli smo čim večjo neodvisnost od operacijskega sistema, zato smo bazo uredili kot skupek slik, shranjenih v formatu JPG. Te v bazo združimo z opisno datoteko, ki aplikaciji pove, kje na disku naj poišče posamezne gradnike. Tako lahko aplikacija deluje tudi, če se del slik poškoduje. Prav tako lahko pozneje preprosto dodajamo nove gradnike in s tem razširjamo uporabnost orodja.

V nasprotju s statičnimi gradniki so dinamični določeni s programsko kodo, zato jih ne moremo tako preprosto dodajati, saj so le skupek funkcij, ki jih kličemo znotraj aplikacije. Program SIN je napisan v jeziku C++ in uporablja knjižnico Qt ter vmesnik OpenGL, s čimer si ponovno prizadevamo za neodvisnost od operacijskega sistema, saj lahko aplikacija z minimalnim dodatnim delom deluje v operacijskem sistemu Windows in Linux, ki je prav tako zelo razširjen. Kot smo že v uvodu omenili, temelji poučevanje s pomočjo orodja SIN na prej pripravljenih scenarijih, katerih podlaga je fotografija realnih krajev morebitne nesreče, ki jo slušatelji dobro poznajo. Scenarij pripravljamo postopoma tako, da na sliko ozadja dodajamo posamezne gradnike, ki so lahko statični in dinamični. Za združevanje gradnikov z ozadjem smo uporabili tehniko zelenega ozadja, ki jo uporabljajo v filmski industriji, igralec pa nastopa pred zelenim ali modrim ozadjem, ki ga nato nadomesti slika ali video, ki se pozneje vstavi na njegovo mesto. Tehniko že dolgo poznajo v filmski industriji, saj so jo uporabljali že pred uporabo računalnikov, danes pa je ena najpogosteje uporabljenih tehnik v filmski produkciji. Za uporabo s tehniko zelenega ozadja smo morali gradnike temu primerno ustvariti, torej smo jih morali postaviti na zeleno ozadje. To je bilo preprosto izvedljivo, saj so gradniki ustvarjeni s pomočjo animacijskega paketa, kar omogoča poljubno izbiro ozadja. Statični gradniki so tako objekti, postavljeni na zeleno ozadje, ki se ob uporabi odstrani s filtrom. Tako ostane le objekt, ki ga lahko vstavimo v katero koli ozadje. Pri tem je zelena barva ozadja takšna, da se v realnem svetu skoraj nikoli ne pojavi, saj bi to sicer lahko vodilo v ustvarjanje neželenih lukenj v objektu.

Program se bo uporabljal tudi za preverjanje pridobljenega znanja, zato ne bo le na računalniku predavatelja, temveč ga bodo imeli tudi slušatelji, s čimer se pojavijo možnosti nenadzorovanega in nedovoljenega kopiranja. Da bi to preprečili, smo aplikacijo opremili tudi z mehanizmi za varnost, ki od uporabnika, če program ni že aktiviran, zahtevajo vstopno geslo. Zaščita poteka na več ravneh, tako da brez gesla ali aktivacije program ne more delovati, tudi če nam uspe najti in kopirati vse potrebne komponente. Pri tem je treba povedati, da je aktivacija vezana na čas, tako da program preneha delovati, ko je na primer tečaj opravljen.



Slika 1:
Arhitektura učne aplikacije SIN
Figure 1:
Architecture of the SIN learning application

