

Fabijan Lukin, Fran Pregernik, Tomislav Sukser

Korisnička dokumentacija za obojivo računarstvo

U Zagrebu, travanj 2006.

SADRŽAJ

| | | |
|-------|---|----|
| 1 | Uvod u obojivo računarstvo | 4 |
| 2 | Zahtjevi za pokretanje obojivog računarstva..... | 5 |
| 3 | Instalacija obojivog računarstva..... | 6 |
| 4 | Grafičko sučelje za simulaciju..... | 7 |
| 4.1 | Izgled i dijelovi simulatora..... | 7 |
| 4.2 | Pokretanje simulacije i opcije simulacije..... | 10 |
| 4.3 | Ubacivanje programa na zid | 12 |
| 4.4 | Pregled sadržaja obojivog računala..... | 16 |
| 5 | SmartClient - Grafičko sučelje za simulaciju bazirano na konceptu klijent-poslužitelj | 18 |
| 5.1 | Izgled i dijelovi simulatora SmartClient | 18 |
| 5.2 | Pokretanje simulacije..... | 21 |
| 5.3 | Ubacivanje programa na zid | 22 |
| 5.4 | Pregled sadržaja obojivog računala..... | 23 |
| 6 | Potpuna klijent-poslužitelj simulacija..... | 25 |
| 6.1 | Parametri i pokretanje poslužitelja | 25 |
| 6.2 | Pokretanje klijenta | 26 |
| 6.3 | Napomena o kompatibilnosti | 28 |
| 7 | Virtualno okruženje i senzori..... | 29 |
| 7.1 | Postavljanje senzora | 29 |
| 7.1.1 | Postavljane senzora prilikom stvaranje nove simulacije..... | 29 |
| 7.1.2 | Pojedinačno postavljanje senzora | 30 |
| 7.1.3 | Grupno postavljanje senzora..... | 31 |
| 7.2 | Prikaz senzora u simulatoru | 32 |
| 7.3 | Isticanje senzora i portala | 33 |
| 8 | Programi | 35 |
| 8.1 | Sigurnost u programima | 35 |
| 8.2 | Parametri simulacije za navedene primjere | 35 |
| 8.3 | Program Breadcrumb | 36 |
| 8.4 | Program Gradient | 37 |
| 8.5 | Program Pulsator..... | 39 |

| | | |
|-------|---|----|
| 8.6 | Program Shadowfax | 41 |
| 8.7 | Koordinatni sustav | 46 |
| 8.7.1 | Trikovi i Gradient na steroidima | 46 |
| 8.7.2 | Postavljanje 3 programa Coordinate Tunnel | 47 |
| 8.7.3 | Bojanje koordinatnog sustava | 49 |
| 8.7.4 | Egzotični primjeri koordinatnog sustava | 51 |
| 9 | Literatura..... | 55 |

1 Uvod u obojivo računarstvo

Nekadašnja moćna stolna računala pozamašnog volumena i potrošnje mogu se danas napraviti veličine zrnca pijeska. Preostaje nam kupiti pola kilograma takvih „zrnaca“, pomiješati ih u boju i obojati zid. Rezultat toga je obojivo računarstvo (izvorno Paintable Computing, izraz koji je primjenio William Joseph Butera u radu Programming a Paintable Computer). Iako tema ovog projekta više liči na soboslikarstvo, spomenuta zrnca, a dalje u tekstu obojiva računala, nisu ograničena na boju već se mogu ugraditi u tkanine, građevinske materijale i sl. Spomenuta obojiva računala neće biti samostalna, ona će biti međusobno bežično povezana, bilo infracrvenom vezom ili radio vezom. Napajat će se iz okoline, otporna na okolinu i jeftina. Možda je ovo ideja za distribuirano računarstvo budućnosti.

Što god nam budućnost donosi, mi ćemo se ovdje pozabaviti sa simulacijom jedne takve hipotetske budućnosti. Otići ćemo u dućan, kupiti pola kilograma obojivih računala i 5 litara boje za zid, doći kući u sobu, zamiješati obojiva računala u boju te obojati zid. Nakon toga ćemo se smjestiti u naslonjač i diviti se našem novom sustavu računala ogromne procesne moći. Zid će zaživjeti.

Vratimo se iz ove utopijske budućnosti (tko si još može priuštiti odmaranje u naslonjaču?!) u našu stvarnost gdje postoji simulator za zidove prepune obojivih računala. Zid ćemo zamijeniti crnom podlogom simulatora, a obojiva računala točkicama raznih boja. Točkice kojima ćemo povremeno mijenjati boju na našoj crnoj podlozi ne zovu se obojiva računala jer mijenjaju boju već zato jer su ubojana u zid, a naše prikazivanje točkica različitim bojama neće biti ništa drugo nego prikazivanje različitih stanja u kojima se pojedino obojivo računalo nalazi. Stoga, pogledajmo što sve možemo napraviti s pola kilograma obojivih računala na zidu...

2 Zahtjevi za pokretanje obojivog računarstva

Za pokretanje simulatora obojivog računarstva mogu se navesti minimalni zahtjevi. Neki preporučljivi zahtjevi se ne mogu lako navesti, zato jer simulator mora biti u stanju simulirati nekoliko tisuća obojivih računala gdje se u svakom računalu nalazi po nekoliko programa za obojiva računala te svako računalo komunicira s nekoliko svojih susjeda i stalno izmjenjuje podatke. Brojke dovoljno govore o kompleksnosti sustava za simulaciju.

Stoga, minimalni zahtjevi za pokretanje i barem djelomično ugodan rad su:

- Pentium (ili srodni procesor), barem 2.0 GHz
- 128 MB slobodne radne memorije
- Nekoliko MB mjesta na disku
- Microsoft Windows XP ili 2003 Server
- Microsoft .NET Framework 2.0 (Runtime)

Ukoliko se želi simulirati više tisuća obojivih računala gdje u svakom se nalazi više programa, preporučljivo je imati slobodno oko 512 MB radne memorije. I na kraju, ukoliko se želi koristiti mogućnost spremanja simulacije u video datoteku (AVI), onda je potrebno red veličine GB slobodnog prostora na disku.

3 Instalacija obojivog računarstva

Program nije potrebno instalirati. Ukoliko su ispunjeni svi preduvjeti iz prethodnog poglavlja, onda je dovoljno mapu Binaries kopirati sa optičkog medija Obojivo računarstvo na željeno mjesto na disku i pokrenuti neku od izvršnih datoteka. U narednim poglavljima možete naučiti koju za što.

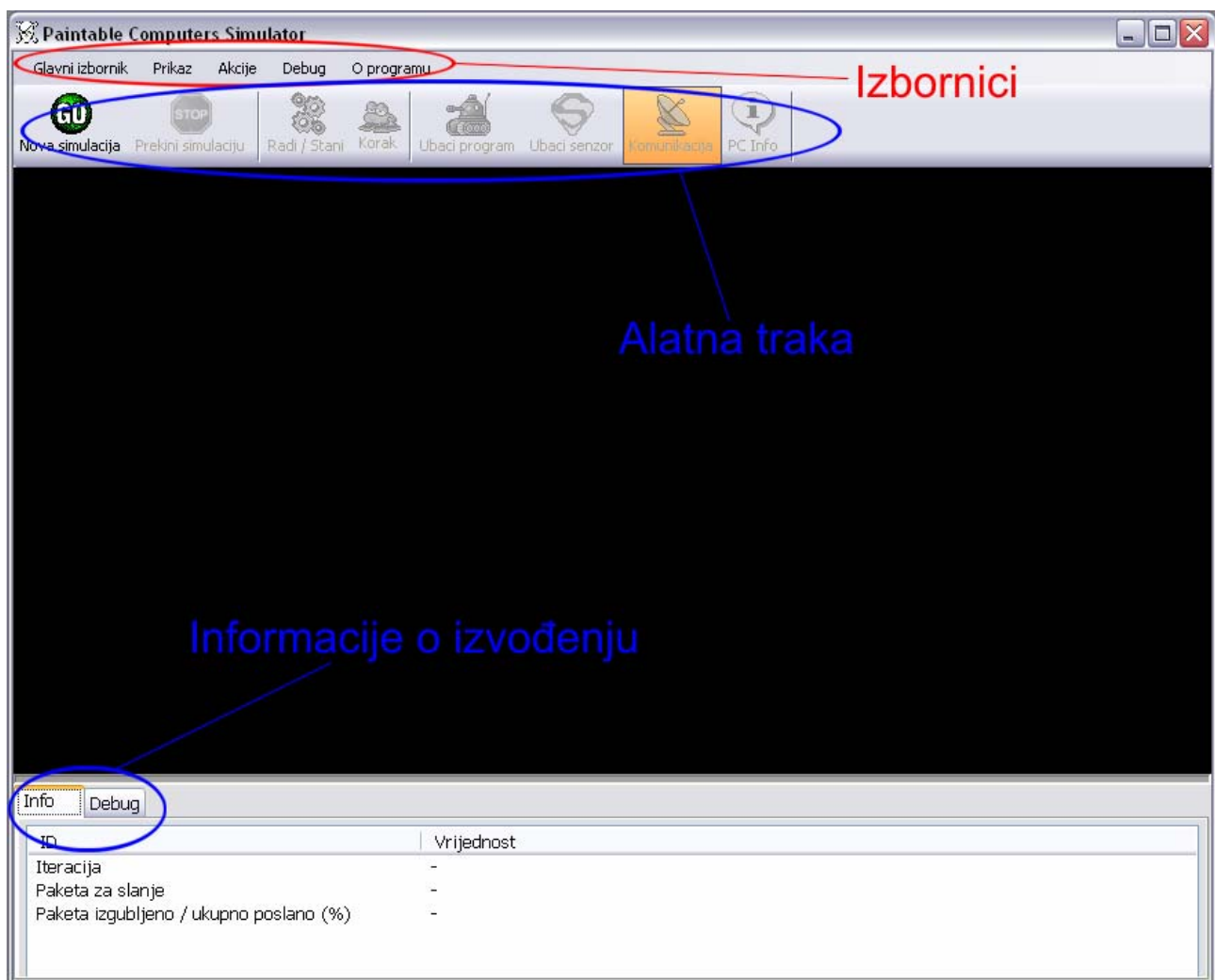
| Name ▲ | Size | Type |
|--|----------|------------------------|
| hr | | File Folder |
| Plugins | | File Folder |
| log4net.dll | 224 KB | Application Extension |
| log4net.xml | 1,258 KB | XML Document |
| PaintableSimulator.Engine.dll | 88 KB | Application Extension |
| PaintableSimulator.Engine.pdb | 246 KB | Program Debug Database |
| PaintableSimulator.Engine.XML | 160 KB | XML Document |
| PaintableSimulator.GUI.exe | 104 KB | Application |
| PaintableSimulator.GUI.exe.config | 1 KB | XML Configuration File |
| PaintableSimulator.GUI.pdb | 84 KB | Program Debug Database |
| PaintableSimulator.GUI.vshost.exe | 6 KB | Application |
| PaintableSimulator.GUI.vshost.exe.config | 1 KB | XML Configuration File |
| PaintableSimulator.GUI.XML | 11 KB | XML Document |
| PaintableSimulator.Server.application | 2 KB | Application Manifest |
| PaintableSimulator.Server.exe | 24 KB | Application |
| PaintableSimulator.Server.exe.config | 2 KB | XML Configuration File |
| PaintableSimulator.Server.exe.manifest | 4 KB | MANIFEST File |
| PaintableSimulator.Server.pdb | 24 KB | Program Debug Database |
| PaintableSimulator.Server.XML | 3 KB | XML Document |
| PaintableSimulator.SmartClient.exe | 456 KB | Application |
| PaintableSimulator.SmartClient.pdb | 216 KB | Program Debug Database |
| PaintableSimulator.SmartClient.XML | 74 KB | XML Document |
| ProcessSimulator.Visualizations.dll | 84 KB | Application Extension |
| ProcessSimulator.Visualizations.pdb | 46 KB | Program Debug Database |
| ProcessSimulator.Visualizations.XML | 4 KB | XML Document |

Sadržaj mape binaries

4 Grafičko sučelje za simulaciju

Simulator obojivog računarstva s kojim bi valjalo započeti uvod u obojivo računarstvo pokreće se iz datoteke PaintableSimulator.GUI.exe. Na početku nas dočeka jedan zid obojan u crno. To nije slučajnost, svi zidovi su crne boje u simulaciji, a određena stanja obojivih računala koja su rasuta po zidu će biti prikazana u raznim bojama, tek toliko da iskoristimo naše vizualne receptore.

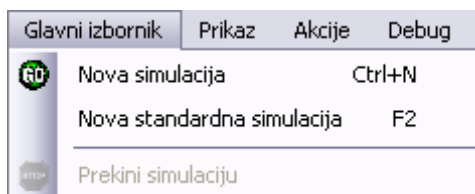
4.1 Izgled i dijelovi simulatora



Osnovni izgled ekrana simulatora obojivog računarstva

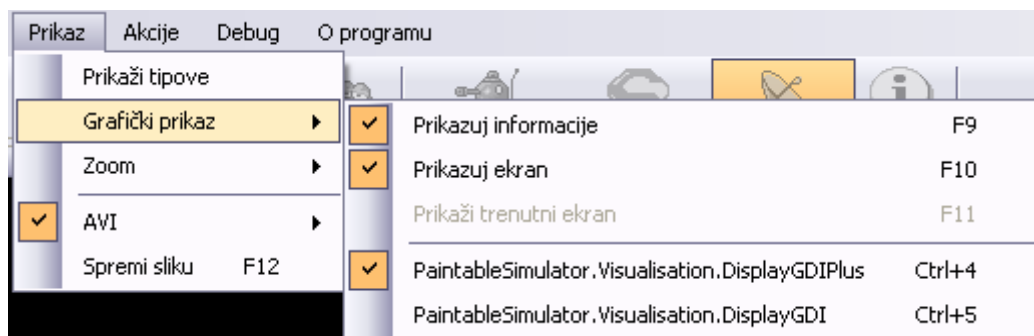
Za početak, možemo objasniti smisao pojedinog izbornika. Glavni izbornik, što se i iz imena da naslutiti, služi za obavljanje osnovnih operacija u simulatoru. Prva stavka u izborniku je pokretanje nove simulacije pod nazivom Nova simulacija. Druga stavka, Nova standardna simulacija, je pokretanje simulacije već predefiniiranih parametara, prvenstveno namijenjena za programere programa i njihovo testiranje. Na kraju, posljednja stavka izbornika služi za prekidanje simulacije. Spomenute stavke za pokretanje nove

simulacije i za prekidanje simulacije imaju svoje kratice na alatnoj traci. Pri dnu ekrana nalaze se informacije o trenutnom stanju simulatora, gdje se može vidjeti koliko je taktova procesora prosječnog obojivog računala proteklo, koliko paketa se u određenom trenutku šalje u čitavoj komunikacijskoj mreži, te postotak izgubljenih paketa kao rezultat fiktivne greške u komunikaciji. Drugi dio pri dnu ekrana (Debug) predstavlja ispis najbitnijih događaja simulatora.



Glavni izbornik simulatora

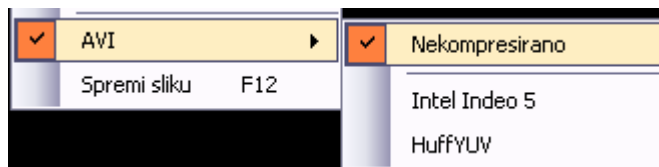
Sljedeći izbornik jest izbornik u kojem određujemo prikaz simulacije. Prva stavka imenom Prikaži tipove je u pravilu izbornik koji se dodatno proširuje i koji ima popis učitanih programa, te time možemo izabrati program čiju vizualizaciju ćemo pratiti na ekranu. Grafički prikaz sadrži mogućnosti s kojima možemo izabrati način vizualizacije. To su mogućnosti pod nazivom „PaintableSimulator.Visualisation.DisplayGDIPlus“ i „PaintableSimulator.Visualisation.DisplayGDI“. Prvi označava korištenje GDI Plus Windows API-ja za iscrtavanje slike na ekran, a drugi označava običan GDI Windows API. Ovisno o samom računalu na kojem se pokreće simulator i njegovim karakteristikama, jedan od ta dva načina je bolji. Stavka podizbornika „Prikazuj informacije“ označava prenose li se informacije simulacije iz same jezgre simulacije do ovog grafičkog sučelja. Ukoliko se te informacije ne prenose, simulacija je ponešto brža. Stavka „Prikazuj ekran“ označava hoće li se ekran iscrtavati tokom simulacije automatski ili ne. Ukoliko se ekran neće automatski prikazivati, pokretanjem stavke „Prikaži trenutni ekran“ prikazujemo sliku koja je rezultat vizualizacije simulacije tog trenutka. Stavka Zoom izbornika Prikaz je vrlo vjerojatno razumljiva sama po sebi, ona služi za uvećanje ili umanjeње prikaza, a njezin je raspon od 10% do 200%.



Izbornik za prikaz

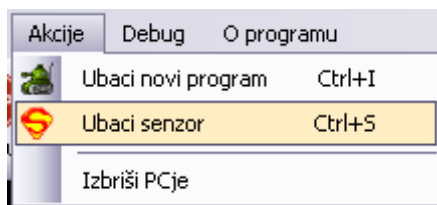
U drugom dijelu izbornika Prikaz ne određuje se način prikaza na ekran nego se nude mogućnosti pohrane vizualizacije u datoteke. Prva mogućnost pohrane je u obliku animacije gdje je rezultat spremanja video datoteka (AVI), a u podizborniku određujemo hoće li to biti nekompresirana video datoteka, ili datoteka kompresirana metodom Intel Indeo 5 ili HuffYUV. Za opcije Intel Indeo 5 i HuffYUV, odgovarajuća potpora tim formatima mora biti instalirana na računalu. Na kraju, posljednja stavka izbornika Prikaz koju opisujemo se zove „Spremi sliku“ koja sprema trenutni prikaz s ekrana u datoteku. Slika se

sprema u formatu PNG (Portable Network Graphics) u podmapu „ScreenShots“ u odnosu na mapu gdje se nalazi opisivani simulator PaintableSimulator.GUI.exe. Ekstenzija koja se dodjeljuje slikama jest bmp.



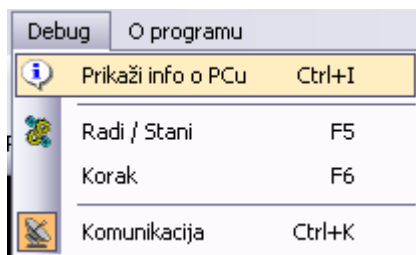
Drugi dio izbornika za prikaz

Sljedeći na redu je izbornik s akcijama. Prva stavka naziva „Ubaci novi program“ služi za ubacivanje novog programa u jedno od računala na već spomenutom zidu. Za ubacivanje senzora u jedno od računala poslužiti ćemo se stavkom „Ubaci senzor“. Šećer na kraju predstavlja mogućnost brisanja obojivih računala sa zida, što je ustvari ništa drugi nego simulacijska zamjena za masovno umiranje označenih računala iz raznih razloga.



Izbornik Akcije

Uz prethodno navedeni izbornik, izbornik Debug je jedan koji će vjerojatno češće biti korišten. Prva njegova stavka pod nazivom „Prikaži info o PCu“ služi za prikazivanje informacija o odabranom obojivom računalu. Sljedeća stavka Radi/Stani služi za pauziranje simulacije gdje se onda može stavkom Korak izvoditi korak po korak simulacije, odnosno služi za nastavak simulacije. Posljednja stavka Komunikacija omogućava ili onemogućava bilokakvu komunikaciju između bilo koja dva obojiva računala.



Izbornik Debug

Stavke alatne trake istih naziva kao i nabrojane stavke iz posljednja dva izbornika imaju istu funkcionalnost kao i stavke u izbornicima, a nalaze se na alatnoj traci zato jer su najčešće korištene. I na kraju, posljednji izbornik O programu služi za prikaz osnovnih podataka o programu i njegovim autorima.

4.2 Pokretanje simulacije i opcije simulacije

Početak ćemo s klikom na alatnu traku, na gumb Nova simulacija. S desne strane nam se otvorio jedan izbornik s raznim opcijama. Za početak ćemo objasniti osnovne opcije. Prva opcija pod nazivom Maksimalni broj PCova određuje maksimalni broj obojivih računala koja će biti postavljena na zid. Stvarni broj će gotovo uvijek biti nešto manji zato da se postigne određena pravilnost u postavljanju računala na zid. Izbornik Početni PCovi određuje vrstu računala koja će biti postavljena na zid. GenericPaintableComputer označava obično obojivo računalo, zatim PressureSensorComputer označava obojivo računalo sa senzorom pritiska, a TemperatureSensorPaintableComputer označava obojivo računalo sa senzorom za temperaturu. Za ovaj primjer ćemo odabrati najobičnije obojivo računalo GenericPaintableComputer. Komunikacija među obojivim računalima je bežična, a na ovom mjestu se može odrediti i njezin radijus. Na kraju, posljednja od osnovnih opcija je pohrana animacije simulacije u video datoteku. Za određivanje datoteke, može se napisati ime u odgovarajuću kućicu, ili se datoteka može odabrati klikom na gumb „...“

Početak simulacije

Osnovno Napredne opcije

Maksimalni broj PCova

2000

Početni PCovi

GenericPaintableComputer

Radius bežične veze

40

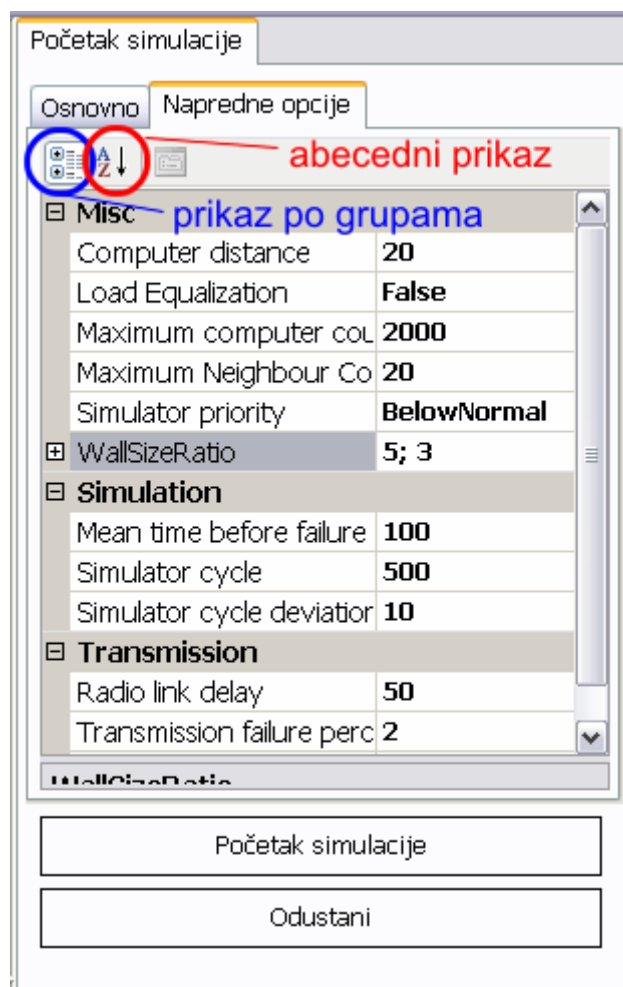
Ispis u AVI datoteku

Početak simulacije

Odustani

Osnovne opcije simulacije

Napredne opcije simulacije mogu se prikazati na dva načina: abecednim redom ili po grupama (označeno crvenom i plavom bojom na slici).



