

Kolokvijum iz Dizajna programskih jezika 1.12.2010

Lanac ishrane

Vuk i zec su oduvek imali suprotstavljene ciljeve. Zec ima za cilj da sebi obezbedi što ugodniji život u smislu: što više kupusa, što manje vukova. Vuk sa druge strane želi što više zečeva. U centralnom delu prozora nalazi se njiva sa kupusom, oko nje je livada. Godišnji prinos iznosi 100 glavica, a kako je ovo ubrzana simulacija, pretpostavimo da godina traje samo 10 sekundi. Ukoliko zec naiđe na njivu, on obično pojede jednu, a ako je baš jako gladan najviše dve glavice kupusa, nakon toga napušta njivu. Ako na njivi nema kupusa, zec čeka dok se situacija ne promeni. Zečiji algoritam kretanja je vrlo jednostavan i odgovara kretanju bilijarske kugle, dakle ukoliko naiđe na njivu ili zidove prozora (kraj sveta), on se odbija pri čemu je izlazni ugao jednak ulaznom. Vuk se kreće na identičan način i njemu nije interesantna njiva. Međutim, ukoliko naiđe na zeca on će ga pojesti bez obzira gde se on nalazi, bilo na njivi, bilo na livadi.

Elementi za implementaciju:

1. Prozor dimenzija 400 x 600 sa iscrtanom njivom dimenzija 100 x 100 na sredini [20%]
2. Svaki zec je zasebna nit pri čemu je potrebno implementirati gore opisano kretanje, detektovanje ulaska u njivu, konzumaciju kupusa i blokiranje ukoliko ga nema dovoljno. Zec grafički izgleda kao krug prečnika 10. Kreirati ukupno 20 zečeva. [20%]
3. Svaki vuk je zasebna nit sa gore opisanim kretanjem, pri čemu se on ne blokira se na njivi već samo prelazi preko nje. Vuk je grafički prikazan krugom prečnika 40 i potrebno je da detektuje zeca kao objekat, a to je kada se dva kruga preseku. U tom momentu zec nestaje sa ekrana, tj. uklanja se iz kolekcije zečeva. [30%]
4. Njiva je jedini deljeni resurs i treba ga zaštititi korišćenjem semafora i listi čekanja (kondicionala), pri čemu postoji samo jedan kondicional, a to je kada nema dovoljno kupusa (primetiti da nema gornje granice, jer zečevi ne donose kupus, već ga samo uzimaju). Gore je objašnjeno da je njiva obnovljivi resurs, tako da i ona treba da bude nit, kako bi sa vremena na vreme povećavala količinu kupusa (100 glavica na svakih 10 sekundi). [30%]

Scheme

Definisati tip podataka za poziciju (position), što je dvodimenzionalna lokacija. Zatim, definisati tipove podataka sledeće oblike:

- Krug se definiše centrom (position) i prečnikom (broj)
- Kvadrat se definiše gornjom levom pozicijom i dužinom stranice (broj)
- Pravougaonik ima gornju levu poziciju, visinu i širinu

Razviti funkciju *in-shape*, koja prihvata oblik i poziciju p, a potom vraća *true* ako je p u okviru oblika, ili *false* u suprotnom.