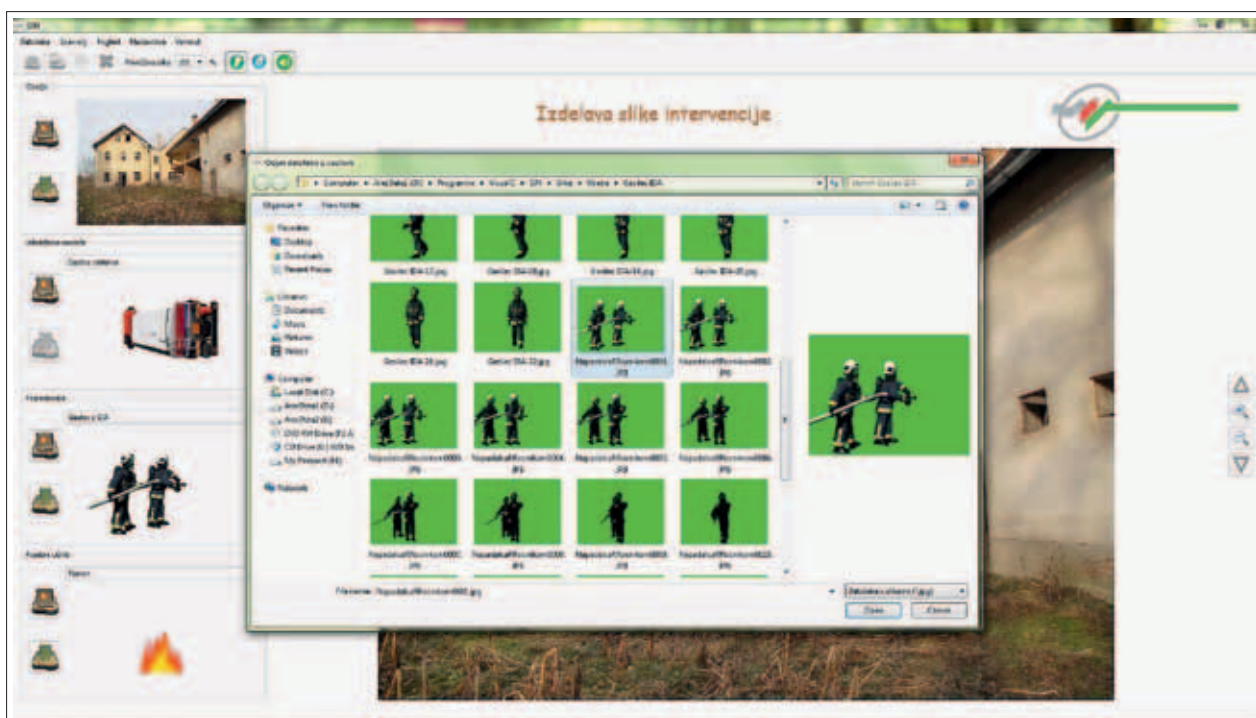
Dodaten cilj, ki smo ga upoštevali pri pripravi učnega sistema, je tudi njegova preprosta uporaba. Gradnikov je veliko, zato bi lahko postalo delo z njimi precej naporno. Da bi se temu izognili, smo gradnike razdelili v štiri logične skupine. Posebna skupina gradnikov je ozadje, saj se to skozi scenarij ne spreminja, druge objekte pa lahko poljubno spreminjamo. Te smo razdelili v skupino gradnikov nesreče, skupino gradnikov posredovalcev, tako oseb kot vozil, in skupino posebnih učinkov [glej sliko 1]. Pri tem je treba poudariti, da so le v zadnji skupini dinamični gradniki, v prvih treh skupinah pa so izključno statični. Čeprav smo se v aplikaciji SIN odpovedali vizualnim učinkom, značilnim za igre, smo vseeno želeli, da slike, ki jih ustvarimo, delujejo čim bolj realno, saj to dodatno povečuje učni učinek pri slušateljih. Za to poskrbijo prav gradniki zadnje skupine, to so posebni učinki, pri čemer gre predvsem za dim in plamene. Ti objekti so definirani z algoritmi in se delno združijo s sliko ozadja, zato jih ni treba postavljati na zeleno ozadje kot statične gradnike, temveč jih preprosto izračunavamo neposredno nad sliko ozadja. V nasprotju s statičnimi objekti je treba dinamične izračunavati v enakomernih, toda dovolj hitrih intervalih. Tako za tekočo sliko potrebujemo najmanj trideset izračunov v sekundi. Še posebno pri dimu gre za precej kompleksne algoritme, zato lahko to pomeni tudi precejšnjo obremenitev glavnega procesorja, kar je treba upoštevati pri dodajanju tovrstnih gradnikov. Če je teh namreč preveč, računalnik ne zmore izračunov opraviti dovolj hitro in slika ni več tekoča. Kar se uporabnika tiče, ta razlike med statičnimi in dinamičnimi gradniki ne čuti. Poleg slikovnih gradnikov imamo v bazi znotraj aplikacije SIN še en tip gradnika, ki se sicer ne izračunava sproti, ga pa ne moremo šteti med statične gradnike. To je zvok, ki smo ga tudi na sliki 1 vključili med dinamične gradnike. Tudi zvok inštruktor izkoristi pri razlagi, še pomembnejši pa je pri preverjanju pridobljenega

znanja, ko ga uporabimo kot faktor motenja, saj pri slušatelju zmanjšuje zbranost in dodatno otežuje reševanje dodeljene naloge.

Povedali smo, da scenarij pripravljamo postopoma. Scenarij si tako najlažje predstavljamo kot zaporedje dinamičnih slik, pri čemer je trenutna slika izhodišče za sliko, ki ji sledi. Ob pripravi nove slike se vanjo samodejno preslikajo vsi gradniki, ki so bili v njeni predhodnici. Te gradnike lahko poljubno spreminjamo, lahko pa jih tudi odstranimo. S scene lahko na primer odstranimo gasilsko vozilo, ker je šlo po vodo, ali ponesrečence, ki so bili odpeljani na varno. Vsak scenarij lahko sestavlja poljubno število slik. Slike lahko dodajamo tudi pozneje, spreminjamo lahko njihov vrstni red, lahko pa jih iz scenarija tudi odstranimo. To nam omogoča, da ustvarimo knjižnico osnovnih scenarijev, ki jih lahko inštruktor prilagaja trenutnim potrebam. Iz osnovne zbirke lahko ustvari veliko podobnih primerov, iz njih lahko pripravi tudi različice za preizkušanje znanja, pri čemer mora slušatelj nato pravilno razporejati enote, ki jih ima na voljo. Čeprav bi lahko slike med seboj povezali po zgladu ključnih slik v računalniški animaciji in bi tako dobili tekočo animacijo, se za to nismo odločili. Predvidevamo namreč, da bo ob vsaki sliki predavatelj razložil dogodek, ki je prikazan, kar lahko traja različno dolgo. Vnaprej določena animacija bi bila v tem primeru prej moteča, kot pa da bi pripomogla k izboljšanju učnega procesa. Tako je predavatelj tisti, ki odloča, kdaj bomo prešli z ene slike na drugo.