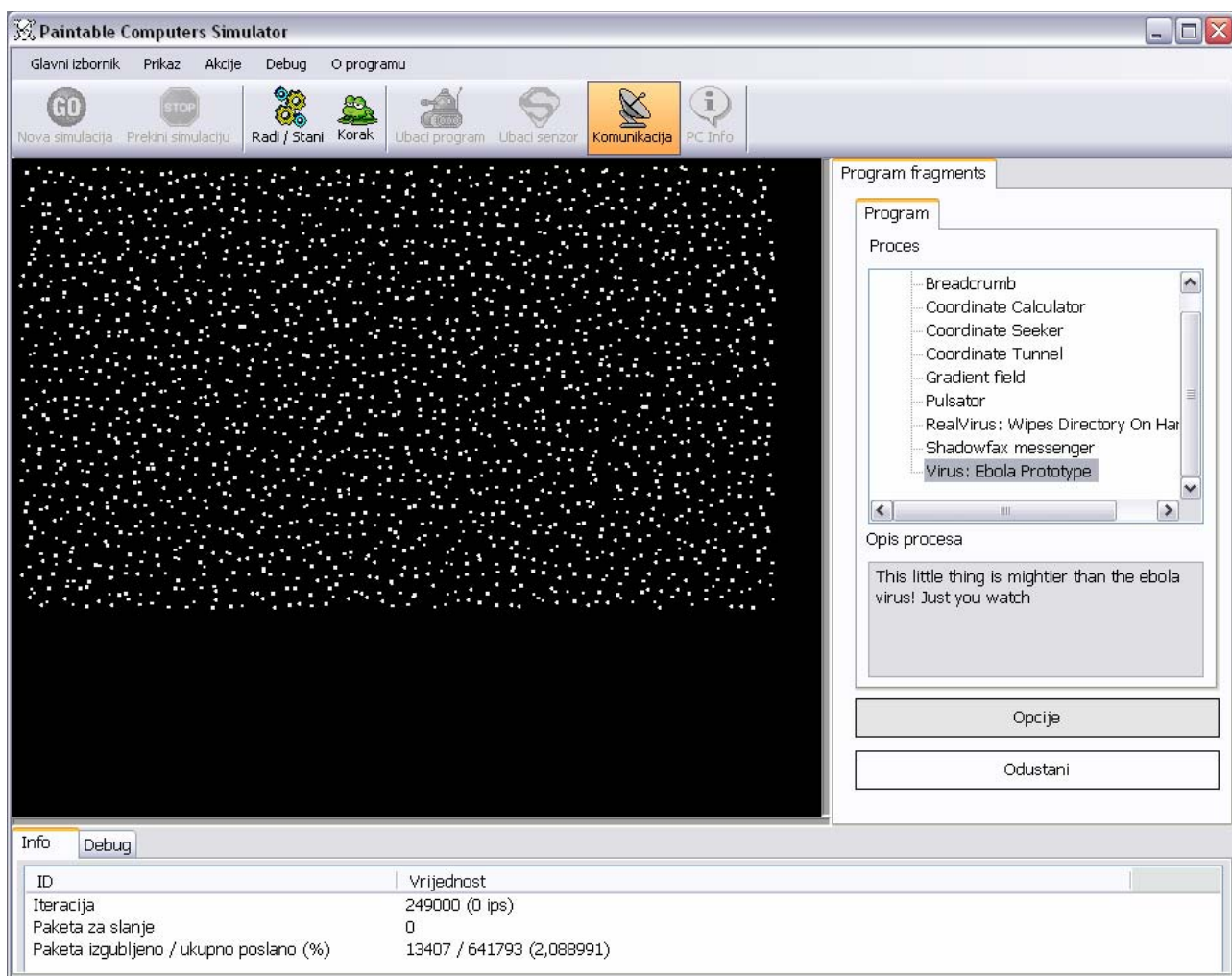
Napredne opcije simulacije

| Opcija | Opis |
|---------------------------------|--|
| Computer distance | Približna vrijednost razmaka između 2 obojiva računala na zidu, u određenim fiktivnim jedinicama za duljinu. |
| Load Equalization | Omogućavanje preraspodjele resursa za simulaciju kako bi se načelno dobila ujednačenija simulacija |
| Maximum computer count | Maksimalan broj obojivih računala na zidu |
| Maximum Neighbour Count | Maksimalan broj susjeda koje jedno računalo može upamtiti, bez obzira na to koliko je radijus komunikacije velik |
| Simulator priority | Prioritet dretve simulatora |
| WallSizeRation | Omjer širine i visine zida za simulaciju |
| Mean time before failure | Prosječno vrijeme života jednog obojivog računala, u ciklusima simulatora |
| Simulator cycle | Jedan ciklus simulatora se promatra kao određen broj taktova prosječnog obojivog računala |
| Simulator cycle deviation | Odstupanje od prosječnog trajanja jednog ciklusa simulatora, isto u taktovima obojivog računala. Razlog uvođenja jest nejednakost brzine takta svih obojivih računala. |
| Radio link delay | Kašnjenje u komunikaciji, isto u taktovima |
| Transmission failure percentage | Postotak krivo prenesenih paketa u komunikaciji. Paketi koji će doživjeti takvu sudbinu određuju se slučajno. |
| Wireless radius | Radijus bežične komunikacije između obojivih računala, u fiktivnim jedinicama za duljinu. |

Opis naprednih opcija simulacije

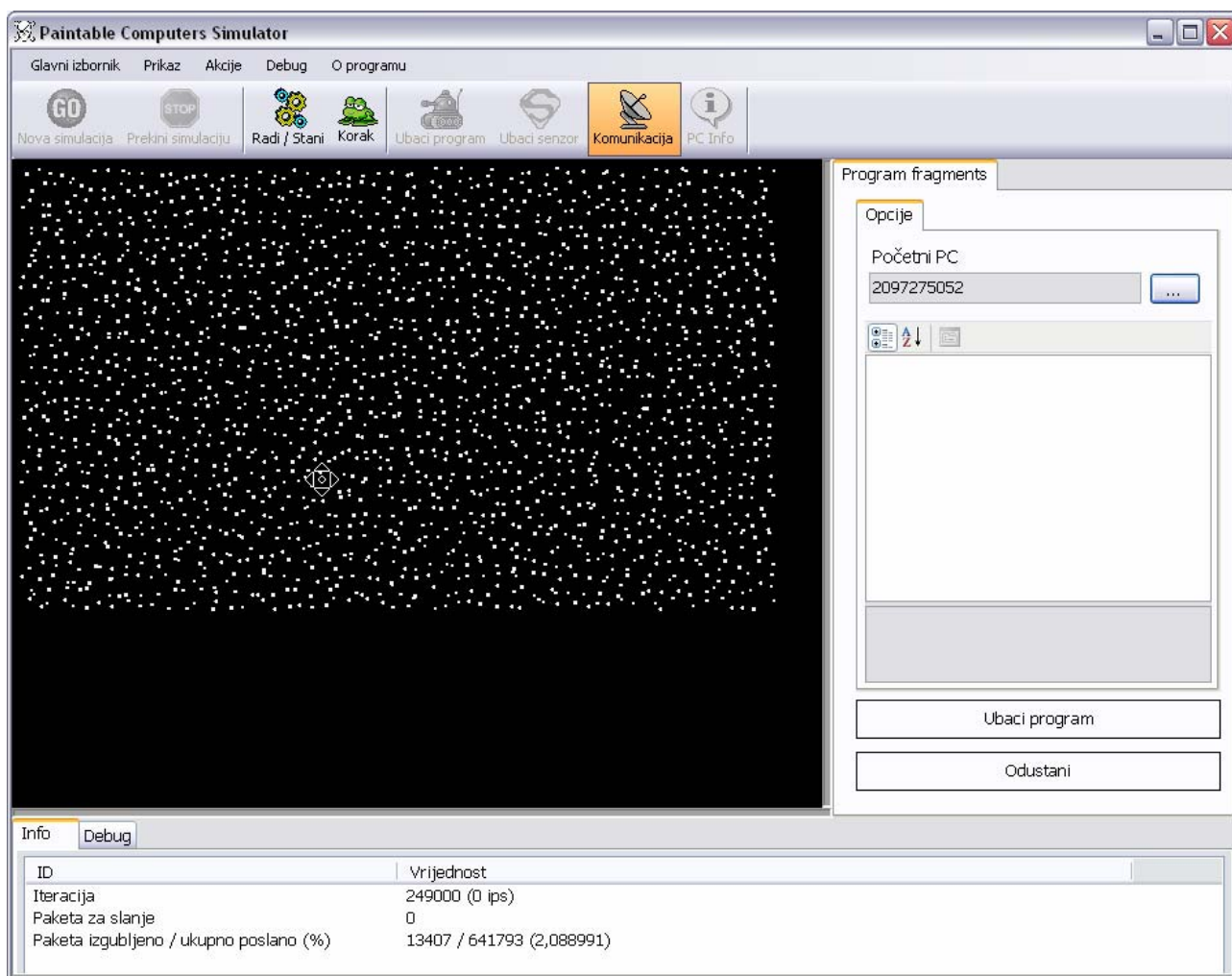
4.3 Ubacivanje programa na zid

Uz ostavljene predefinirane postavke, sad je red da pokrenemo simulaciju klikom na gumb Početak simulacije. Nakon toga smanjit ćemo prikaz na 50% odabравši Zoom i 50% iz izbornika Prikaz. Za prvi puta, privremeno ćemo zaustaviti simulaciju klikom na gumb Radi/Stani, te ćemo kliknuti na gumb Ubaci program kako bi ubacili jedan program na naš zid. Iz dobivenog popisa programa s desne strane, odabrat ćemo Virus: Ebola Prototype, te kliknuti na gumb Opcije, kako bi odredili obojivo računalo gdje ćemo ubaciti željeni program, te eventualno njegove opcije.



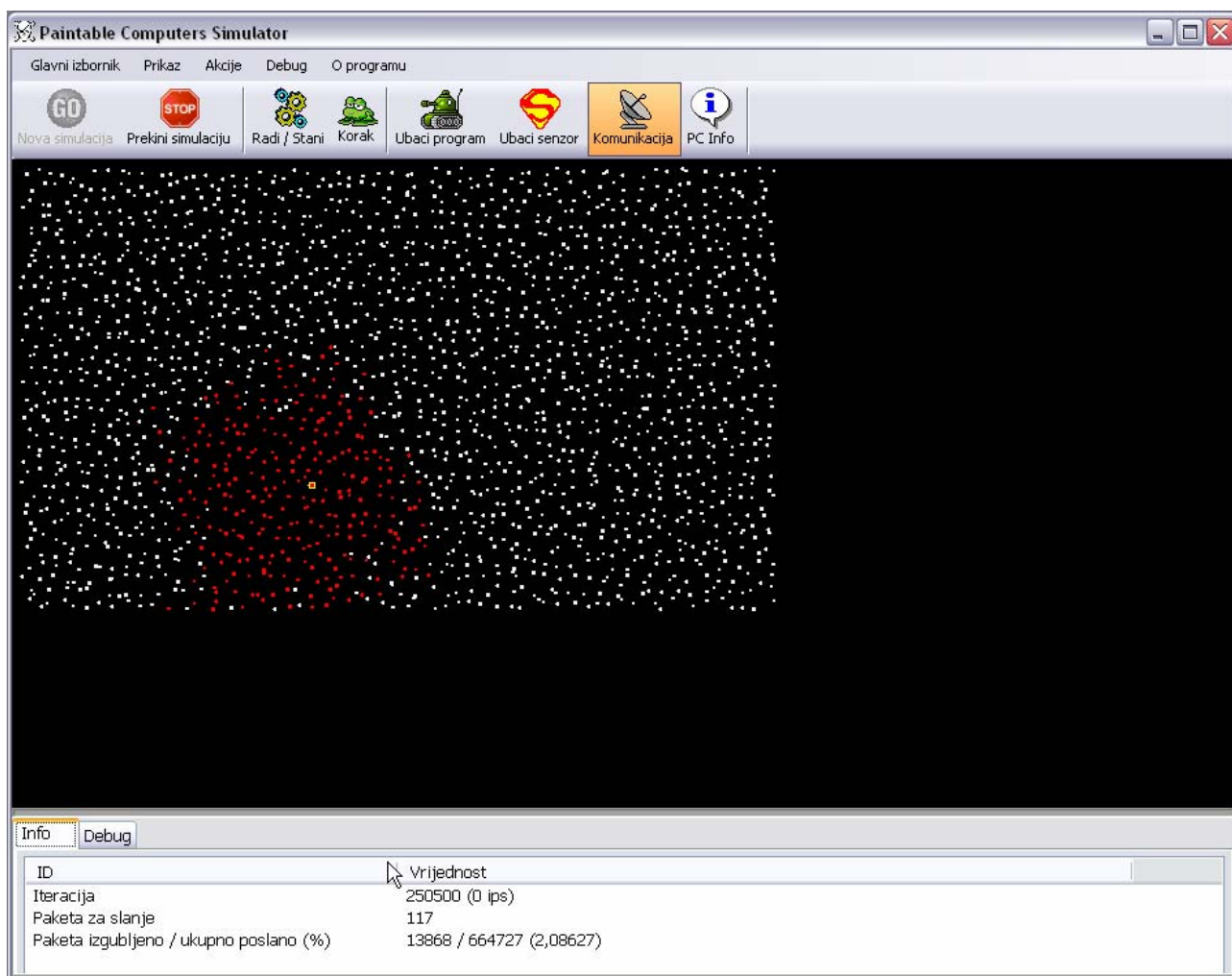
Odabir programa za ubacivanje

Odabrani program Virus: Ebola Prototype nije nešto što ima dodatnih opcija, pa ćemo zato odmah krenuti na odabir obojivog računala za ubacivanje. U tu svrhu ćemo kliknuti na gumb „...“ te ćemo onda kliknuti negdje na naš zid na jedno od naših računala, kao što je prikazano na slici.



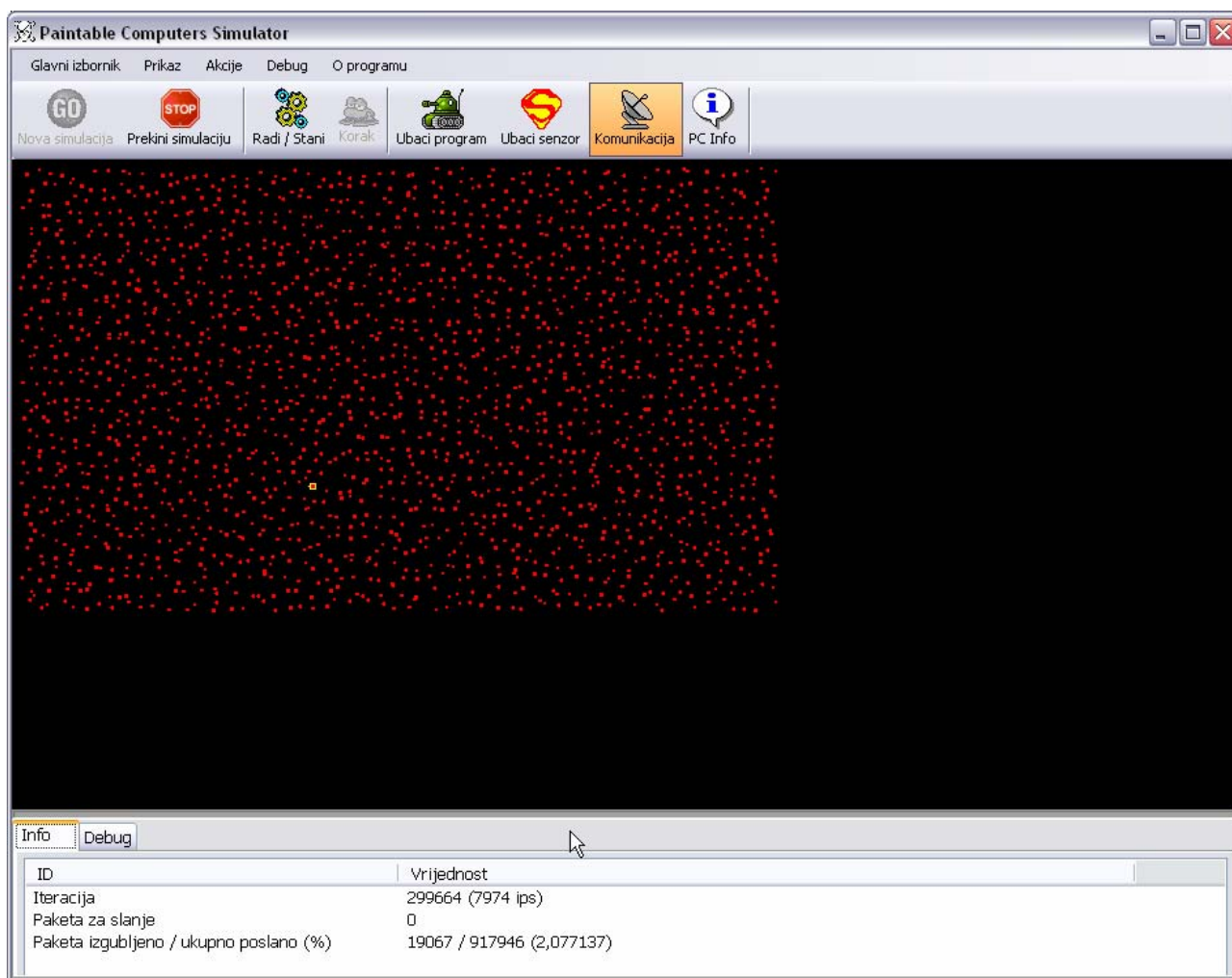
Odabir početnog računala za program

Na kraju ćemo kliknuti na gumb Ubaci program te ćemo učitati program u odabrano obojivo računalo. Zatim, ostaje nam još da isprobamo izvođenje simulacije korak po korak nekoliko puta, što ćemo napraviti s nekoliko klikova na gumb korak. Žuti kružić oko jednog obojivog računala označava da je to obojivo računalo portal, tj. ono preko kojega je neki program ušao na zid, te ovisno o namjeri programa, proširio se dalje.



Obavljenih par koraka simulacije

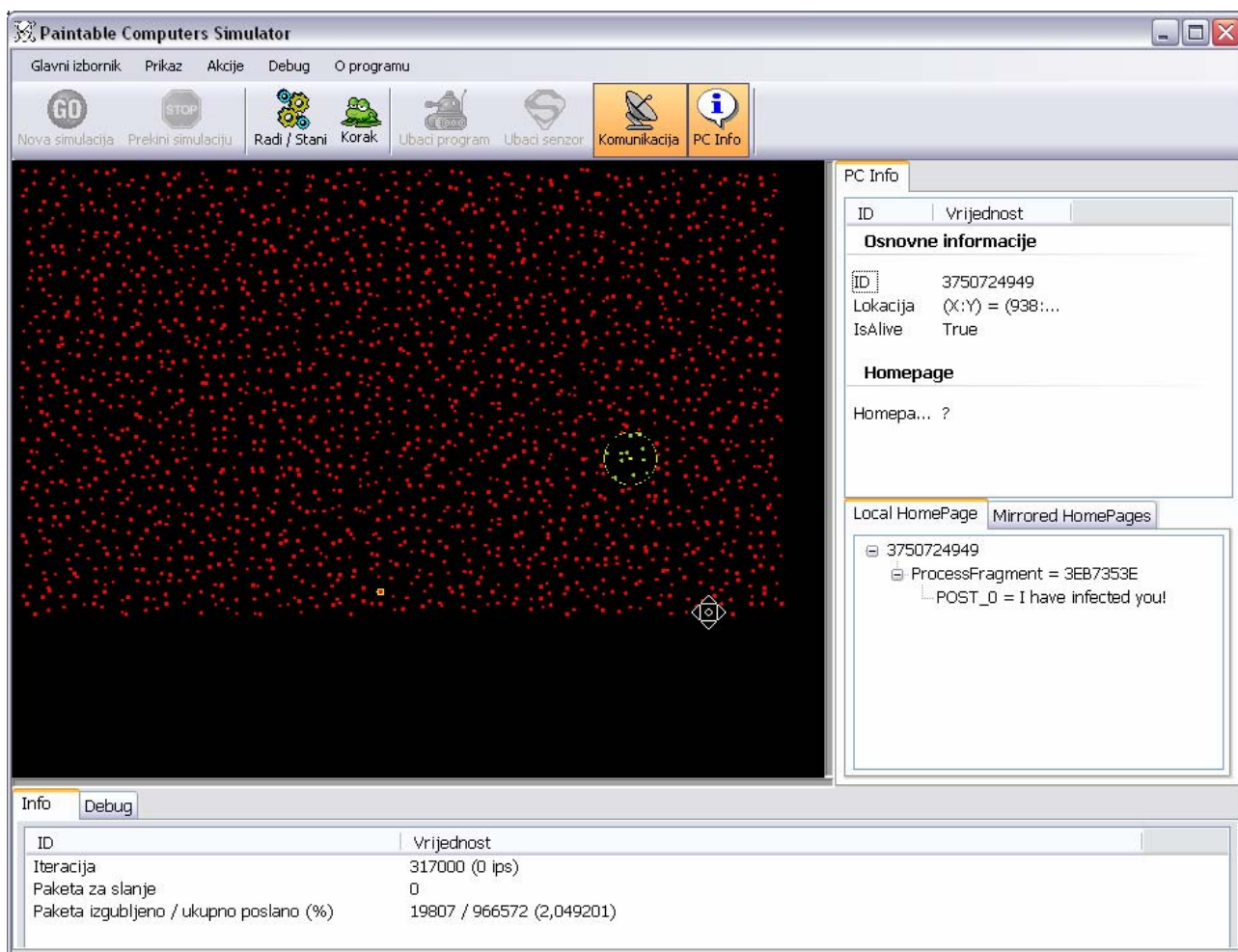
Sad možemo pogledati širenje dotičnog programa (virusa) u punoj snazi simulacije, pa ćemo stisnuti gumb Radi/Stani. Za nekoliko trenutaka cijeli zid bit će prekriven ovim programom.



