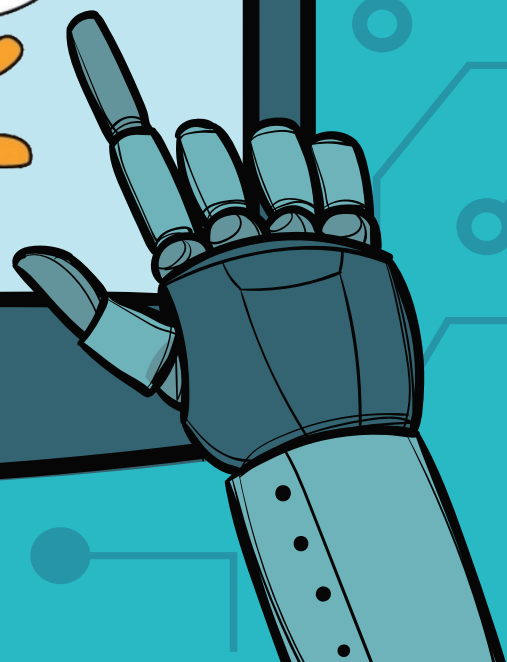
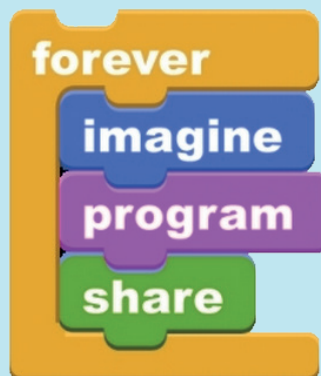


Mali programeri

NAUČI DA RAZMIŠLJAŠ  
PROGRAMERSKI

SCRATCH



**Autori:**

Biljana Popović

Janko Vukotić

Marija Mugoša

**Ilustracije, dizajn, prelom:**

Biljana Popović

**Tehnička pomoć:**

Ilija Ivanov

Skripta je nastala za potrebe realizacije Projekta Mali programeri koji je finansiran od strane Crnogorskog Telekomu u okviru konkursa "Za svako dobro".

Takođe, skripta se može koristiti kao pomoćno nastavno sredstvo za realizaciju Obrazovno-vaspitni ishod 2 u VIII razredu (Na kraju učenja učenik će biti osposobljen da izvrši korake algoritma/programa i napiše jednostavan algoritam/program sa zadatim skupom instrukcija).



za svako dobro



Udruženje profesora  
INFORMATIKE

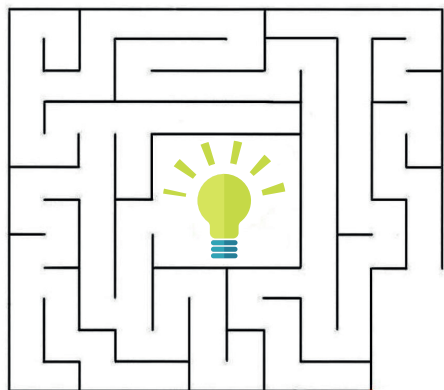
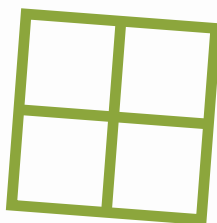
# ALGORITAM I PROGRAMIRANJE

Ako želimo da neko uradi nešto za nas možemo to uraditi na više načina. Npr. „gladan sam, možeš li mi napraviti sendvič“; „nacrtaj kuću“; „postavi to na onu policu“... onaj kome govori-mo uglavnom zna šta očekujemo da uradi i bez preciznih objašnjenja. Ali da bismo programi-rali kompjuter moramo zadati korake i to po redosledu kojim se izvršavaju. Koraci koje zadajemo prilikom izvršenja nekog zadatka čine ALGORITAM.



## Programer & robot

Kroz ovu igru uvježbaćeš davanje komandi koje robot treba da uradi. Dva učenika sjede jedan naspram drugog. Jedan učenik će biti programer i davaće uputstva kako da drugi učenik, robot slikar, nacrti prikazane sličice. Potrebni su papir i olovka.



## Izazov

Da li možeš žmureći naći put iz lavirinta, slušajući uputstva (gore, dole, lijevo, desno)?



Algoritam je ključni pojam u računarskoj obradi podataka jer je računarski program izvjestan algoritam koji računaru objašnjava koje korake (naredbe) i kojim redoslijedom treba da obavlja. Tako se algoritmom može smatrati bilo koji niz instrukcija koji se može uraditi na računaru - mašini.

Ovdje možemo doći do zaključka da računari, ma koliko često i mi sami rekli da su pametni, zapravo nijesu, jer ne mogu samostalno da razmišljaju. Računari su mašine koje prate određene naredbe i tako obavljaju određene zadatke.

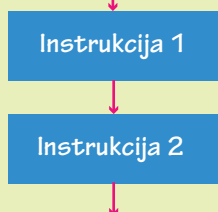
Pod algoritamskom (programskom) strukturom podrazumijevamo više koraka (komandi programskog jezika) koji čine jednu cjelinu. Postoje tri elementarne algoritamske strukture:

**Linijski algoritam** - izvršava se korak po korak, od prvog do posljednjeg, i svaki od koraka se izvršava tačno jedan put;

**Ciklični algoritmi** - u njima primjećujemo ponavljanja u zavisnosti od nekog uslova, u upotrebi je više cikličnih struktura: dok-WHILE; do-UNTIL; brojačka struktura-FOR.

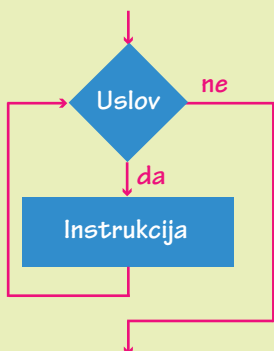
**Uslovni (razgranati) algoritmi** - to su rješenja zadataka koji zavise od nekog uslova. U uslovnim algoritmima neke naredbe će se izvršiti, a neke neće, u zavisnosti od ispunjenosti uslova.

### Linijski algoritam

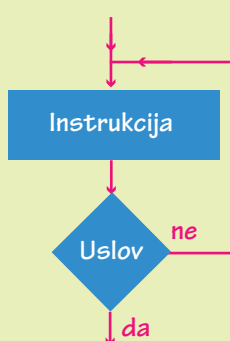


### Ciklični algoritam

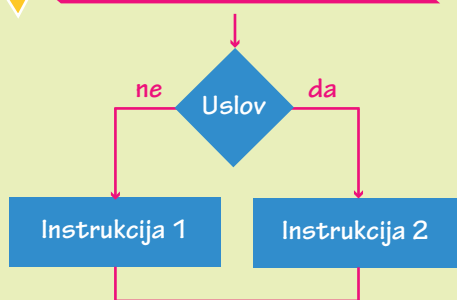
sa preduslovom



sa postuslovom



### Uslovni algoritam



Svako od nas može riješiti određeni problem na svoj način, tačnije može postojati više algoritama za rješavanje istog problema.

