

Zadatak POKLON	Autor: Frano Mihaljević
-----------------------	--------------------------------

Prvo nacrtamo kvadrat stranice 100 piksela koji omeđuje poklon (pisanje možemo skratiti korištenjem naredbe REPEAT) te potom pomaknemo kornjaču naprijed za 70 piksela, okrenemo se za 90 stupnjeva udesno i nacrtamo srednji dio poklona (tri crte duljine 50, 40 i 50 piksela uz okretanje kornjače za 90 stupnjeva).

potrebno znanje: osnovne naredbe za pomicanje kornjače

Zadatak KOŠARKA	Autor: Frano Mihaljević
------------------------	--------------------------------

Nacrtamo pravokutnik koji prikazuje granice košarkaškog terena, pomaknemo kornjaču za 35 piksela unaprijed i nacrtamo pravokutnik koji označava lijevi reket. Pri crtanju tog pravokutnika stanemo na sredini treće stranice gdje pomoću naredbe CIRCLE nacrtamo kružnicu radijusa 15 piksela. Nakon toga kornjaču pomaknemo na donji desni kut desnog reketa i ponovimo postupak, ali pravokutnik crtamo u drugom smjeru.

Potom pomaknemo kornjaču na polovicu donje stranice terena i nacrtamo centralnu liniju i kružnicu.

Ovo je samo jedna od mogućnosti rješavanja ovog zadatka. Elementi košarkaškog terena mogu se crtati raznim načinima i redoslijedom.

potrebno znanje: osnovne naredbe za pomicanje kornjače, crtanje kružnice

Zadatak BUBA	Autor: Marija Gegić
---------------------	----------------------------

Najprije koristeći naredbu CIRCLE nacrtamo kružnicu polumjera 100 piksela. Zatim 6 puta ponavljamo crtanje noge. Za svaku nogu se potrebno prvo koristeći naredbu PU pomaknuti 100 piksela do ruba kružnice, nacrtati crtu duljine 50 piksela, nakon toga se pomaknuti 10 piksela bez ostavljanja traga, kako bismo došli u središte kružnice koju je potrebno nacrtati i zatim je ispuniti naredbom FILL. Nakon toga nacrtamo glavu bube te na sličan način kao što smo nacrtali noge nacrtamo i ticala, pazeći da je crta koja spaja glavu bube i ispunjenu kružnicu duljine 30 piksela.

potrebno znanje: osnovne naredbe za pomicanje kornjače, crtanje kružnice, ispuna površine

Zadatak TALOS**Autor: Ivan Paljak**

Među najjačim oružjima iz arsenala dobrog natjecatelja sasvim sigurno se nalaze **papir i olovka**. Vjerojatno najbolji pristup rješavanju ovog zadatka je konstrukcija rješenja na papiru. Za osvajanje svih bodova trebalo je paziti da se iskoriste svi tipovi tetromina te da ploča bude u potpunosti popločena. Ovakav pristup relativno nas brzo vodi do rješenja budući da su dimenzije ploče odabrane tako da postoji velik broj optimalnih rješenja.

Nakon što riješite neki zadatak, dobra je strategija uvesti neka nova (otežavajuća) ograničenja. Primjerice, ovdje se prirodno nameću pitanja poput "Što kada bi se, umjesto jednog konkretnog rješenja, tražio ukupan broj različitih rješenja?", "Kako bismo pristupili zadatku kada bi dimenzija ploče bila fiksna ali mnogo veća (npr 100x100)?" ili "Što kada dimenzija ploče ne bi bila fiksna, već bi se nalazila u ulazu?".

Odgovore na gore postavljena pitanja, naravno, ostavljamo čitateljici za vježbu.

potrebno znanje: osnovne naredbe za pomicanje kornjače, konstrukcijski algoritmi.