Izgled zida nekoliko trenutaka nakon nastavljjanja simulacije

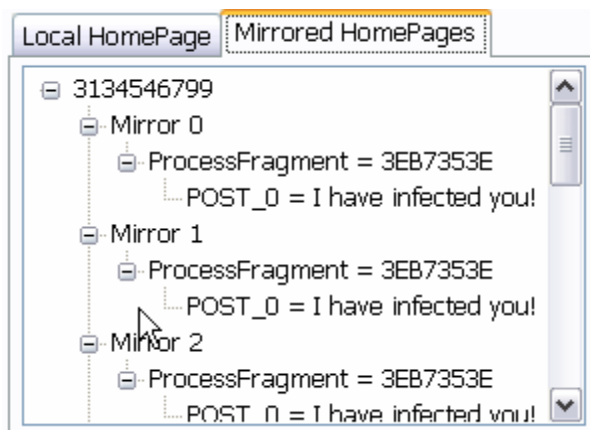
4.4 Pregled sadržaja obojivog računala

Još jedna podosta zanimljiva stvar jest pogledati što svako od obojivih računala ima na sebi. Zato ćemo kliknuti na gumb PC Info na alatnoj traci te s mišem doći nad računalo koje želimo promotriti. Oko njega će se prikazati jedan kružić koji predstavlja radijus komunikacije, a obojiva računala drugačije obojana su ona koja su susjedi sa trenutno odabranim obojivim računalom.



Informacije o obojivom računalu

S desne strane nalaze se podaci o tom računalu, gdje ID označava neki slučajni broj, koji bi valjalo promatrati kao serijski broj jednog obojivog računala. Lokacija u obliku koordinata određuje položaj računala na zidu, u fiktivnim jedinicama za duljinu. Zadnja stavka IsAlive označava je li to obojivo računalo živo. Pri dnu, nalazi se sadržaj lokalne podatkovne stranice (Local HomePage), te se nalazi sadržaj zrcaljenih podatkovnih stranica od susjeda odabranog obojivog računala.



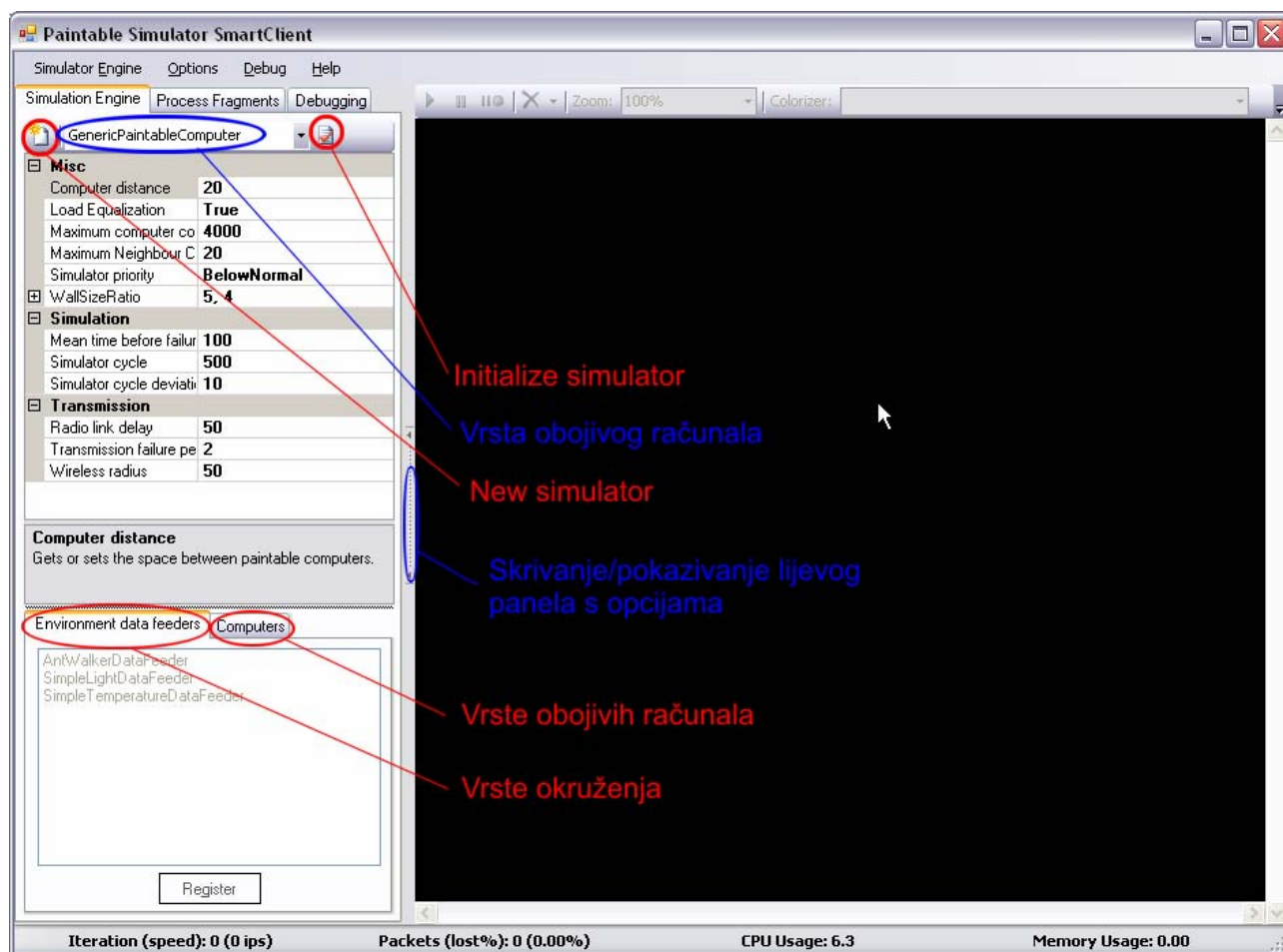
Informacije o zrcaljenim podatkovnim stranicama odabranog obojivog računala

5 SmartClient - Grafičko sučelje za simulaciju bazirano na konceptu klijent-poslužitelj

Drugi simulator koji je namijenjen i za rad u okruženju klijent-poslužitelj, može u potpunosti samostalno pokrenuti simulaciju i u tom načinu rada se ne razlikuje mnogo od funkcioniranja prethodno opisanog simulatora. On se krije iza izvršne datoteke PaintableSimulator.SmartClient.exe.

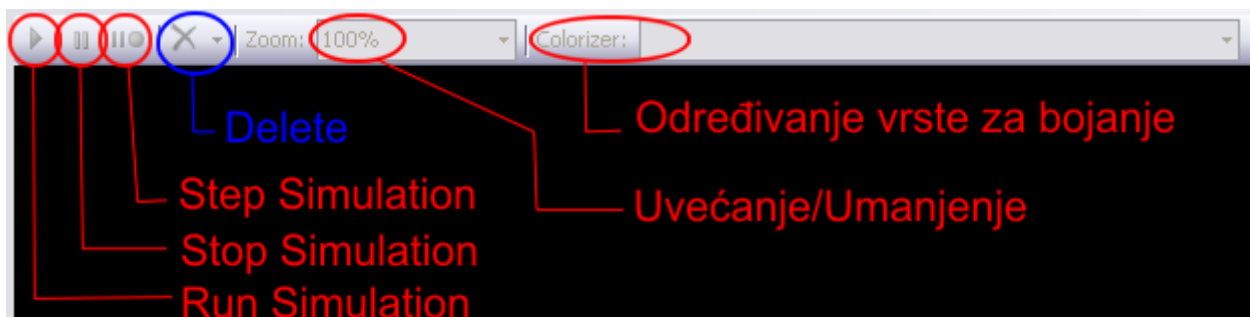
5.1 Izgled i dijelovi simulatora SmartClient

Simulator SmartClient se sastoji od nekoliko osnovnih dijelova. Jedan od njih je izbornik, koji će biti opisan kasnije detaljno. Drugi dio je lijeva traka s opcijama i alatima gdje se mogu odrediti parametri simulacije (Simulation Engine), tipovi obojivih računala i okruženje (Environment data feeders i Computers), mogu se odrediti opcije programa i učitati programi (Process Fragments), te se može iskoristiti skroman način proučavanja svakog zasebnog obojivog računala (Debugging). Najveći i najbitniji dio je velika crna površina, koja ne zahtijeva mnogo objašnjenja – to je zid gdje će se obojiva računala nastaniti.



Izgled simulatora SmartClient

Preostaje još kratki opis gornje alatne trake. Prva 3 gumba s lijeve strane imaju iste funkcije kao i istoimene stavke iz izbornika Simulator Engine. Zatim, Delete sa svoja 3 različita načina omogućava uklanjanje obojivih računala sa zida. Zoom služi za uvećanje ili umanjeње prikaza zida. Na kraju, stavka Colorizer označava koji će se program isticati na zidu, odnosno koji od učitanih će se bojati.



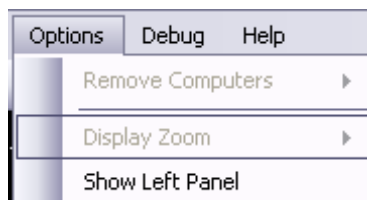
Gornja alatna traka

Stavke izbornika su po svojoj funkcionalnosti gotovo identične onima iz prethodno opisanog simulatora, te se ovdje neće detaljno razmatrati. Stavka New Simulator služi za stvaranje nove simulacije, pri čemu se prvo moraju odabrati parametri simulacije. Sljedeće po redu je Initialize Simulator, a to je stvaranje novog zida sa obojivim računalima. Dalje dolaze 3 stavke Run, Stop i Step Simulation čije su funkcije za pokretanje i nastavljjanje simulacije, druga stavka je za zaustavljanje simulacije (pauziranje), dok je posljednja za izvođenje jednog koraka simulacije. Sve navedene opcije imaju svoje gumbе na raznim alatnim trakama, što se može vidjeti na prethodnim slikama.



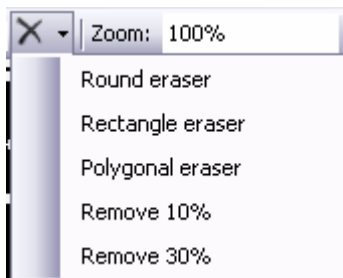
Izbornik Simulator Engine

Sljedeći se izbornik (Options) sastoji od 3 stavke. Prva navedena imena Remove Computers predstavlja način brisanja računala sa zida. Druga stavka Zoom određuje povećanje ili umanjeње prikaza zida s obojivim računalima na ekranu. Posljednja stavka (Show Left Panel) utječe na vidljivost lijeve trake s alatima i opcijama.

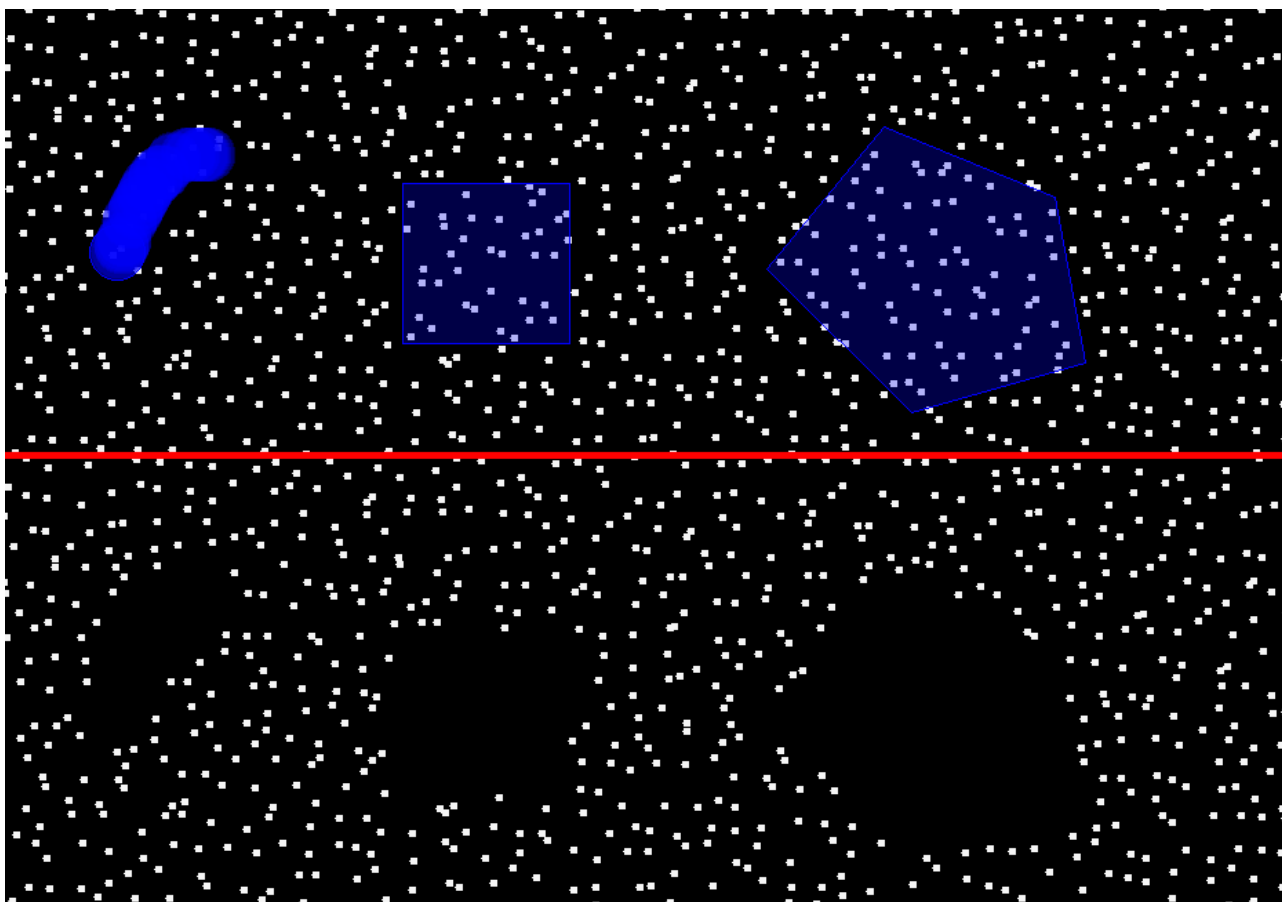


Izbornik Options

Postoji 5 načina brisanja obojivih računala sa zida. Prvi način je Round eraser, što je izvedeno u obliku kružića koji briše sva obojiva računala na koja naleti. Drugi način (Rectangle eraser) predstavlja brisanje svih obojivih računala koja se nađu unutar pravokutnika koji se crta između dvije točke. U oba slučaja, nakon odabira načina brisanja, potrebno je pritisnuti lijevu tipku miša, povući miša do željenog odredišta te otpustiti lijevu tipku miša. Treći način brisanja obojivih računala sa zida je crtanje poligona. Poligon se crta tako da se sa klikom na lijevu tipku miša na mjestu gdje želimo postaviti točke koje razapinju poligon. Kada završimo s izborom točaka i nacrtamo željeni poligon, klikom na desnu tipku miša, označena računala bit će obrisana. Posljednja dva načina brisanja uklanjaju 10%, odnosno 30% računala sa zida slučajnim odabirom.



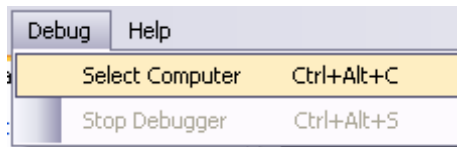
Načini brisanja obojivih računala sa zida



Prva tri načina brisanja na djelu, redom kako su navedeni

Izbornik Debug je vrlo siromašan, no sasvim dovoljan. Prva njegova stavka (Select Computer) dozvoljava odabir obojivog računala mišem. Sadržaj podatkovne stranice tog

računala i zrcaljenih podatkovnih stranica računala susjeda bit će prikazan u novom prozoru. Stavka izbornika Stop Debugger zatvara sve debug prozore koji otvoreni na opisani način.

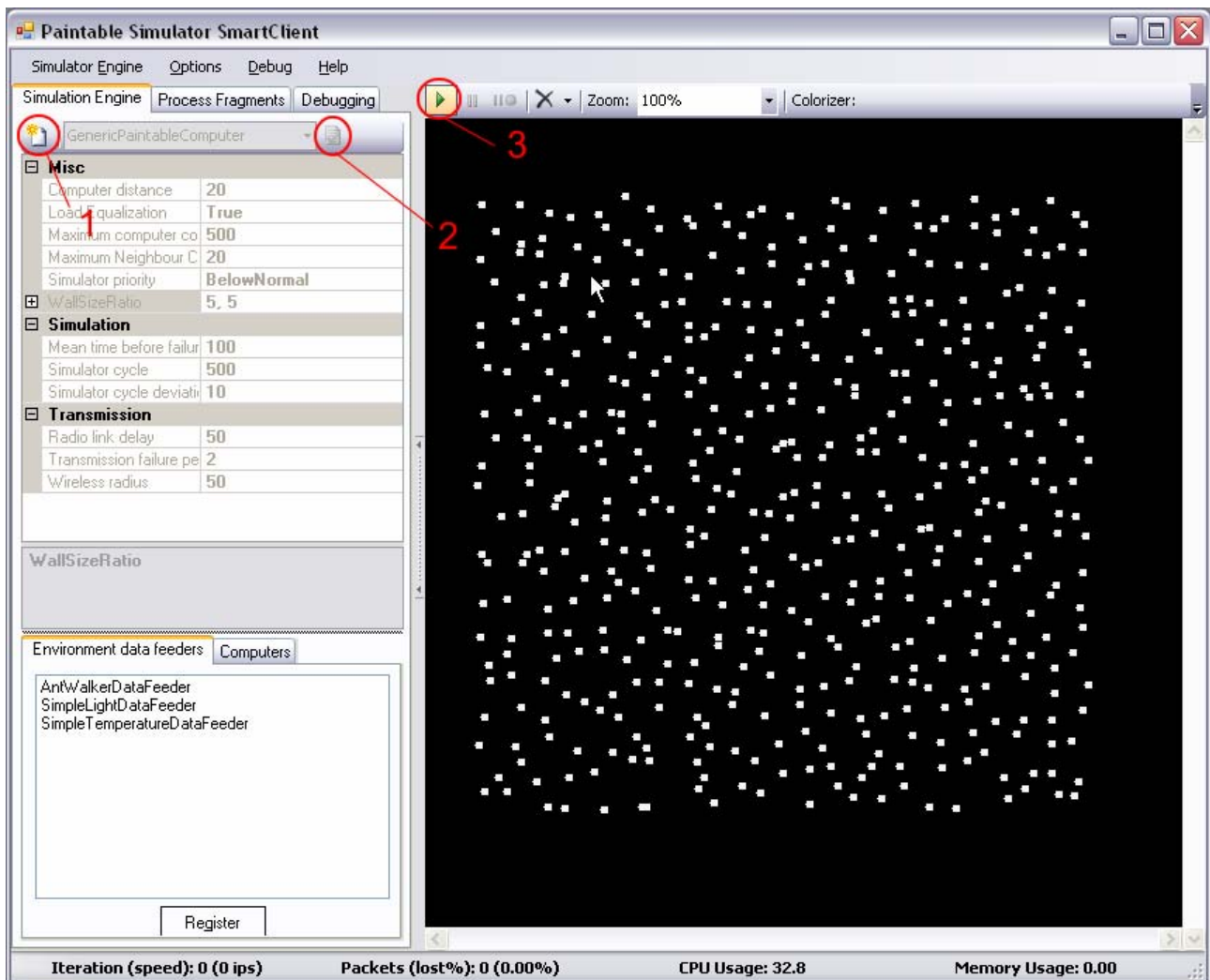


Izbornik Debug

Na kraju, izbornik Help sadrži samo osnovne podatke o programu i njegovim autorima.

5.2 Pokretanje simulacije

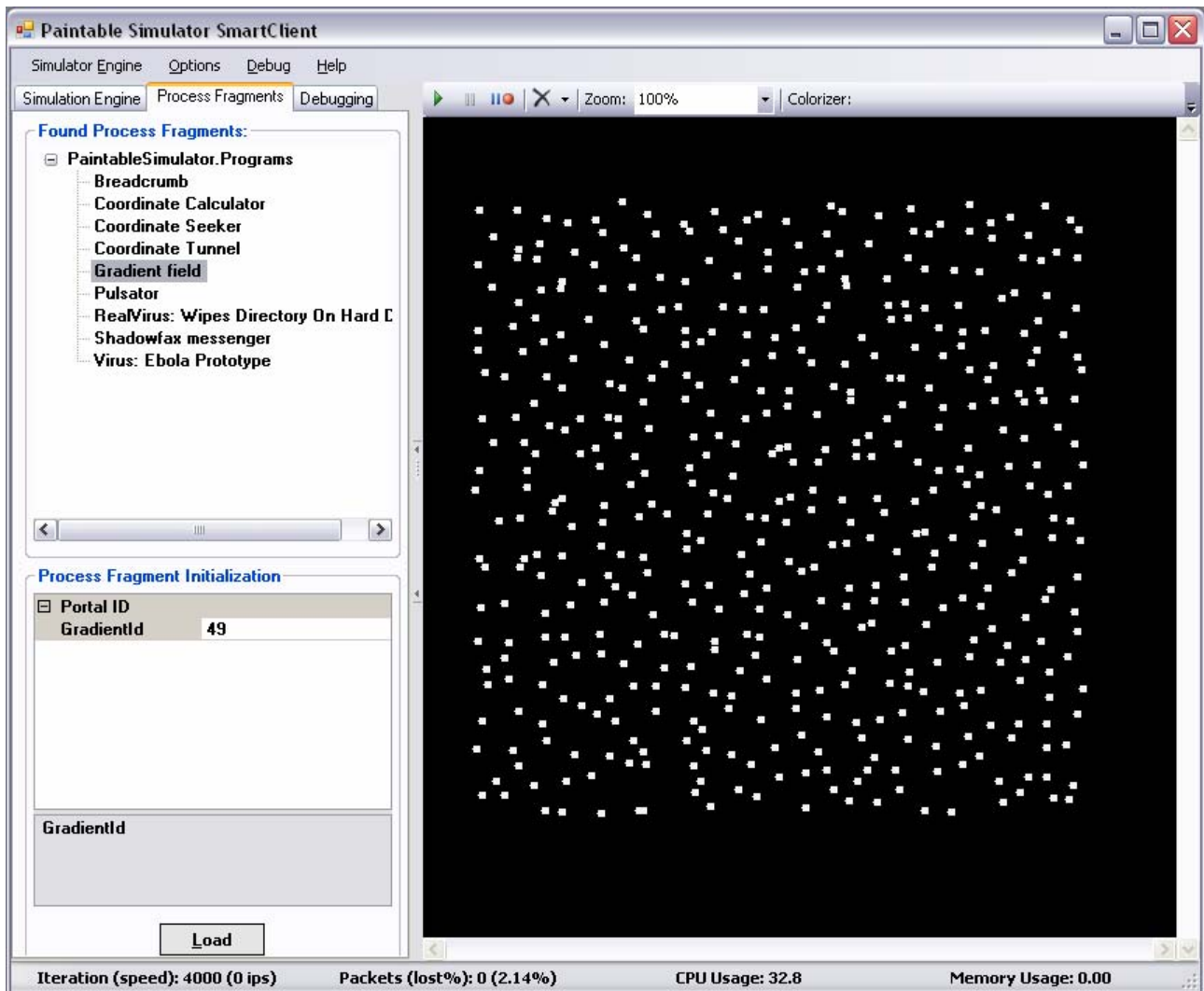
Na početku kliknut ćemo na gumb New Simulator (1), zatim ćemo podesiti parametre simulacije prema slici (parametri su objašnjeni u opisu simulatora), te ćemo kliknuti na gumb Initialize Simulator (2). Na kraju ćemo kliknuti na pokretanje simulacije.