# VOŽNJA BIKIKLA

ČAS  
1

Da li možemo ispisati linijski algoritam za bezbjednu vožnju bicikla?

Jedan algoritam za vožnju bicikla može izgledati ovako:



Koliko još različitih rješenja (agoritama) ima za zamjenu gume? Probaj da daš svoje rješenje.

Šta će se desiti ako preskoči-mo neki korak u postupku mijenjanja gume?



## Zamjena gume

Treba provjeriti gume bicikla, ako je guma pukla zamijeni je novom. U suprotnom idi i vozi biciklo. Koju strukturu algoritma ćeš koristiti?



# SCRATCH

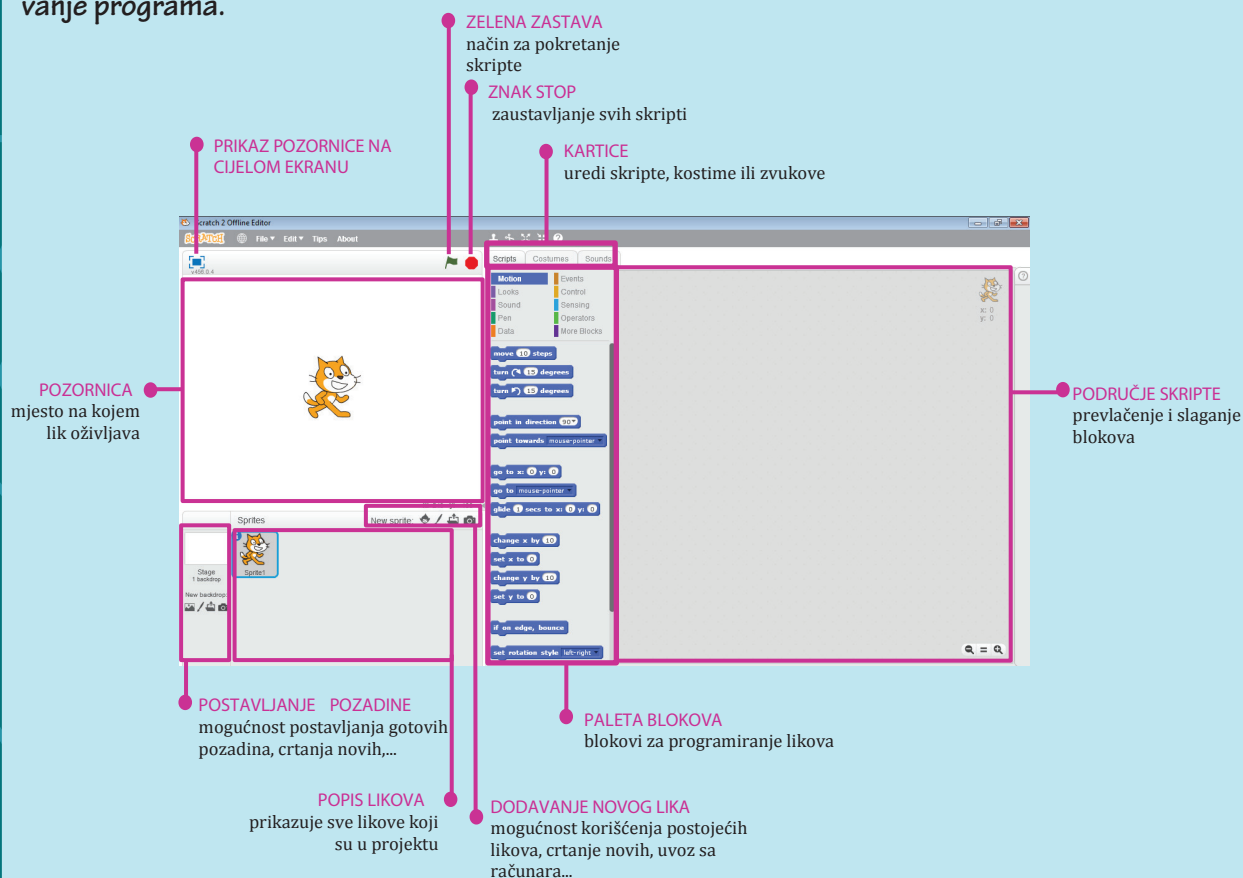
ČAS  
2

Scratch je grafički programski jezik koji omogućava lako kreiranje interaktivnih priča, igara i animacija kao i dijeljenje projekata sa drugim programerima preko weba. Scratch se koristi u više zemalja za početno učenje programiranja. Originalna uputstva i sam program mogu se preuzeti sa web sajta <https://scratch.mit.edu/>

## OSNOVNI ELEMENTI SCRATCH PROJEKTA

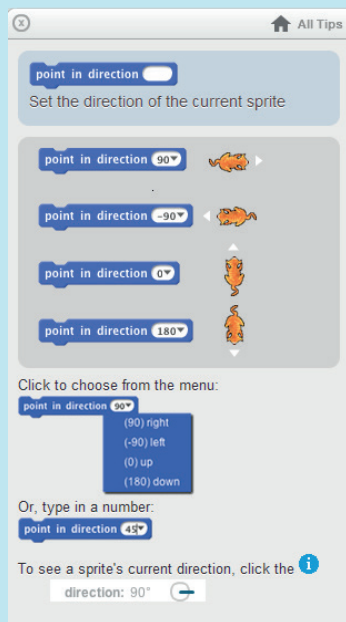
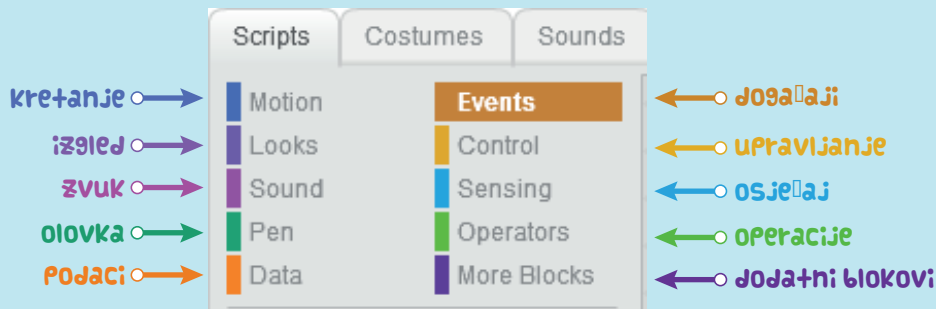
Scratch projekat sastoji se od objekata koji se nazivaju **LIKOVİ** (ili sprajtovi, oblici). Lik može da izgleda kao osoba, predmet, životinja ili bilo šta drugo. Novi lik je moguće izabrati iz biblioteke, nacrtati ga korišćenjem alatki za crtanje, učitati ga sa svog računara ili ga slikati kamerom. Na jednoj pozornici može se naći više likova. Liku se pridružuju instrukcije koje ukazuju kako će se kretati, kako će reagovati na druge likove i koji zvučni efekti će ga pratiti.

