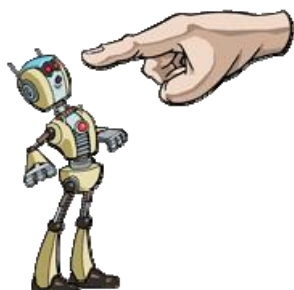


A MAILA.

1


1. ARIKETA. INSTRUKZIOAK




Gure ordenagailuari eman diezazkiokegun agindurik errazenak dira instrukzioak.

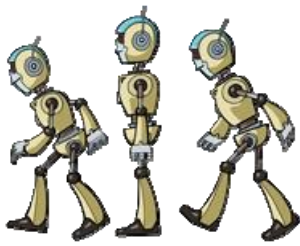
Programa guztiek dute bloke nagusi bat *main* (nagusia) terminoarekin hasi eta *end main* (nagusiaren amaiera) terminoarekin amaitzen dena. Programa abian jartzen denean, bloke horretako instrukzioak exekutatu dira lehenik eta behin.

Laukitxo bat aurrera egiteko aginduko diogu robotari.

Main eta *end main* artean dagoen muntatze-linea arrastatuko dugu gure erreminta-kutxako *EGIN AURRERA* botoia . *EGIN AURRERA* instrukzioan 1 zenbakia dago: *Egin aurrera 1 laukitxo* esan nahi du horrek.


PLAY  botoian egingo dugu klik. Robotak aurrera egingo du, baina ez du lortuko zoruan dagoen gurutze-itxurako seinaleraino iristea. Jarraibide gehiago eman beharko dizkiogu, bada.

Begiratu pantailaren eskuineko panelari, programazio-panelari. Programazio-lengoaia (JavaScript, adibidez) erabiliko bagenu, horiexek izango lirateke programaren kodeko lerroak.



2. MAILA. 2. ARIKETA. SEKUENTZIAK

Gure helburua lortzeko, instrukzio asko batu eta errepikatu beharko ditugu. Horixe da, hain zuzen, sekuentzia bat: instrukzioen zerrenda batua.

Bigarren ariketan, batu EGIN AURRERA eta BIRATU  1 instrukzioak, gurutzeraino iristeko.

2

3. ARIKETA. PARAMETROAK

Instrukzio baten portaera aldarazten duten balioak dira parametroak (berdin dio zenbakia edo letra den).

Egin dezagun aurreko ariketa, baina 3 instrukzio erabili beharrean, bakarra erabiliz.

EGIN AURRERA instrukzioa muntatze-linean ipiniko dugu eta, ondoren, *EDITATU* ikonoraino (beheko aldeko arkatzeraino) arrastatuko dugu.



Leiho bat irekiko da eta, 1 zenbakia dagoen lekuan 3 idatzi eta gero, *OK* botoian sakatuko dugu.

Instrukzioa aldatu egin dela ikusiko dugu, lehen 1 zegoen lekuan orain 3 dago eta. Orain, instrukzio horrek *Egin aurrera 3 laukitxo* esan nahi du.

Barruan zenbakia duten instrukzio-kutxa guztietan dago guk aldatzeko moduko parametro bat. Horretarako, *EDITATU* ikonora arrastatu behar dugu instrukzioa.

4. ARIKETA. PARAMETRO GEHIAGO

Gure programa ulerterraza izan dadin, ahalik eta instrukzio gutxien erabiltzea komeni da. Parametroak baliatuko ditugu horretarako.

Ariketa honetan, 5 instrukzio soilik erabiliko ditugu helmugara (gurutzera) heltzeko.

5. ARIKETA. SEKUENTZIEN AMAIERA

Oso garrantzitsua da gure instrukzioak jartzeko ordena. Bi instrukzio berdin ordena desberdinean ipintzen baditugu, emaitza oso desberdina izango da.

Instrukzioak ordena zehatz batean ipini baino lehenago, horiek exekutatzeko ordena egokia ote den pentsatu behar dugu.

Erabili orain arte ikasitako guztia, ahalik eta instrukzio gutxien erabilita iristeko amaieraraino.

3

B MAILA.

6. ARIKETA. FUNTZIOAK

Programazio-lengoaia guztietan oso garrantzitsua da funtzioaren kontzeptua.

Beharbada, programa exekutatzen dugunean instrukzio multzo bera erabiliko dugu behin baino gehiagotan. Instrukzio-blokea behin eta berriro idazten ez ibiltzeko daude funtzioak.

Bloke nagusitik (*main*) banatutako instrukzio-multzoa da funtzioa.

Funtzioa (*function*) terminoaren eta funtzioaren amaierako markaren (*function end*) artean kokatuko ditugu instrukzio guztiak.

Exekutatu nahi dugunean, programazio-linearen barruan ipini beharko dugu funtzio-blokea `function A`, beste edozein instrukzio balitz bezala.