Izgled simulatora pri pokretanju simulacije

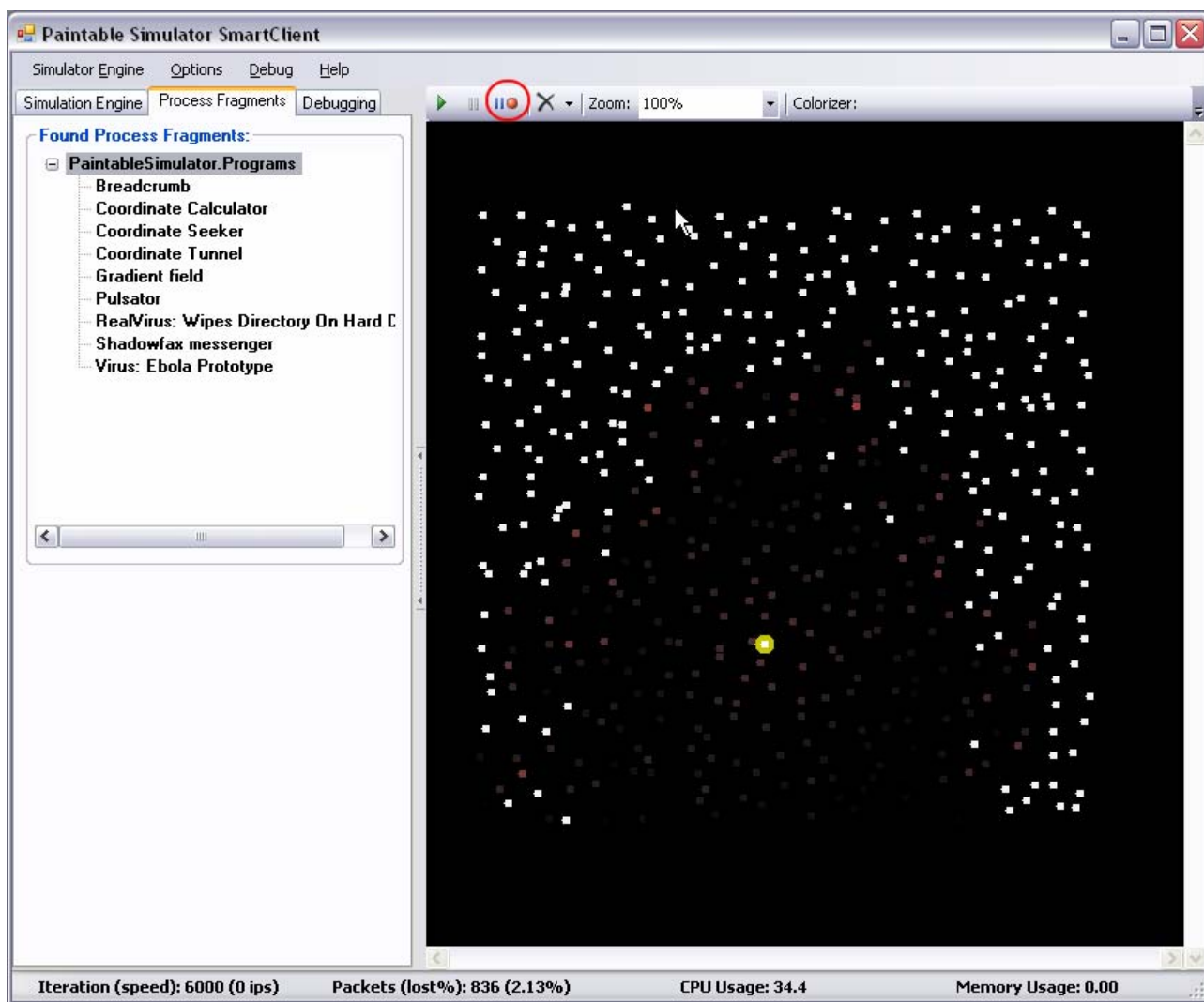
5.3 Ubacivanje programa na zid

Zatim ćemo pauzirati simulator kako bi opet mogli vidjeti izvođenje simulacije korak po korak. U traci s lijeve strane odabrat ćemo stranicu Process Fragments gdje ćemo uzeti program imena Gradient Field za ubacivanje. Njegove opcije dolje prikazane ćemo ostaviti kakve jesu, te ćemo kliknuti na gumb Load.



Odabir programa za ubacivanje

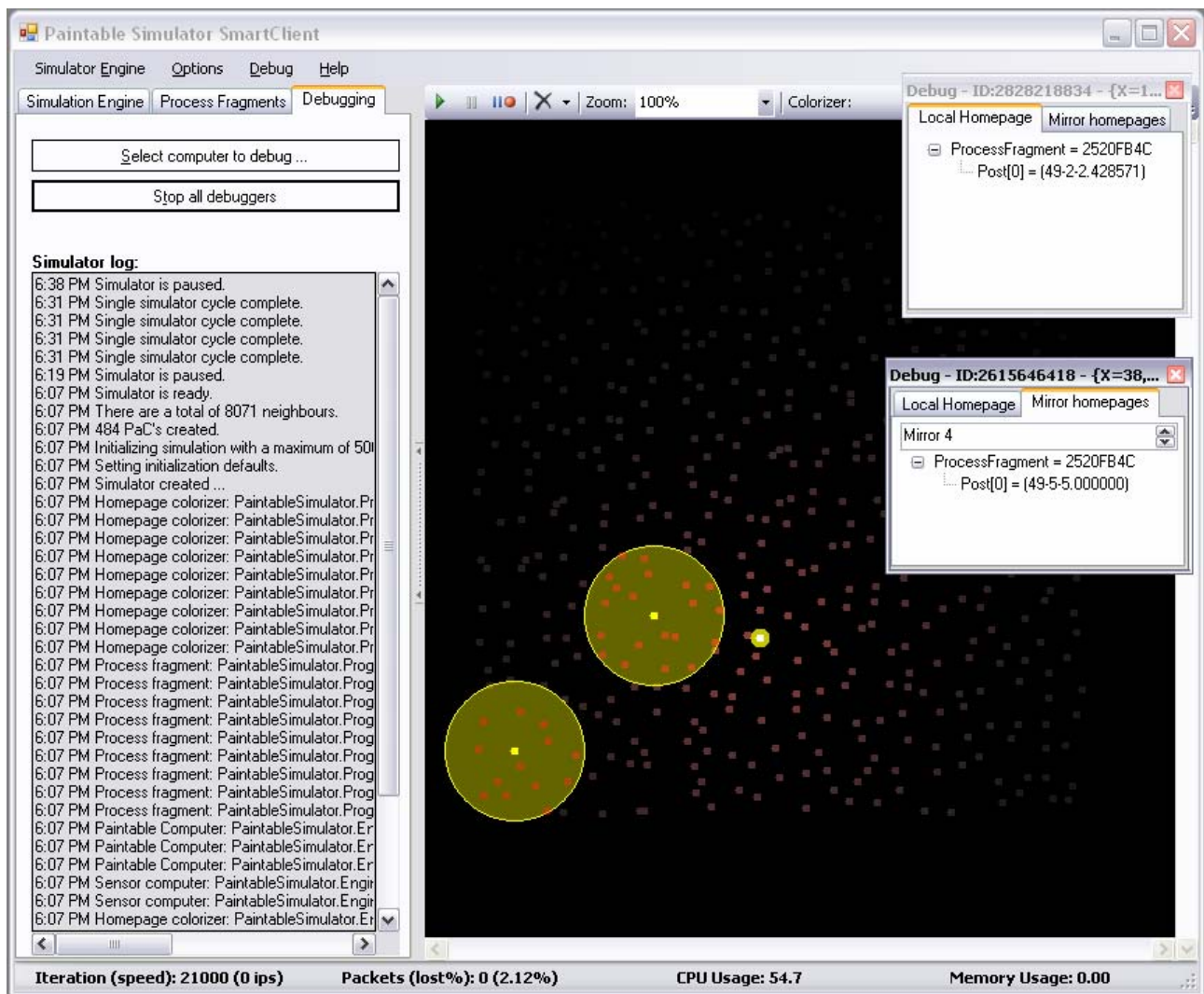
Nakon što smo kliknuli na load otići ćemo mišem nad jedno obojivo računalo na zidu te ćemo kliknuti na njega lijevom tipkom miša. Nakon toga ćemo kliknuti nekoliko puta na gumb Step Simulation i rezultat će biti poput ovog na slici.



Simulacija nakon izvođenja nekoliko koraka simulacije

5.4 Pregled sadržaja obojivog računala

Nakon nekoliko obavljenih koraka, pustit ćemo simulaciju da se dalje izvodi. Kad program raširi preko cijelog zida, kliknut ćemo na gumb Select Computer to Debug... na stranici Debugging s lijeve strane. Zatim ćemo kliknuti na jedno računalo na zidu. Taj postupak ćemo ponoviti za još jedno računalo za koje ćemo prikazati zrcaljene podatkovne stranice njegovih susjeda, a rezultat bi trebao biti kao na slici.



Podatkovne stranice i zrcaljane podatkovne stranice u simulaciji

6 Potpuna klijent-poslužitelj simulacija

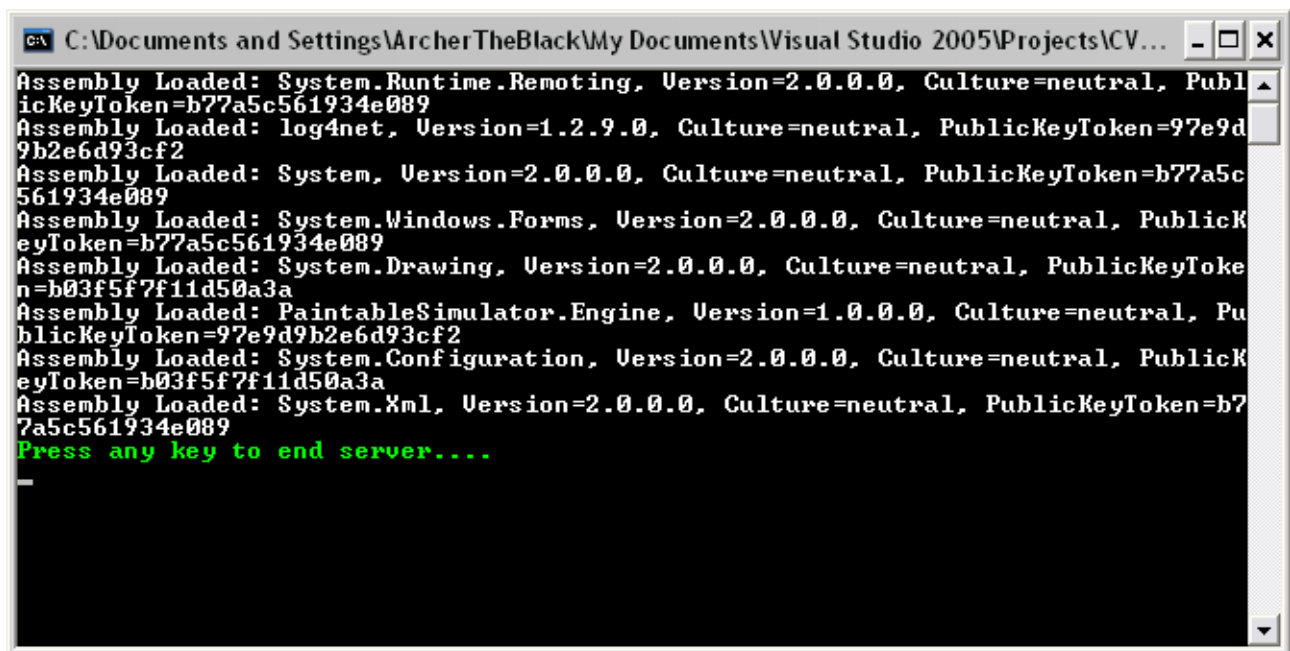
6.1 Parametri i pokretanje poslužitelja

Poslužitelj je program koji je namijenjen za komunikaciju s klijentom na način da je simulacija pokrenuta na poslužitelju, a do klijenta dolaze samo podaci bitni za prikaz simulacije. Parametri koji su bitni za komunikaciju nalaze se u datoteci `PaintableSimulator.Server.exe.config`. U njoj se nalaze parametri s kojima će poslužitelj biti pokrenut. Datoteka je u obliku XML dokumenta. Najbitniji dio te datoteke je prikazan ovdje:

```
...  
  
<PaintableSimulator.Server.Properties.Settings>  
  
    <setting name="ServerPort" serializeAs="String">  
  
        <value>8989</value>  
  
    </setting>  
  
    <setting name="ProcessFragmentsDirectory" serializeAs="String">  
  
        <value>ProcessFragments</value>  
  
    </setting>  
  
    <setting name="MachineName" serializeAs="String">  
  
        <value>localhost</value>  
  
    </setting>  
  
</PaintableSimulator.Server.Properties.Settings>  
  
...
```

U postavci `ServerPort` nalazi se vrijednost 8989 što je predefinirana vrijednost za pristup na kojem će klijent uspostaviti vezu s poslužiteljem, a u postavci `MachineName` nalazi se vrijednost `localhost`, što označava ime računala na kojem će poslužitelj biti pokrenut. Za naš primjer, koristit ćemo te predefinirane vrijednosti. Za promjene parametara navedenih u konfiguracijskoj datoteci može se koristiti bilo koji tekstualni editor.

Poslužitelj je komandno-linijska aplikacija koja se pokreće iz izvršne datoteke `PaintableSimulator.Server.exe`. Nakon podešenja poslužitelja, možemo ga pokrenuti. Pri tome će rezultat izgledati kao na slici. Izvođenje poslužitelja se prekida pritiskom na bilo koju tipku.

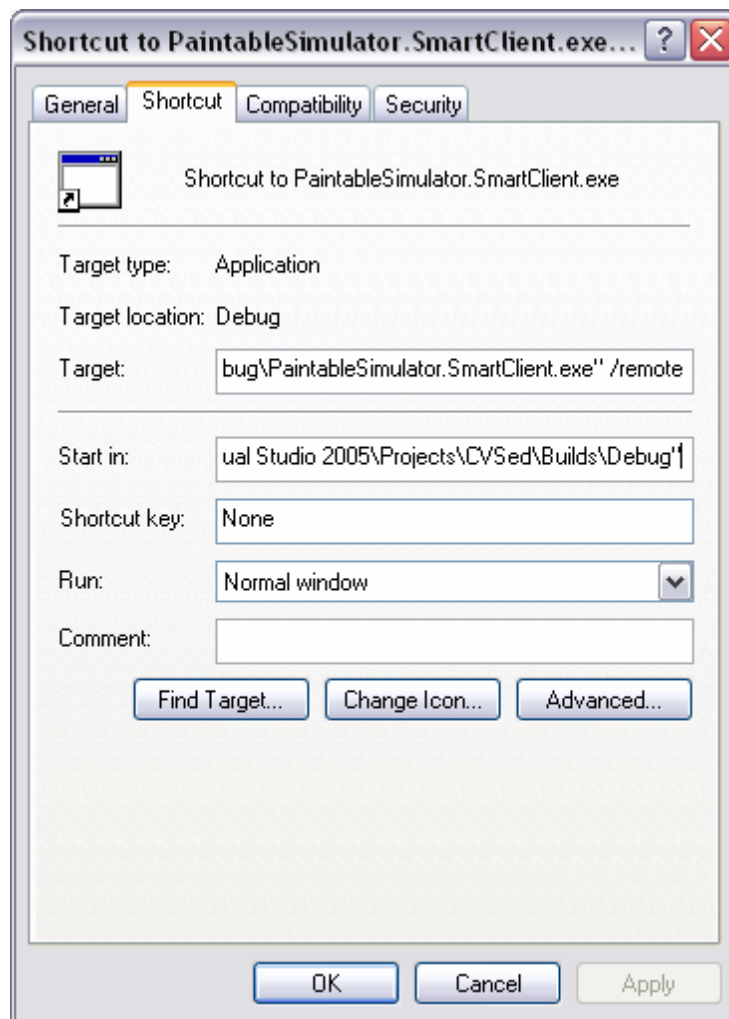


```
C:\Documents and Settings\ArcherTheBlack\My Documents\Visual Studio 2005\Projects\CV...
Assembly Loaded: System.Runtime.Remoting, Version=2.0.0.0, Culture=neutral, Publi
icKeyToken=b77a5c561934e089
Assembly Loaded: log4net, Version=1.2.9.0, Culture=neutral, PublicKeyToken=97e9d
9b2e6d93cf2
Assembly Loaded: System, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b77a5c
561934e089
Assembly Loaded: System.Windows.Forms, Version=2.0.0.0, Culture=neutral, PublicK
eyToken=b77a5c561934e089
Assembly Loaded: System.Drawing, Version=2.0.0.0, Culture=neutral, PublicKeyToke
n=b03f5f7f11d50a3a
Assembly Loaded: PaintableSimulator.Engine, Version=1.0.0.0, Culture=neutral, Pu
blicKeyToken=97e9d9b2e6d93cf2
Assembly Loaded: System.Configuration, Version=2.0.0.0, Culture=neutral, PublicK
eyToken=b03f5f7f11d50a3a
Assembly Loaded: System.Xml, Version=2.0.0.0, Culture=neutral, PublicKeyToken=b7
7a5c561934e089
Press any key to end server....
-
```

Pokrenuti poslužitelj

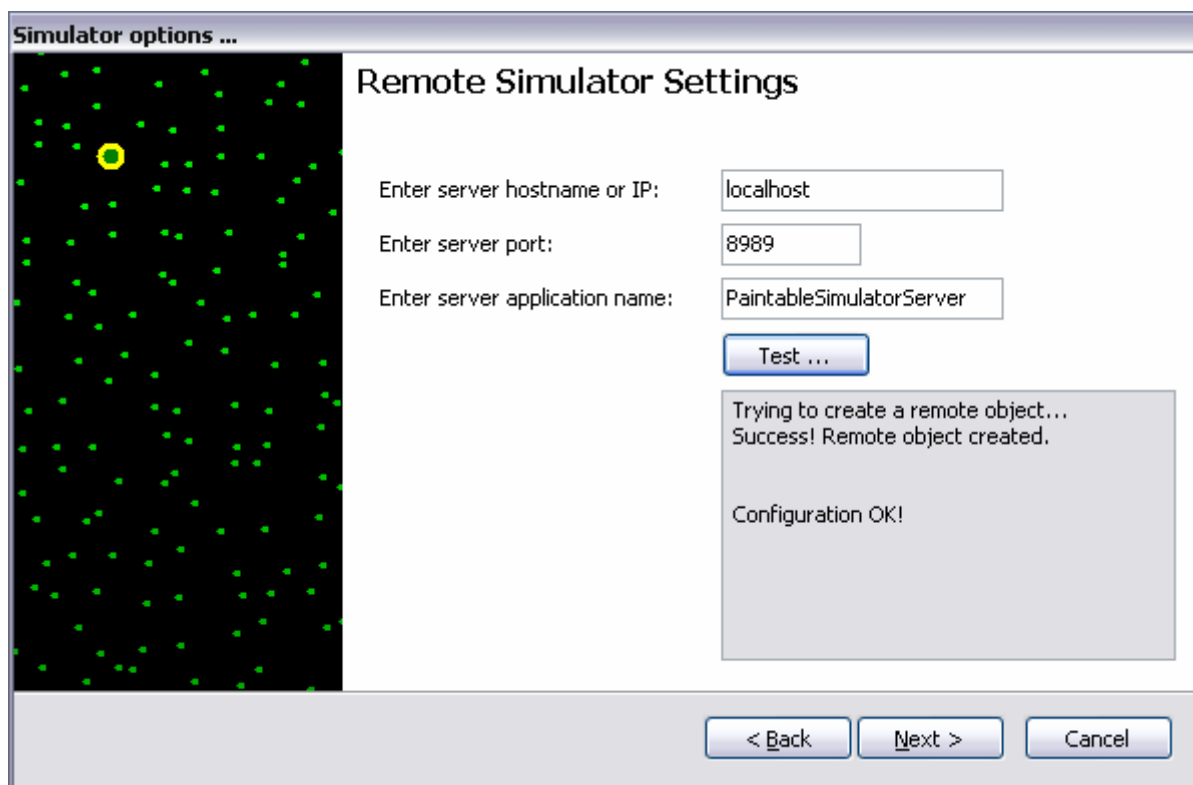
6.2 Pokretanje klijenta

Za pokretanje klijenta u okruženju klijent-poslužitelj potrebno je izvršnu datoteku PaintableSimulator.SmartClient.exe pokrenuti s prekidačem /remote. Prilično jednostavan način jest napraviti kraticu pomoću naredbe Create Shortcut. Nakon toga, potrebno je promijeniti sadržaj kratice tako da se nakon imena izvršne datoteke doda prekidač „/remote“, kao što je prikazano na slici.



Stvaranje kratice za klijenta

Nakon toga, možemo pokrenuti klijenta pomoću te kratice. Na prvom prikazanom prozoru, valja kliknuti na Next. U novoprikazanom prozoru valja podesiti postavke za spajanje na poslužitelj (one moraju biti iste kao što su definirane u konfiguracijskoj datoteci poslužitelja). Postavka Server Application Name bi trebala imati vrijednost „PaintableSimulatorServer“, što joj je i predefinirana vrijednost. Nakon toga možemo kliknuti na gumb Test koji će provjeriti stanje veze i poslužitelja.



Podešavanje klijenta

Klikom na gumb Next otvara se novi prozor u kojem se ispisiuje sažetak podešavanja, te se klikom na Finish u tom prozoru dolazi do glavnog prozora aplikacije SmartClient. Nakon toga, način rada u aplikaciji SmartClient jednak je onom opisanom u prethodnom poglavlju.

6.3 Napomena o kompatibilnosti

Postoje izvjesne situacije u kojima će ostvarenje veze klijenta prema poslužitelju biti problematično. Jedan od mogućih problema je greška prilikom testiranja postavki u podešenju klijenta, no sam klijent može se najnormalnije pokrenuti i komunicirati s poslužiteljem. Drugi mogući problem jest nemogućnost ostvarenja veze uopće. Problemi vezani uz to leže u korištenim Microsoftovim tehnologijama, gdje je uspostavljanje veze vrlo teško ili nemoguće ukoliko se bilo koja od aplikacija nalaze iza sigurnosne zaštitne stijene. Problem je ponajviše izražen ukoliko se koristi Windows 2003 Server, dok na operacijskom sustavu Windows XP ima manje problema prilikom uspostave veze.