Opis ponašanja lika zove se **SKRIPTA** (to je u stvari program) i zadaje se slaganjem blokova jedan uz drugi u veće blokove (kao kada slažemo kockice ili puzzle). Kada se uradi dvoklik na skriptu, Scratch izvršava akcije zadate blokovima od vrha prema dnu. Ovaj proces je u stvari izvršavanje programa.



# SCRATCH BLOKOVİ

Da bi naš lik uradio ono što mi želimo, potrebno je prevlačiti odgovarajuće blokove iz palete blokova u polje za unos skripti. Blokovi su podijeljeni u 10 kategorija različitih boja.



Kako biste otkrili šta određeni blok omogućava, kliknite na njega desnim klikom miša, i odaberite opciju HELP na padajućem meniju.

Da bismo napravili skriptu, potrebno je da za aktivan objekat prenesemo odgovarajuće naredbe iz palete naredbi u oblast skripti, i povežemo ih.

Kada blok pomjerate u područje skripti, prikladno mjesto za spajanje s drugim blokovima biće označeno bijelom bojom.



touching mouse-pointer ?

move 10 steps

point in direction 90°

Neki blokovi imaju tekstualna polja unutar sebe, u kojima možemo unositi željene vrijednosti, dok će neki imati padajuće menije u kojima je potrebno izabrati neku od predefinisanih vrijednosti.

say Hello! for 2 secs

play sound meow

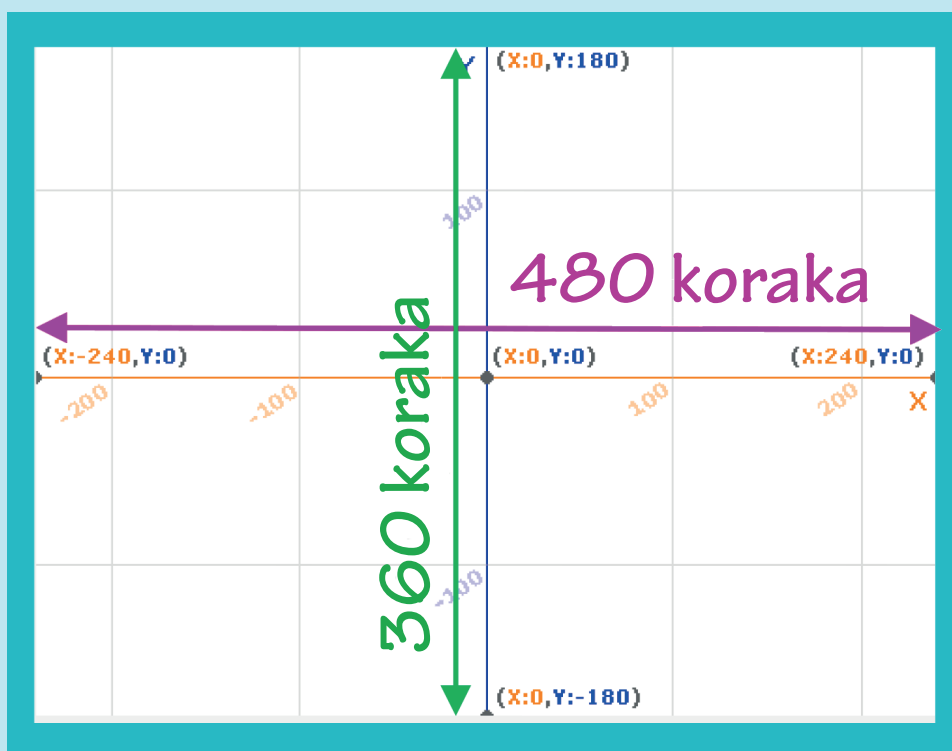
wait 1 secs

## POLOŽAJ LIKA NA POZORNICI

Kada započnemo novi projekat, lik je uvijek u sredini ekrana. Za odeđivanje tačnog položaja lika Scratch koristi Dekartov pravougli koordinatni sistem. Svaka tačka na pozornici određena je sa dva broja.

U Dekartovom koordinatnom sistemu definisane su dvije ose (x osa, y osa) i one su u Scratch-u podijeljene na korake. Centar koordinatnog sistema nalazi se u tački u kojoj su vrijednosti x-ose i y-ose jednake nuli. X osa predstavlja horizontalu, a Y osa vertikalu.

Na lijevoj strani X ose, X koordinata će biti negativna, dok će po desnoj strani biti pozitivna. Y koordinata ima negativnu vrijednost na donjoj strani Y ose, a pozitivnu na gornjoj.



