

Opisi rješenja 5. razred

Šiljilo

Potrebno znanje: osnovne naredbe za kretanje kornjače, crtanje mnogokuta (četverokuta), osnovne operacije za kontrolu tijeka programa (REPEAT), crtanje kružnice

Jedno moguće rješenje je da počnemo crtati obrub šiljila počevši od donjeg lijevog kuta šiljila. Sve potrebne veličine za obrub vidljive su na skici. Potrebo je jedino primijetiti da je širina obruba na gornjoj i donjoj strani jednaka $2 * (:b + :d + :e)$.

Kad smo nacrtali cijeli obrub, ponovno smo u donjem lijevom kutu i krećemo crtati oštricu. Dižemo se za visinu dna šiljila $:c$ i pomičemo udesno za $:b$, kao što smo napravili i ranije, ali dodatno se uz dizanje olovke pomičemo udesno za $:d$ da bismo uvukli oštricu šiljila u odnosu na tijelo. Pomoću naredbe repeat crtamo lijevu polovicu oštrice šiljila pa se pomičemo udesno za $:e$ i crtamo i desnu polovicu šiljila.

Sada slijedi crtanje vijka. Pomičemo se za polovinu širine polovice oštrice udesno pa dižemo olovku i idemo za $:a - :f$ prema gore. Spuštamo olovku i crtamo kružnicu promjera $:e$ Pomičemo se za $:e / 2$ prema gore i prema dolje da bismo crtali liniju koja predstavlja okomiti polumjer kružnice. Zakrećemo se za 90 stupnjeva udesno i idemo za $:e / 2$ udesno i ulijevo da bismo crtali vodoravni polumjer kružnice.

Grčka

Potrebno znanje: osnovne naredbe za kretanje kornjače, osnovne operacije za kontrolu tijeka programa (REPEAT, IF), ispunjavanje omeđenog dijela ekrana bojom (SETF, FILL), osnovne matematičke operacije

U zadatku je potrebno ponavljati zadani segment naizmjenično okrenut prema lijevo i prema desno pa si postupak možemo olakšati stvaranjem dviju procedura koje su u službenom rješenju nazvane DESNI_SEGMENT $:a :b :c$ i LIJEVI_SEGMENT $:a :b :c$. U njima je nacrtan dio koji se ponavlja okrenut na odgovarajuću stranu.

Te dvije procedure potrebno je ponavljati naizmjenično $:n$ puta. ponavljanje se može napraviti na razne načine, a jedan od njih korišten je u službenom rješenju. Korištena je naredba REPEAT u kojoj se provjerava je li trenutni redni broj ponavljanja djeljiv s 2. Redni broj ponavljanja dobiva se naredbom REPCOUNT, a ostatak pri dijeljenju provjerava se naredbom REMAINDER. Naredbom IF odlučuje se koji od segmenata se trenutno crta.

Nakon crtanja svih segmenata crta se obrub te se odgovarajući dijelovi slike ispunjavaju crnom odnosno narančastom bojom pomoću naredbi: SETFC "BLACK, SETFC "ORANGE i FILL.

Za osvajanje 20% (8) bodova u zadatku nije bilo potrebno ponavljati uzorak jer je vrijedilo $n = 1$.

Za ostvarivanje dodatnih 20% (8) bodova crtanje je bilo nešto jednostavnije jer su vrijednosti varijabli a , b i c bile međusobno jednake the je bilo garantirano da je n paran broj te nije bilo potrebno provjeravati parnost trenutno nacrtanog segmenta jer se moglo crtati dva po dva segmenta.

Za ostvarivanje dodatnih 20% (8) bodova također je bilo garantirano da je n paran broj te nije bilo potrebno provjeravati parnost trenutno nacrtanog segmenta, ali su varijable a , b i c imale međusobno različite vrijednosti.

Zmija

Potrebno znanje: osnovne naredbe za kretanje kornjače, crtanje mnogokuta, osnovne operacije za kontrolu tijeka programa (REPEAT, FOR, IF), promjena boje olovke (SETPC)

Pri rješavanju zadatka pomaže ako smo izdvojili kod za crtanje romba u zasebnu proceduru ROMB $d:k$ koja crta romb sa stranicom duljine d takav da mu je manji kut jednak k stupnjeva.

U glavnoj proceduri ZMIJA $d:k:n$:smjer :kp crtamo zmiju. Na početku postavljamo boju olovke na crnu, okrećemo se u desno za :smjer kako bismo bili spremni za crtanje prvog romba te postavljamo varijablu :lijevo na vrijednost "true".