Aztertu irudia. Ikusten duzu errepikatzen den patroirik? Konturatzen bazara, robotak L bat eginez mugitu behar du. Ondoren, bira egiten du. Eta ondoren beste L bat egiten du.

L formako mugimenduak funtzio bat balira bezala kodetu ditzakegu. Hala, instrukzioak linea nagusian jartzen ditugunean, hiru instrukzio baino ez ditugu jarri beharko: funtzioa, biratu eta berriro funtzioa.

7. ARIKETA. FUNTZIOAK

Programazioan pentsatzea eta errepikatzen den patroia aurkitzea da zailena. Aztertu egoera hau eta, zailegia irizten badiozu, egin klik goiko aldean dagoen «mailako laguntza» botoian.

Zoruan dauden gezien irudiak errepikatzen diren patroiak aurkitzen lagunduko dizu; izan ere, gure programazio-lineak optimizatzen lagunduko diguten funtzioak sortzeko balio izango digute.

8. ARIKETA. FUNTZIOAK

Aurkitu errepikatzen den patroia, «mailako laguntza» botoia erabili gabe.

Pista bat: gure funtzioa 3 aldiz errepikatzen da.

9. ARIKETA. FUNTZIOAK

Elementu berri bat: energia-kapsulak. Ariketa honetan, amaierako markaraino iritsi behar dugu. Alabaina, oraingoan energia-kapsula bat dagoen laukitxo guztietatik igaro behar dugu.

Funtzioei esker, instrukzio gutxiago erabiliko ditugu eta linea nagusiko programazioa ulergarriagoa izango da, nahiz eta robotaren mugimenduak nahasiak direla pentsatu.

Betiko moduan, errepikatzen den patroia aurkitu behar dugu.

10. ARIKETA. FUNTZIOAK

Ariketa honetan, beste ibilbide bat egingo dugu kapsulak biltzeko. Dena dela, arazoa ebazteko bidea 9. ariketan kontsultatu daiteke.

C MAILA.

11. ARIKETA. ERREPIKAPENAK

Zer gertatzen da instrukzio-blokea birritan edo hamar bider segidan errepikatzea nahi badugu?

Errepikapenek lagunduko digute arazo hori konpontzen. Beste funtzioetan bezala, behin eta berriro errepikatzen diren funtzioak bi marken artean egongo dira; kasu honetan, *repeat* eta *end repeat*, instrukzio-blokearen hasiera eta amaiera adierazteko.

Zenbat bider errepikatzea nahi dugun adierazteko (*loop*), instrukzioa *Editatu* laukira arrastatuko dugu, balioa aldatzeko.

Ohar zaitez, funtzioak ez bezala, *repeat* blokea *main* bloke nagusiaren barruan dagoela.



Lehen ariketa honetan, instrukzio bat (metro bat aurrera egitea) hiru bider errepikarazi behar duzu.

12. ARIKETA. ERREPIKAPENAK

Konturatu errepikapenak beti jarraian exekutatzen direla, etenik gabe. Erabili «mailako laguntza», pista bat emango dizu eta.

13. ARIKETA. ERREPIKAPENAK

Aztertu egoera, segidan gertatzen diren patroiak aurkitzeko.

Zoaz energia-kapsula bat daukaten laukitxo guztietara eta har ezazu (erabili  instrukzioa kapsula biltzeko, behin kapsula duen laukitxoan zaudenean). Kapsula askatu nahi duzunean, berriz, erabili  instrukzioa.

Energia-kapsulak banaka bildu eta gurutzearen amaierako markaren gainean utzi behar dituzu.

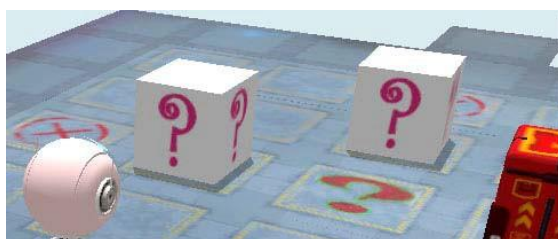
14. ARIKETA. ERREPIKAPENAK

Ariketa honetan, hiru energia-kapsula bildu eta pasilloan dagoen E markaren gainean utzi behar dituzu. Ez egin kasurik ateari, bakarrik irekitzen da.

Erabili *mailako laguntza*, pistak emango dizkizu eta.

15. ARIKETA. ERABAKIAK

Programazioaren funtsezko faseetako bat da, hari esker programak benetan moldakorak izatea lortzen delako, hainbat eta hainbat aukera eta egoeretara egokitzeko modukoak.