### Poveži blokove

idi 10 koraka  
ako si na kraju, okreni se

move 10 steps  
if on edge, bounce

idi 20 koraka  
okreni se za 15°

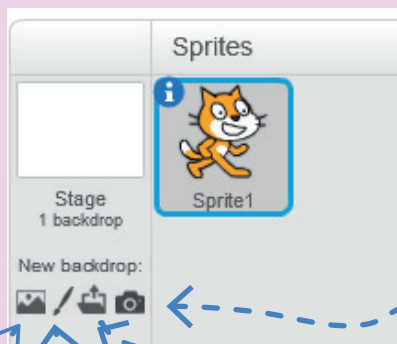
move 20 steps  
turn 15 degrees

idi 10 koraka  
okreni se ka strelici miša

move 10 steps  
point towards mouse-pointer



# ODABIR POZORNICE



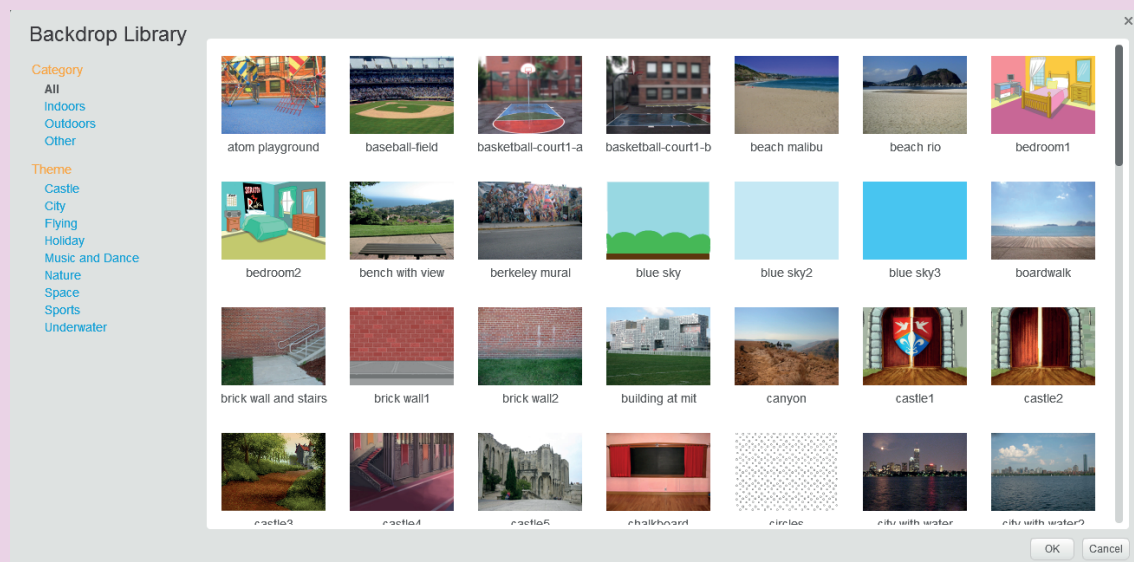
Sa lijeve strane lika nalazi se ikonica za podešavanje pozadine. Klikom na ikoncu možemo dodati neku od slika iz biblioteke, koje su podijeljene u kategorije. Takođe možemo i sami nacrtati pozornicu po kojoj želimo da se naš lik kreće.

Slikati sa aparata

Editor, za crtanje

UPLOAD, biranje sa računara

Biblioteka pozornica

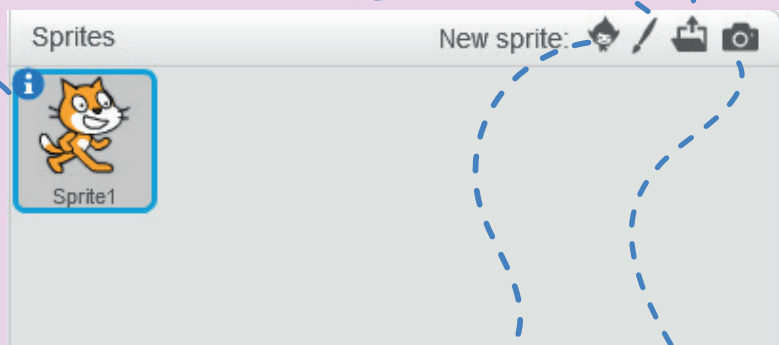


## BIRAM TEBE - izbor lika



Osnovni lik u Scratch projektu je mačka. Za kreiranje novih likova je potrebno kliknuti na neki od ovih dugmića. Moguće je nacrtati sopstveni kostim za lik na sceni korišćenjem editora za crtanje, uvesti kostim iz postojeće baze likova koji dolaze uz program ili je moguće prevući neki kostim direktno sa nekog Web sajta otvorenog u Web browser-u.

### Editor, za crtanje

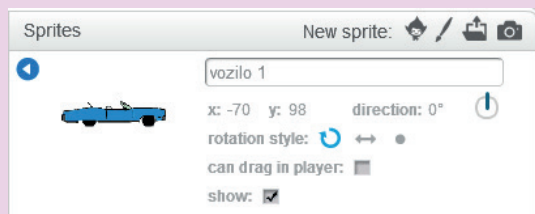


UPLOAD,  
biranje sa računara

Slikati sa  
aparata

Biblioteka  
likova

Informacije  
o liku



Brisanje lika možemo uraditi na dva načina, odabirom makaza s Alatne trake i klikom na lik, a druga mogućnost je desnim klikom miša na lika i odabir opcije delete.



# LETIM, NE STAJEM!

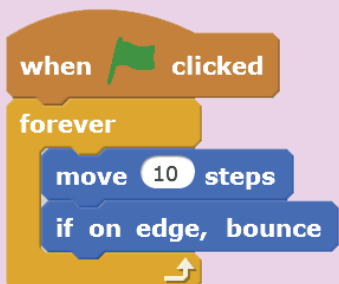
Probaćemo da napravimo program da leptir stalno leti. Iz biblioteke likova pronađi leptira i postavi ga na pozornicu. Nađi odgovarajuću pozadinu za pozornicu i kreni da slažeš svoj prvi program.

Program će se pokrenuti klikom na zelenu zastavicu. Leptir će se kretati po 10 koraka, sve dok ne dođe do ivice, zatim će se okrenuti i nastaviti u suprotnom pravcu. Za ovaj program nam je potrebna petlja **FOREVER** koja će se ponavljati beskonačno, tj. dok ne prekinemo program klikom na crveno dugme.

Pomoć za svaki blok je moguće dobiti izborom stavke HELP u kontekstnom meniju, koji dobijamo nakon desnog klika na blok.

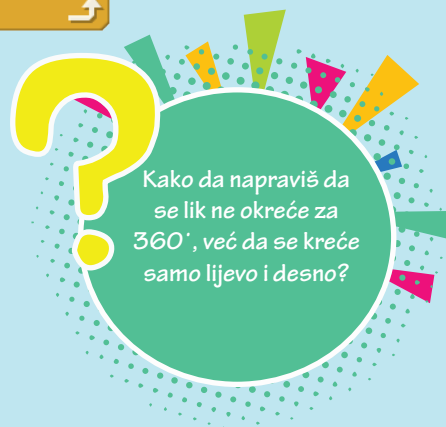
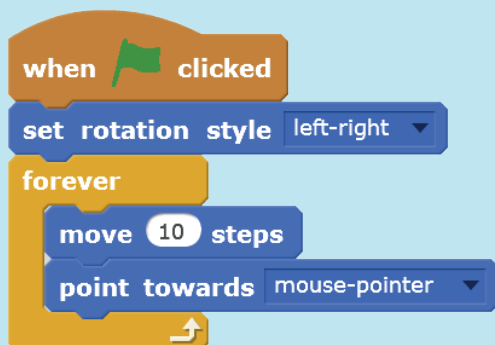
## Oblačići govora

Sada želimo da dodamo oblačić s govorom leptira u zadatom vremenskom okviru od 2 sekunde. Prvi oblačić želimo da kaže **Zdravo!**, a zatim drugi **Želiš li vidjeti moj let?** i nakon toga počinje let.



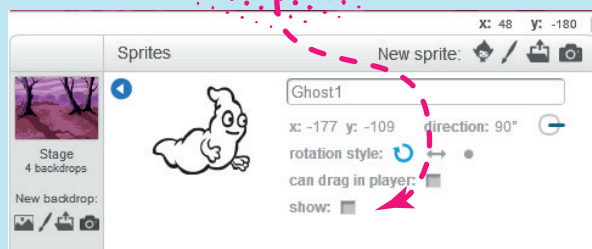
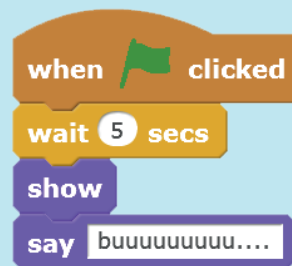
## PRATI ME

Napravićemo program u kome će lik pratiti kretanje strelice miša po pozornici. Program se pokreće na klik zelene zastavice.