7 Virtualno okruženje i senzori

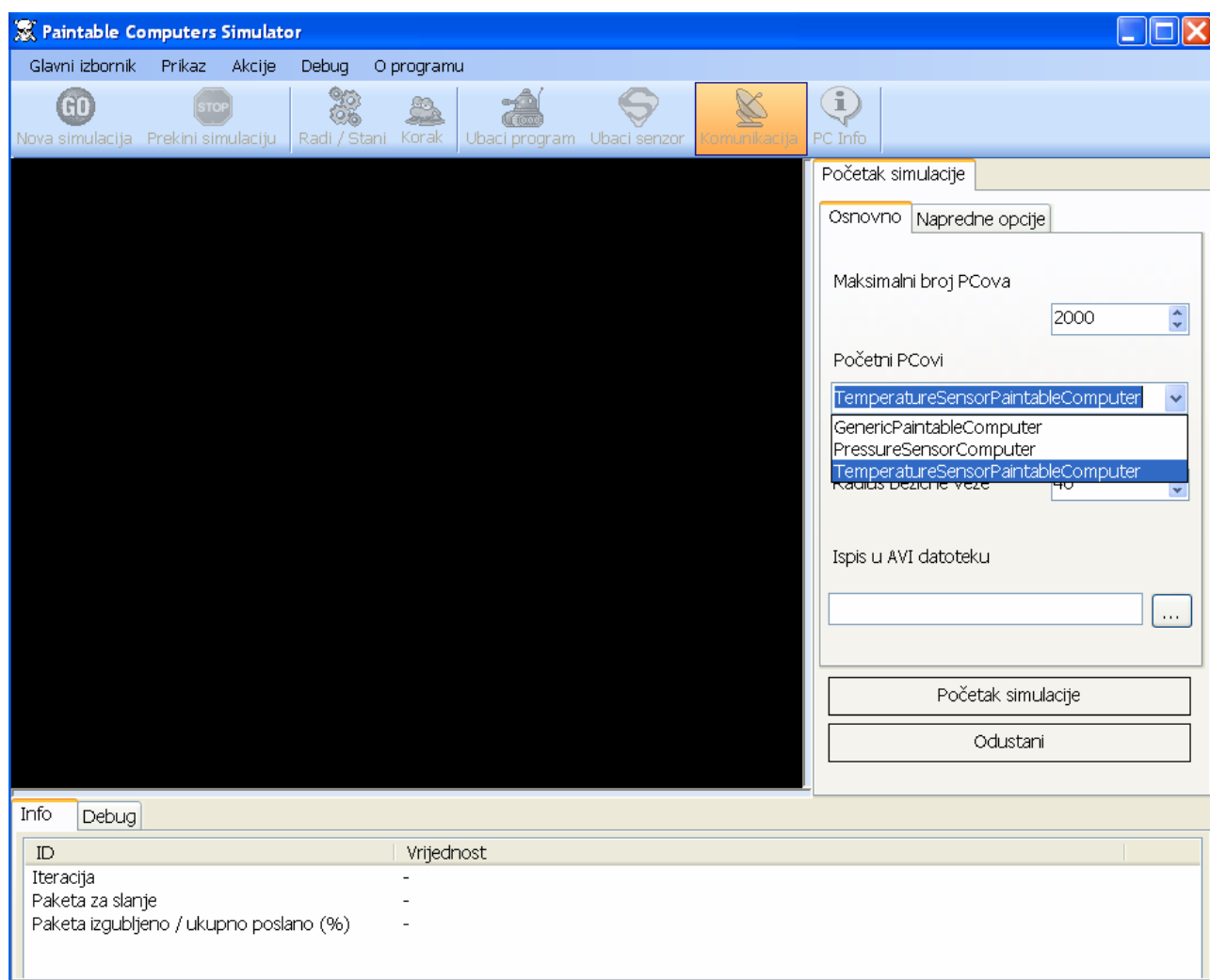
Simulator omogućava prikupljanje informacije i njihovo prikazivanje. To rade senzori. Oni su obična obojiva računala koja sadrže i posebne dijelove kojima mogu prikupljati informacije, i na osnovu toga prikazati informaciju korisniku u obliku boja.

7.1 Postavljanje senzora

Senzori se mogu postavljati pojedinačno na određena mjesta, može se određeni broj senzora pseudoslučajno rasuti po simulatoru, a može se i prilikom stvaranja nove simulacije definirati da sva obojiva računala ujedno budu i senzori.

7.1.1 Postavljanje senzora prilikom stvaranje nove simulacije

Prilikom stvaranja nove simulacije, moguće je odabrati kojom vrstom obojivih računala će se popuniti simulator. Tako je moguće popuniti simulator i sensorima. Potrebno je samo odabrati senzor za početno obojivo računalo.

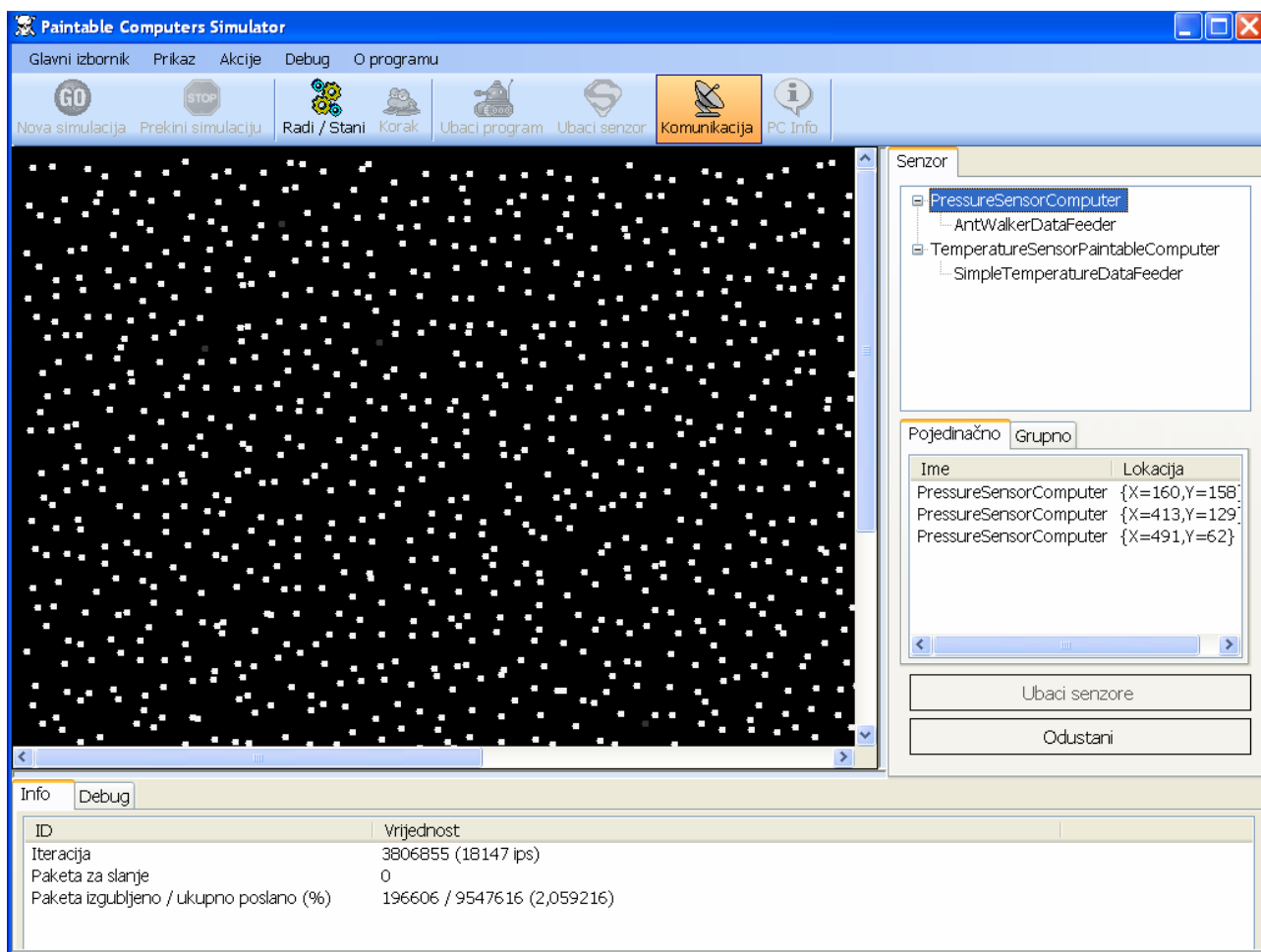


Prikaz postavljanja senzora pri stvaranju nove simulacije

7.1.2 Pojedinačno postavljanje senzora

Senzori se mogu postavljati i ručno na pozicije koje korisnik odabere. Pri tome treba naglasiti da se senzori ne mogu postavljati na mjesta gdje već postoji senzor ili obojivo računalo.

Dodavanje senzora možete započeti izbornikom (Akcije) pa izaberite (Ubaci senzor). Drugi način je da izaberete alat (Ubaci senzor) na traci s alatima. Na desno strani pojaviti će vam se opcije za ubacivanje senzora.



Prikaz pojedinačnog postavljanja senzora

Na vrhu su sam prikazani dostupni senzori i koje podatke su u stanju primati i prikazivati. Označite senzor koji želite dodati i mišem ih postavite među ostala obojiva računala. Lista senzora i odgovarajućih lokacija gdje će biti postavljeni prikazuju se u donjoj listi pod (Pojedinačno).

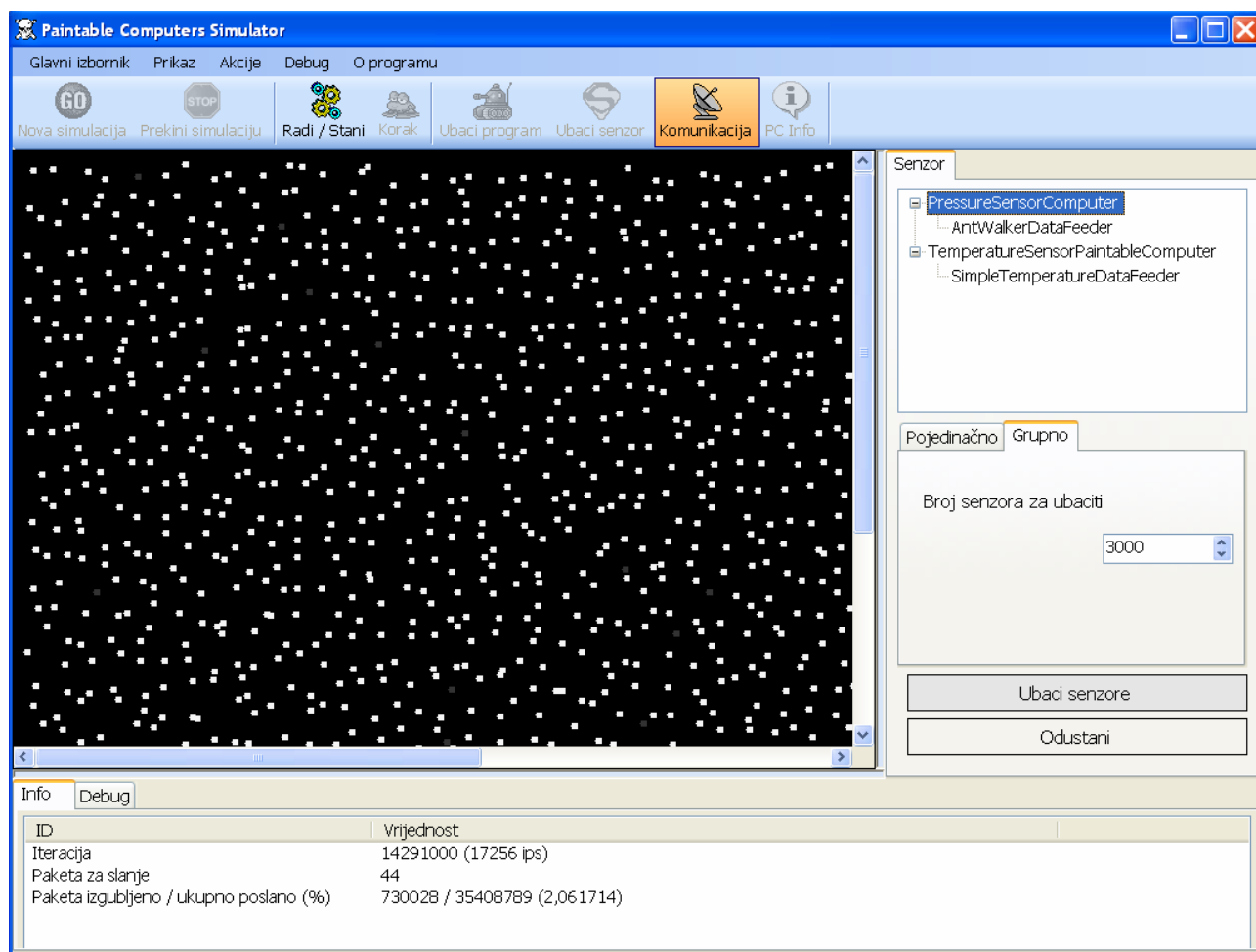
Niste ograničeni na postavljanje samo jednog senzora. Ako želite, možete označiti neki drugi senzor i njega dodati u simulator.

Kad ste završili s postavljanjem, kliknite na (Ubaci senzore) i senzori će se ubaciti u simulator. Tada će se i prikazati na ekranu.

7.1.3 Grupno postavljanje senzora

Ova opcija vam omogućuje da određeni broj senzora slučajnim uzorkom raspršite unutar simulatora.

Grupno dodavanje senzora započinje se na isti način kao i dodavanje pojedinačnih senzora. Kada se pojave opcije za postavljanje senzora, izaberite senzor i kliknite na (Grupno).

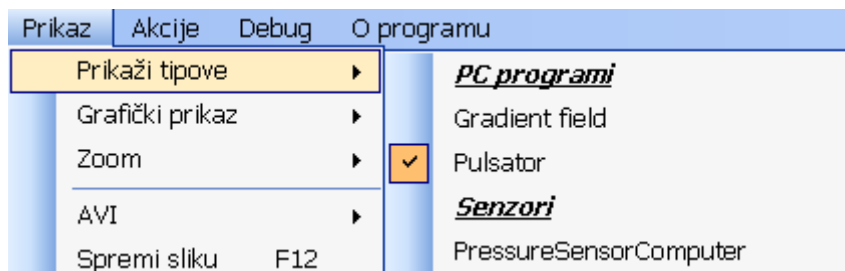


Prikaz grupnog postavljanja senzora

Unutar grupnih opcija odaberite koliko senzora želite dodati. Nakon toga kliknite na (Ubaci senzore) i pogledajte kako je senzore simulator slučajno rasporedio.

7.2 Prikaz senzora u simulatoru

Istovremeno u simulatoru možete imati više programa i senzora i ne želite gledati sve u isto vrijeme. Zato je moguće izabrati samo jednoga kojega želite promatrati. To se radi pomoću izbornika (Prikaz) te podizbornika (Prikaži tipove). Tu se smještaju svi programi i senzori koje je moguće prikazati u simulatoru.

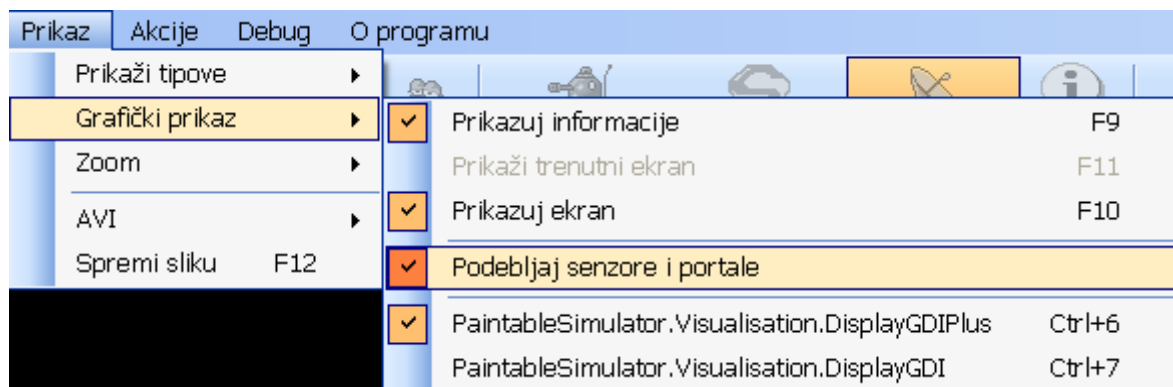


Izbornik za prikaz senzora u simulatoru

Ako želite prikazati senzor ili program, pronađite ga u ovom izborniku i označite.

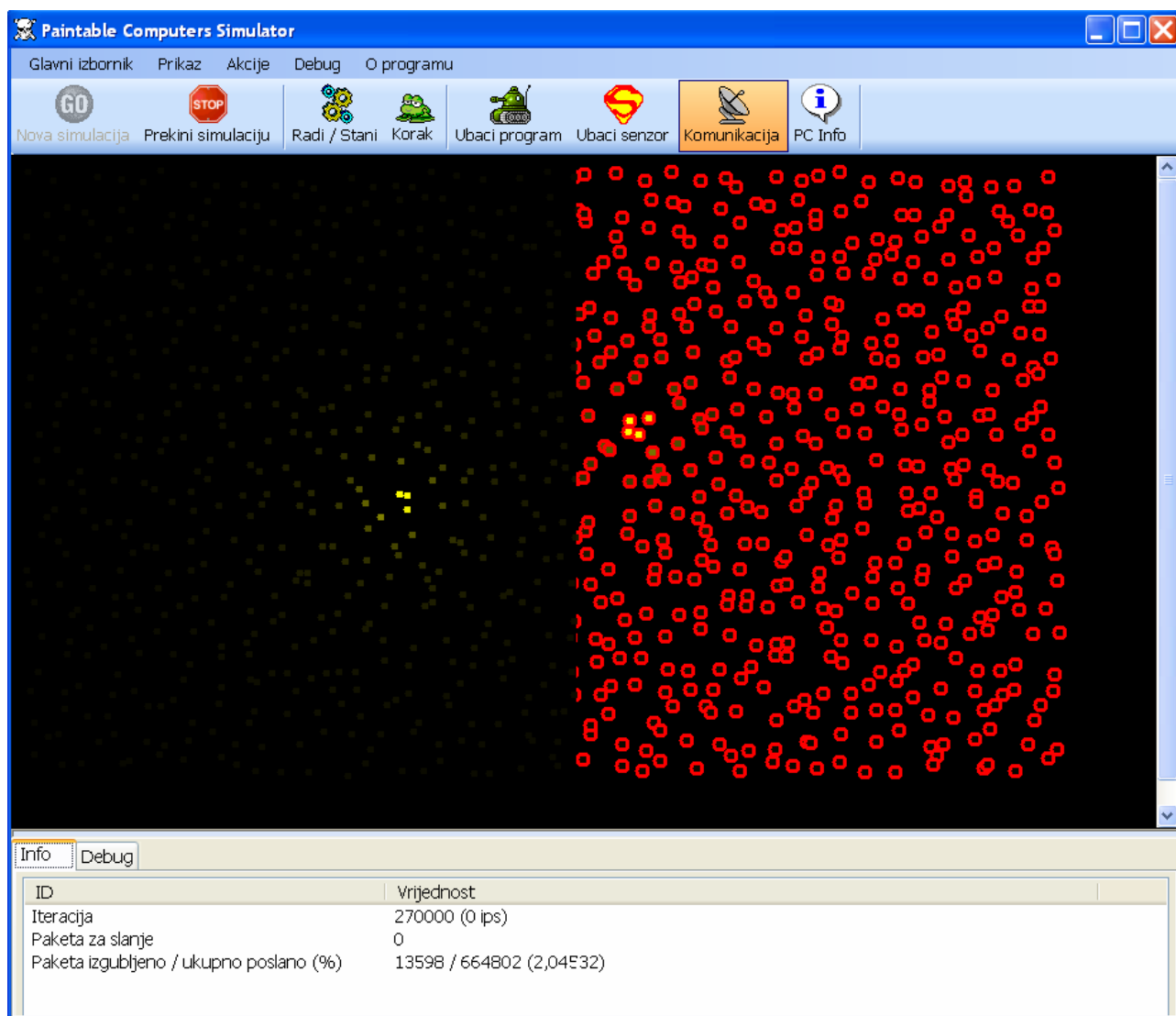
7.3 Isticanje senzora i portala

Ako nema mnogo senzora teško ih je pronaći u moru obojivih računala. Zato postoji opcija koja ističe senzore i portale programa. Zove se (Podebljaj senzore i portale) i nalazi se u izborniku (Prikaz) pa podizborniku (Grafički prikaz).



Opcija za isticanje senzora i portala

Međutim, ako ima mnogo senzora to ne izgleda lijepo pa ju je poželjno isključiti. Primjer i razliku možete vidjeti na slici ispod.



Kontraprimjer za korištenje podebljavanja senzora

8 Programi

Programi (izvorno process fragments) su zamišljeni kao minijaturne aplikacije predviđene za rad u neugodnom ali distribuiranom okolišu. Svaki program može se naseliti na računalo, može zatražiti kopiranje sebe na susjedna računala, može pisati po podatkovnoj stranici računala na kojem se nalazi, te može čitati zrcaljene podatkovne stranice susjednih računala. Programi na različitim računalima (ili unutar jednog računala) kao takvi ne mogu direktno komunicirati, nego komunikaciju moraju ostvariti preko podatkovnih stranica. Dakle, programi mogu jedino komunicirati s operacijskim sustavom, a operacijski sustav se zajedno sa sklopovljem obojivog računala brine o tome da se podatkovne stranice zrcale susjednim računalima te da se programi koji su zatražili prijenos (kopiranje) na neko drugo računalo prenesu na njega.

Programi barataju vrlo ograničenim podacima. Osnovno što im je na raspolaganju su broj obojivog računala na kojem se nalaze, te podaci o susjedima – njihove zrcaljene podatkovne stranice. Programi koji međusobno komuniciraju preko zrcaljenih podatkovnih stranica, koji je usput rečeno najefikasniji način, moraju voditi računa o kašnjenju zrcaljenja podatkovnih stranica. Programi, kao ni obojiva računala, nisu svjesni svog položaja u prostoru, tj. jedino što mogu saznati jest udaljenost nekog drugog računala u broju skokova koji su potrebni da jedan podatak prijeđe s jednog računala na drugo. Broj skokova (engl. hops) je ustvari broj potrebnih kopiranja podatka s jednog susjeda na drugog korištenjem zrcalne podatkovne stranice. Ukoliko je razmještaj obojivih računala približno homogen, broj skokova se može koristiti kao neka određena mjera udaljenosti dvaju računala, što je iskorišteno i za stvaranje koordinatnog sustava.