Zadatak KORMILO**Autor: Marija Gegić**

Uz pomoć naredbe `CIRCLE` nacrtamo dvije kružnice, radijusa `:r1` i `:r2`. Zatim uz pomoć neke petlje, primjerice `REPEAT`, `:n` puta ponavljamo crtanje crte duljine `:r1` koja izlazi iz središta kružnice. Nakon toga se potrebno pomaknuti za `:r2-:r1` kako bismo došli na poziciju iz koje možemo nacrtati dva jednakostranična trokuta stranice duljine `:d`. Nakon toga se vraćamo u središte kružnica i okrećemo za kut `360/:n`, kako bi sve crte bile pravilno raspoređene.

potrebno znanje: petlje

Zadatak ZMIJA**Autor: Antea Hadviger**

Pogledajmo najprije podzadatke opisane u sekciji BODOVANJE. Za rješavanje prvog podzadatka trebamo osmisliti algoritam kojim crtamo zmiju koja se sastoji od k trokuta, a savija se nakon njih n ($k > n$). U drugom podzadatku nije potrebno paziti na savijanje zmije, ali je potrebno paziti na orijentacije trokuta. Jasno je da nezavisnim rješavanjem ovih podzadataka dobivamo i rješenje cijelog zadatka.

Orijentaciju trokuta jednostavno možemo odrediti tako da provjerimo parnost indeksa elementa u listi :l na kojem se trenutno nalazimo. Tako se skupine trokuta opisane neparnim elementima liste :l crtaju ulijevo, dok se skupine trokuta opisane parnim elementima crtaju udesno.

Dodatno, da bismo pazili na savijanje zmijske, potrebno je pamtit i koliko smo do sada trokuta nacrtali te saviti zmijsku nakon svakog n-tog nacrtanog trokuta (neovisno o orijentaciji).

potrebno znanje: rad s listama, petlje

Zadatak SPOJI	Autor: Marija Gegić
----------------------	----------------------------

Za jednostavno rješavanje ovog zadatka, korisno je primijetiti da spajanja koja se crtaju iznad kvadrata i spajanja koja se crtaju ispod kvadrata ne utječu jedna na druge, pa ih možemo zasebno rješavati tako da prvo nacrtamo sva spajanja iznad kvadrata, pa nakon toga sva spajanja ispod.

Kada želimo spojiti dva kvadrata a i b, na visinu na kojoj ćemo ih spojiti utječu samo do sada nacrtana spajanja kvadrata x i y za koje vrijedi da postoji presjek intervala [a b] i [x y]. Pamtim li u dodatnoj listi na kojoj smo visini nacrtali koje spajanje, onda za svako novo spajanje možemo stoga proći po svim prethodnim spajanjima i za svako od njih provjeriti postoji li presjek sa spajanjem kojeg trebamo nacrtati, te ukoliko postoji, provjeriti je li visina na kojoj je to spajanje nacrtano najveća do sada. Visina na kojoj ćemo nacrtati novo spajanje bit će za 10 piksela veća od te najveće visine.

Za implementacijske detalje pogledajte službeno rješenje.

potrebno znanje: petlje, liste

Zadatak KUGLICE	Autor: Ivan Paljak
------------------------	---------------------------

Da biste osvojili 10 bodova na ovom zadatku, dovoljno je poznavati samo osnovne naredbe za pomicanje kornjače za crtanje pravokutnika visine 225 i širine 450 kojem fali gornja stranica. Za osvajanje 30 bodova na zadatku dovoljno je poznavati osnovne naredbe za rad s listama te osnove koordinatne grafike. Ovaj dio zadatka u službenom je rješenju implementiran procedurom pocetno_stanje.

Za osvajanje ukupno pola bodova na zadatku (80) bilo je potrebno riješiti nešto lakšu varijantu zadatka u kojoj su sve prepreke horizontalne. Tada je dovoljno za svaku kuglicu pronaći **najvišu** prepreku koja se nalazi direktno ispod nje, a ako takva prepreka ne postoji zaključujemo da kuglica pada na pod. Zgodan