## Ne želim biti sam!

I dok naš lik prati kretanje strelice po sceni, nakon 5 sekundi se pojavljuje novi lik na sceni i govori BUUUUUUU...



## SMJER STRELICE

Pokušaj da napraviš skriptu kojom se lik vodi po ekranu strelicama sa tastature.

Postavi kao pozadinu underwater3 iz biblioteke pozadina.

Liku ribe pridruži naredbu

when space key pressed

U zavisnosti od smjera kretanja (gore, dolje, lijevo i desno) lik se pomjera za po 20 koraka. Kada dođe do ivice, okrene se i nastavi u suprotnom pravcu.

when down arrow key pressed  
change y by -20  
if on edge, bounce

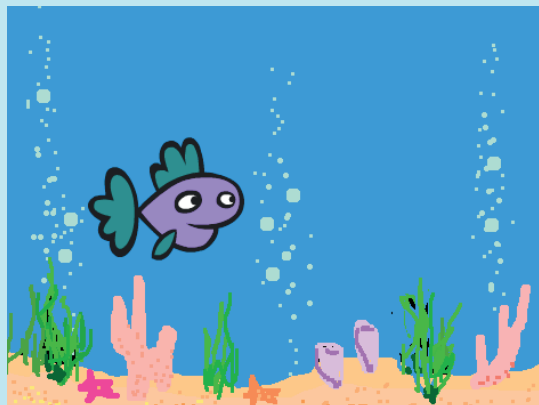
when up arrow key pressed  
change y by 20  
if on edge, bounce

when left arrow key pressed  
change x by -20  
point in direction -90  
if on edge, bounce

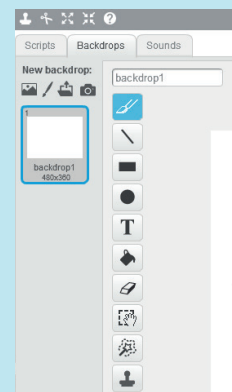
when right arrow key pressed  
change x by 20  
point in direction 90  
if on edge, bounce

Kako napraviti da se lik okrene u smjeru strelice?

## Docrtaj pozadinu



Nacrtaj par prepreka na postojećoj pozadini koje tvoj lik treba zaobići, krećući se strelicama po pozornici.



# GUŽVA U SAOBRAĆAJU

4

Pokušaj da napraviš skriptu u kojoj lik uspješno prelazi ulicu, dok se dva vozila kreću gore dolje neprestano. Ukoliko lik dođe u kontakt sa nekim vozilom igra staje.

Kako da napraviš da se lik ne okreće za 360°, već da se kreće samo lijevo i desno?

## Skripta za lika koji prelazi ulicu

when up arrow key pressed

change y by 10

when down arrow key pressed

change y by -10

when right arrow key pressed

point in direction 90

change x by 10

when left arrow key pressed

point in direction -90

change x by -10

when clicked

forever

if touching vozilo 1 ? or touching vozilo 2 ? then

stop all

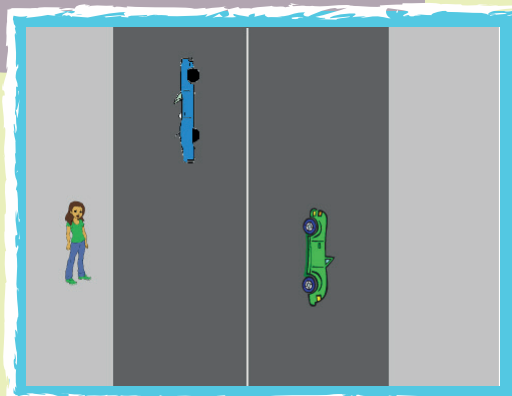
## Skripta za vozila

when clicked

forever

move 5 steps

if on edge, bounce





# IDEMO U OBILAZAK

5

Naš lik nas vodi u kratku šetnju. Za ovu aktivnost potreban nam je određeni lik i nekoliko različitih pozornica. Lik se kreće po pozornici pravolinijski, kada dođe do ruba postojeće pozornice, lik se vrati na početnu koordinatu i pozornica se promijeni. Na početku svake pozornice lik nas pozdravi i kaže nam gdje se trenutno nalazimo, a zatim krene u šetnju do sledeće destinacije.

Naredba  
REPEAT UNTIL  
opisuje ciklus čiji broj  
ponavljanja nije poznat, već  
će zavisiti od toga da li je  
ispunjen uslov za  
izlazak iz ciklusa.

when  clicked

forever

repeat until touching edge ?

move 5 steps

switch backdrop to beach malibu

go to x: -200 y: -65

think ja sam na plazi for 2 secs

repeat until touching edge ?

move 5 steps

switch backdrop to canyon

go to x: -200 y: -65

think ja sam u kanjonu for 2 secs

repeat until touching edge ?

move 5 steps

switch backdrop to basketball-court1-b

go to x: -200 y: -65

think ja sam na terenu for 2 secs

repeat until touching edge ?

move 5 steps

Pomjeraj se po 5 koraka dok  
ne dođeš do kraja pozornice.

Potrebno je da se pozornica  
promijeni svaki put kada lik dođe  
do ruba postojeće.

