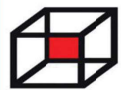




IVAN ŠIMAC
RUŽICA KAMENJAŠEVIĆ

SCRATCH

PRIRUČNIK



ZAJEDNICA TEHNIČKE KULTURE
PRIMORSKO-GORANSKE ŽUPANIJE



REPUBLIKA HRVATSKA
Ministarstvo znanosti i obrazovanja

Ivan Šimac
Ružica Kamenjašević

SCRATCH

PRIRUČNIK



REPUBLIKA HRVATSKA
Ministarstvo znanosti i
obrazovanja

Rijeka, 2022.

Ivan Šimac, mag. educ. inf.
Ružica Kamenjašević, dipl. ing.

PRIRUČNIK SCRATCH

Recenzija

Dr.sc. Sabrina Šuman

Lektura

Tamara Cikuša, mag. educ. philol. croat.
i mag. educ. hist.

Dizajn, slog i prijelom

Element d.o.o. Zagreb

Izrada i izdavanje priručnika financirana je sredstvima Ministarstva znanosti i obrazovanja Republike Hrvatske u sklopu projekta „Umrežimo s(v)e“ u području izvaninstitucionalnoga odgoja i obrazovanja djece i mladih u školskoj godini 2021./2022.

Tisak

Element d.o.o. Zagreb

Predgovor

Draga učenice,

Dragi učenice,

Pred tobom se nalazi priručnik za osnovno ovladavanje programskim jezikom Scratch.

Programiranje je vještina koja je u današnjim danima vrlo tražena. Budućnost donosi potrebu za još većim brojem programera.

Programirati nije jednostavno. Samim time moraš znati da je potrebno mnogo vježbe da bismo ovladali vještinom programiranja. Mnogi programeri svoje su znanje stekli metodom „pokušaja i pogreške“. Neka ta metoda bude i tvoj način rada s računalom. Neka te ne obeshrabre mnoge greške koje će tvoji programi imati. Veliki broj grešaka znači da marljivo radiš. Jer onaj koji ne radi, nema niti grešaka.

Nakon nekog vremena, imat ćeš još više grešaka. Ali one više neće biti osnovne, već će sigurno biti puno složenije, baš kao i tvoji programi.

Marljivo radi, griješi i ispravljaj.

Budi sigurna/siguran da ćeš ovladati jednom od najtraženijih vještina današnjice.

Ne boj se programerskih grešaka, samo tako možeš postati vrhunska programerka/programer.

Autori

Sadržaj

1. Programiranje i algoritam	7
2. Scratch	9
2.1. Izgled glavnog prozora	10
2.2. Pokretanje i zaustavljanje programa.....	11
2.3. Promjena i brisanje lika	11
2.4. Promjena pozadine.....	11
2.5. Naredbe	12
3. Projekti	16
3.1. Nacrtaj kvadrat.....	16
3.2. Rak koji pleše	17
3.3. Izradi čestitku	18
3.4. Igra skrivača.....	20
3.5. Izradi kviz.....	22
3.6. Pile koje preskače jaja	24
3.7. Igrica odbijanja od prepreke	27
3.8. Ja u igrici pucanja balona	29
3.9. Pričajmo sa Scratch-om.....	31
3.10. Zaspirajmo instrumente	32
3.11. Tvoj projekt	36
4. Literatura	37
Bilješke	38



1. Programiranje i algoritam

Kako bismo mogli krenuti izučavati vještinu programiranja, moramo znati što nam ta riječ znači. Iako će svatko od nas izreći smisao programiranja svojim riječima, što je svakako i cilj nastave, ovo je pravo mjesto da damo formalniju definiciju programiranja.

Programiranje je skup radnji koje za cilj imaju stvoriti program koji će se izvršavati na stroju (računalu, mobitelu, automobilu, ...). Skup radnji uključuje proces modeliranja okoline, pisanje ključnih riječi određenog programskog jezika, testiranje napisanih ključnih riječi (drugi riječima – testiranja samog programa), otklanjanje pronađenih grešaka te daljnje održavanje i promjenu programa u ovisnosti o promijenjenim uvjetima gdje se program koristi.

Izrecimo to i skraćeno. Programiranje je postupak pisanja naredbi u određenom programskom jeziku koji poznaje točno zadane riječi kako bismo dobili program koji će izvršiti ono što se od njega očekuje.

Rekli smo, točno što se od njega (programa) očekuje. Očekuje li se od vas da uspješno napišete zadaću? Svakako, sigurno ste u sebi odgovorili.

Svi imamo isti cilj. Dobro napisati zadaću. A što ako netko od nas zadaću krene pisati odmah nakon nastave? A netko se prvo odmori, pa tek onda prione poslu. A netko prvo ode na trening i tek tada krene pisati zadaću. Hoće li svi oni uspješno odrediti zadaću?

Ako zanemarimo moguće nesnalaženje u gradivu i pretpostavimo kako vi posjedujete jednako znanje, tada da, svi će uspjeti i doći do krajnjeg cilja – dobro napisana zadaća.

Kako sada to? Raznim putevi a isti konačni cilj.

E, dragi čitatelju, to je ljepota programiranja. Do jednog rješenja možemo doći na onoliko načina koliko nas ima. Postavljanjem jednog zadatka u učionici, dobili smo toliko različitih rješenja koliko je vaših predivnih glavica.

Ako želimo definirati taj postupak, tada ćemo ga nazvati algoritmom.

Algoritam je skup precizno definiranih postupaka koje moramo obaviti točno zadanim redoslijedom kako bismo došli do rješenja. Nemojte se zabuniti, rekli smo da imamo mnogo različitih rješenja koje nas dovode do istog rješenja – to znači da za jeda problem imamo mnogo različitih algoritama.

Jedan od klasičnih primjera prikaza algoritma jest kuhanje čaja. U tablici ćemo dati prikaz dva različita algoritma s točno definiranim koracima koji nas moraju dovesti do ispijanja toplog čaja.

Uvidjet ćete kako imamo različite korake, ali jednak rezultat.

Algoritam 1	Algoritam 2
Otvori ormar i uzmi posudu za vodu	Otvori ormar, uzmi posudu za vodu, vrećicu čaj i šalicu
Dođi do slavine i ulij vodu u posudu	Stavi vrećicu čaja u šalicu
Upali štednjak i stavi vodu da se zagrije do ključanja	U posudu ulij vodu
Otvori ormar i uzmi šalicu	Posudu stavi na štednjak i upali ga
Otvori ormar i uzmi omiljeni čaj	Pričekaj da voda proključa
U šalicu stavi vrećicu s čajem	Kada voda proključa, ulij vodu u šalicu i popij svoj čaj
Kada voda proključa, ulij vodu u šalicu	
Pričekaj nekoliko minuta i popij fini čaj	

Možete li uvidjeti ljepotu programiranja? Toliko puteva a isti cilj. Idemo putovati programiranjem zajedno. Do istog cilja, raznim putevima.



2.

Scratch

Programskih jezika ima mnoštvo, oko 700 različitih. Naučiti bilo koji programski jezik je lako, ali samo ako smo ovladali logikom programiranja. Logika programiranja znači da moramo shvatiti kako računalo razmišlja. Ukoliko to shvatimo, unutar godine dana možemo ovladati s nekoliko programskih jezika. Početniku ne treba biti cilj ganjati različite programske jezike. Cilj je zaustaviti se na jednom, dobro razumjeti logiku programiranja jer tu leži moć za daljnje učenje.

Mi ćemo se pozabaviti Scratch-om.