8.1 Sigurnost u programima

Programi koji se mogu učitati u simulator moraju biti benigni za simulator i platformu na kojoj je simulator pokrenut, što znači da ne smiju zahtijevati nikakve ulazno-izlazne operacije, niti bilo kakve druge operacije koje bi mogle rezultirati neželjenim učincima na sustavu na kojem je simulator pokrenut. Za provjeru toga se brine jedan dio simulatora koji će blokirati učitavanje i pokretanje programa koji bi mogli naštetiti simulatoru ili operacijskom sustavu. Razlog ovome sigurnosnom dijelu je mogućnost zlorabe izvođenja simulacije na poslužitelju, gdje bi zločesti korisnik mogao u svojem programu kojeg želi testirati jednostavno napisati kod za brisanje neke od sistemskih mapa. Time su ustvari programi ograničeni na svoj svijet obojivog računarstva. Kao primjer može se pokušati pokrenuti program pod imenom RealVirus (pokretanje se ipak izvodi na vlastitu odgovornost jer autori simulatora ne odgovaraju za propuste).

8.2 Parametri simulacije za navedene primjere

Za navedene primjere izuzev koordinatnog sustava biti će parametri simulacije, kao što je prikazano na slici.

| | |
|-------------------------|-------------|
| Misc | |
| Computer distance | 20 |
| Load Equalization | True |
| Maximum computer co | 1000 |
| Maximum Neighbour C | 20 |
| Simulator priority | BelowNormal |
| WallSizeRatio | 5, 5 |
| Simulation | |
| Mean time before failur | 100 |
| Simulator cycle | 500 |
| Simulator cycle deviat | 10 |
| Transmission | |
| Radio link delay | 50 |
| Transmission failure pe | 2 |
| Wireless radius | 50 |

Parametri simulacije za primjere

8.3 Program Breadcrumb

Program Virus: Ebola Prototype prikazan je već u prikazu rada simulatora te neće biti ponovno prikazan. Njegov je smisao samo da se naseli na sva moguća obojiva računala koja su u dosegu onog računala u koje je on ubačen.

Program koji se širi na sličan, djelomice pravilan način u svom procesu širenja liči na mrvicu kruha po čemu je i dobio ime (Breadcrumb). Parametar programa imena CrumbNumber određuje jačinu bojanja i opadanje intenziteta proširene mrvice kruha. Program i njegovo širenje se mogu vrlo lijepo vizualizirati ukoliko zid načinimo kvadratičnim, te sam program učitamo u jedno obojivo računala što bliže sredini tog zida.

Process Fragment Initialization

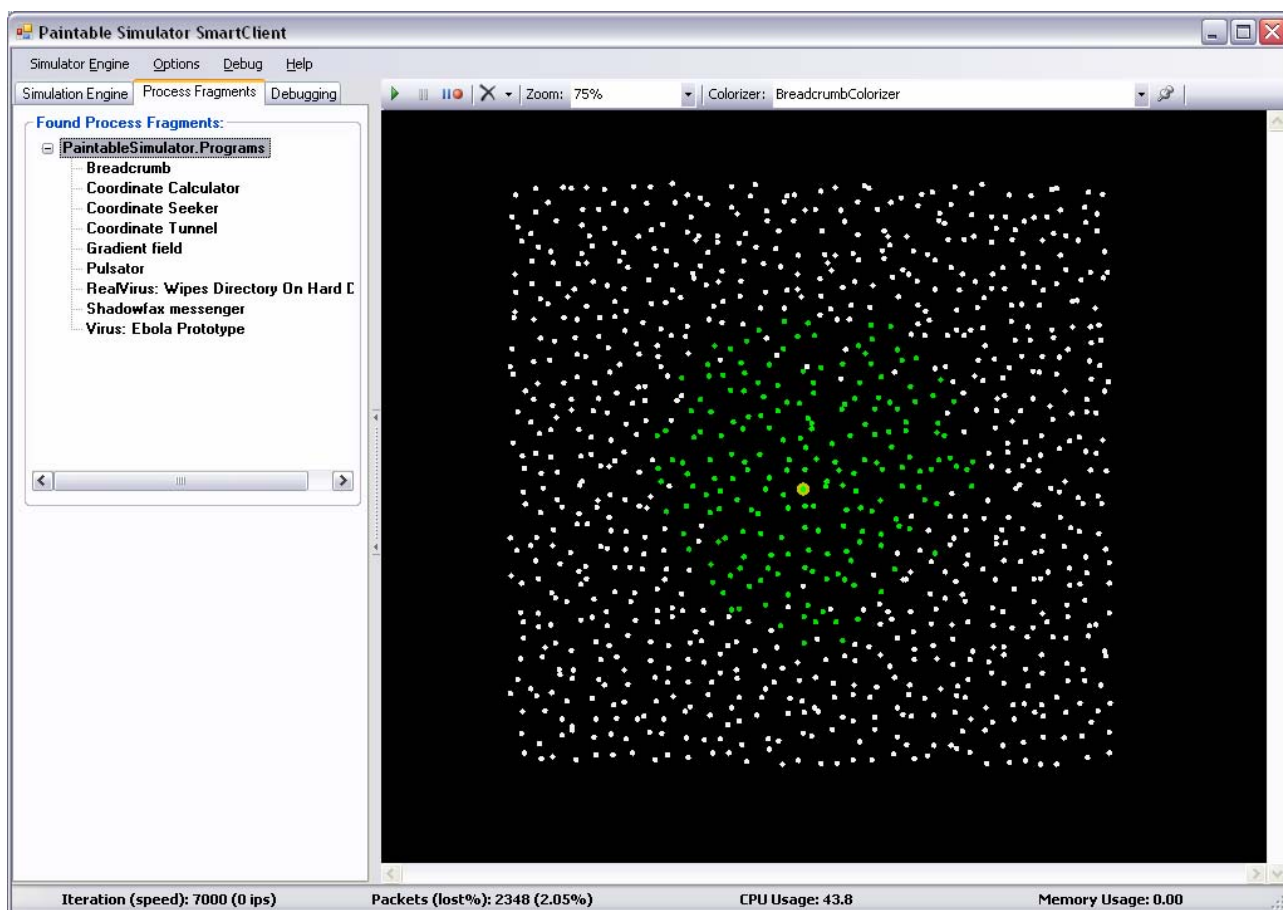
| | |
|-------------|---|
| Misc | |
| CrumbNumber | 0 |

CrumbNumber

Load

Parametri programa Breadcrumb

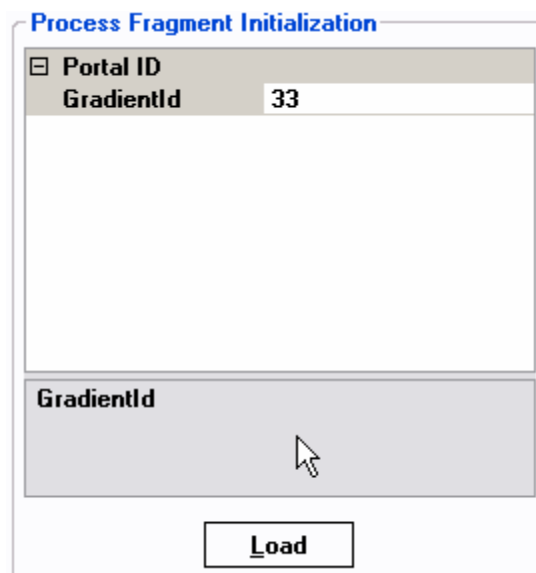
Nakon što se program učitava u jedno obojivo računalo te poslije nekoliko koraka simulacije, dobije se rezultat koji je prikazan na slici.



Rezultat nekoliko koraka simulacije programa Breadcrumb

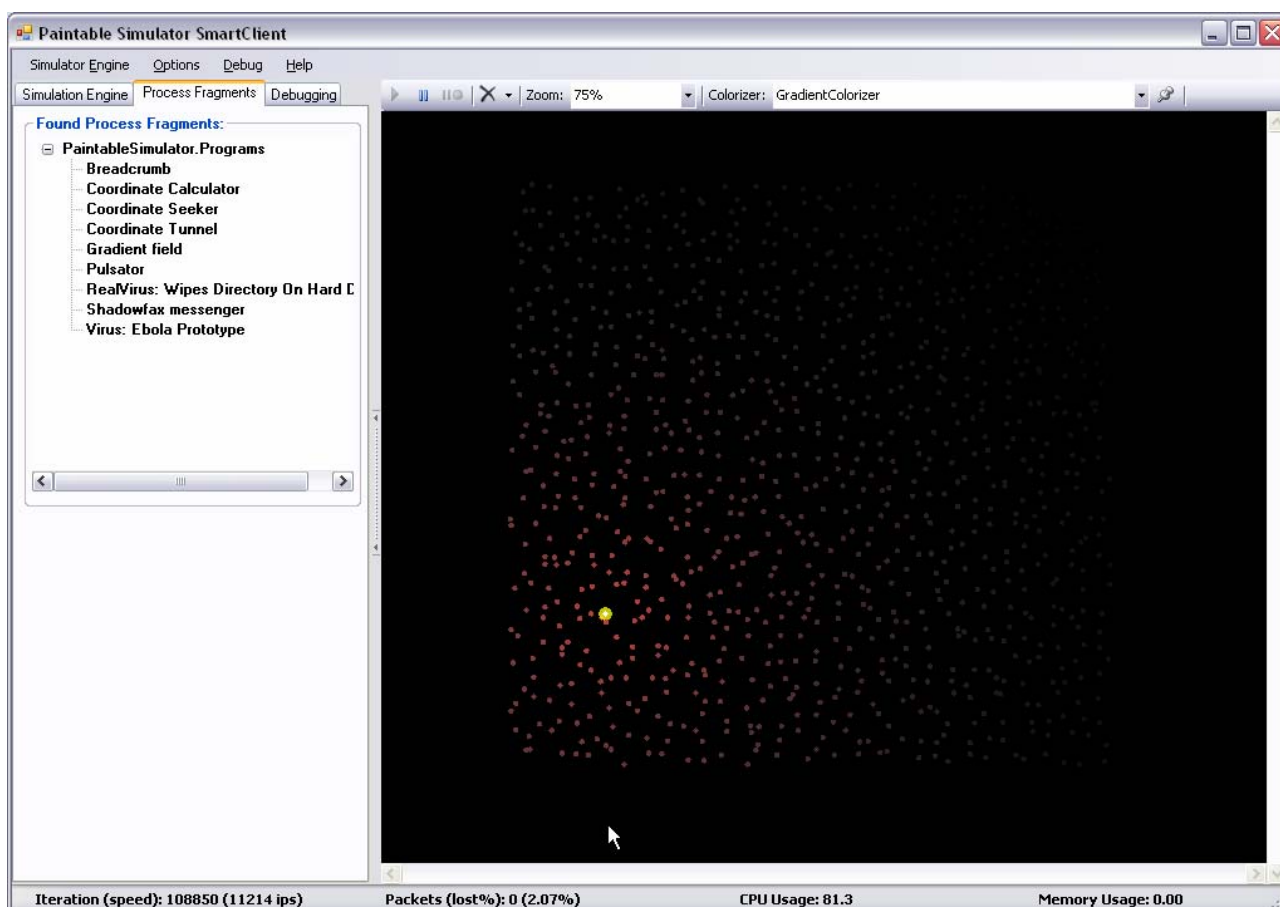
8.4 Program Gradient

Željeli bi obojati obojiva računala tako da rezultat liči na jedno svjetlo čiji se intenzitet smanjuje s udaljenošću. Upravo program Gradient služi da ostvarimo tu želju. Njegovo je svojstvo, za razliku od stvarne svjetlosti to da intenzitet opada linearno udaljenosti od izvora, pri čemu nam izvor predstavlja obojivo računalo – portal, u kojeg je program na početku učitani. Udaljenost po kojoj ovaj program računa intenzitet mjeri se u već spomenutim skokovima. Parametar GradientId koji se podešava jest identifikacijski broj pojedinog programa i on se slučajno generira, no korisnik može po želji ga promijeniti.



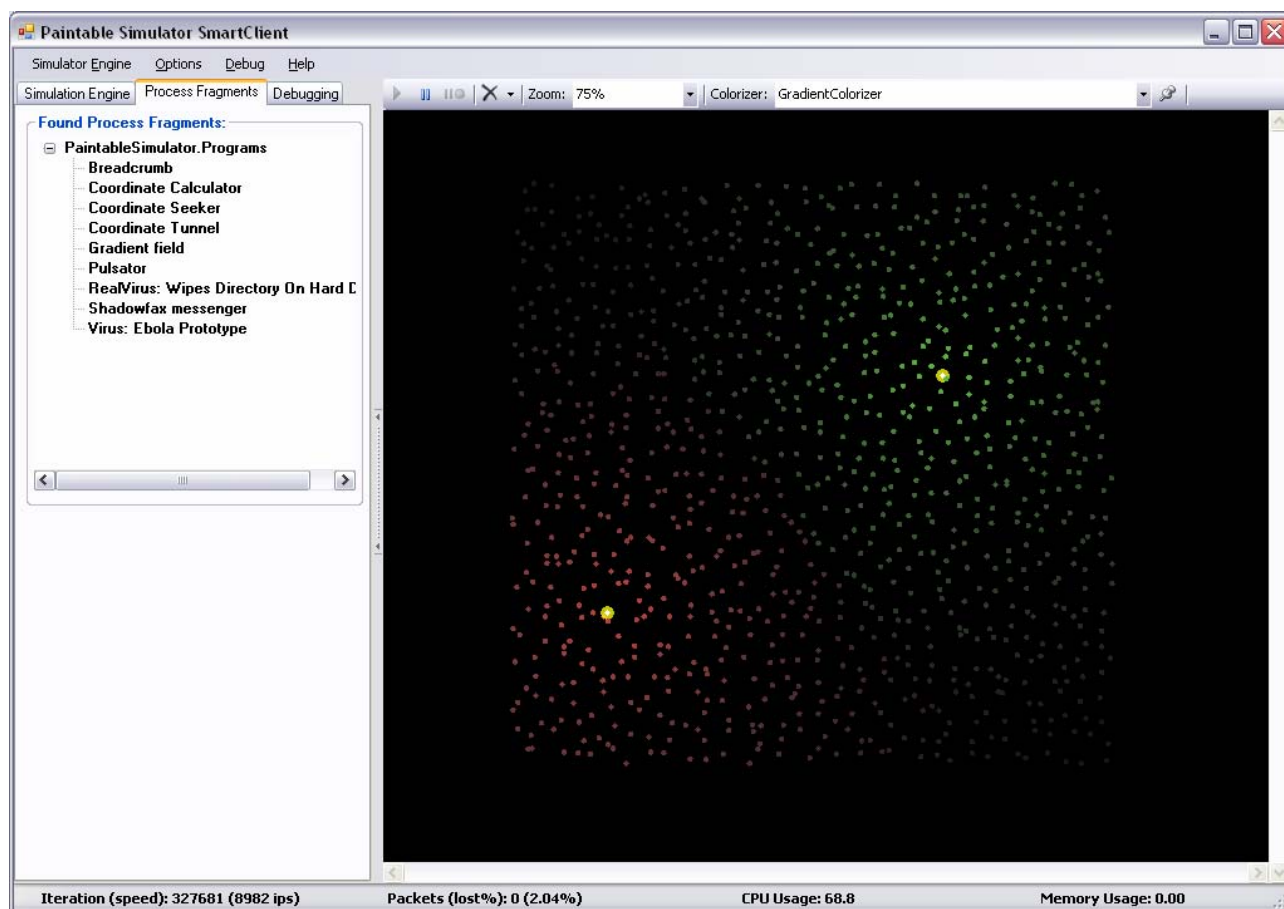
Parametri programa Gradient

Na slici je prikazan ekran nekoliko trenutaka nakon pokretanja simulacije i učitavanja programa Gradient.



Izvođenje programa Gradient

Nakon toga, možemo ubaciti još jedan program iste vrste na naš zid na neko drugo mjesto, pri čemu valja paziti da identifikacijski brojevi budu različiti. Svako računalo će se obojiti u onu boju čiji portal je bliži.



Učitavanje drugog programa Gradient

Taj program sam za sebe nije pretjerano zanimljiv, no on je ključna komponenta za ostvarenje koordinatnog sustava o kojem će biti riječi kasnije.

8.5 Program Pulsator

Krenuvši sa zanimljivim i poprilično beskorisnim primjerima, ne možemo izostaviti program Pulsator koji oponaša na neki način rasprostiranje valova u prostoru, kao što to izgleda kad bacimo kamenčić u vodu. Najbolji prikaz toga je bio obojati valne fronte u različite boje. Postavke programa koje se mogu podešavati prikazane su na slici.

Process Fragment Initialization

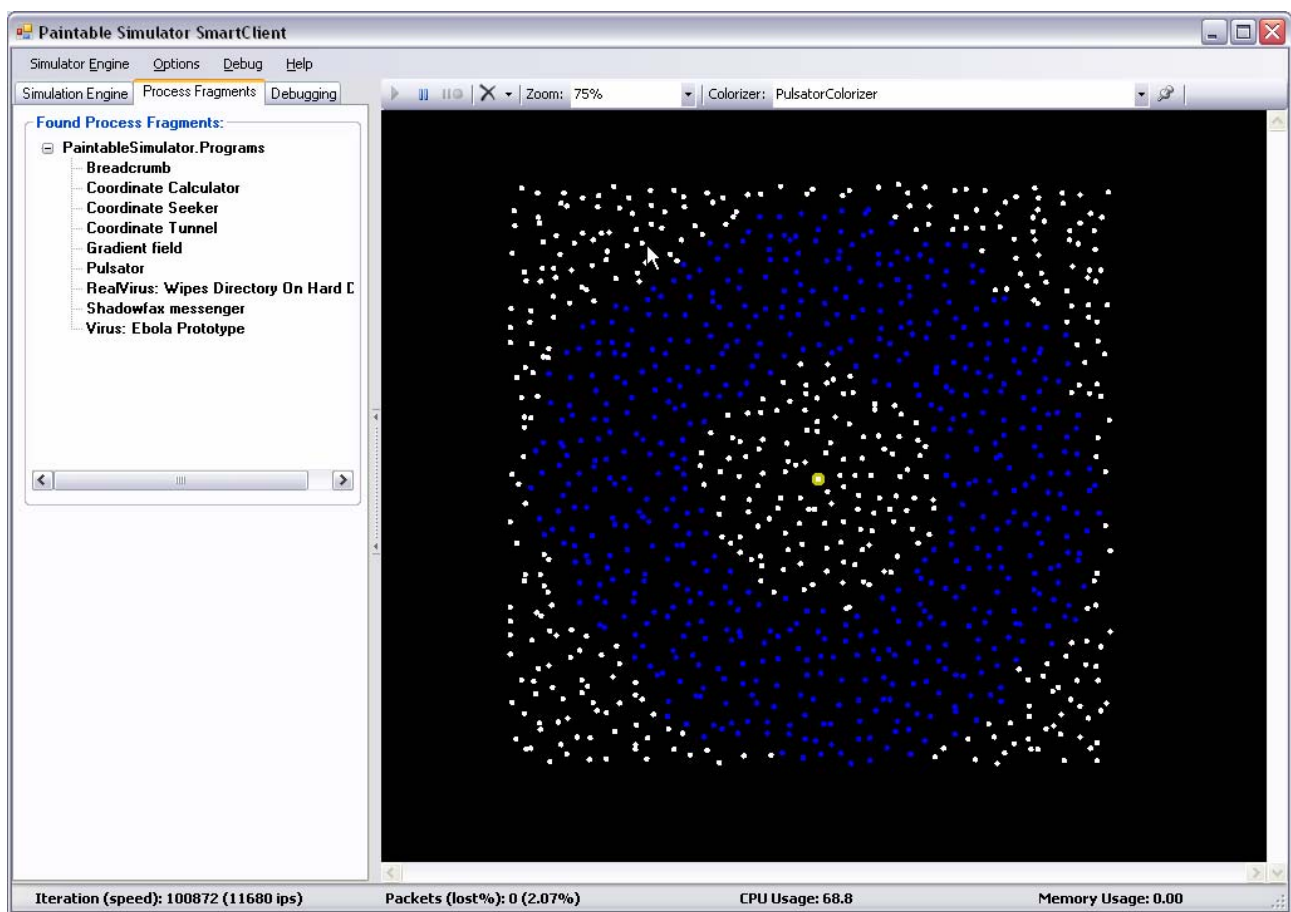
| | | |
|--|------------------|-----|
| <input type="checkbox"/> Color propagation delay | PropagationDelay | 5 |
| <input type="checkbox"/> Maximal Radius in Hops | MaxHopCount | 200 |
| <input type="checkbox"/> Pulsing frequency | PulsAfter | 25 |

MaxHopCount

Load

Postavke programa Pulsator

Postavka PropagationDelay označava vrijeme zadržavanja stare boje prije nego se nova prikaže (veći broj povećava kružni vijenac). Vrijeme se mjeri u broju pokretanja programa. Sljedeća postavka MaxHopCount se ne koristi. Posljednja postavka PulsAfter označava vrijeme mjereno u pokretanju programa koje će čekati pri generiranju sljedeće valne fronte. Rezultat izvođenja programa prikazan je na sljedećoj slici.



Program Pulsator na djelu

8.6 Program Shadowfax

A sada nešto potpuno drugačije. Jedan mnogo ozbiljniji primjer. Program je inspiriran brzim Gandalfovim konjem iz kultnog djela J.R.R.Tolkiena – Gospodar prstenova. Možda njegov najbolji opis bi bio citatom "He would not tolerate a bridle or saddle, and only carried Gandalf by choice..." Nešto slično bi se moglo reći i za ovaj program čije je ime preuzeto iz Tolkienovog remek-djela. Naime, to je program koji služi za komunikaciju između dva obojiva računala, pri čemu se brine da se poruke prenose što brže, dakle, najkraćom stazom. Uz sve to, Shadowfax nudi 64 kanala za komunikaciju čiji su redni brojevi od 0 do 63, te se staze različitih kanala mogu najnormalnije preplitati.



Shadowfax

Osnovna podjela Shadowfaksa jest na 3 načina rada. Prvi je Listener (prijemnik), drugi je Initiator (predajnik) i treći je Carrier (nosilac). Da bi natjerali stvar da proradi, moramo podesiti jednog Shadowfaksa na sljedeći način: postaviti njegovu osobnost (Personality) na vrijednost Listener, i podesiti njegovo ime. U našem primjeru njegovo ime (Name) će biti Marica, dok opcije Channel (broj kanala) i Destination (odredište) nemaju nikakvo značenje za prijemnik. Nakon odabranih opcija stavljamo taj program u jedno obojivo računalo na zidu. U našem primjeru, to će biti dolje lijevo na zidu.

Process Fragment Initialization

| | |
|---|----------|
| <input type="checkbox"/> Channel | |
| Channel | 0 |
| <input type="checkbox"/> Name | |
| Name | Marica |
| <input type="checkbox"/> Personality of Shadowfax | |
| Personality | Listener |
| <input type="checkbox"/> Sending to | |
| Destination | |
| Name | |

Load

Podešavanje prijemnika

Na neko drugo mjesto na zidu moramo postaviti predajnika. Dakle postavljamo mu osobnost na Initiator, zatim mu moramo podesiti njegovo vlastito ime (Name) gdje ćemo napisati Ivica, ime predajnika (Destination) će odgovarajuće tome biti Marica, odabrat ćemo kanal broj 6. Sad možemo postaviti taj program u neko računalo na zidu, što ćemo i napraviti. Gađat ćemo jedno računalo u donjem desnom kutu zida.

