

## 8. razred

### Čaša

**Potrebna znanja:** osnove rada s koordinatnom grafikom, osnovne naredbe za kretanje kornjače, osnovne operacije za kontrolu tijeka programa (naredba IF)

Počinjemo crtati od pravokutnog dna čaše se se pomičemo na gornju stranu pravokutnika iz koje prema gore crtamo stalak čaše.

Sada crtamo obris gornjeg dijela čaše. Poziciju na kojoj se stalak spaja s trokutastim gornjim dijelom čaše spremamo u varijablu :po. Zatim dižemo pero kornjače i pomičemo se ravno gore za :v kako bismo došli do ruba gornjeg dijela čaše. Tada se okrećemo u lijevo, spuštamo pero kornjače i idemo naprijed za :a to lijevog kuta gornjeg dijela čaše. kako bismo nacrtali stranicu trokuta kojoj nije zadana duljina pomoću naredbe SETPOS vraćamo se na poziciju :po. Postupak se ponavlja za desnu stranu gornjeg dijela čaše određenu varijablom :b. Dok crtamo desni dio gornjeg dijela čaše i nalazimo se u desnom kutu pomoću naredbe CIRCLE crtamo krug polumjera :r.

Ukoliko je varijabla :p veća od 0, potrebno je nacrtati i sadržaj čaše. Sadržaj se crta istom strategijom kao i gornji dio čaše jer je proporcionalan. Potrebno je samo odrediti koeficijent proporcionalnosti :k. Površina cijelog gornjeg dijela čaše koju ćemo označiti s :pov iznosi  $(a + b) * v / 2$ , a površina tekućine je zadana kao :p. Koeficijent proporcionalnosti površina je jednak kvadratu koeficijenta proporcionalnosti stranica koji tražimo pa koeficijent :k možemo dobiti formulom  $\text{SQRT}(:pov / :p)$ . Kad znamo koeficijent, možemo izračunati duljine stranica trokuta koji prikazuje tekućinu i nacrtati ju na isti način kao što smo crtali i gornji dio čaše.

U testnim primjerima vrijednim 20% (6) bodova vrijedilo je :r = 0 i :p = 0 pa nije trebalo crtati krug ni tekućinu u čaši.

U testnim primjerima vrijednim dodatnih 40% (12) bodova vrijedilo je :p = 0 što znači da se krug crta, a tekućina ne.

### Strip

**Potrebno znanje:** osnovne naredbe za kretanje kornjače, osnovne operacije za kontrolu tijeka programa (naredba REPEAT), crtanje kružnih lukova

Za pojednostavljivanje glavne procedure korištena je procedura VALOVI :e :f :x, pomoćna procedura koja crta valovite stranice osmerokuta. Varijable :e i :f u njoj određuju dimenzije ponavljajućih poluelipsi, a :x označava koliko će ih se crtati u nizu.

Počinjemo crtati s desnog kraja donje vodoravne stranice zamišljenog osmerokuta. U glavnoj proceduri crtaju se najprije jedna stranica s :n poluelipsi, zatim tri stranice s :m poluelipsi, zatim jedna

s :n, te na kraju ponovno tri stranice s :m poluelipsi. Na kraju se postavljamo okumito na zadanu stranicu zamišljenog osmerokuta i udaljavamo od nje te crtamo :n kružića s polumjerom :f.

U testnim primjerima vrijednim 20% (8) bodova vrijedilo je :n = 1 i :m = 1, što znači da je oblačić izgledao kao šesterokut s po jednom poluelipsom na svakoj stranici te se sa strane nalazio samo jedan kružić.

U testnim primjerima vrijednim dodatnih 20% (8) bodova vrijedilo je :n = 1 i :m > 1, što znači da se sa strane oblačića nalazio samo jedan kružić, ali je na vodoravnim stranicama oblačića bio više od jedne poluelipse.

## Burger

**Potrebno znanje:** osnove rada s listama dubine 1 i riječima, naredbe za kontrolu tijeka programa (FOR, IF), crtanje kružnih lukova

Ključ rješavanja zadatka je nacrtati dvije različite verzije mesa i rajčice, ovisno o tome nalaze li se oni neposredno ispod salate u sendviču ili ne.

Potom prolazimo po listi pamteći prethodno nacrtani element (točnije, samo je li prethodni element bio salata) i crtamo sastojke redom.

U slučaju da se meso nalazi ispod salate, potrebno je “obrisati” dijelove kružnog luka koji bi se inače nalazili unutar salate. To je moguće napraviti tako da nakon crtanja mesa, prethodno nacrtanu salatu obojimo crnom bojom i ponovno ju nacrtamo i obojamo, koristeći drugu boju pera, ali zelenu boju ispune. Potom još jednom nacrtamo tu istu salatu koristeći crnu boju pera i pomaknemo se na mjesto idućeg sastojka.

Dodatno, uočena činjenica da su dimenzije donjeg peciva iste kao i dimenzije rajčice olakšava rješavanje zadatka, s obzirom da se ista funkcija može koristiti za crtanje oba sastojka, samo je donje pecivo potrebno obojiti drugom bojom.