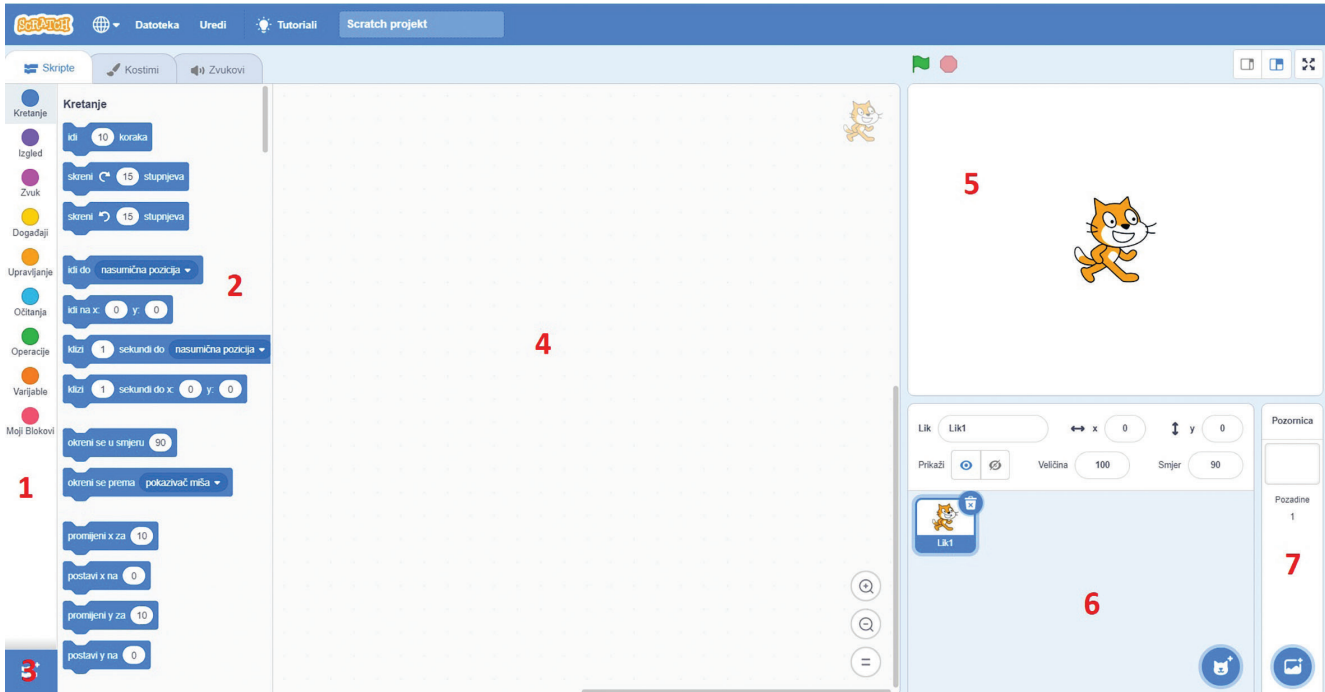
Scratch programski jezik pojavio se 2003. godine. Svjetski je raširen i koristi ga veliki broj djece u početnom stadiju učenja programiranja.

Popularnost leži u tome što se programiranje svodi na povlačenje blokova koje stvara konačni proizvod – toliko željeni program. Uz to što je vizualni programski jezik, preveden je i na hrvatski jezik.

Blokovi uvelike pomažu jer su naredbe obojane različitim bojama. Svaka boja ima određenu svrhu. Također, blokovi imaju i specifičan izgled na koji se može vezati samo određeni blok, što početnicima pomaže u odabiru pravilnog slijeda naredbi. Još jednom napomenimo, pravilni slijed naredbi jednako algoritam.

Scratch možemo koristiti kao Web2.0 alat (dostupan online, bez preuzimanja, kroz preglednik) ili kao program koji instaliramo na računalo pa ga pokrećemo lokalno, bez potrebe za mrežom.

2.1. Izgled glavnog prozora



Objasnimo koji dio označavaju brojevi.

- 1 – Blokovi naredbi, lako ih prepoznamo po bojama
- 2 – Naredbe unutar jednog bloka
- 3 – Odabir dodatnih naredbi (glazba, olovka, umjetna inteligencija)
- 4 – Područje za slaganje programa (dodavanje blokova – naredbi)
- 5 – Pozornica, tu se odvija naš program
- 6 – Područje za manipulaciju likovima
- 7 – Područje za manipulaciju pozadine pozornice



2.2. Pokretanje i zaustavljanje programa

Tijekom pisanja programa, svakako je dobro provjeriti koliko smo uspješno izradili program. Da bismo to mogli, program pokrećemo pritiskom na zelenu zastavicu. Ukoliko program želimo zaustaviti, tada pritisnemo crveni gumb.

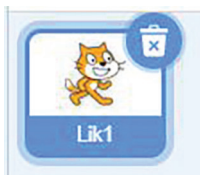


Potrebno je nadodati i naredbu koja će učiniti da zelena strelica postane aktivna, a to možemo učiniti dodavanjem naredbe iz bloka događaji.



2.3. Promjena i brisanje lika

U dijelu prozora gdje nam je omogućeno promjena lika, klikom na gumbić možemo promijeniti lik. Moguće je učitati i svoju fotografiju.



Klikom na koš za smeće, lik koji nam nije potreban, uklanja se.

Lik se dodatno može povećati ili smanjiti pomoću gumbića veličina. Gumbić smjer zaokrenut će ga u nekom smjeru. O tome ćemo više kasnije u priručniku.

2.4. Promjena pozadine



Klikom na ovaj gumbić, možemo promijeniti pozadinu gdje nam se nalazi lik. Također, kao i kod lika, postoji opcija učitavanja vlastite pozadine.

2.5. Naredbe

U ovom ćemo dijelu prokomentirati najčešće korištene naredbe. One su složene po blokovima i obojane su posebnim bojama.

2.5.1. Kretanje

Kretanje

idi 10 koraka

pomakni se onoliko koraka koliko smo napisali

skreni 15 stupnjeva

skreni udesno određeni broj stupnjeva

skreni 15 stupnjeva

skreni ulijevo određeni broj stupnjeva

idi do nasumična pozicija

pojavi se negdje na ekranu; svaki puta negdje drugdje

idi na x: 0 y: 0

pojavi se na točno definiranoj poziciji; 0,0 označava sredinu

klizi 1 sekundi do nasumična pozicija

putuj određeni broj sekundi do slučajne (nasumične) pozicije na ekranu

klizi 1 sekundi do x: 0 y: 0

putuj određeni broj sekundi do točne pozicije na ekranu

okreni se u smjeru 90

okreni lik u određenom smjeru

okreni se prema pokazivač miša

prati pomak miša

promijeni x za 10

pođi desno za deset koraka

postavi x na 0

ako gledamo lijevi i desni dio pozornice, budi točno na sredini

promijeni y za 10

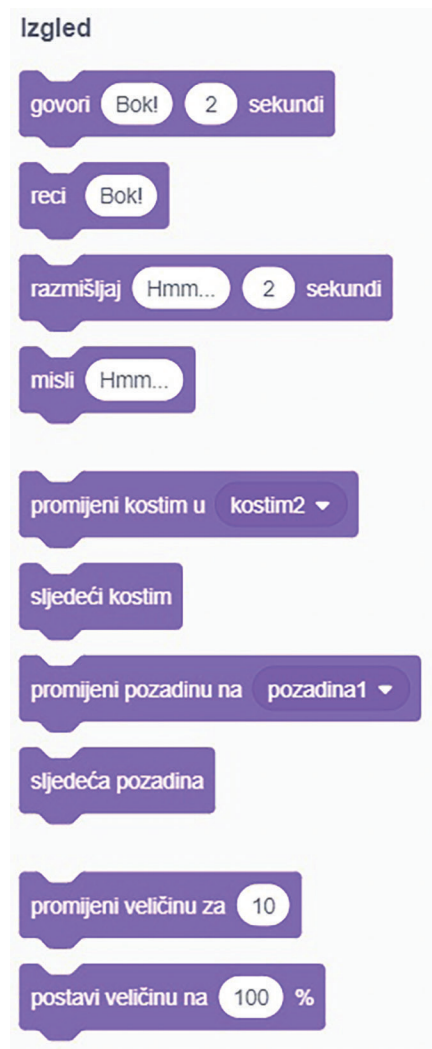
pođi gore za deset koraka

postavi y na 0

ako gledamo gornji i donji dio pozornice, budi točno na sredini



2.5.2. Izgled



govori nešto određeni broj sekundi

govori nešto zauvijek (dok ne dođe nova naredba za govor)