Statični gradniki

Kot smo že večkrat poudarili, so statični gradniki umetno generirani gradniki, postavljeni na zeleno ozadje, real-



Slika 2: Pogled na bazo gradnikov z napadalcema ob požaru

Figure 2: An overview of building components. The top window: firefighter thumbnail gallery.

no upodobljeni in postavljeni v najrazličnejše položaje. Ti gradniki sestavljajo posebno bazo gradnikov, ki se lahko poljubno dopolnjuje. Uporabnik lahko iz baze poljubno izbira tiste gradnike, ki jih za scenarij potrebuje. Na sliki 2 lahko vidimo primer izbire gradnika napadalcev z izolirnima dihalnima aparatoma, ki spadata v skupino gradnikov reševalcev. Ko je gradnik enkrat izbran, se zeleno ozadje odstrani, s čimer je gradnik pripravljen za dodajanje na sceno.

Znotraj aplikacije SIN delujeta dva modula. Prvi je namenjen intervencijam ob nesrečah z nevarnimi snovmi, drugi pa intervencijam ob požarih. Poleg po običajnih gradnikih se modula razlikujeta tudi po opisu nevarnosti, ki jo scenarij opisuje. Tako imamo v modulu za nesreče z nevarno snovjo dodatne gradnike, to so table za opis

nevarnosti, s katerimi morajo biti opremljena vsa vozila za prevoz nevarnih snovi. Od tega so table transportnih razredov in oznake nevarnosti tovora nespremenljive ter shranjene enako kot drugi statični gradniki, table za opis nevarnosti s Kemlerjevim številom in številom UN, ki je koda nevarne snovi, pa se ustvarjajo po potrebi, saj je mogočih veliko kombinacij. Ne glede na to, za kateri tip nesreče gre, lahko gradnike posredovalcev delimo na dve podskupini. Prva podskupina so gradniki oseb, ki jih lahko vidimo na sliki 3, druga pa so vozila in orodja, ki jih pri tem uporabljajo.

Znotraj podskupine človeških gradnikov najdemo gasilca v ognjevarni zaščitni obleki z izolirnim dihalnim aparatom ali brez njega in v zaščitni obleki za posredovanje v nesrečah z nevarno snovjo, v skupini pa najdemo tudi



Slika 3:
Gradniki človeških
posredovalcev

Figure 3:
A pool of emergency
responders



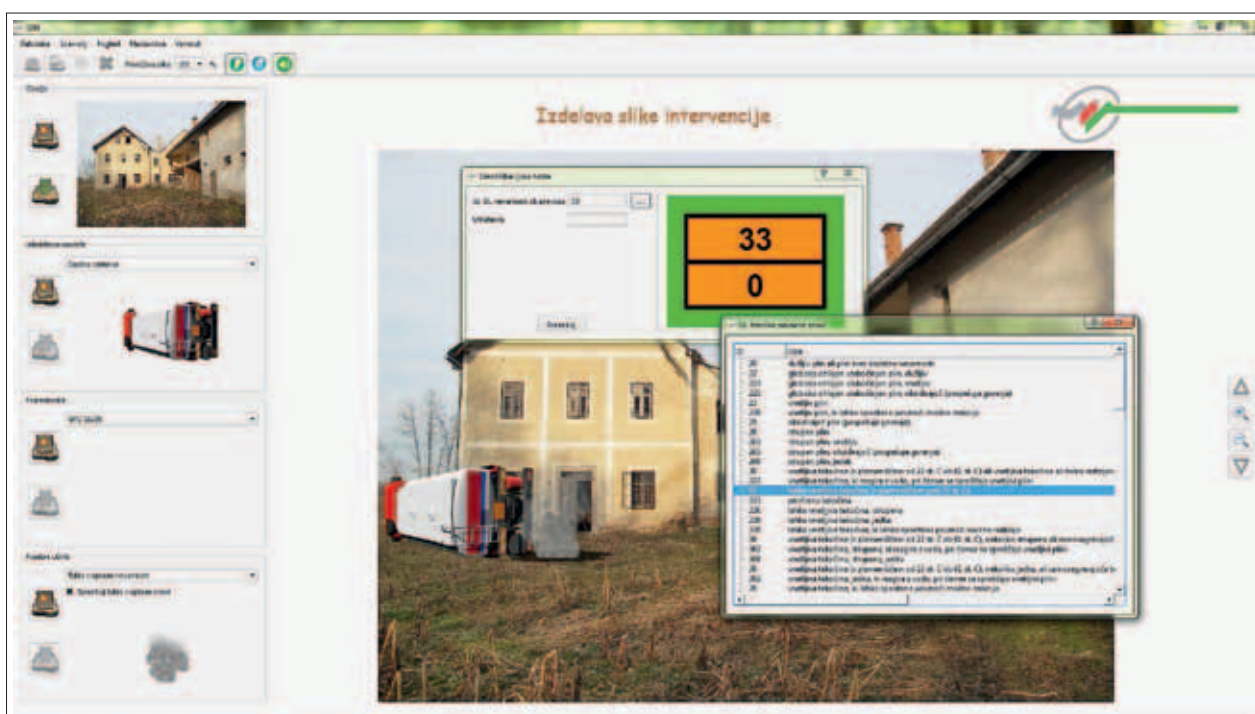
Slika 4:
Statični gradniki nesreče
Figure 4:
Static elements of an
accident