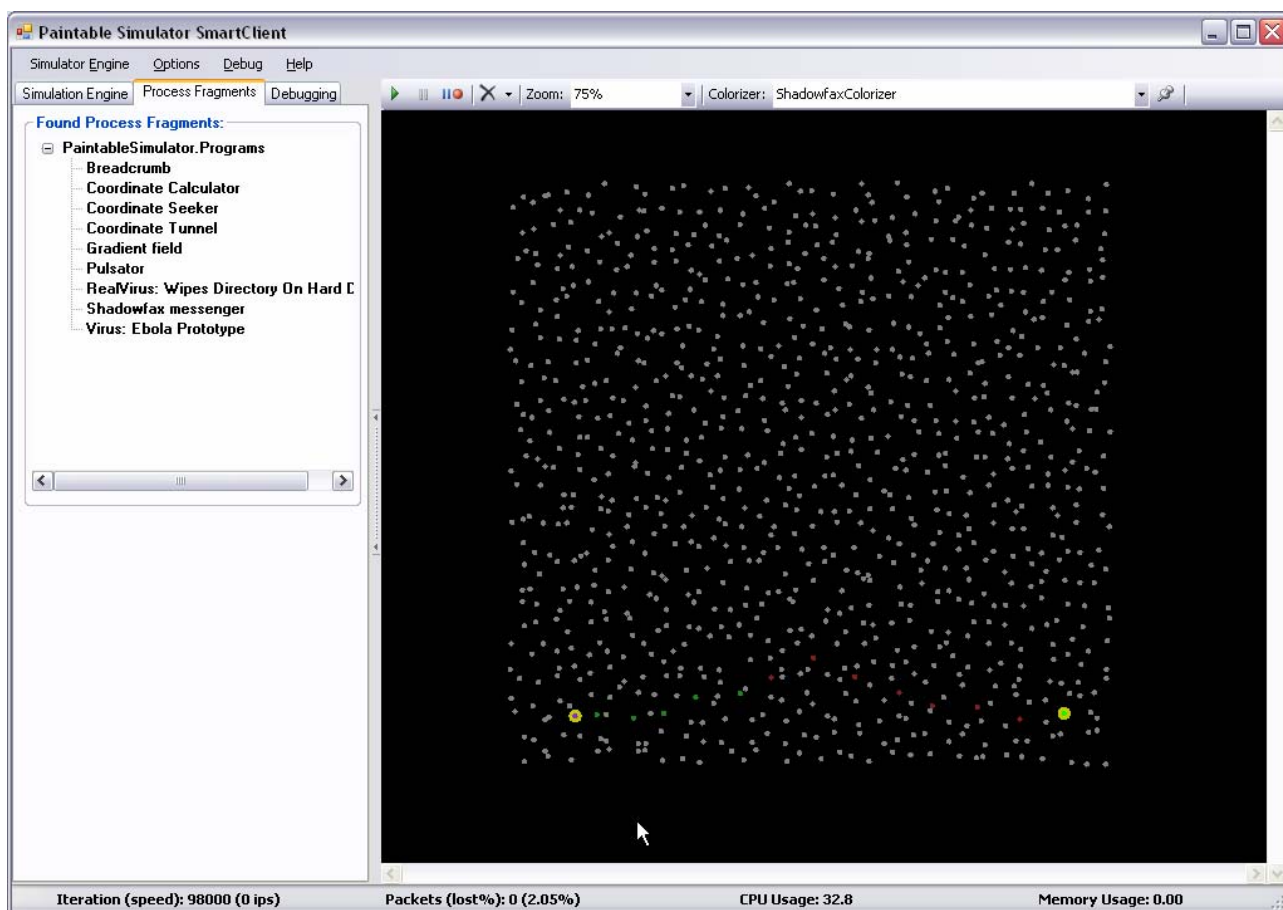
Process Fragment Initialization

| | | |
|---|-----------|--|
| <input type="checkbox"/> Channel | | |
| Channel | 6 | |
| <input type="checkbox"/> Name | | |
| Name | Ivica | |
| <input type="checkbox"/> Personality of Shadowfax | | |
| Personality | Initiator | |
| <input type="checkbox"/> Sending to | | |
| Destination | Marica | |
| | | |
| Channel | | |

Load

Podešavanje predajnika

Nakon nekoliko trenutaka Shadowfax se raširio po cijelom zidu tražeći najbolju stazu. Prijemnik neko vrijeme čeka i skuplja informacije o najboljoj stazi, te pri isteku tog vremena, odabire najbolju koju je primio, te počinje slati odgovor odabranom najboljom stazom. Predajnik po primitku te poruke isto zna najbolju stazu te je koristi za slanje svoje poruke prijemniku i tako dalje. Jedna zanimljiva stvar koju valja spomenuti je ta da nigdje nismo ubacivali program nosilac (Carrier). Za njegovo se ubacivanje pobrinuo sam predajnik (Initiator). Na slici je primjer uspostavljene komunikacije između Ivica i Marice, gdje crvena obojiva računala označavaju put gdje je prošla poruka od Ivica prema Marici, a zelena obojiva računala predstavljaju putovanje Maričinog odgovora Ivici.



Uspostavljena komunikacija

Nakon ove komunikacije, možemo još pokušati isprobati dodavanje jednog para prijemnika i predajnika, čije će postavke biti kao ove prikazane na slici.

Process Fragment Initialization

| | |
|---|----------|
| <input type="checkbox"/> Channel | |
| Channel | 0 |
| <input type="checkbox"/> Name | |
| Name | Bob |
| <input type="checkbox"/> Personality of Shadowfax | |
| Personality | Listener |
| <input type="checkbox"/> Sending to | |
| Destination | |
| Destination | |
| <input type="button" value="Load"/> | |

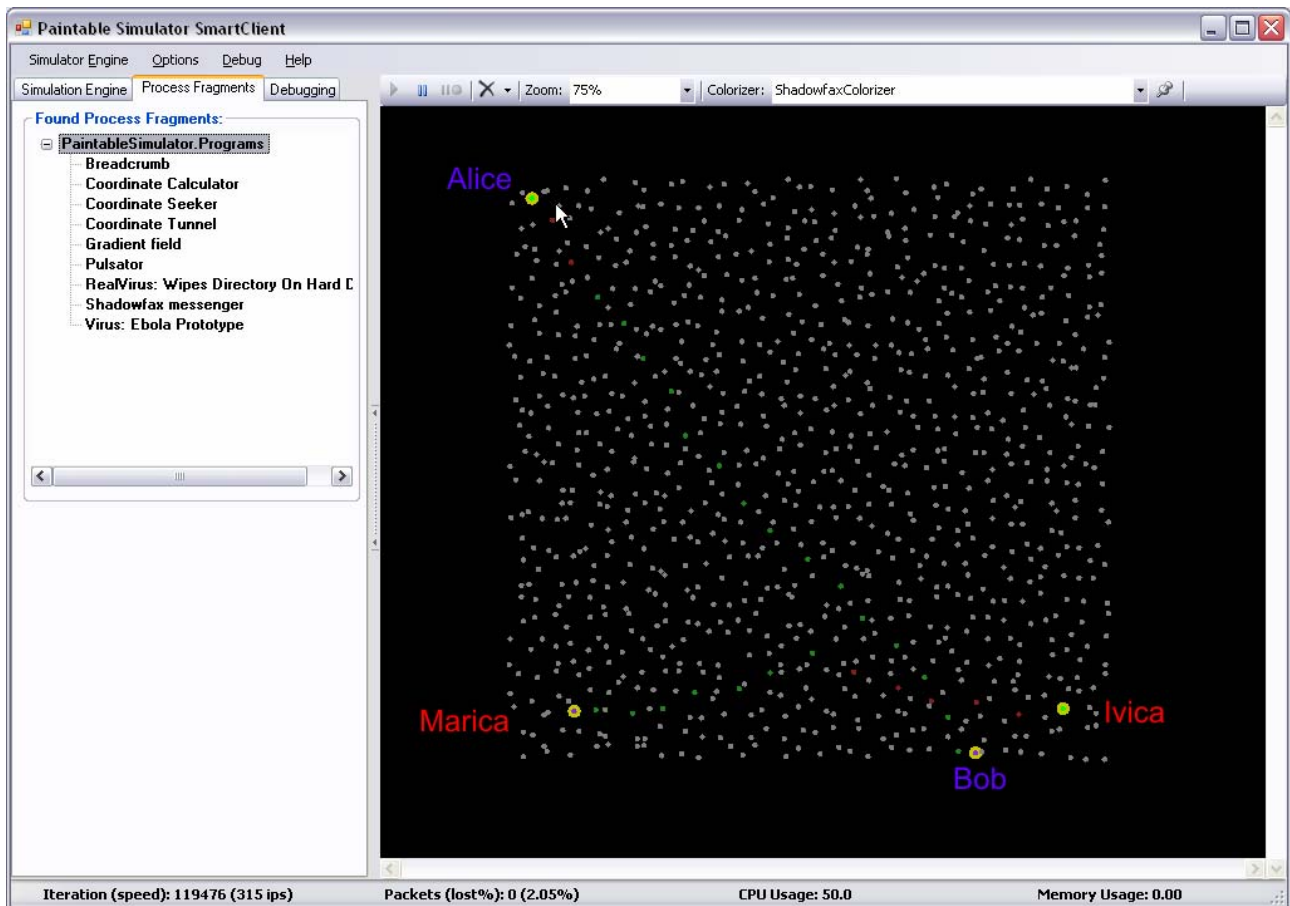
Process Fragment Initialization

| | |
|---|-----------|
| <input type="checkbox"/> Channel | |
| Channel | 13 |
| <input type="checkbox"/> Name | |
| Name | Alice |
| <input type="checkbox"/> Personality of Shadowfax | |
| Personality | Initiator |
| <input type="checkbox"/> Sending to | |
| Destination | Bob |
| Channel | |
| <input type="button" value="Load"/> | |

Postavke drugog para prijemnika i predajnika

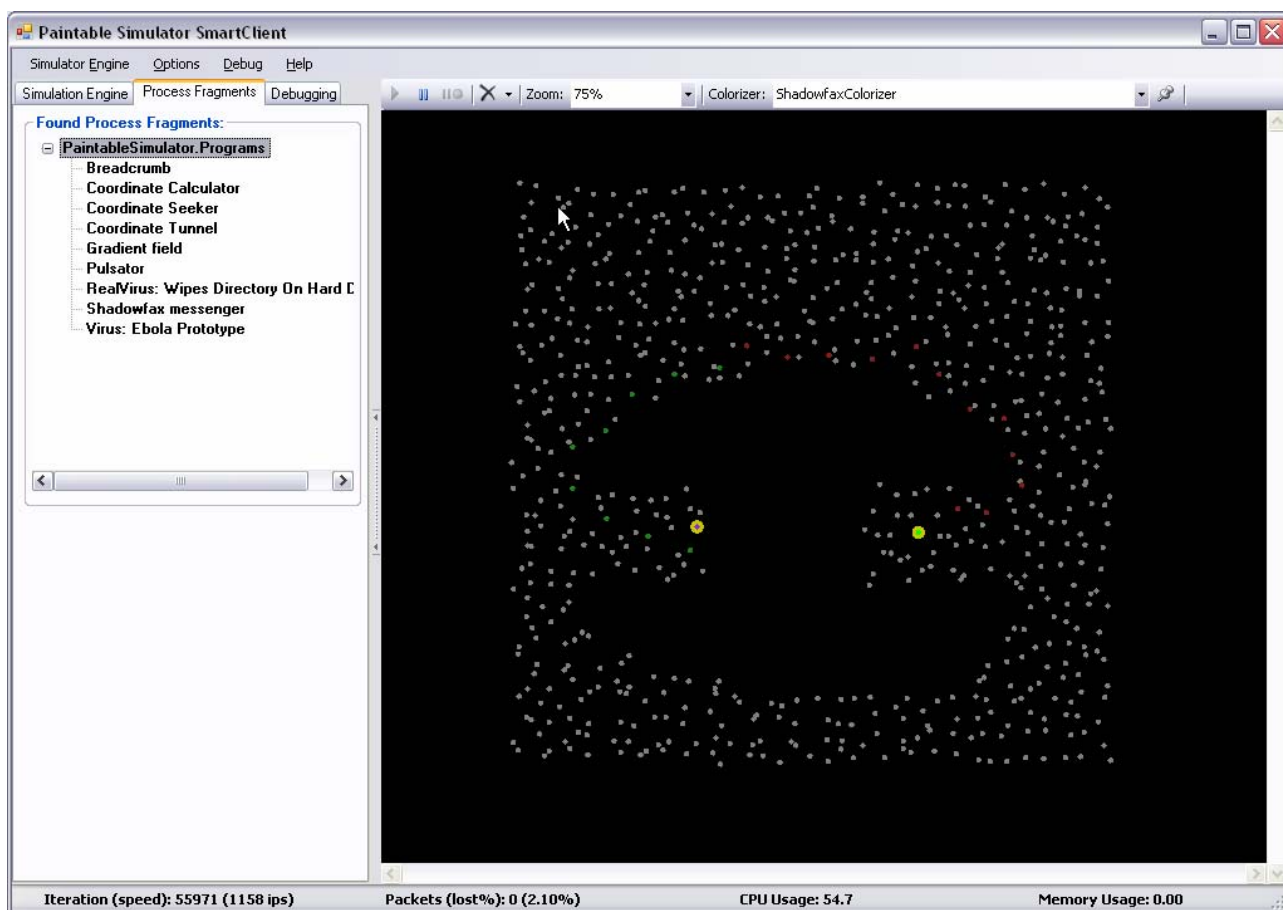
Prijemnik ćemo smjestiti u donji desni dio ekrana, ne jako daleko od prethodnog prijemnika, a predajnik ćemo smjestiti u gornji lijevi dio ekrana. Nakon nekog vremena,

izgled ekrana će biti kao na slici, gdje su i označena imena prijemnika i predajnika (ne, to simulator ne ispisiše, to je ipak samo korištenje alata za obradu slike).



Komunikacija dva para prijemnika i predajnika na dva kanala

Na kraju ćemo odabrati jedan ponešto egzotičan primjer gdje ćemo obrisati određena računala sa zida, te ćemo stvoriti jedan par prijemnik-predajnik kao u prethodnom primjeru. Jedino što je nejasno sa slike je li odabrani put uistinu najkraći put? Autoru ovog programa to definitivno ne izgleda kao najkraće, ali svakako djeluje dovoljno dobro.



Egzotičan primjer programa Shadowfax

8.7 Koordinatni sustav

Negdje prije u tekstu je spomenuto kako programi u obojivim računalima znaju samo svoje međusobne udaljenosti (tj. mogu saznati ukoliko se napišu takvima, no vidjeli smo da nam gradijent služi za to), znaju identifikacijski broj računala na kojem se nalaze i sposobni su preko zrcaljenih podatkovnih stranica komunicirati sa susjedima. Da stvar još više izdramatiziramo, reći ćemo da nam identifikacijski broj nije nešto što ćemo koristiti, dapače, on nam je samo višak za izgradnju koordinatnog sustava.

8.7.1 Trikovi i Gradient na steroidima

U cijelom postupku postavljanja koordinatnog sustava postoje samo 2 vrlo jednostavna trika. Korisnik mora za prve dvije točke koordinatnog sustava reći njihove prave koordinate kao npr. (0, 0) i (4,3), ubaciti ih na neka dva obojiva računala na zidu, te na kraju, samo postaviti treću točku koordinatnog sustava tako da nije kolinearna s prve dvije navedene. Gore opisani program Gradient jest taj koji je sposoban računati udaljenosti, no u ovom slučaju on nam je samo baza cijelog problema, pa se za ovaj problem koristi Coordinate Tunnel koji je ništa više nego poboljšani Gradient. Njegove postavke su točka koju on predstavlja u koordinatnom sustavu (StartPoint), zatim njegov tip (Tip) koji može biti prvi, drugi ili treći (First, Second, Third respektivno). Na kraju ostaju 2 identifikacijska broja gradijenta koje u pravilu nije potrebno mijenjati.

Za prve dvije točke (First i Second), potrebno je upisati koordinate koje će nam odrediti izgled konačnog koordinatnog sustava. Za treću točku, koordinate se automatski računaju pri čemu od dvije moguće točke (u općemu slučaju se kružnice sijeku u dvije točke), program izabire „bolju“. Nakon što postavimo ta 3 programa u obojiva računala na zid, iskoristit ćemo program imenom Coordinate Calculator koji će za svako računalo više ili manje uspješno izračunati njegove koordinate u našem koordinatnom sustavu. Točke s koordinatama manjima od nule se ne bojaju, a ostale se bojaju u diskretnim koracima koji su određeni s cjelobrojnim dijelom svake od koordinata pojedinog obojivog računala.

8.7.2 Postavljanje 3 programa Coordinate Tunnel

Za ovaj primjer, povećat ćemo broj računala koja ćemo koristiti u simulaciji kako bi naš koordinatni sustav izgledao što pravilnije, stoga, valja postavke simulacije namjestiti kao što je to prikazano na slici.

| GenericPaintableComputer | |
|--------------------------|-------------|
| Misc | |
| Computer distance | 20 |
| Load Equalization | True |
| Maximum computer co | 4000 |
| Maximum Neighbour C | 20 |
| Simulator priority | BelowNormal |
| WallSizeRatio | 1, 1 |
| Simulation | |
| Mean time before failur | 100 |
| Simulator cycle | 500 |
| Simulator cycle deviat | 10 |
| Transmission | |
| Radio link delay | 50 |
| Transmission failure pe | 2 |
| Wireless radius | 50 |

Postavke simulacije

Napravit ćemo jedan vrlo jednostavan koordinatni sustav. U donji lijevi dio zida ubacit ćemo jedan program Coordinate Tunnel čija će točka biti (0,0) i bit će prvi (First). U gornji desni rub ubacit ćemo program Coordinate Tunnel čija će točka biti (5,5) a bit će drugi (Second). Na kraju, ubacit ćemo treći Coordinate Tunnel (Third) negdje u gornji lijevi dio ekrana. Na slici su prikazane postavke sva 3 programa za ubacivanje.

Process Fragment Initialization

| | |
|--------------------|-------|
| Misc | |
| StartPoint | 0, 0 |
| Tip | First |
| Portal ID's | |
| GradientId1 | 82 |
| GradientId2 | 82 |

StartPoint

Load

Process Fragment Initialization

| | |
|--------------------|--------|
| Misc | |
| StartPoint | 5, 5 |
| Tip | Second |
| Portal ID's | |
| GradientId1 | 23 |
| GradientId2 | 23 |

StartPoint

Load

Process Fragment Initialization

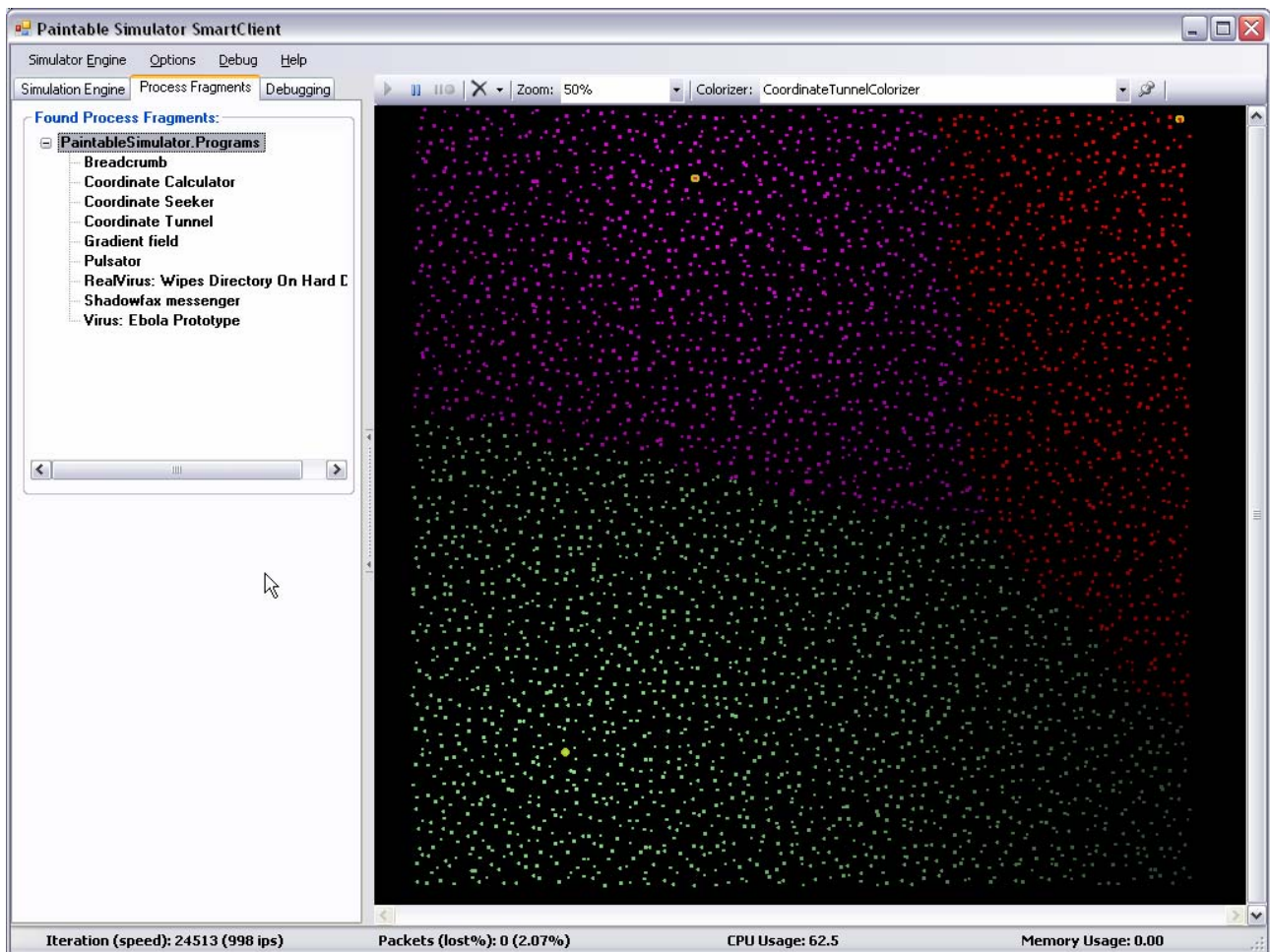
| | |
|--------------------|-------|
| Misc | |
| StartPoint | 0, 0 |
| Tip | Third |
| Portal ID's | |
| GradientId1 | 50 |
| GradientId2 | 50 |

Tip

Load

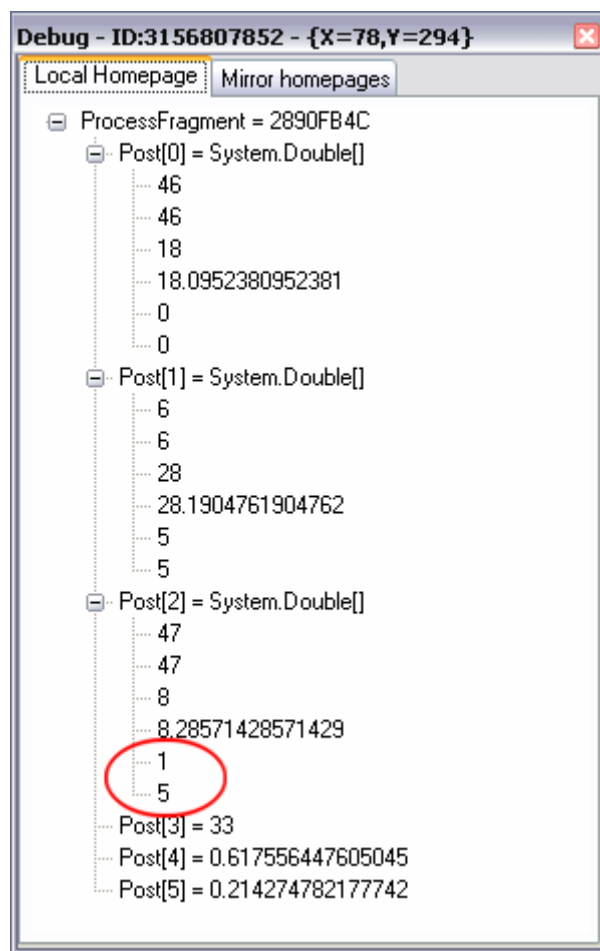
Postavke sva 3 programa Coordinate Tunnel

Nakon ubacivanja sva 3 programa, naš zid će poprimiti izgled poput ovog na slici.