Gure robotek aurrera egiten dutenean, gerta daiteke kutxak aurkitzea ala ez aurkitzea. Ez dago lehenago jakiterik eta, beraz, bi aukeretako edozein gertatzen denean zer egin behar duen programatu beharko dugu. Irudian, galdera-ikurra duten kutxa zuriek esaten digute leku horretan kutxa bat egon daitekeen ala ez.

Zoruan galdera-ikurra duen laukitxo batera heltzen garenean, erabakia hartu behar dugu. Gure kodea exekutatzen dugunean, bi posizioetakoren batean ipiniko da kutxa, ausaz; horregatik, kutxa edozein lekutan dagoela funtzionatu behar du gure kodeak.

Erabaki bat hartzeko, instrukzio berri bat erabiliko dugu: *if*.



If hitzak *baldin eta* esan nahi du. Ikonoak esaten digu aurrera egitean oztopoa aurkitu dugula eta, beraz, itzulpen osoa hauxe izango litzateke: «baldin eta aurrera egiten dugunean robotaren aurrean objektu bat aurkitzen badugu».

repeat instrukzioan bezala, *if* baldintza betez gero exekutatzen diren instrukzio guztiak bi ikurren artean kokatuko ditugu: *if* hasieran eta *end if* amaieran.

Egin aurrera eta pausatu galdera-ikurraren gainean. Adierazi programazioan zer egin behar den robotaren aurrean oztopoa aurkituz gero. Gero, eta erabaki-bloketik kanpo, osatu zer mugimendu egin behar dituen robotak erabakia exekutatu (edo ez exekutatu) eta gero.

16. ARIKETA. ERABAKIAK

Ariketa honetan, erabakia hartu eta gero exekutatu behar den programazioa ebatziko dugu.

Zailtasuna: erraza.

17. ARIKETA. ERABAKIAK

Tranpak egiteak ez du balio, amaierako gurutzeetako batzuetara iristen saiatuz erabaki-laukiagatik pasatu gabe.

Horrela jokatzek baduzu, programak ez digu proba onartuko.

Galderaren gaineko ekintza bat egin behar dugu *if* exekutatu baino lehen.

Zailtasuna: ertaina.

18. ARIKETA.- ERABAKIAK.

Ariketa honetan, erabakiak eta funtzioak lotuko ditugu. Bereizita ikusi ditugun egitura horiek guztiak bilduta ager daitezke programetan, pentsa ditzakegun konbinazio guztietan. Eta programazioa zailtzen du horrek. Horregatik, hona hemen aholku on bat: arazo handi bat badugu, hobe arazo txiki askotan zatitzen badugu; errazago programatuko dituzu eta zailagoa izango da hutsegite logikoak egitea.

Erabili robotaren mugimendua optimizatzeko funtzioak erabakiaren laukitxora iritsi arte, eta hartu gero zure ustez hartu beharreko erabakia.

AMAIERA

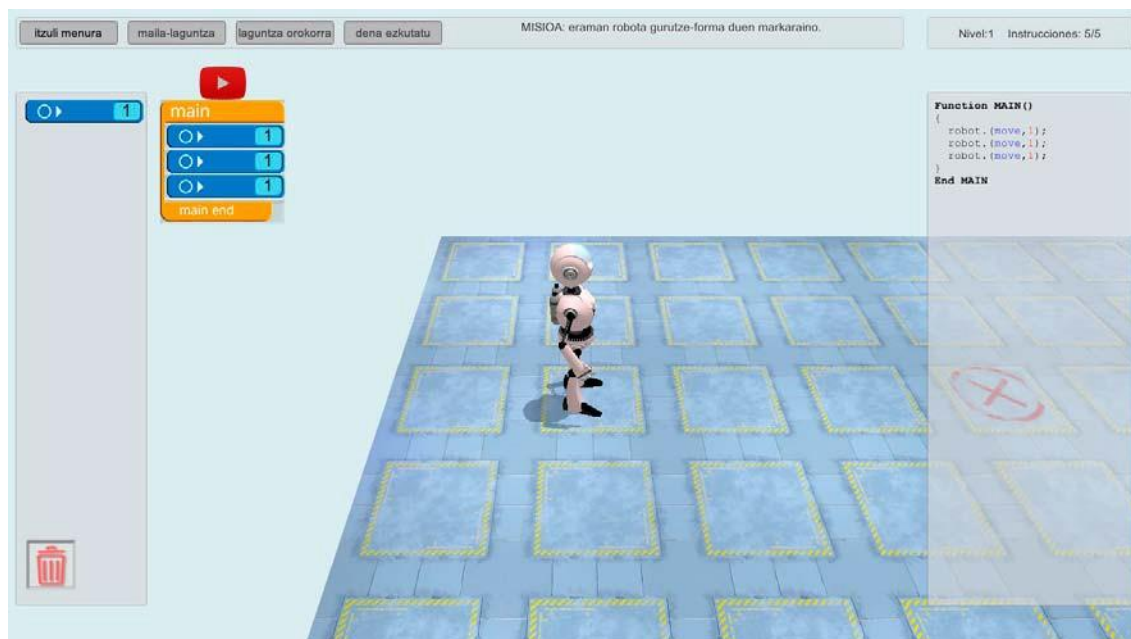
Gure ibilbidearen amaieraraino heldu zara. Ez da zaila izan eta programazio-lengoaia GUZTIEN oinarrian dauden egitura nagusiak ulertzeko balio izan dizu. Egin nahi duzun aplikazioak aginduko dizu zer lengoaia erabili beharko duzun. Jokoak sortu nahi badituzu, irudiak erabiltzea errazten duten lengoaiak erabili beharko dituzu; errealtate birtualeko aplikazioak egin nahi badituzu, 3D objektuak erabiltzeko aukera ematen dizun lengoaiaren bat erabili beharko duzu, eta gestiorako aplikazioa egin nahi baduzu, datu-baseak erabiltzeko prest dagoen lengoaia bat erabili beharko duzu.