slična kao gorna naredba, drugačiji oblačić

natpis do nove naredbe za govor ili razmišljanje

promjena kostima lika

promjena kostima u iduću koji dolazi prema rasporedu u Scratchu

promjena pozadine gdje se nalazi lik

promjena pozadine u iduću koja dolazi prema

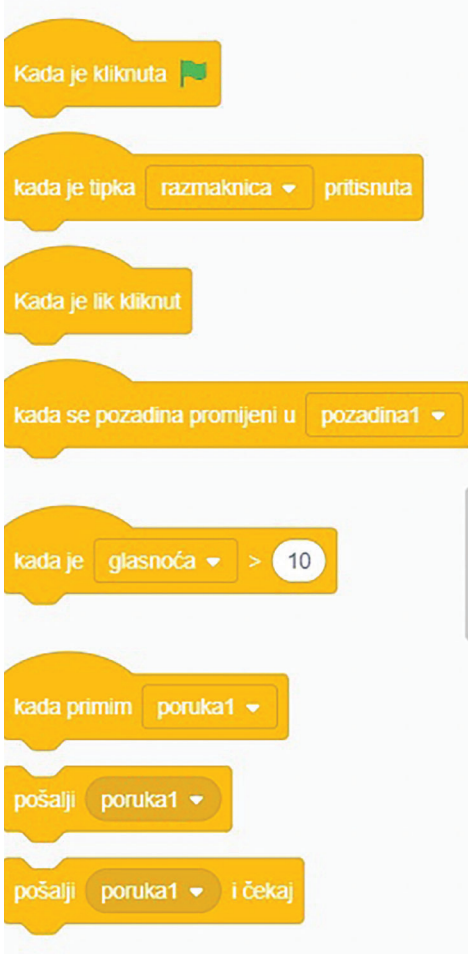
promjena veličine lika u pikselima (jednak omjer kao idi 10)

promjena veličine u postocima (ako je nešto 1 cm, nakon naredbe bit će 1.1 cm)

2.5.3. Zvuk

Naredbe zvuka su shvatljive same po sebi pa ih nećemo posebno objašnjavati.

2.5.4. Događaji



pokreće program; potrebno je kliknuti na zastavicu iznad lika

program pokreće tipka koja je odabrana

program pokreće klikanje na lika

idući korak slijedi nakon promjene pozadine

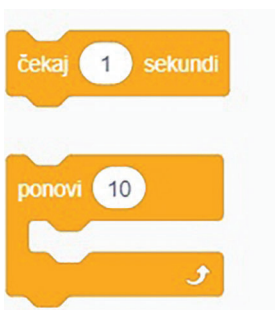
ako je zadovoljen uvjet, tada se pokreće druga naredba

ako je dobivena poruka, slijedi novi korak (naredba)

pošalji poruku liku kao okidač za daljnju naredbu

pošalji poruku liku i čeka njegov primitak za daljnju akciju

2.5.5. Upravljanje



prije iduće akcije (naredbe) čekaj određeni broj sekundi

petlja; ponovi točno poznati broj puta



petlja; ponavljaj nepoznati broj puta; prekini ovisno o nekom uvjetu



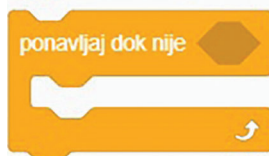
grananje; ako se zadovolji neki uvjet, zada naredba



grananje; ako se zadovolji neki uvjet, tada naredba; inače druga naredba



čekaj dok se ne zadovolji određeni uvjet



petlja; ponavljaj dok nije zadovoljen uvjet

Upravljanje nam je jako važno. Petlja je algoritamska struktura koja će nekoliko puta ponoviti sve naredbe. Mi navedemo naredbe jednom, a petlja ih ponovi onoliko puta koliko nam treba.

Struktura ako onda znači da provjeravamo neki uvjet i ako je on istinit, tada će se izvršiti onaj dio naredbi pod onda. Ako pada kiša, onda ću uzeti kišobran.

Struktura ako onda inače znači da ako je uvjet istinit, tada se izvršava onda. Ako je uvjet lažan (nije zadovoljen), tada se izvršava inače.

Ako pada kiša, uzet ću kišobran. Inače neću. Pogledate kroz prozor i vidite da je sunce što znači da ne uzimate kišobran.

Do ostalih ćemo izbornika (blokova naredbi) doći kasnije tijekom praktičnih zadataka.

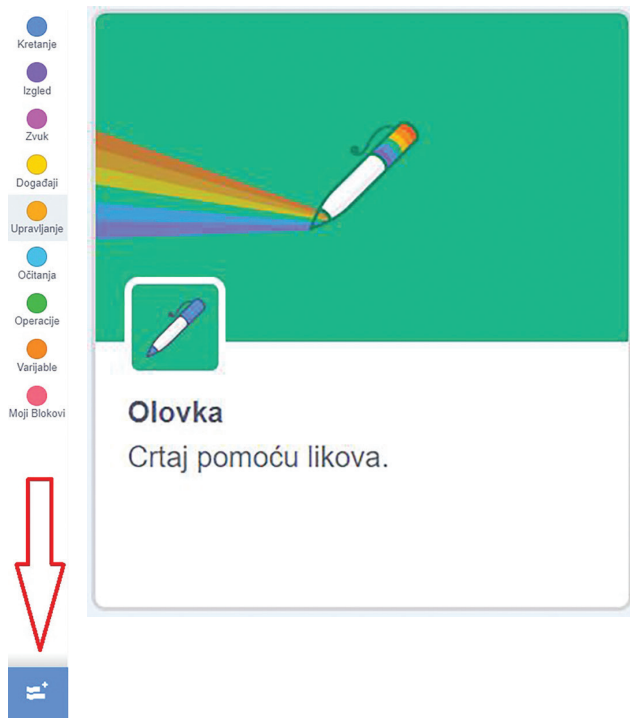
3.

Projekti

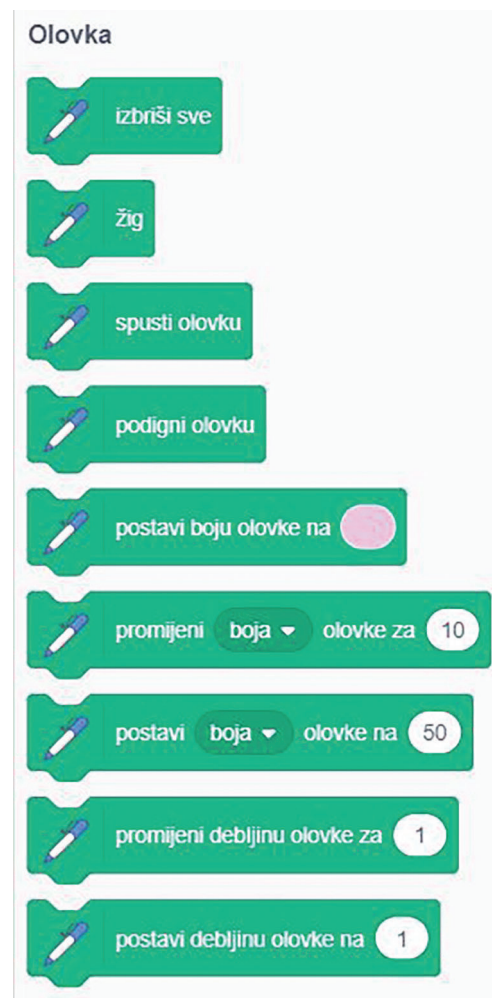
3.1. Nacrtaj kvadrat

Razmislimo. Potreban nam je lik, olovka koja će crtati i petlja koja će jedno crtanje linije ponoviti četiri puta. Kvadrat ima četiri kuta po 90 stupnjeva.