vodjo intervencije, reševalca, policista ter napadalca pri posredovanju ob požarih. Kot je razvidno s slike 3, lahko gradniki drug drugega tudi zakrivajo in tako ustvarjajo občutek treh dimenzij. Zakrivanje dosežemo z vrstnim redom izrisa, ki je določen z vrstnim redom dodajanja gradnikov, ki pa ga lahko uporabnik pozneje tudi spreminja. V skupini vozil in pripomočkov najdemo gasilska vozila s cisterno GVC 16/25 in GVC 24/50 ter osnovno gasilsko vozilo GVV1, v tej skupini pa so lahko tudi zastavica za označitev poveljniškega mesta in podobni drobni objekti. V skupini gradnikov nesreče so večinoma stavbe in vozila, ki jih lahko vidimo na sliki 4.

Kot smo že omenili, nastopa pri nesrečah z nevarno snovjo še ena skupina gradnikov, to so table za opis nevarnosti. Ne glede na to, ali gre za nespremenljive ta-

ble ali table, ki jih uporabljamo znotraj aplikacije, teh ne moremo uvrstiti v nobeno izmed opisanih skupin, zato jih znotraj aplikacije SIN uvrstimo v skupino posebnih učinkov. Če izberemo pripravo table za opis nevarnosti, se odpre posebno okno za uporabo, kar lahko vidimo na sliki 5.

Pri pripravi tabel si uporabnik lahko pomaga s kodami nevarnosti in njihovimi pomeni, kar dodatno olajša njegovo delo (glej sliko 5). Ko je tabla pripravljena, se v ničemer več ne razlikuje od drugih statičnih gradnikov. Čeprav lahko s statičnimi gradniki ustvarimo sceno, ki je bolj realna kot pri primerljivih aplikacijah, pa ti niso dovolj, da bi lahko sestavljali realno sceno tudi pri intervencijah ob požarih. V ta namen imamo na voljo še drugo skupino gradnikov, to so dinamični gradniki.



Slika 5: Priprava table za opis nevarnosti znotraj aplikacije SIN
Figure 5: Creation of the labelling summary table within the SIN application

Dinamični gradniki

Predstavitev ognja in dima znotraj grafičnih aplikacij je velik izziv za računalniško grafiko. Prav zato so se že zelo zgodaj pojavili prvi algoritmi, kako ustvariti sliko plamenov ali dima, ki bi delovala realno. Z razvojem strojne opreme, ki je postajala vedno hitrejša, so postali ti postopki uporabni tudi v računalniških igrah, ki zato delujejo tudi vedno bolj realno. Žal so ti algoritmi povezani z veliko parametri, s katerimi uravnavamo videz tako dima kot plamenov in se lahko pri vsakem dogodku zelo razlikujejo. Če so parametri nastavljeni slabo, je rezultat, ki ga dobimo, temu primerno slab. To je še posebno pomembno pri ustvarjanju dima, ki je posebno zahtevno področje, saj so algoritmi, ki ustvarjajo realno podobo dima, še vedno zelo počasni, posebno če je dim gost. V aplikaciji SIN se pojavljajo vse oblike dima, od rahlih, komaj vidnih meglic do gostega črnega dima, zato smo morali določiti algoritem, ki to ustvari. Pri tem je pomembno, da ga lahko inštruktor upravlja s čim manj parametri. V našem primeru smo vse parametre zmanjšali le na barvo in gostoto dima, ki ju nadzorujemo z drsnikom. Da smo lahko to naredili, smo morali predelati algoritem za prostorsko upodabljanje (Levoy, 1988), ki je bil izvirno razvit za vizualizacijo medicinskih podatkov, vendar se je hitro prilagodil tudi za ustvarjanje vseh vrst oblakov in dima. Žal je izvirni algoritem kljub hitremu razvoju strojne opreme še vedno veliko prepočasen, da bi ga lahko uporabili v realnem času, poleg tega naša aplikacija deluje v ravnini in ne v prostoru. Tako se izvirno dim ustvari s pomočjo implicitnih krogel v prostoru, ki imajo vnaprej definirani barvo in prosojnost. Skozi te krogle pošljemo posebne žarke, vzdolž katerih kumuliramo barvo in gostoto dima. Ta izračun je treba narediti za vsako točko v sliki, ki jo dim pokriva. Gostota v točki določa, koliko ozadja bo skozi dim vidnega, to pa se nato kombinira z barvo dima, da lahko točko slike pravilno obarvamo. Pri tem je treba poudariti, da se implicitne krogle s časom neprestano premikajo po vnaprej definiranih poteh, ki jih nekoliko spreminjamo s šumom. Pri tem je treba dodati, da tudi kroglja nima povsod na svojem površju enake barve, temveč ji dodamo šum (Ebert in sod., 2003). Ustvarjanje šuma, ki je značilen za naravno okolje, je časovno precej potratno, še posebno pa to velja za šum, ki ga potrebujemo pri ustvarjanju dima, saj mora biti ta tridimenzionalen, da dobimo prostorski občutek v ustvarjeni sliki. To velja tudi za naš primer, saj želimo, da bo dim deloval čim bolj realno, čeprav ga dodajamo na fotografijo. Da bi dobili potrebno hitrost izrisa, smo krogle zamenjali z implicitnimi krogi, ki jih izračunavamo s pomočjo senčilnikov na grafičnem procesorju, za šum pa uporabljamo vnaprej izračunane vrednosti. Za opis barve posamezne točke uporabljamo model RGBA, pri katerem je barva določena s tremi osnovnimi komponentami, in sicer rdečo (R), zeleno (G) in modro (B), ter prozornostjo (A). Večja ko je prozornost točke, več ozadja lahko vidimo. Prozornost točke izračunavamo po modelih za ustvarjanje oblakov, kumulacijo barve in gostote, ki prozornost zmanjšuje, pa izračunavamo s pomočjo posebnih vmesnikov na grafični kartici. Grafična kartica je zelo hitra, zato nam tak način omogoča skoraj enako kakovost