ja sam u kanjonu

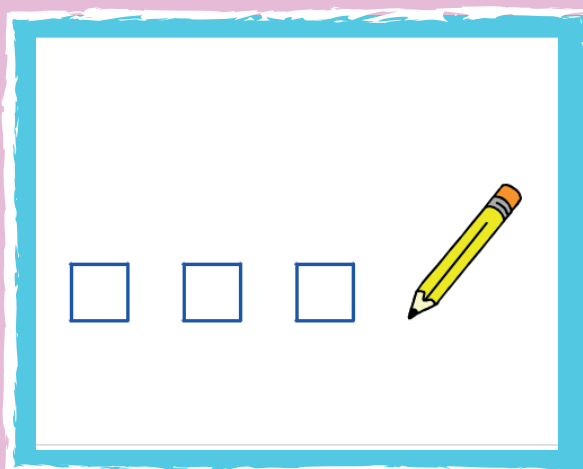
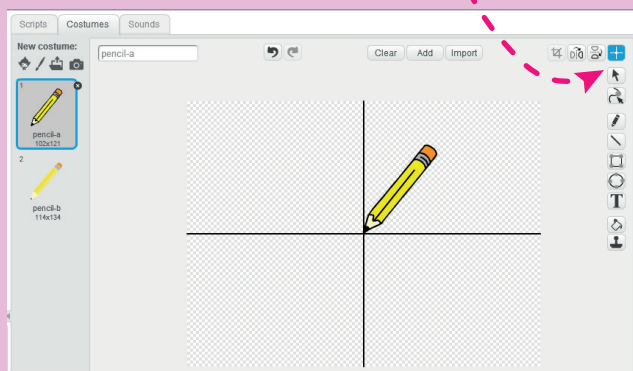
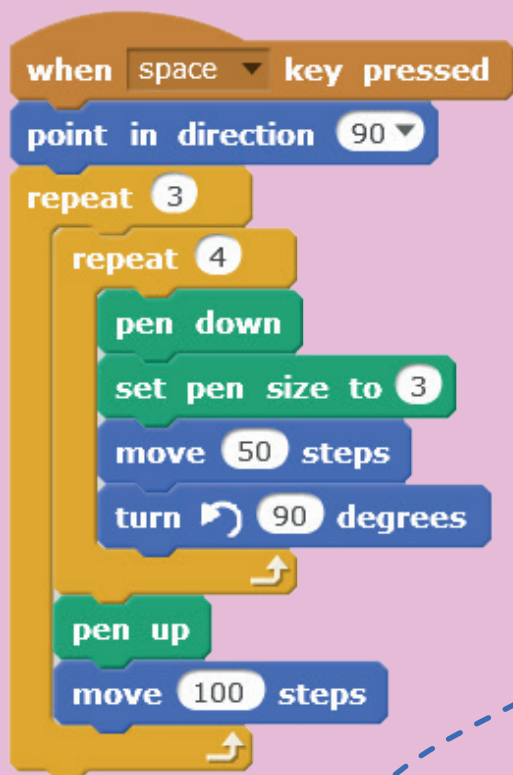


# CRTANJE OLOVKOM

Napravićemo program koji crta kvadrat. Za lika ćemo izabrati olovku iz biblioteke likova. Olovka je uvijek usmjerena na 90°. Crtaju se tri kvadrata, stranice dužine 50 koraka, sa razmakom od 100 koraka između kvadrata. Debljina olovke je 3.

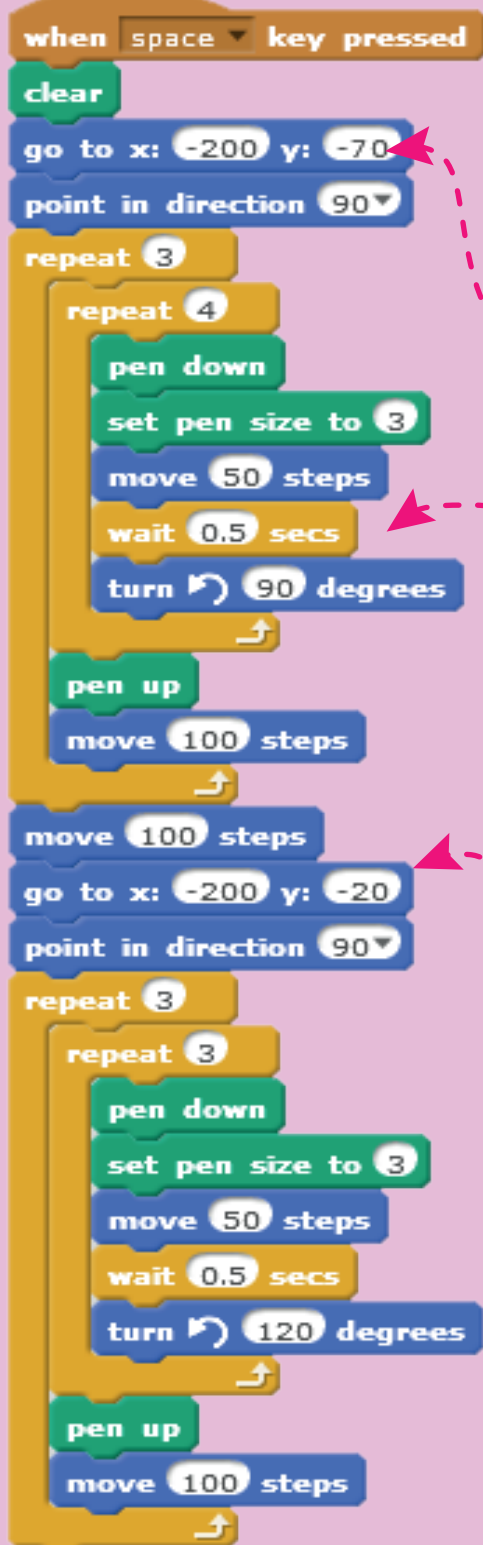
Za ovaj program nam je potrebna petlja **REPEAT**. Program se pokreće na taster **SPACE**.

Da bi olovka crtala vrhom grafitu biramo karticu **Costumes**, zatim **Set costume center** i pozicioniramo centar koordinatnog sistema u željenoj tački.



Da bismo znali pod kojim uglom se olovka okreće na svakom ćošku i crta sledeću stranicu, koristimo formulu  $180 - (n - 2) \times 180 \div n$ , gdje je  $n$  broj





## Tri kućice

Probaj da nadogradiš postojeći program tako da se pritiskom na taster SPACE nacrtaju tri kućice.

Ovaj blok pomjera lika do određene pozicije, u našem slučaju x: -200 y: -70.

Napravićemo malu pauzu od 0,5 sekundi da bismo vidjeli pokrete olovke na pozornici.

Potrebno je da olovka počne sa iste tačke po x: osi, dok po y: osi dodajemo 50 koraka stranice kocke i dobijemo -20.



# ČUVAJ SE LOPTE

Jedna od omiljenih igrica sve djece je hvatanje ili izbjegavanje određenih objekata, nakon čega dobijamo ili gubimo neke poene ili živote.

Hajde da napravimo da se naš lik kreće pomoću strelica lijevo-desno i izbjegava lopte koje nasumično padaju sa vrha.

Da bi igra bila zanimljivija dodaćemo i VARIJABLU/PROMJENLJIVU život, tj. da igra startuje sa tri života i svaki put ako lopta takne naš lik život se smanji za jedna bod. Kada se izgube svi životi igra prestaje i pojavljuje se tekst Game Over.



U kategoriji Data napravićmo novu varijablu i nazvati je Život.