Do olovke dolazimo na sljedeći način:



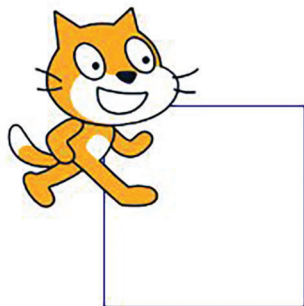
Nakon toga, dobivamo sljedeći izbornik.



Svakako prvo probaj samostalno riješiti zadatak. Nakon nekoliko pogrešaka, posavjetuj se s dijelom rješenja.



Rješenje.



3.2. Rak koji pleše

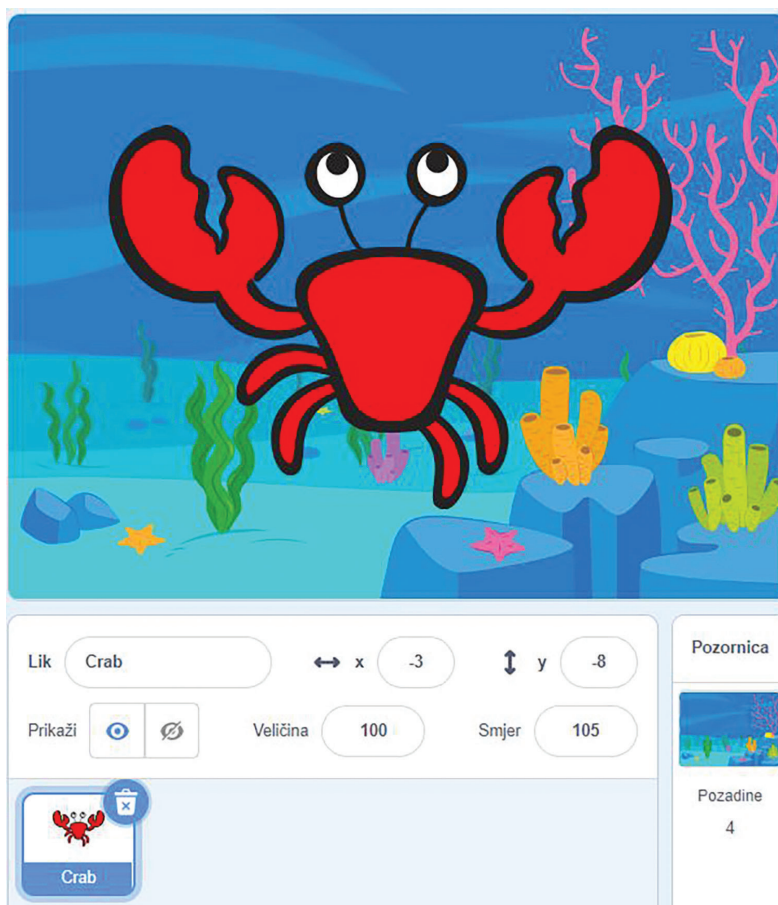
Razmislimo. Potreban nam je rak, nekoliko njegovih različitih kostima. Želimo da se pomiče nasumično (slučajnim pojavljivanjem po ekranu). Postavimo i zanimljivu glazbu.

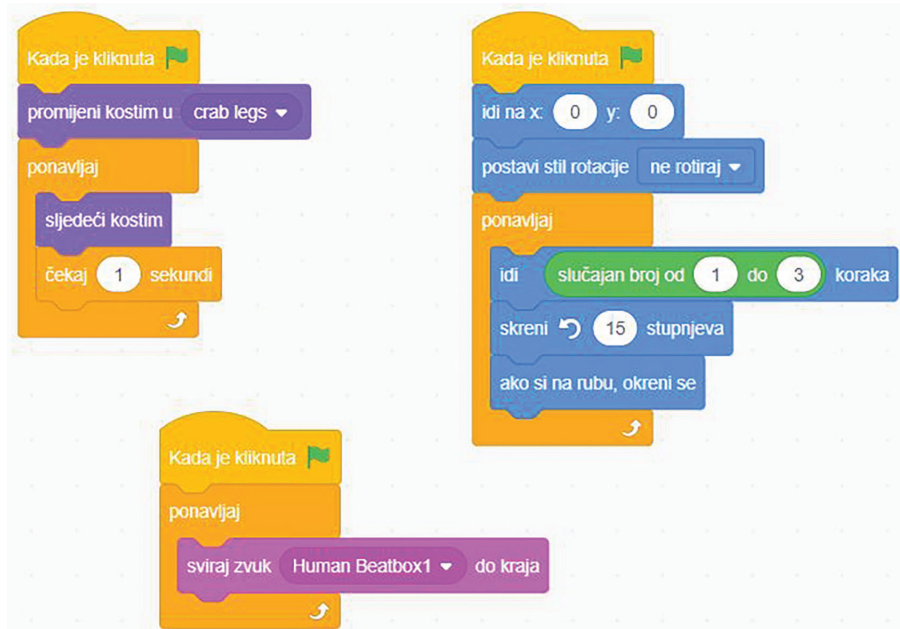
Rješenje.

Postavimo raka na sredinu (blok kretanje).

Ponavljaj dok ne ugasimo program da se rak pomiče na nekoliko slučajnih pozicija. Također, neka glazba svira.

Uz pomicanje po ekranu, neka rak mijenja svoj izgled.





3.3. Izradi čestitku

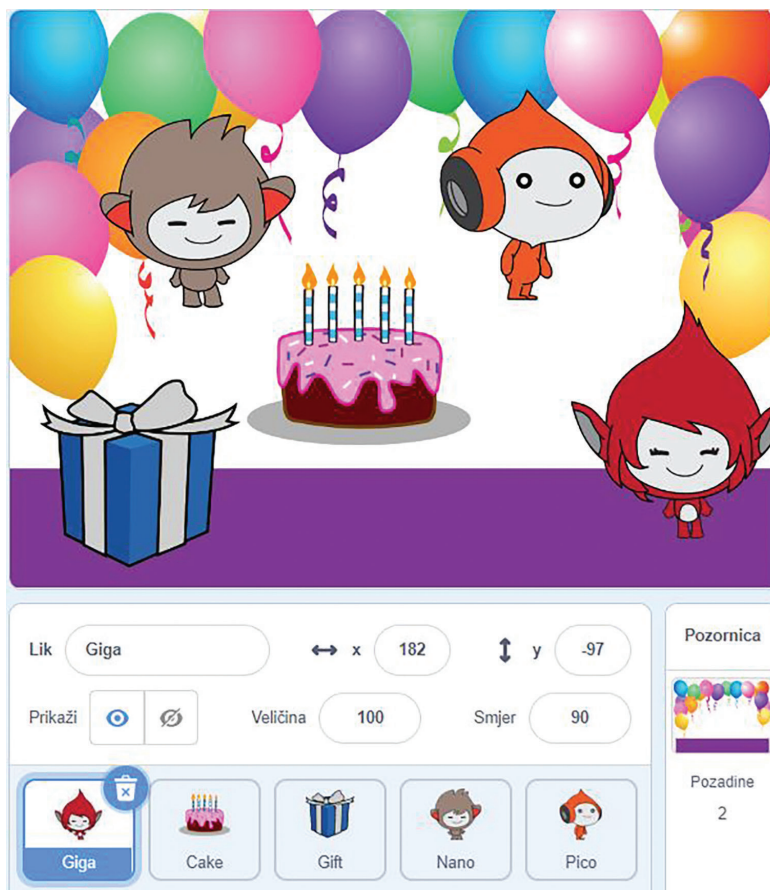
Razmislimo. Scratch nudi i gotove animacije. To nisu animacije po definiciji, već statički tekst ili fotografija koja se pomiče stvarajući iluziju kontinuiranog kretanja.