dima kot s klasičnim prostorskim upodabljanjem, vendar v realnem času, kar pomeni, da je izračun dovolj hiter, da lahko z njim pripravljamo gladko animacijo tudi pri več izvorih dima. Oblak dima je značilno predstavljen z največ sedemdeset implicitnimi krogi, ki so različnih velikosti, saj se radij kroga s časom od njegovega nastanka nenehno povečuje, kar daje občutek širjenja dima po prostoru. Vsak krog znotraj oblaka se tudi premika. Njegovo pot nadzoruje sistem delcev (Reeves, 1983), ki je značilen pripomoček pri ustvarjanju ognja ali dima, razvit že leta 1983. Sistem delcev v vnaprej določenih presledkih ustvarja točko, ki nato potuje skozi prostor. Enačba, ki določa njeno pot, hkrati tudi določa njen namen uporabe. Sistemov delcev namreč ne uporabljamo le za ustvarjanje dima ali ognja, temveč tudi eksplozij, slapov in podobnega. V enačbi za določitev poti delca so tudi naključne spremenljivke, ki omogočajo, da ima vsak delec svojo pot. V našem primeru vsak delec predstavlja središče implicitnega kroga. Na sliki 6 lahko vidimo ustvarjanje oblaka dima po opisanem postopku.



Slika 6: Sistem implicitnih krogov za ustvarjanje dima
Figure 6: System of implicit circles needed for smoke generation

Vsak delec ima svojo življenjsko dobo, ki se naključno določi in lahko traja od nekaj milisekund do sekunde. Največ delcev ima kratko življenjsko dobo, zato je največ implicitnih krogov pri izvoru, zaradi česar je tam dim tudi najbolj gost. Oblak je treba na koncu še obdelati z nizkim sitom (Sonka in sod., 2008), da zakrijemo meje med posameznimi krogi in dobimo končno sliko oblaka. Z velikostjo sita spreminjamo tudi strukturo oblaka, ki variira od rahle meglice do gostega, valečega se dima (glej sliko 7). Velikost sita je tretji parameter, ki ga ima uporabnik na voljo. Velikost sita si lahko predstavljamo kot velikost kvadrata, ki ga postavimo na oblak dima in znotraj njega poenotimo barvo oblaka. Po izračunu novih vrednosti se sito premakne za eno slikovno točko v vodoravni ali navpični smeri.



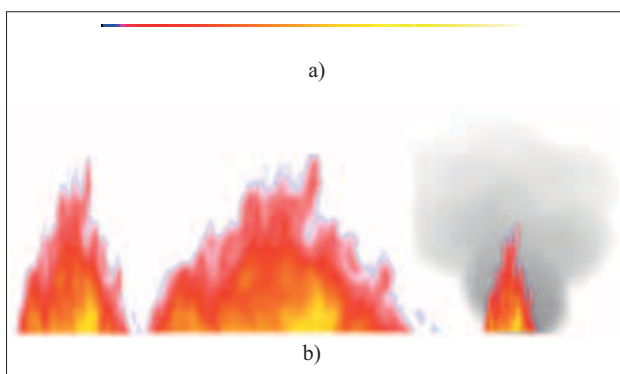
Slika 7:
Različni tipi dima, ki jih dosežemo s spreminjanjem parametrov znotraj SIN-a
Figure 7:
Various types of smoke, obtained by the changing of two parameters within the SIN

Na sliki 7 lahko vidimo štiri popolnoma različne oblake, ki jih dobimo s spreminjanjem prosojnosti in barve oblaka. Pri tem smo pri pripravi aplikacije SIN izhajali iz domneve, da bomo potrebovali le oblake dima pri požarih, zato lahko uporabnik izbira le med odtenki sive barve. Velikost siva vpliva na zamegljenost oblaka, saj večja kot je, bolj zamegljen je oblak. Oblak dima lahko v nasprotju s statičnimi gradniki večamo ali zmanjšujemo tudi ločeno v vodoravni in navpični smeri, kar nam omogoča ustvarjanje še raznovrstnejših oblik. To velja za vse dinamične gradnike. Dodatno nam pri ustvarjanju realne slike pomaga tudi to, da lahko oblake dima med seboj kombiniramo, s čimer dobimo še več oblik. Pri tem ne smemo pretiravati, saj lahko preveliko oblakov tako upočasnijo izračune, da animacija dima ni več gladka. To je odvisno od računalnika, na katerem aplikacija poteka.