implementacijski trik jest tretirati pod kao horizontalnu prepreku koja se proteže od (0,0) do (450,0) te na taj način eliminirati poseban slučaj. Pronalazak najviše prepreke koja se nalazi direktno ispod loptice možemo implementirati na razne načine. Jedan od (računalno efikasnijih) načina oslanja se na matematičku analizu problema. Naime, označimo li početnu poziciju kuglice kao $(x_k, 225)$, a krajnje točke horizontalne prepreke kao (x_l, y) i (x_r, y) , jasno je vidljivo da se prepreka nalazi ispod kuglice ako i samo ako vrijedi

$$x_l \leq x_k \leq x_r$$

Sada je samo potrebno pronaći onu prepreku koja se nalazi ispod kuglice, a ima **maksimalnu** vrijednost y-koordinate. Ovo jednostavno radimo jednim prolaskom kroz listu u kojoj se nalaze prepreke.

Valja istaknuti i možebitno jednostavniji postupak pronalaska najviše prepreke koja se nalazi ispod kuglice. Mogli bismo, primjerice, nakon crtanja terena spuštati kornjaču sa početne pozicije kuglice prema dnu sve dok ne naiđemo na piksel crne boje. Tim smo postupkom sigurno pronašli poziciju na koju će kuglica pasti.

Konačno, za osvajanje svih bodova potrebno je algoritam proširiti na prepreke koje nisu nužno horizontalne. Algoritam pada loptice možemo opisati sljedećim pseudokodom:

```
pad_loptice(x, y): // vraca završnu poziciju loptice
    p = sljed_prepreka(x, y) // p = ((x1, y1), (x2, y2))
    ako vodoravna(p):
        vrati (x, y)
    ako nagnuta_lijevo(p):
        vrati pad_loptice(x1-5, y1)
    inače:
        vrati pad_loptice(x2+5, y2)
```

Funkcije `vodoravna()` i `nagnuta_lijevo()` svode se na usporedbu y-koordinata krajnjih točaka prepreke, dok je funkcija `sljed_prepreka()` nešto manje očita. Pronalazak sljedeće prepreke svodi se na filtriranje onih koje se nalaze direktno ispod kuglice (ovaj je dio ekvivalentan provjeri iz prethodnog podzadatka koji nosi 80 bodova) te pronalazak one prepreke na koju će kuglica najprije pasti. Zamislimo li pravac koji prolazi središtem kuglice, a okomit je na x-os, vidjet ćemo da on siječe sve prepreke koje se nalaze direktno ispod loptice. Prepreka koju tražimo je ona prepreka čije sjecište sa spomenutim pravcem ima **najveću** y koordinatu.

Sada se postavlja pitanje kako odrediti sjecište okomitog pravca i prepreke. S ciljem širenja znanja, u nastavku ćemo opisati matematički pogled na problem, no

imajte u vidu da smo sjecište mogli tražiti i na druge načine (primjerice micanjem kornjače do crnog piksela ili binarnom pretragom).

Sve točke nekog pravca, pod uvjetom da nije okomit na x-os, možemo opisati jednadžbom oblika $y = ax + b$, gdje nam parametar b govori u kojoj točki pravac sječe y-os, dok nam parametar a regulira njegov nagib. Značenje parametra a možemo na intuitivniji način shvatiti zamislimo li kornjaču koja se kreće po danom pravcu. Ako se x koordinata kornjače prilikom kretanja po pravcu promijenila za x' , tada se njena y koordinata promijenila za $a \cdot x'$. Jednadžbu pravca na kojem leži prepreka s krajnjim točkama u (x_1, y_1) i (x_2, y_2) možemo jednostavno odrediti riješimo li sustav jednadžbi

$$y_1 = a \cdot x_1 + b$$

$$y_2 = a \cdot x_2 + b$$

po nepoznanicama a i b . Sada kada znamo jednadžbe pravaca na kojima leže prepreke, dovoljno je u svaku jednadžbu uvrstiti x-koordinatu kuglice kako bismo dobili visinu na kojoj bi kuglica pala na tu prepreku. Ovaj je dio u službenom rješenju implementiran funkcijom `sjeciste()`.

potrebno znanje: Napredniji rad s listama, koordinatnom grafikom i dobro poznavanje matematike.