Izradite pozadinu s natpisom *sretan rođendan*, tko slavi, koji je rođendan... Nadodajte i nekoliko likova koji će, kada pritisnuti, nešto napraviti.

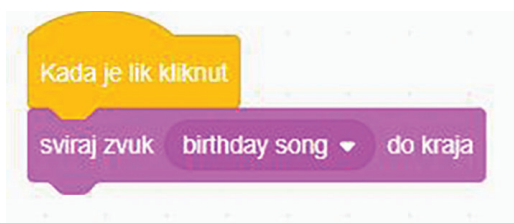
Rješenje.

Kada kliknemo na tortu, svirat će pjesmica *Sretan rođendan*. Pritiskom na poklon, neka promijeni veličinu i boju. A pritiskom na likove: okret cijeli krug, i izgovaranje rečenice SRETAN ROĐENDAN.

Budimo oprezni da uvijek kliknemo na lik kako bi poprimio naredbe koje smo mu namijenili.



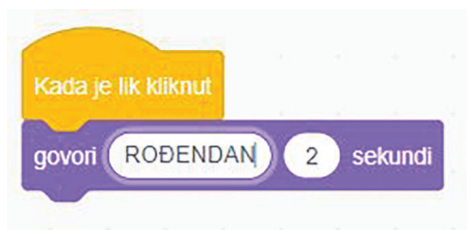
Torta



Lik 1



Lik 2



Lik 3



3.4. Igra skrivača

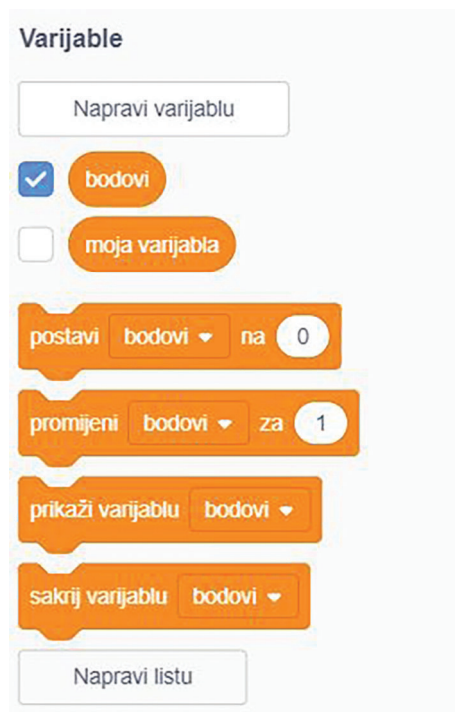
Razmislimo. Želimo napraviti igricu gdje se lik pojavljuje i nestaje. Kada ga kliknemo, osvajamo bod.

Ovdje uvodimo koncept varijable. Varijabla je vrijednost u programu koja se mijenja. Rezultat se svakako mijenja (naravno, može ostati i jednak ukoliko nogometaši ne daju gol ili ukoliko mi ne dotaknemo našeg lika) i tu promjenu nazivamo varijablom. Na početku igrice, rezultat je nula, što znači da našoj varijabli moramo postaviti početnu vrijednost na nula.

Klikom na blok varijable, dobivamo izbornik.

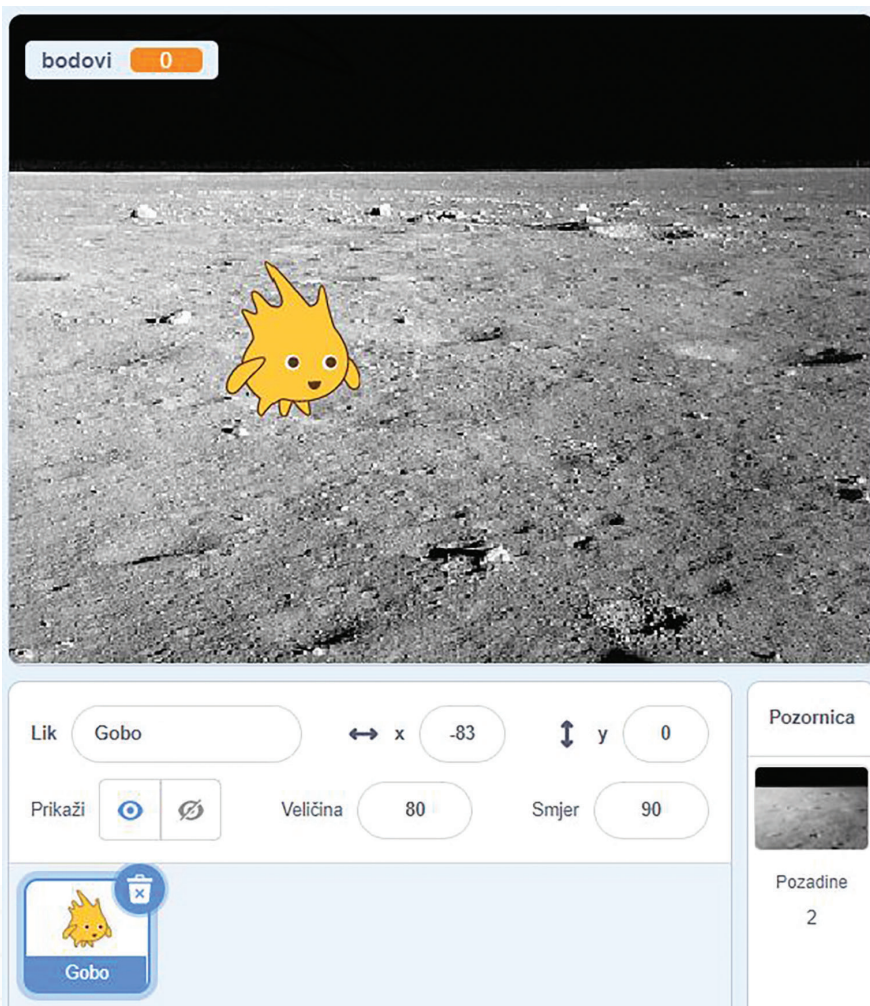
Napravi varijablu služi da stvorimo svoju vlastiti varijablu. Početak igre uvijek prima prvu naredbu (postavi varijablu na nula). U ovisnosti o igri, varijablu možemo povećati za jedan ali i više.

U našoj igrici, lik se slučajnim rasporedom pomiče po pozornici; naš klik povećava bodove za jedan.





Rješenje.



```
Kada je kliknuta
postavi bodovi na 0
idi na x: 0 y: 0
prikaži
govori Klikni me za bodove 2 sekundi
ponavljaj
  sakrij
  čekaj 1 sekundi
  idi na x: slučajan broj od -200 do 200 y: slučajan broj od -140 do 140
  prikaži
  čekaj 0.7 sekundi
Kada je lik kliknut
promijeni bodovi za 1
sviraj zvuk zoop
```

3.5. Izradi kviz

Razmislimo. Želimo osmisliti pitanja i pričekati da korisnik odgovori. Ovdje je najteži dio osmisliti pitanje. Pitanje mora biti takvo da na njega svi mogu znati odgovoriti. Pitanje poput *Što sam jučer jeo?* svakako nije dobro jer nije opće poznato. Pitanje poput *Koji je glavni grad Hrvatske?* svakako nudi samo jedan točan odgovor. Logika će nam biti sljedeća. Obradit ćemo sve moguće točne odgovore, a za sve netočne dati jedan odgovor. Odgovor moramo uzeti iz bloka očitavanja te ga usporediti sa znakom jednakosti iz bloka operacije. Prisjetimo se, odgovor je u našem slučaju varijabla.