Dim lahko kombiniramo tudi z ognjem. Ustvarjanje ognja je bilo veliko bolj preprosto in se prav tako izračunava na grafični kartici. Glavna zamisel je, da spreminjamo gradientno sliko ognja po precej zahtevnih, vendar hitrih enačbah, ki sestavljajo plamen realnega videza.

Na sliki 8 b vidimo tri slike plamenov, ki jih dobimo le z večanjem ali manjšanjem izvirnega plamena. Algoritem za ustvarjanje ognja je veliko preprostejši od algoritma ustvarjanja dima, zato posebnih nastavitev za lastnosti plamena nimamo.

Poleg gradnika plamena imamo tudi združen gradnik dima in ognja, ki ponuja možnost spreminjanja že opisanih parametrov dima in položaja plamena glede na ta oblak dima.



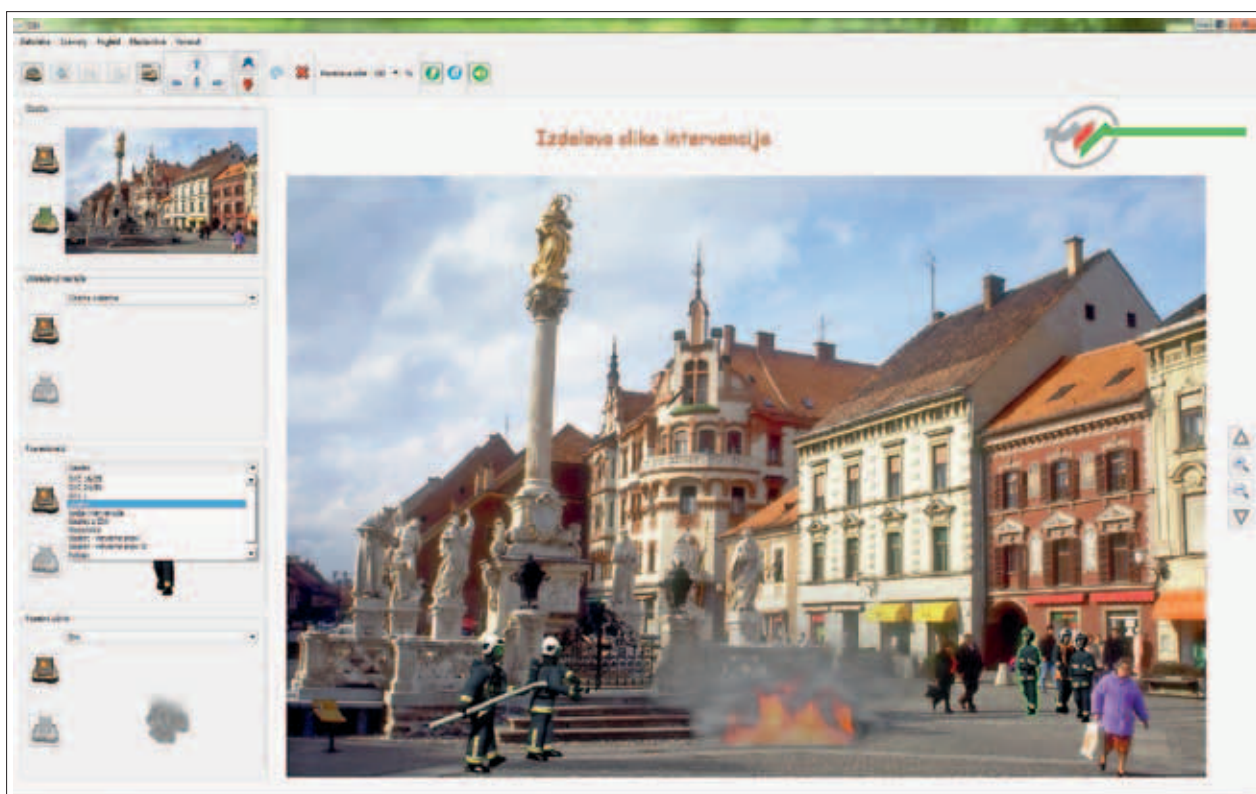
Slika 8: Ustvarjanje plamenov a) izhodiščna gradientna slika, b) različni tipi plamenov, ki jih lahko dobimo
Figure 8: Flame generation a) starting gradient image, b) various flame types that can be generated

Tako algoritem za ustvarjanje dima kot tudi algoritem za ustvarjanje ognja potrebujeta za svoje delo grafično kartico, ki dovoli programiranje grafičnega procesorja, kar pomeni problem pri starejših prenosnikih z integrirano grafično kartico. Ta problem smo rešili z vnaprej izračunano osnovno animacijo gradnikov ognja in dima. Tako dobimo animacijo, ki je sicer enako kakovostna, le da je pri določenih nastavitvah parametra dima viden preskok s konca na začetek izračunane animacije, ki pa ni preveč moteč.