Izgled zida nakon ubacivanja sva 3 programa Coordinate Tunnel

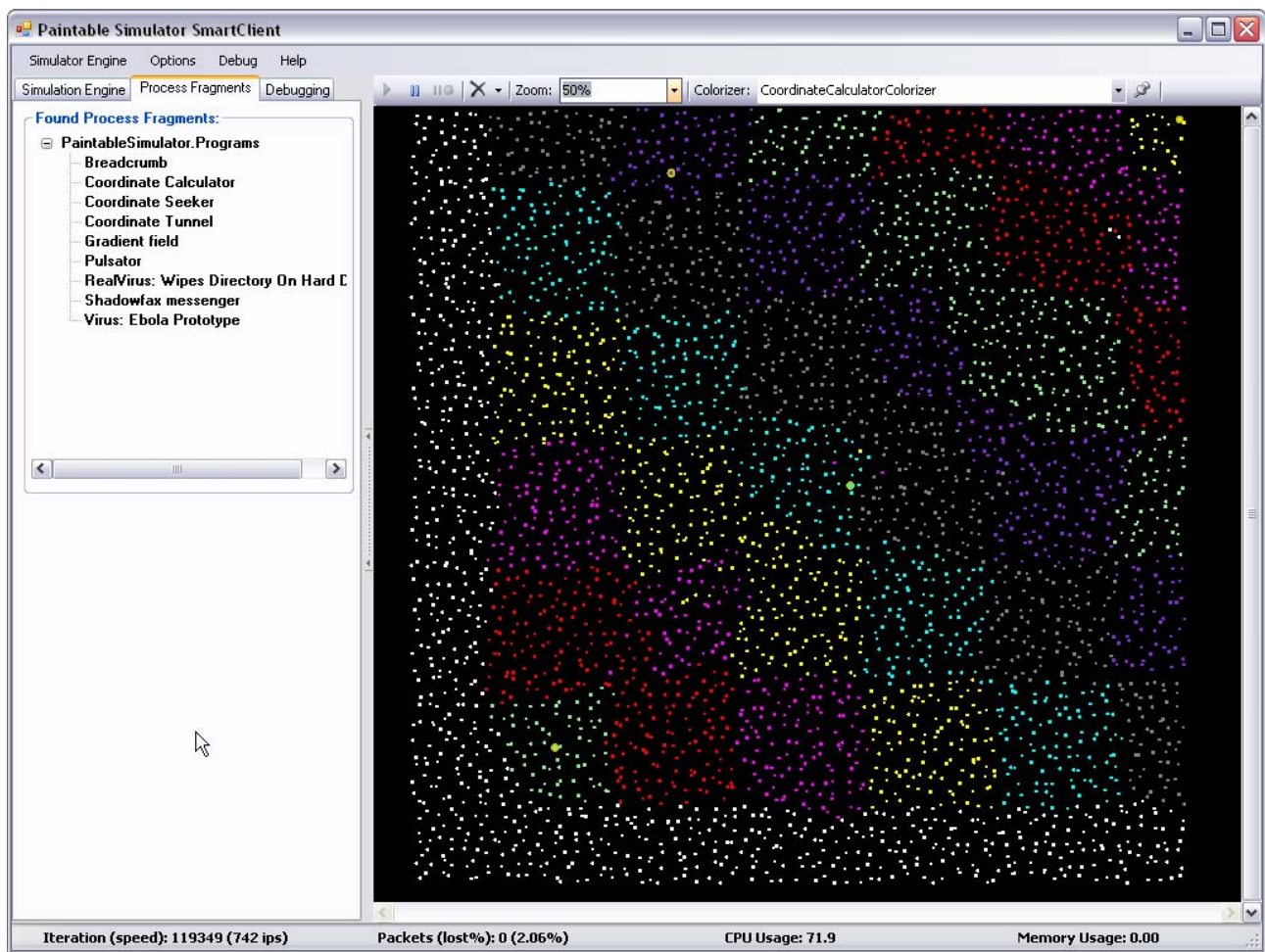
U prozoru za Debug, možemo pogledati koordinate treće točke koju je izračunao program Coordinate Tunnel. Koordinate (1,5) čine se dobrima za treću točku.



Prozor Debug i koordinate treće točke.

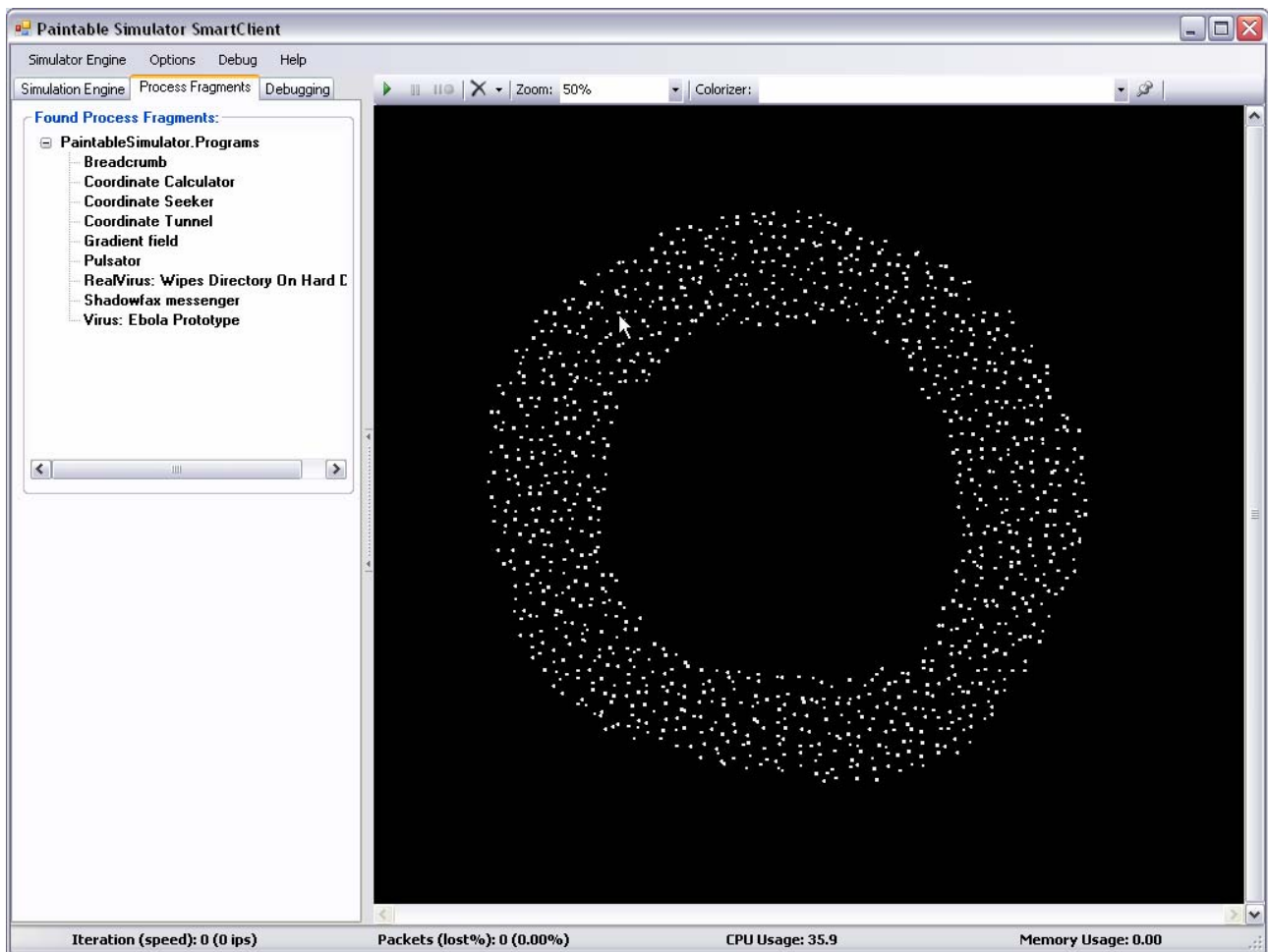
8.7.3 Bojanje koordinatnog sustava

Sada je u našu simulaciju potrebno ubaciti jedino program koji će za svako obojivo računalo odrediti njegove koordinate u razapetom koordinatnom sustavu te obojati svaku od tih različitih koordinata u neku boju. Program koji trebamo učitati zove se Coordinate Calculator i nema parametara za podešavanje. Mi ćemo ga ubaciti negdje blizu sredine zida. Rezultat bojanja vidljiv je na sljedećoj slici.



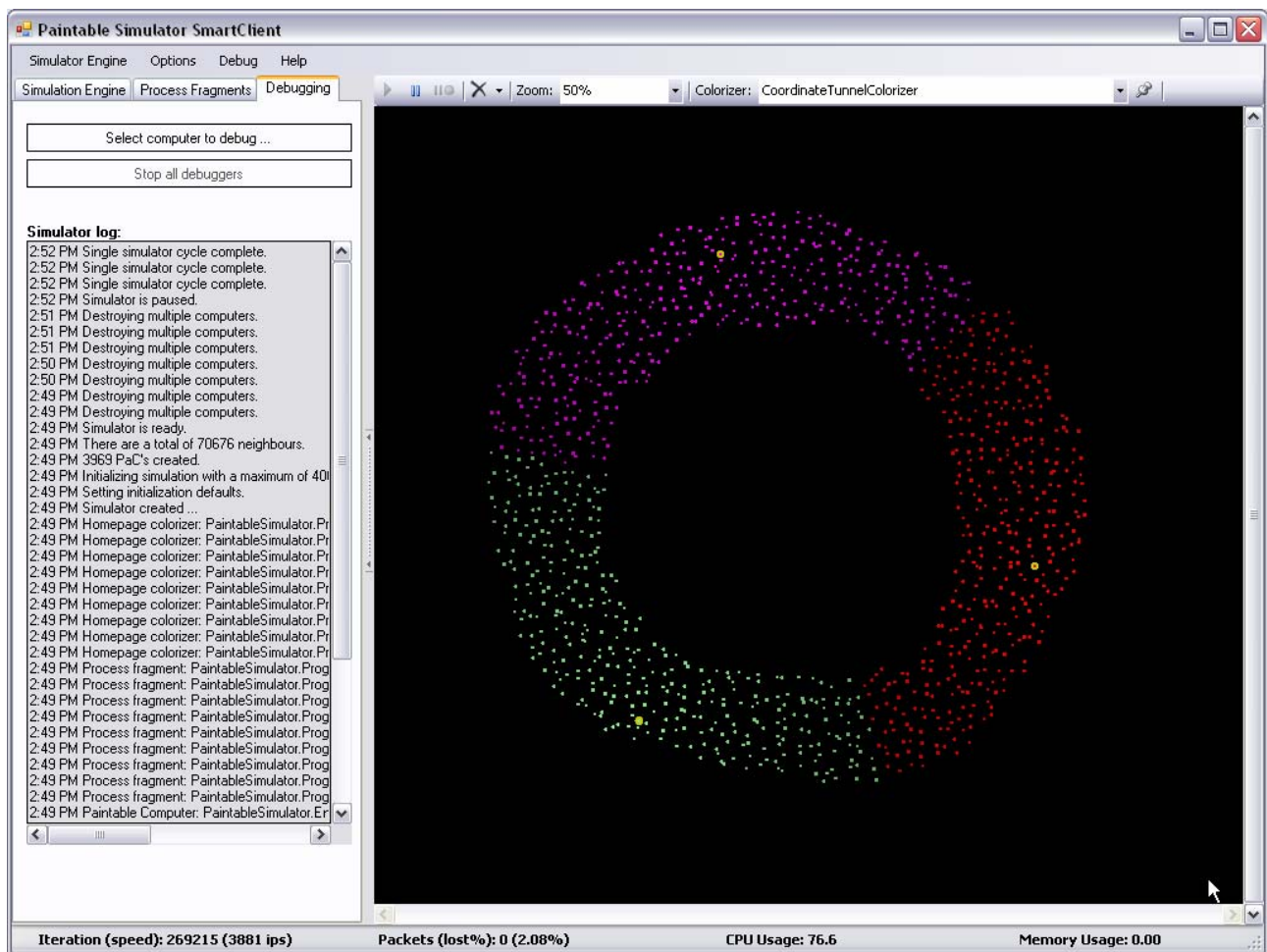
Obojani koordinatni sustav

8.7.4 Egzotični primjeri koordinatnog sustava



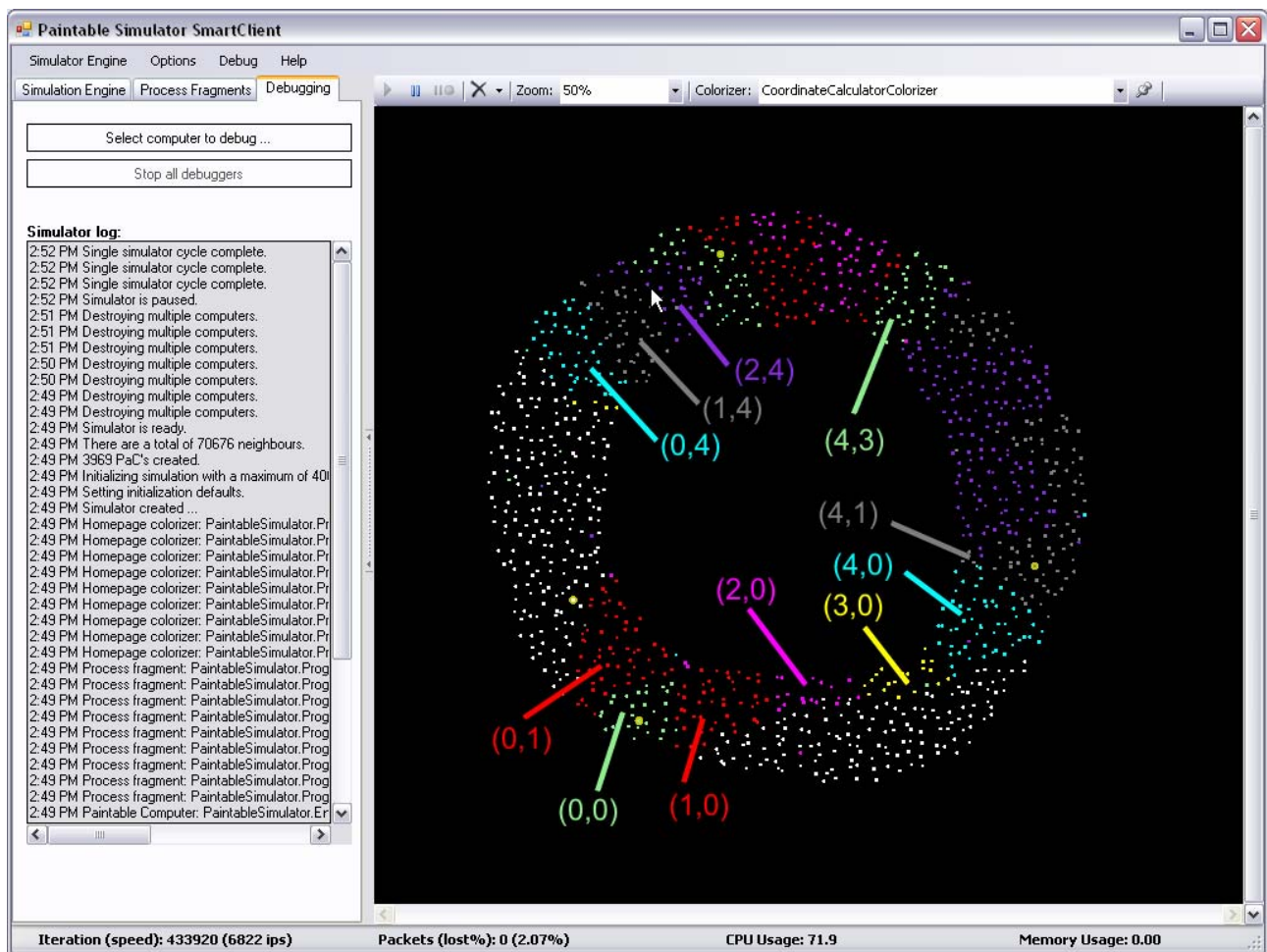
Izrezani zid

Na početku ćemo iz zida alatima za rezanje napraviti nešto što liči na automobilsku gumu. Nakon toga ćemo ubaciti 2 točke sa koordinatama $(0,0)$; $(5,0)$ i treću da se sama pronađe. Na slici su prikazana mjesta ubacivanja pojedine točke (prva točka je portal u zelenom području, druga točka je portal u crvenom području i treća je portal u ljubičastom području). Treća točka je poprimila koordinate $(3,5)$.



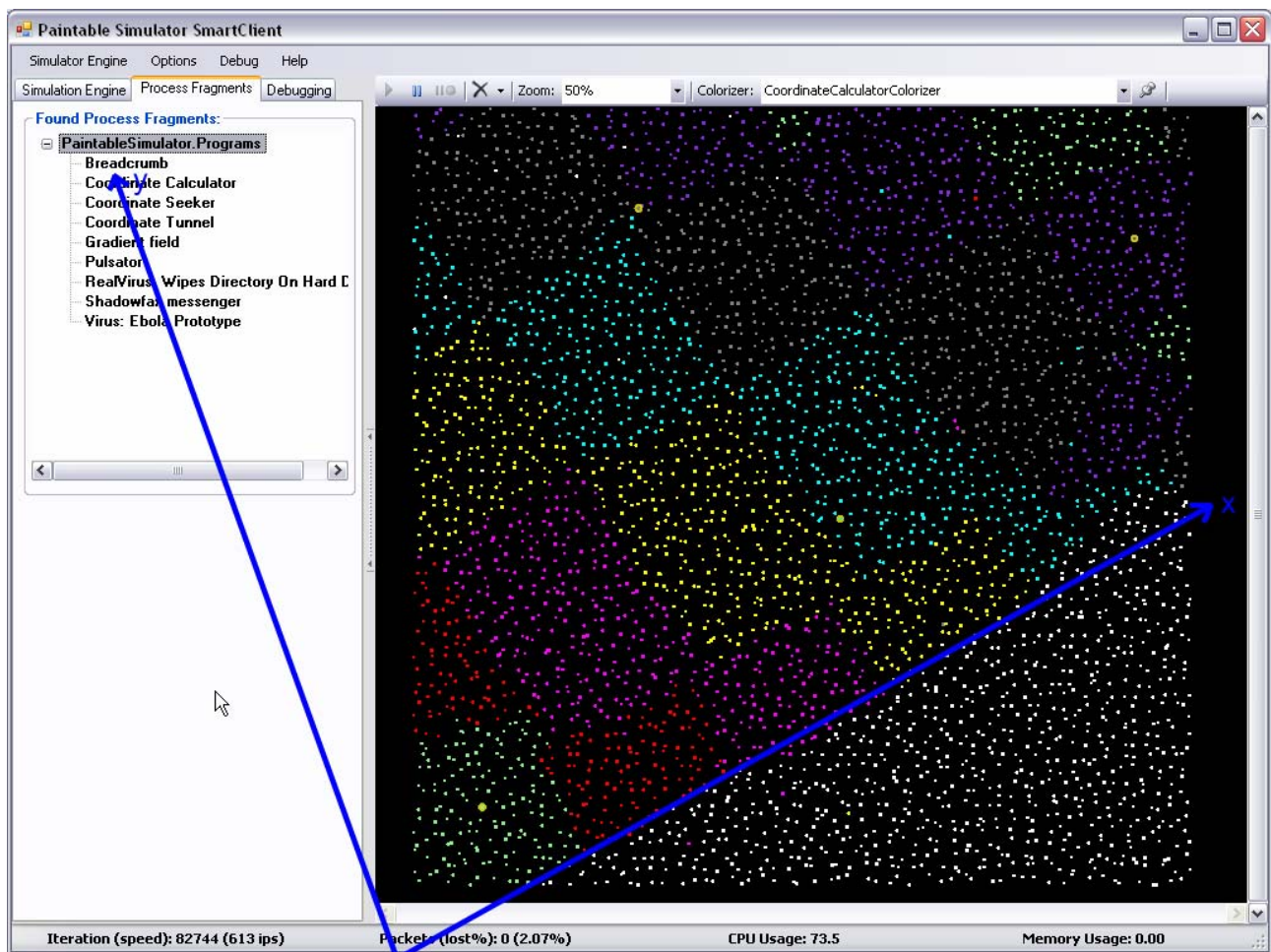
Izgled zida nakon učitavanja točaka s koordinatama

Nakon obavljenog postupka valja ubaciti program Coordinate Calculator na ovaj zid da pokuša iscrtati koordinatni sustav. Na sljedećoj slici je prikazan koordinatni sustav sa ponekom izdvojenom točkom. Sve u svemu, pogreška u određenim koordinatama i nije toliko ogromna.



Egzotičan primjer s označenim nekim koordinatama

Na kraju, ponavljamo prvi primjer s koordinatnim sustavom, samo ovaj puta ćemo ga nakositi. Na slici su označene osi koje smo željeli postići.



Kosi koordinatni sustav

9 Literatura

- William Joseph Butera: Programming a Paintable Computer
- J.R.R. Tolkien: The Lord of the Rings
- Microsoft Developer Network: <http://msdn.microsoft.com>
- Wikipedia (Trilateration): <http://en.wikipedia.org/wiki/Trilateration>