Rješenje.



NAPOMENA

Scratch ne razlikuje mala i velika slova.

✓

```

Kada je kliknuta
  izbriši sve
  pitaj Kako se zoveš? i čekaj
  govori spoji Dobar dan, odgovor 2 sekundi
  pitaj Koji je glavni grad Hrvatske? i čekaj
  ako odgovor = Zagreb onda
    govori Točan odgovor 2 sekundi
  inače
    govori Pokušaj ponovno! 2 sekundi
  
```



Ili ako imamo dva točna odgovora.

```
Kada je kliknuta
  izbriši sve
  pitaj Kako se zoveš? i čekaj
  govori spoji Dobar dan, odgovor 2 sekundi
  pitaj KOLIKO JE 2+2? i čekaj
  ako odgovor = 4 ili odgovor = četiri onda
    govori Točan odgovor 2 sekundi
  inače
    govori Pokušaj ponovno! 2 sekundi
```

Morali smo obraditi korisnički unos koji može biti znamenkom ili slovima. Pravi programer mora misliti na sve moguće slučajeve.

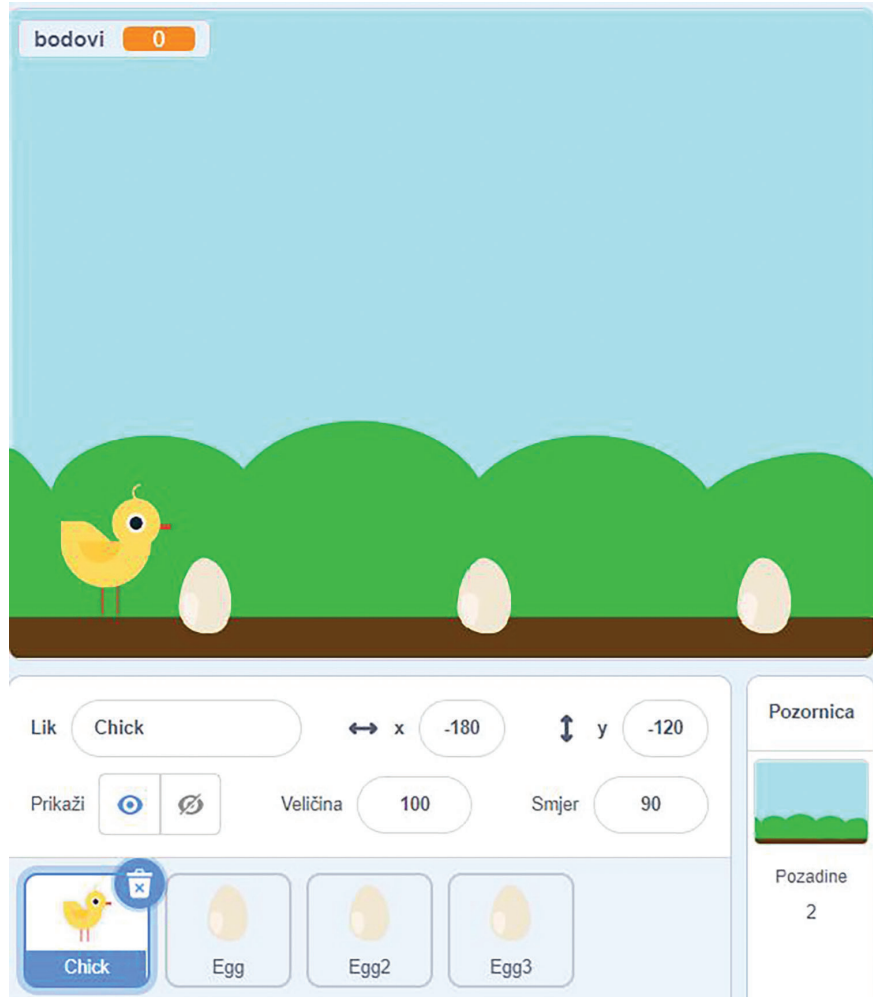
Dodatno. Probaj izraditi ovaj zadatak s varijablom bodovi. Svaki put kada korisnik odgovori točno, neka se broj bodova poveća za jedan.

bodovi 1

```
Kada je kliknuta
  postavi bodovi na 0
  izbriši sve
  pitaj Kako se zoveš? i čekaj
  govori spoji Dobar dan, odgovor 2 sekundi
  pitaj KOLIKO JE 2+2? i čekaj
  ako odgovor = 4 ili odgovor = četiri onda
    govori Točan odgovor 2 sekundi
    promijeni bodovi za 1
  inače
    govori Pokušaj ponovno! 2 sekundi
```

3.6. Pile koje preskače jaja

Razmislimo. Želimo napraviti igricu u kojoj, kada pritisnemo tipku, pile poskoči preko jajeta. Ukoliko to uspije, osvaja bod, inače igrica završava. Treba nam brojač rezultat, pile, jaja. Postavit ćemo tri jaja. Neka svako jaje dolazi sekundu nakon onog prethodnog.



Rješenje.

Najprije programirajmo jaja koja naizgled putuju.