Rombove koji čine tijelo zmije crtamo pomoću FOR petlje. U svakom izvršavanju petlje najprije crtamo romb, a zatim odlučujemo kamo se trebamo okrenuti. Okrećemo se nakon crtanja svakog romba, osim nakon zadnjeg pa prije okretanja provjeravamo je li redni broj romba koji se crta, nazvan i , manji od ukupnog broja rombova n .

Nakon crtanja romba provjeravamo stanje varijable :lijevo koja prati crtamo li trenutno zmiju koja skreće lijevo ili ne. Ako skreće lijevo i vrijedi :smjer-:kp < -90 spremamo vrijednost "false" u varijablu lijevo kako bismo se prebacili u desne okrete. Ako zmija trenutno skreće u desno i vrijedi :smjer+:kp > 90, vraćamo se na okrete u lijevo i postavljamo varijablu :lijevo na "true". Nakon ove korekcije smjera radimo okret tako što provjerimo trenutno stanej varijable :lijevo te ako je vrijeme za okret u lijevo obavimo okret i smanjimo vrijednost varijable :smjer za :kp. Kod okreta u desno povećavamo varijablu :smjer za :kp.

Na samom kraju postavljamo boju olovke na crvenu i crtamo jezik zmije prema skici.

Za ostvarivanje ukupno 20% (12) bodova, zmija nije mijenjala smjer pa se moglo izostaviti dio rješenja koje prati trenutni smjer i vrstu zakretanja te je vrijedilo $k = 0$ što znači da se nije moralo crtati rombove nego samo crte duljine $2*d$.

Za ostvarivanje ukupno 50% (30) bodova, vrijedilo je samo $k = 0$ pa je bilo potrebno pratiti smjer i vstu zakretanja, ali nije bilo potrebno crtati rombove.

Piramida

Potrebno znanje: osnovne naredbe za kretanje kornjače, crtanje mnogokuta, osnovne operacije za kontrolu tijeka programa (REPEAT, WHILE, IF), ispunjavanje omeđenog dijela ekrana bojom (SETF, FILL)

Za uspješno rješavanje zadatka prvo je potrebno otkriti koliko je široka baza piramide. Budući da piramida nije nužno dovršena, moramo otkriti koliko bi joj bila baza kad bi bila potpuna te zatim crtati cigle od dna prema vrhu dok ih sve ne iskoristimo.

Širinu baze otkrivamo isprobavajući sve moguće piramide od najmanje moguće do prve veće koja sadržava barem n cigli. Za početak postavljamo na 1 varijablu `:baza` koja će nam predstavljati širinu baze, odnosno najdonjeg reda cigli, za piramidu koju trenutno isprobavamo. Varijablu `:maks` koja nam predstavlja ukupan broj cigli u trenutno promatranoj piramidi također postavljamo na 1 jer nam najmanja moguća piramida ima samo jednu jedinu ciglu pa nam je ta cigla ujedno i baza čitava piramida. WHILE petljom povećavamo bazu za 1 te ju pribrajamo varijabli `:maks` dok god vrijedi $n > :maks$. Nakon završetka petlje po varijabli `:baza` znamo koliko ćemo cigli nacrtati u najnižem redu piramide.

Sad možemo krenuti s crtanjem. Potrebna nam je petlja u kojoj ćemo crtati cigle red po red prema vrhu, svaki put smanjujući bazu za 1 te pazеći da prestanemo crtati kad ponestane cigli. U službenom rješenju to je učinjeno pomoću beskonačne WHILE petlje koja crta red cigli, provjeravajući nakon svake je li potrebno zaustaviti program, te nakon nacrtanog reda smanjuje broj cigli koji se crta u sljedećem redu. Prilikom crtanja cigli pratimo koliko smo neobojenih cigli nacrtali za redom pomoću varijable `:c`. Nju postupno povećavamo te ispunjavamo ciglu ukoliko je $c = k$. Kad smo ispunili ciglu bojom vraćamo `:c` na 0 kako bismo ponovno počeli brojati.

Za osvajanje 30% (21) bodova, k je bio jednak 1 pa se bojala svaka cigla i crtala se potpuna piramida pa nije trebalo otkrivati kolika je baza.

Za osvajanje 50% (35) bodova, crtala se potpuna piramida, no trebalo je pratiti koju ciglu je potrebno obojiti.