## Golf

**Potrebno znanje:** koordinatna grafika, rad s listama većih dubina, ispunjavanje omeđenog dijela ekrana nekom bojom (FILL), osnovne operacije za kontrolu tijeka programa (naredba IF)

Za rješavanje zadatka prvo je potrebno nacrtati teren, tj. rupu (krug ispunjen crvenom bojom) i štapove.

Zatim je potrebno simulirati igru svakog igrača, kretanjem po terenu s podignutim perom za 0.5 piksela sve dok je zadovoljeno sljedeće:

- loptica nije ušla u rupu (odnosno ne nalazi se na crvenom pikselu)
- loptica nije udarila u neki štap (odnosno ne nalazi se na crnom pikselu)
- loptica nije iskoristila svu snagu

Pritom boju piksela na kojem se kornjača nalazi provjeravamo naredbom PIXEL.

Nakon svakog udarca u štap i odbijanja loptice, potrebno je nastaviti simulaciju na isti način.

Prilikom simulacije, za svakog igrača je potrebno brojati broj udaraca loptice u štapove te provjeriti je li loptica upala u rupu, kako bi se na kraju mogao odrediti pobjednik uspoređivanjem tih vrijednosti te ispisati broj odbijanja loptice.

Pritom se u listi pamte sve ključne pozicije kretanja kornjače (početak, svaka točka odbijanja, točka ulaska u rupu ili prestanka kretanja) kako bi se naknadno putanja loptice mogla nacrtati spajanjem tih točaka redom uz korištenje koordinatne grafike.

Kako bi se mogao odrediti kut odbijanja, potrebno je odrediti u koji je štap loptica udarila te kut između putanje loptice i dužine.

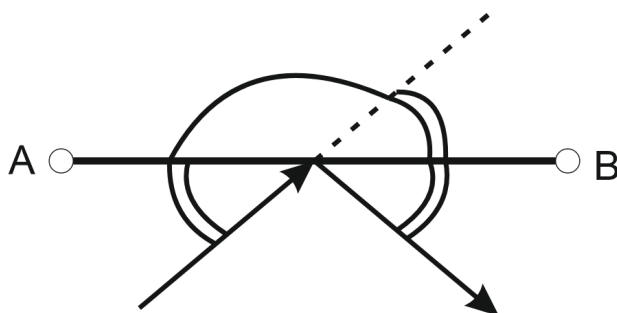
Označimo sa A i B krajnje točke dužine u koju je loptica udarila (onim redom kako su navedene u listi) te sa X točku u kojoj se udarac dogodio.

Da bismo odredili u koji je štap loptica udarila, dovoljno je primijetiti kako je  $d(A, X) + d(X, B) = d(A, B)$ , pri čemu je  $d(A, B)$  udaljenost točaka A i B. Zato je potrebno provjeriti je li zadovoljena navedena jednakost za svaku dužinu (uzimajući u obzir mala odstupanja s obzirom na nepreciznost koordinatne grafike). Tražena dužina je ona za koju ta jednakost (uz malo odstupanje) vrijedi.

Nakon što smo odredili dužinu, potrebno je odrediti kut između putanje loptice i dužine korištenjem trenutnog smjera kornjače (heading) i funkcije towards. Pritom razlikujemo slučajeve kada se točka A nalazi s lijeve ili desne strane u odnosu na dosadašnji smjer kretanja loptice, a o kojem je slučaju riječ možemo saznati iz vrijednosti kutova  $\alpha_1$  i  $\alpha_2$  iz rješenja.

Zatim, ovisno o vrijednostima  $\alpha_1$  i  $\beta_1$ , odnosno  $\alpha_2$  i  $\beta_2$  (ovisno o slučaju), određujemo smjer skretanja i veličinu kuta prilikom odbijanja.

Na slici se nalazi jedan primjer odbijanja. U ovom slučaju se točka A nalazi lijevo u odnosu na dosadašnji smjer kretanja loptice, a vrijedi  $\beta_2 < \beta_1$ . Kut  $\alpha_2$  je na skici označen jednostrukim lukom, a  $\beta_2$  je označen dvostrukim lukom (pojavljuje se na tri mjesta na skici).



Za osvajanje 20% (14) bodova u slučaju kada je lista :l prazna, dovoljno je nacrtati teren (odnosno samu rupu), putanju kretanja obje loptice s obzirom na zadani kut prilikom ispućavanja i snagu loptice te provjeriti je li neka od loptica upala u rupu. Broj odbijanja svake loptice će biti jednak 0.

Za osvajanje 20% (14) bodova u slučaju kada lista :l ima točno 1 element, dovoljno je nacrtati teren koji se sastoji od jednog štapa i rupe, putanju kretanja obje loptice (pritom će se svaka loptica odbiti najviše jednom) te provjeriti je li neka od loptica upala u rupu. Broj odbijanja svake loptice će biti jednak 0 ili 1.