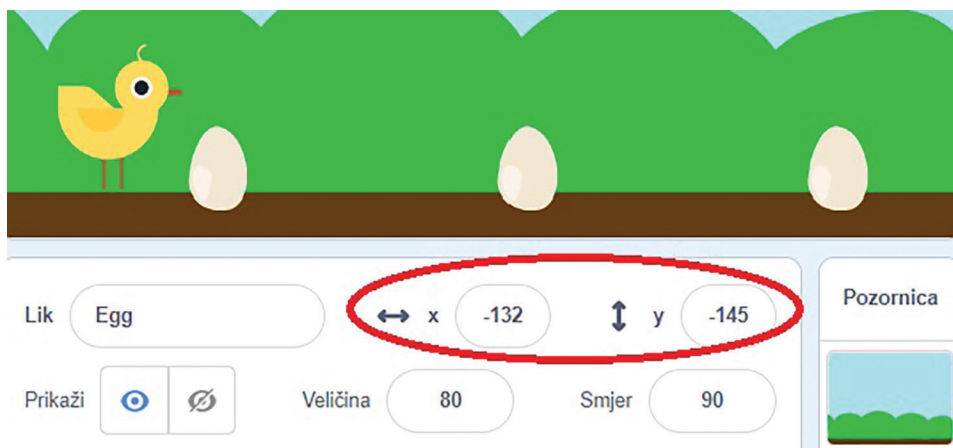
Jaje 1



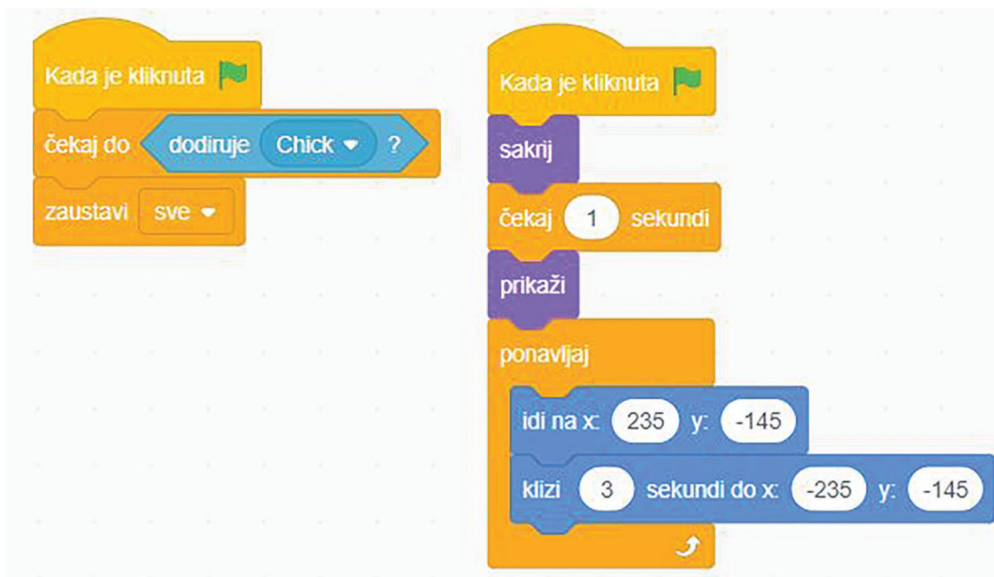


Pogledajmo prvi dio kôda (naredbi). Kada jaje dotakne pile, igra se zaustavlja. Inače, cijelo vrijeme putuj prema piletu. Kako znamo gdje je pile, odnosno do kuda jaje mora putovati?

Ukoliko lik postavimo negdje na pozornicu, Scratch automatski ispisuje njegove koordinate (brojeve koji jednoznačno opisuju njegov položaj). Moguće je da u školi niste učili koordinatni sustav. U tom nam slučaju uopće nije važno razumijevanje tog koncepta. Za sada, samo preuzmite te brojke.



Jaje 2



Ono je moralo pričekati jednu sekundu i tek se tada pojaviti na pozornici.
 Jaje 3 čeka još dodatnu sekundu, ukupno dvije.

The image shows two Scratch code snippets. The left snippet is for a 'Kada je kliknuta' (When clicked) event, containing a 'čekaj do dodiruje Chick?' (wait until touched by Chick?) block and a 'zaustavi sve' (stop all) block. The right snippet is also for a 'Kada je kliknuta' event, containing a 'sakrij' (hide) block, a 'čekaj 2 sekundi' (wait 2 seconds) block, a 'prikaži' (show) block, a 'ponavljaj' (repeat) loop with 'idi na x: 235 y: -145' (go to x: 235 y: -145) and 'klizi 3 sekundi do x: -235 y: -145' (slide 3 seconds to x: -235 y: -145) blocks.

Pile

The image shows two Scratch code snippets for a 'Pile' character. The left snippet is for a 'Kada je kliknuta' (When clicked) event, containing an 'idi na x: -180 y: -120' (go to x: -180 y: -120) block and a 'postavi bodovi na 0' (set score to 0) block. The right snippet is for a 'kada je tipka razmaknica pritisnuta' (when space key is pressed) event, containing a 'sviraj zvuk Chirp' (play sound Chirp) block, a 'ponovi 10' (repeat 10) loop with 'promijeni y za 10' (change y by 10) block, another 'ponovi 10' (repeat 10) loop with 'promijeni y za -10' (change y by -10) block, and a 'promijeni bodovi za 1' (change score by 1) block.

Primijetimo, pritiskom na tipku razmaknicu, pile skakuće.

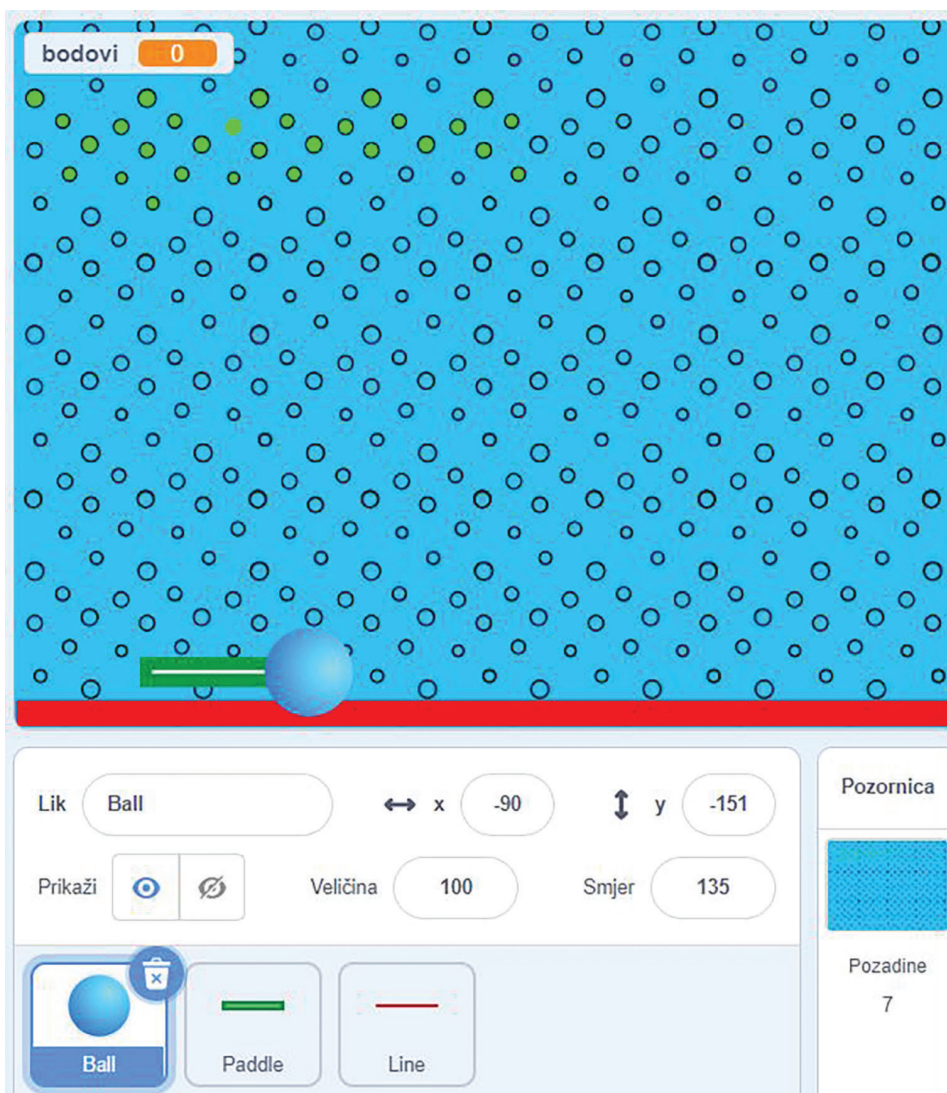


3.7. Igrica odbijanja od prepreke

Razmislimo. Treba nam predmet koji putuje po pozornici, lik koji će predstavljati podlogu za odbijanje i predmet koji će zaustaviti igricu ako je dotaknut. Nadodajmo i varijablu bodovi koji će brojiti svako uspješno odbijanje. Pomakom miša pomiče se i objekt za odbijanje.

Izradimo pozornicu kao na fotografiji.

Rješenje.



Lopta

The code consists of two main event-driven blocks. The first block, 'Kada je kliknuta' (When clicked), contains a 'ponavljanje' (repeat) loop. Inside the loop, there is an 'ako dodiruje Paddle ? onda' (if touches Paddle? then) block. This block contains: 'promijeni bodovi za 1' (change score by 1), 'skreni slučajni broj od 170 do 190 stupnjeva' (turn random angle from 170 to 190 degrees), 'idi 15 koraka' (move 15 steps), and 'čekaj 0.5 sekundi' (wait 0.5 seconds). The second block, also 'Kada je kliknuta', contains 'okreni se u smjeru 45' (turn in direction 45) and another 'ponavljanje' loop with 'idi 15 koraka' (move 15 steps) and 'ako si na rubu, okreni se' (if at edge, turn).