New Variable

Variable name:

☒ For all sprites ☐ For this sprite only

OK Cancel

Scripts Costumes Sounds

Motion Looks Sound Pen Data&Blocks Events Control Sensing Operators Robots

Make a Variable

☒ **životi**

set **životi** to 0

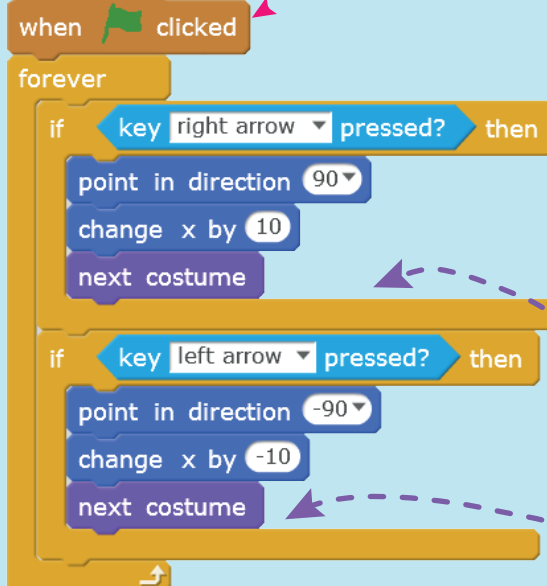
change **životi** by 1

show variable **životi**

hide variable **životi**

Make a List

Make a Block



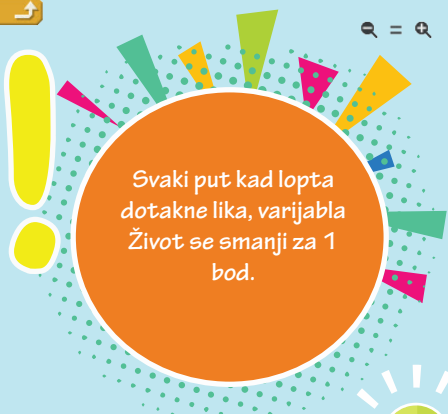
Da bi naš lik pomjerao određene djelove i na taj način zaista pravio prave pokrete kretanja, koristimo za promjenu kostima.

Blokovi koje dobijamo prilikom kreiranja varijable omogućavaju nam da pratimo rezultate, zadamo početne vrijednosti, mijenjamo vrijednosti u zavisnosti od potreba, ukonimo ili prikažemo.

```

when green flag clicked
  set zivoti to 3
  set y to 180
  set x to pick random -220 to 200
  forever
    change y by -10
    if y position < -170 then
      set y to 180
      set x to pick random -220 to 200
    if touching Avery Walking ? then
      change zivoti by -1
      set y to 180
      set x to pick random -220 to 200
  
```

Ovaj blok nasumično pozicionira našeg lika po X osi, tj. po horizontali kako bi lopta svaki sledeći put izlazila sa različite lokacije.



```

when green flag clicked
  set zivoti to 3
  wait 1 secs
  set y to 180
  set x to pick random -220 to 200
  forever
    change y by -10
    if y position < -170 then
      set y to 180
      set x to pick random -220 to 200
    if touching Avery Walking ? then
      change zivoti by -1
      set y to 180
      set x to pick random -220 to 200
  
```

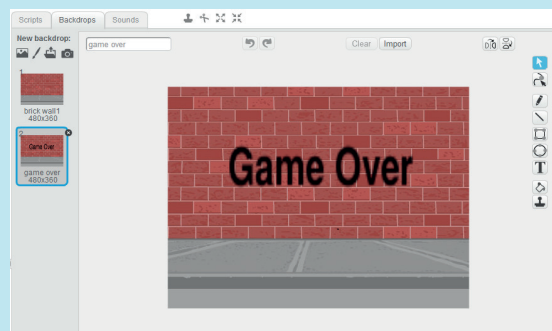
## Odluči kad je kraj!

I kad naš lik izgubi sve živote moramo ga obavijestiti da je igri došao kraj. Napravi još jedan STAGE koji će se pojaviti kada broj života bude jednak nuli i tada će sve radnje stati.

```

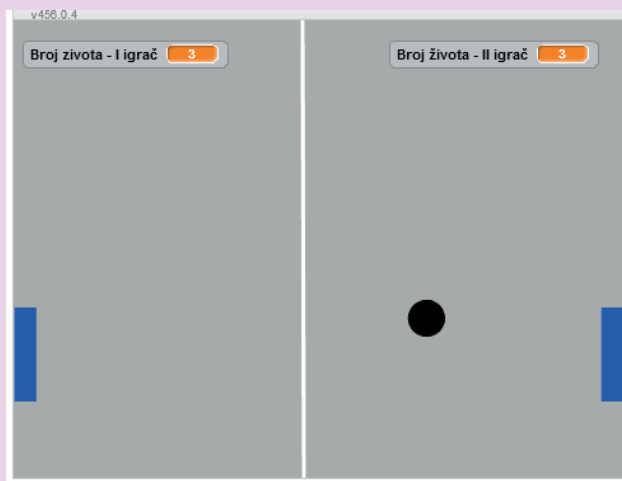
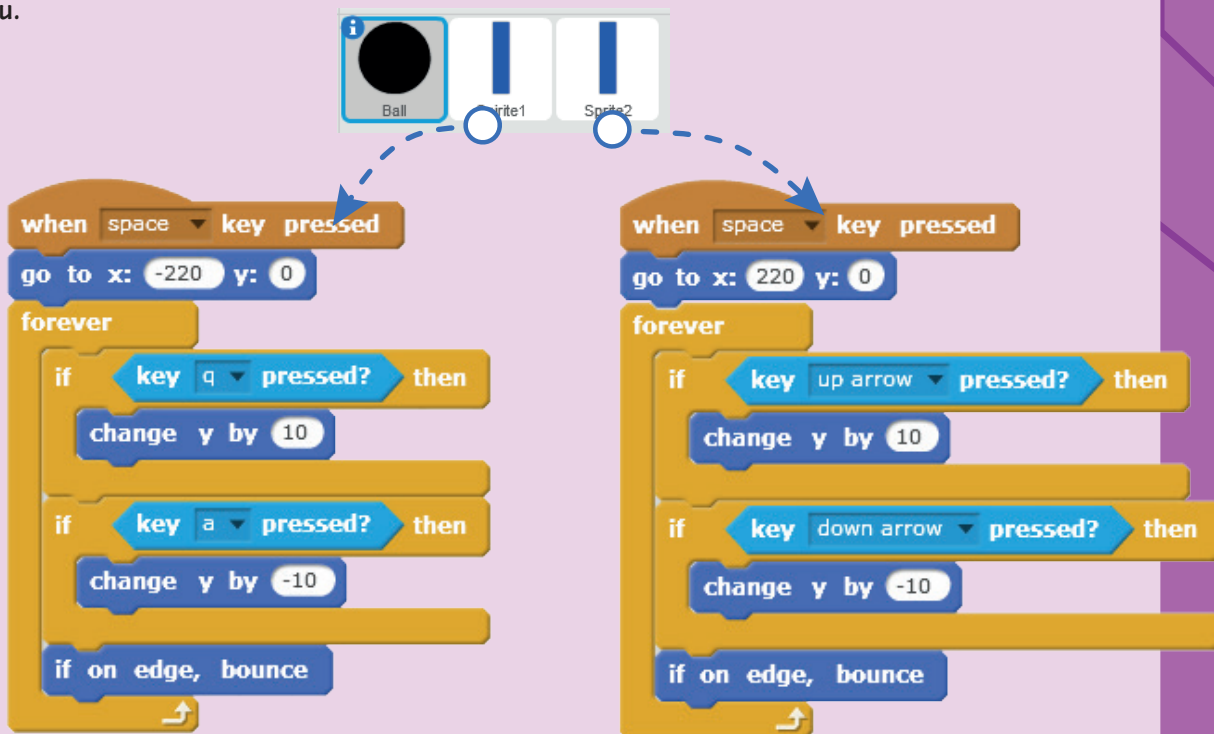
when green flag clicked
  switch backdrop to brick wall1
  wait until zivoti = 0
  switch backdrop to game over
  stop all
  