Priprava scenarija nesreče in posredovanja

Scenarij znotraj aplikacije SIN je skupek dinamičnih slik oziroma scen, ki prikazujejo neki tip nesreče, ki jo je treba rešiti. Ozadje znotraj scenarija je vedno enako, zato se je zanj treba odločiti ob pripravi prve scene. Ozadja ne moremo preoblikovati z zmanjševanjem ali katerim drugim oblikovanjem, temveč je vedno nespremenjeno. Orodja, ki jih imamo na voljo za pripravo scenarija, so zbrana na levi strani okna aplikacije in so razdeljena v štiri področja (slika 9), kar sovпада z delitvijo že opisanih statičnih in dinamičnih gradnikov. Znotraj vsakega razdelka ima uporabnik na voljo osnovna ukaza, in sicer za nalaganje gradnika ter njegovo postavitev na sceno. To sta tudi edina ukaza, ki ju lahko uporabimo pri gradniku fotografije ozadja, pri drugih skupinah gradnikov pa je treba gradnik, ki ga želimo dodati na sceno, izbrati iz padajočega menija. Gradnike lahko nalagamo in jih dodajamo na sceno šele, ko je postavljena slika ozadja. Po postavitvi gradnika na sceno ta postane aktivni gradnik, ki ga lahko premikamo, večamo ali manjšamo, spreminjamo vrstni red njegovega izrisa, ga vrtimo ali zberšemo. Vsak aktivni gradnik je okvirjen z zeleno obrobo, ki se prilega njegovim konturam, da ga lahko ločimo od drugih. Našteti ukazi so dostopni prek orodne vrstice, postavljene pod menijem. Dodatne opcije, ki jih ukaz ponuja, se pokažejo ob njegovi aktivaciji. Tako nam na primer ukaz za premikanje gradnika omogoča njegovo natančno pozicioniranje in kontrolo za določanje vrstnega reda njegovega izrisa (glej sliko 9), s čimer ga postavi pred ali za druge objekte. Oboje je ob normalnem delovanju skrito.

Ko smo s položajem gradnika na sceni zadovoljni, kliknemo z miško na ozadje, s čimer zelena obroba okrog



Slika 9: Razširjena oblika orodij za nadzor nad posameznimi gradniki
[vir fotografije ozadja: <http://www.slovenijaturizem.com>]

Figure 9: Extended view of the tools for the control of building components
[background image source: <http://www.slovenijaturizem.com>]

objekta izgine, objekta pa ne moremo več spreminjati. Lahko kliknemo tudi na kakšen drug gradnik, ki bo tako postal aktiven, kar bo omogočalo njegove spremembe. To pomeni, da lahko objekte tudi pozneje spreminjamo, kar je še posebno koristno ob prehodu na novo sceno. Ko namreč izberemo ukaz za ustvarjanje nove scene, se vanjo preslikajo vsi gradniki, ki jih lahko z izjemo ozadja tudi spreminjamo. Za tako opcijo smo se odločili, ker se zaporedni sceni med seboj bistveno ne razlikujeta, navadno je dodan še kak dodatni gradnik, drugi gradniki pa so enaki kot na prejšnji sceni, mogoče le nekoliko premaknjeni. Tak pristop omogoča hitro pripravo scenarijev in tudi spreminjanje že obstoječih. Scenarij pripravljamo sceno po sceno, pri tem pa lahko vrstni red posameznih scen spreminjamo, lahko jih iz scenarija tudi brišemo ali dodajamo nove. Dinamične slike se ob poučevanju prikazujejo v celozaslonskem načinu, tako da orodja niso moteča. V scene lahko dodajamo tudi zvok, kar omogoča pripravo dodatnih scenarijev nevarnosti, ki se na začetku kažejo z zvokom, kot je na primer puščanja plina, ki se razvije v eksplozijo. Tako bi lahko na prvih scenah slišali le zvok, ki se s scene na sceno intenzivira in se čez nekaj časa ob neukrepanju spremeni v eksplozijo. Pripravo scenarijev lahko uporabljamo ne le za poučevanje novih postopkov, temveč tudi za analize intervencij in preverjanje znanja, saj lahko pripravimo tečajniku scenarij s problemi, ki ga mora dopolniti z rešitvami. Delo s scenariji je zelo preprosto, zato je SIN primeren za uporabo ne glede na stopnjo računalniške pismenosti slušateljev.