Dena dela, lengoaia guzti-guztietan arazo berak aurkituko dituzu eta hemen ikasi dituzun egiturak erabili beharko dituzu haiek ebazteko.

Soluzioak:

Programazioan beti gertatzen den bezala, edozein arazotarako hamaika soluzio dagoen arren, irudi hauetan maila bakoitzerako soluzioetako bat aurkituko duzu.

1 MAILA.



8

2 maila:





9

3 maila:



10



10



6. maila:



11

7 maila:



8 maila:



12

9 maila:



10 maila:



13

11. maila: bi aukera.





14

12 maila:



13 maila: bi aukera.



16



16

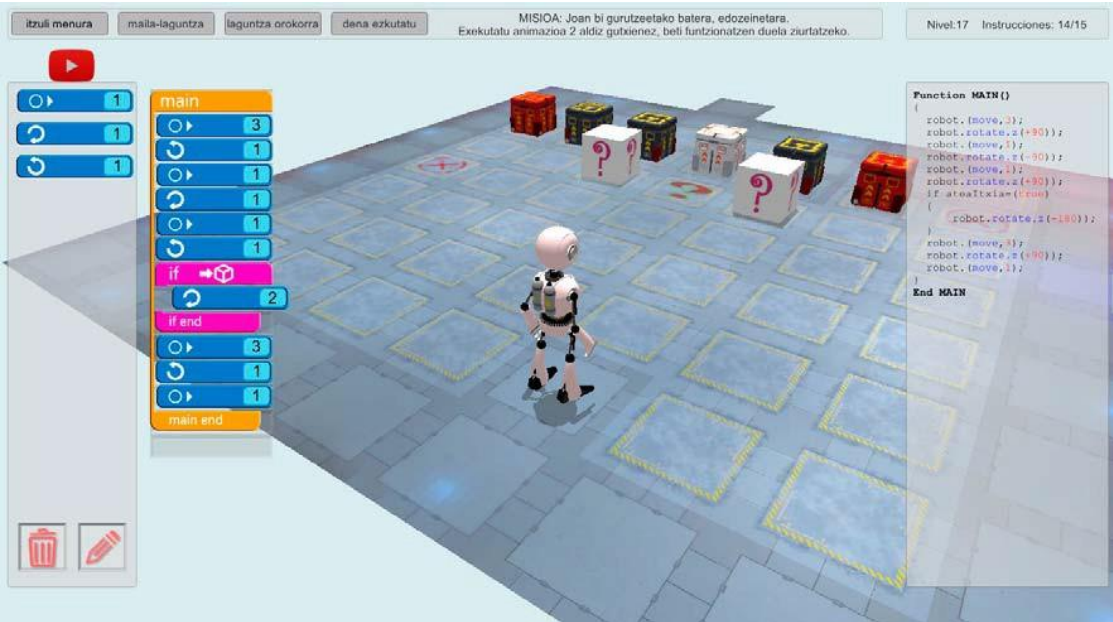


16 maila:



17

17 maila:



18 maila:

MISIOA: Joan bi gurutzetako batera, edozeinetara.
Exekutatu animazioa 2 aldiz gubanez, biki funtzionatzen duela ziurtatzeko.

Nivel:18 Instrucciones: 18/20

Function MAIN()
{
 functionA();
 functionA();
 functionA();
 robot.(move,1);
 if @tealtzia=(true)
 {
 robot.rotate.z(+90);
 robot.(move,1);
 }
 robot.(move,3);
}
End MAIN

Function A()
{
 robot.(move,1);
 robot.rotate.z(+90);
 robot.(move,1);
 robot.rotate.z(-90);
}
End A

Oharra: Erabili A, Q eta gezi-teklak kamera mugitzeko eta ariketaren ikuspegi hobea izateko (ikus laguntza orokorrean).