```

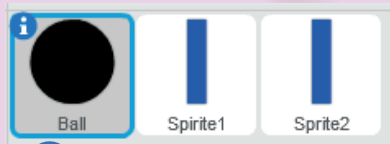
Svaka naredna lopta ima istu skriptu, samo je razlika u bloku WAIT koji pravi pauzu između pojavljivanja lopti.



# ICE HOCKEY

Napravi igricu Ice Hockey tako što ćeš na suprotnim stranama terena postaviti dvije pločice za dva igrača. Pločice sam nacrtaj u [Editoru za crtanje](#). Igrači se pokretima pločica (dolje-gore) dodaju hokejaškim pakom (lopticom), sa mogućnošću da se loptica nasumično odbija od ivica terena. Oba igrača imaju mogućnosti za po tri promašaja loptice (tri života), nakon čega gube partiju.





when space key pressed

switch backdrop to backdrop1

set x to 0

set y to 0

point in direction pick random 90 to 180

set Broj života - I igrač to 3

set Broj života - II igrač to 3

forever

move 10 steps

if on edge, bounce

if touching Sprite1 ? or touching Sprite2 ? then

play drum 13 for 0.1 beats

if x position < -220 and not touching Sprite1 ? then

change Broj života - I igrač by -1

if x position > 220 and not touching Sprite2 ? then

change Broj života - II igrač by -1

if Broj života - I igrač = 0 or Broj života - II igrač = 0 then

switch backdrop to backdrop2

stop all

Svaki put kada loptica udari u pločicu čuje se neki zvučni signal.

# LAVIRINT

9

Gobo ima zadatak da stigne do zelene zastavice. Svaki put kad takne crveni zid ili neku od buba vrati se na početak.

Nacrtaj u Editoru za crtanje pozornicu sa preprekama (zidovima), kao i pozornicu koja će se pojaviti kada Gobo stigne do zelene zastavice, kao povratna informacija igraču da je uspio.

Možeš li dodati zvučne efekte u ovoj igrici?  
Neka se čuje neki ton svaki put kad Gobo takne zid ili bubu.

when clicked

switch backdrop to costume1

set x to -221

set y to -162

forever

if key right arrow pressed? then

point in direction 90

change x by 5

if key left arrow pressed? then

point in direction -90

change x by -5

if key up arrow pressed? then

change y by 5

if key down arrow pressed? then

change y by -5

if touching Beetle ? or touching Beetle2 ? or touching Beetle3 ? or touching color red ? then

set x to -221

set y to -162

if touching color green ? then

switch backdrop to backdrop1

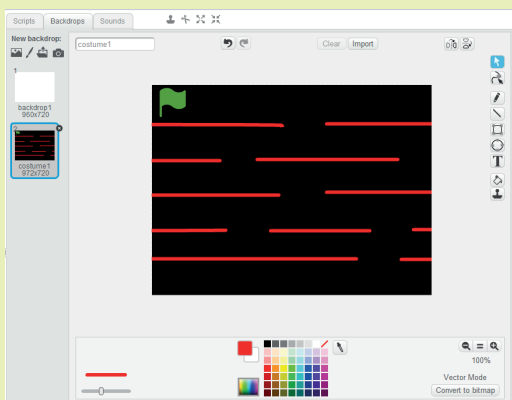
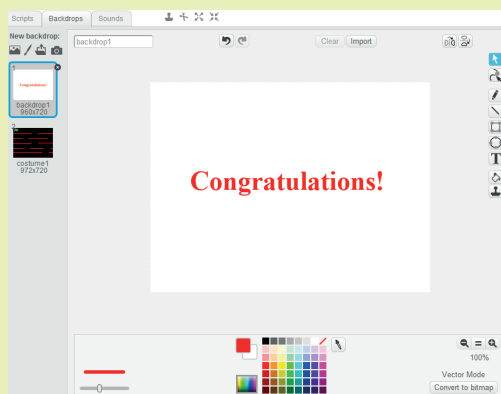
Gobo Beetle Beetle2 Beetle3

Ukoliko naš lik dotakne neku od buba koje se kreću kao prepreka scenom ili sami zid lavirinta, vratićemo ga na početak.



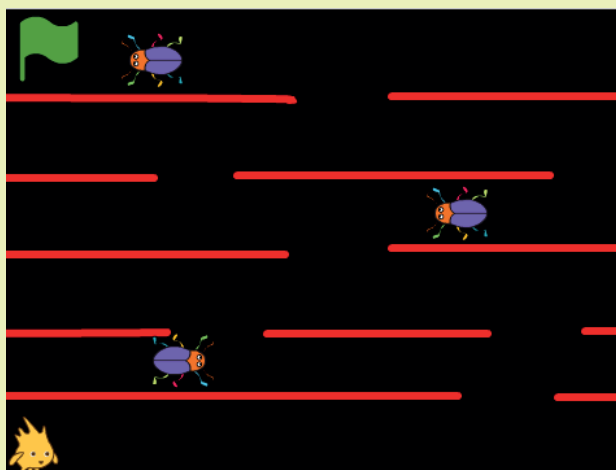
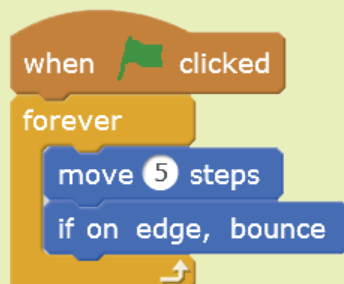
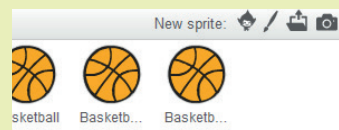
## Nacrtaj svoju pozornicu

Pokušaj da napraviš pozornicu koji će sa satojati od linija, debljinu i boju odlaberi sam. Vodi računa da otvor mora biti dovoljno velik da lik svojom veličinom može proći kroz njega, a opet ne prevelik da bude previše lako za proći ;)



## Skripta za bube

Dodaj na par mjesta novog lika, koji će se kretati lijevo-desno i na taj način otežavati igraču da stigne do zelene zastavice.



Pokušaj da dodaš zvuk koji se čuje svaki put kad lik dotakne zid lavirinta.