Sklepne misli

Aplikacija SIN je bila zasnovana kot pomoč inštruktorju pri razlagi zahtevnih postopkov za poučevanje pravih postopkov pri nesrečah z nevarno snovjo in pri požaru. Uporablja se lahko za poučevanje postopkov prvih posredovalcev in tudi gasilske taktike. Posebna prednost orodja SIN pred podobnimi rešitvami je priprava scenarijev znotraj realnega okolja, ki ga slušatelji poznajo. To lahko dosežemo le tako, da kot ozadje vzamemo fotografijo realnega okolja, zato dobimo orodje, ki ima zelo široko področje uporabnosti, ki je omejeno le s skupinami gradnikov. Gradniki so umetno ustvarjeni objekti, postavljeni na zeleno ozadje in združeni s fotografijo ozadja s pomočjo preizkušene tehnike iz filmske industrije, in sicer metode zelenega ozadja. Znotraj aplikacije lahko ustvarimo dinamične slike, ki jih lahko predvajamo le znotraj aplikacije SIN, lahko pa ustvarimo tudi običajne slike, ki jih lahko uporabimo tudi znotraj drugih aplikacij ali kot slikovne zbirke na spletu. Preprosto je tudi dodajanje novih statičnih gradnikov, saj je struktura zbirke slik gradnikov določena s posebno datoteko, ki jo lahko tudi spremenimo in nove gradnike dodamo v skupino ponesrečencev ali posredovalcev. Žal to ni mogoče za bolj kompleksne, to je dinamične gradnike, saj bi to zahtevalo posege v izvorno kodo, kar bi presegalo povprečno znanje uporabnika na področju računalništva. Priprava scenarijev je preprosta in hitra, kar omogoča hitro

spreminjanje že pripravljenih scenarijev ter pripravo novih. Okolje scenarijev ni vnaprej definirano, zato lahko orodje uporabimo tudi v druge namene, na primer kot pripomoček za analizo opravljenih intervencij, še posebno, ko so zahtevnejše.

Pri preverjanju znanja je še precej možnosti za izboljšavo in nadaljnje delo, saj to ob razvoju orodja ni bila glavna naloga. Če razlago inštruktorja posnamemo, lahko scenarij opremimo tudi s tem zvočnim podatkom in se lahko tako aplikacija uporablja tudi za dobro predpripravo na izobraževanje ali za poučevanje na daljavo. Tako lahko dodatno izboljšamo pripravljenost gasilskih enot in osnovna izobraževanja, ki potekajo v organizaciji posameznih občinskih gasilskih zvez, saj lahko s pomočjo orodja SIN pripravimo kakovostno multimedijско gradivo, ki bi bilo veliko bolj zanimivo od običajnega. Z uporabo tako pripravljenega gradiva ne bi le izboljšali pripravljenosti gasilskih enot, z njim bi lahko pritegnili tudi mlade člane, saj bi lahko aplikacijo uporabljali tudi v gasilskih krožkih po osnovnih šolah ter tako povečali zanimanje najmlajših za požarno varnost in tudi za gasilsko službo. Možnosti uporabe aplikacije SIN za izboljšanje znanja pri gasilcih in za povečanje požarne varnosti je veliko. Aplikacija je rezultat domačega dela, zato jo lahko tudi preprosto dopolnjujemo in prilagajamo potrebam, ki bi se pozneje pojavile, s čimer bi lahko njihovo uporabnost še povečali.

Zahvala

Avtorji se zahvaljujemo Upravi Republike Slovenije za zaščito in reševanje za financiranje ter Izobraževalnemu centru za zaščito in reševanje na Igu za strokovno sodelovanje in pomoč, še posebno pa Milanu Dubravcu, ki je z nami delil svoje dolgoletne izkušnje na področju izobraževanja gasilcev.

Viri in literatura

1. Backlund, P., Engström, H., Gustavsson, M., Johansson, M., Lebram, M., and Sjörs, E., 2009. SIDH: A Game-Based Architecture for a Training Simulator, *International Journal of Computer Games Technology*, vol. 2009, članek ID 472672.
2. Ebert, D. S., Musgrave, F. K., Peachey, D., Perlin, K., Worley, S., Mark, W. R., Hart, J. C., 2003. *Texturing & Modeling A Procedural Approach*, tretja izdaja, Morgan Kaufman Publishers.
3. Levoy, M., 1988. Display of Surfaces from Volume Data, *IEEE Computer graphics and applications*, vol. 8, št. 3, str. 29–37.
4. Lucas, J., Thabet, W., 2008. Implementation and Evaluation of a VR Task-Based Training Tool For Conveyor Belt Safety Training, *ITcon*, vol. 13, str. 637–659.
5. Querrec, R., Buche, C., Maffre, E., Chevallier, P., 2003. SecuReVi: Virtual Environments For Fire Fighting Training, *VRIC 2003 (Colloque Laval Virtual 2003 Cinquieme Conférence Internationale Sur La Réalité Virtuelle)*, Laval, France 14-16, str. 169–175.
6. Reeves, W. T., 1983. Particle Systems - A Technique for Modeling a Class of Fuzzy Objects, *ACM Transactions on Graphics*, vol. 2, št. 2, str. 91–108.
7. Sherman, W. R., Penick, M. A., Su, S., Brown, T., Harris, F. C., 2007. VR fire: an immersive visualization experience for wildfire spread analysis. In: *IEEE virtual reality conference*. Charlotte, NC, str. 243–246.
8. Sonka, M., Hlavac, V., Boyle, R., 2008. *Image processing, analysis and machine vision*, tretja izdaja, Thompson Learning.
9. Splet 1: <http://www.commandsim.com/>, 2011.
10. Splet 2: <http://www.rescuesim.com/>, 2011.
11. Tate, D. L., Sibert, L., King, T., 1997. Using Virtual Environments to Train Firefighters, *IEEE Computer Graphics and Applications*, vol. 17, str. 23–29.