Paddle (lik za odbijanje)

Crvena crta (ako je dotaknemo, zaustavljamo igricu)

The code block is 'Kada je kliknuta' (When clicked) followed by a 'ponavljanje' (repeat) loop containing 'postavi x na miš x' (set x to mouse x).

The code consists of two event-driven blocks. The first block, 'Kada je kliknuta' (When clicked), contains a 'ponavljanje' (repeat) loop with an 'ako dodiruje Ball ? onda' (if touches Ball? then) block, which contains 'zaustavi sve' (stop all). The second block, also 'Kada je kliknuta', contains 'postavi bodovi na 0' (set score to 0), 'čekaj do bodovi > 5' (wait until score > 5), and 'sljedeća pozadina' (next background).

Primijetimo kako nije važno na koji ćemo lik vezati naredbu za postavljanje brojača rezultat na nula (slično kao što nije ni vama važno tko broji kada igrate neke društvene ili sportske igre; bitno je da netko krene brojati), važno je samo da se nalazi nakon naredbe za pokretanje igrice.



3.8. Ja u igrici pucanja balona

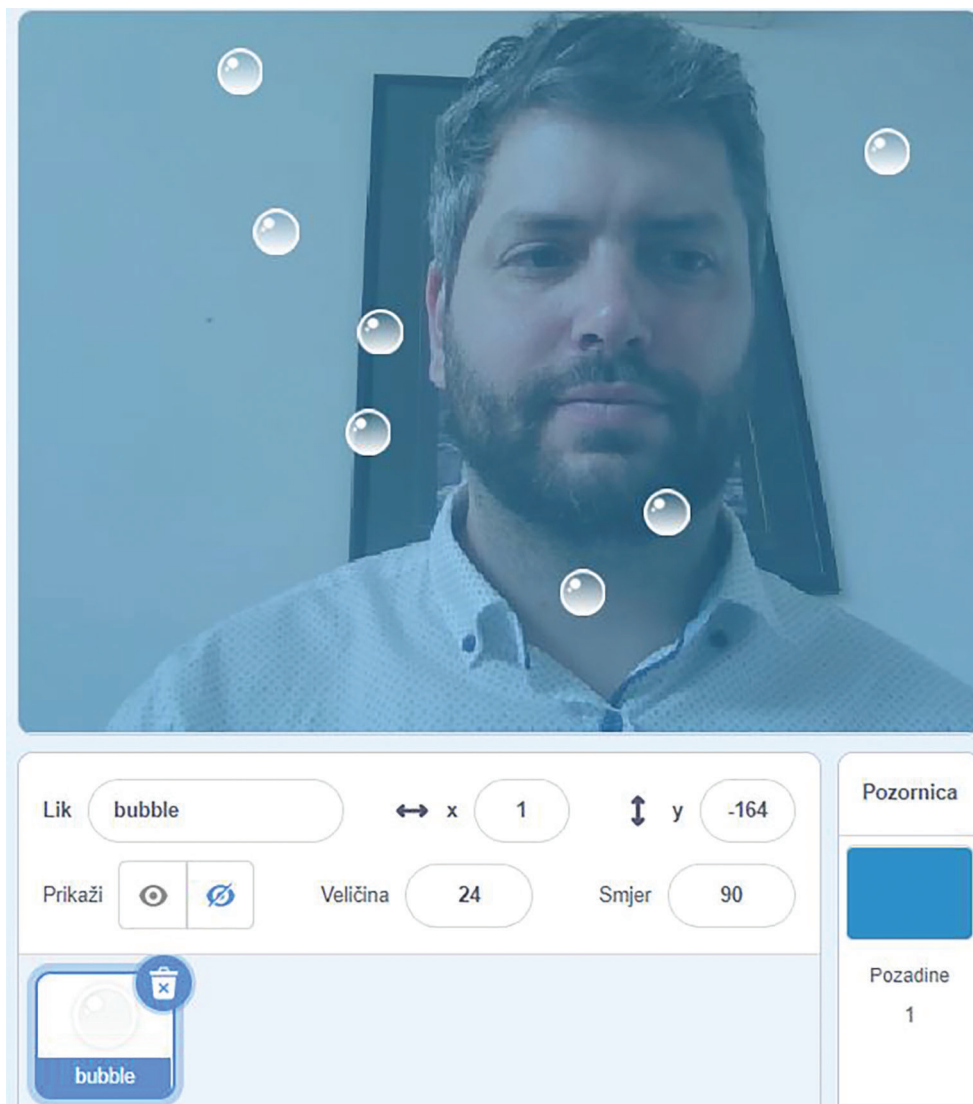
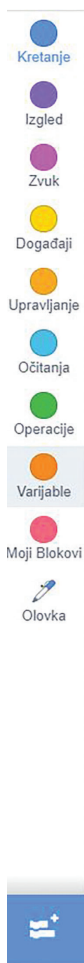
Razmislimo. Dosadilo nam je stalno imati neke druge likove u igrici, hajmo nadodati sebe. Nema boljega nego napušemo balone od sapunice i trudimo se da ih čim više popucamo.

Trebaju nam kamera, baloni.

Rješenje.

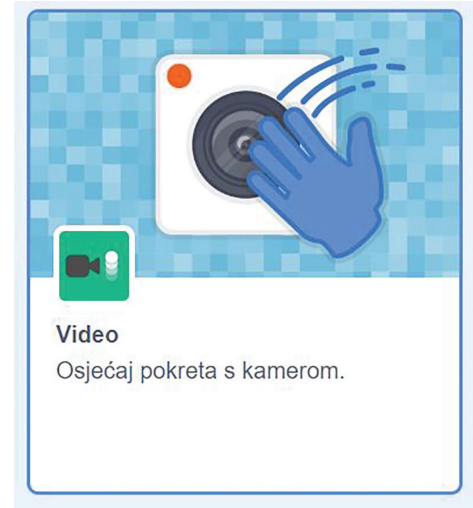
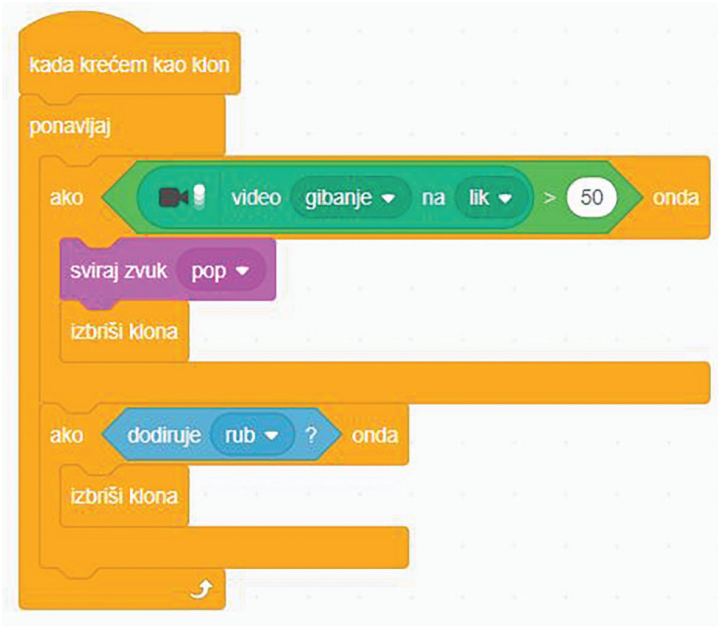
Prvo je potrebno ubaciti proširenje.

Klikom na plus, dobivamo izbor-nik.



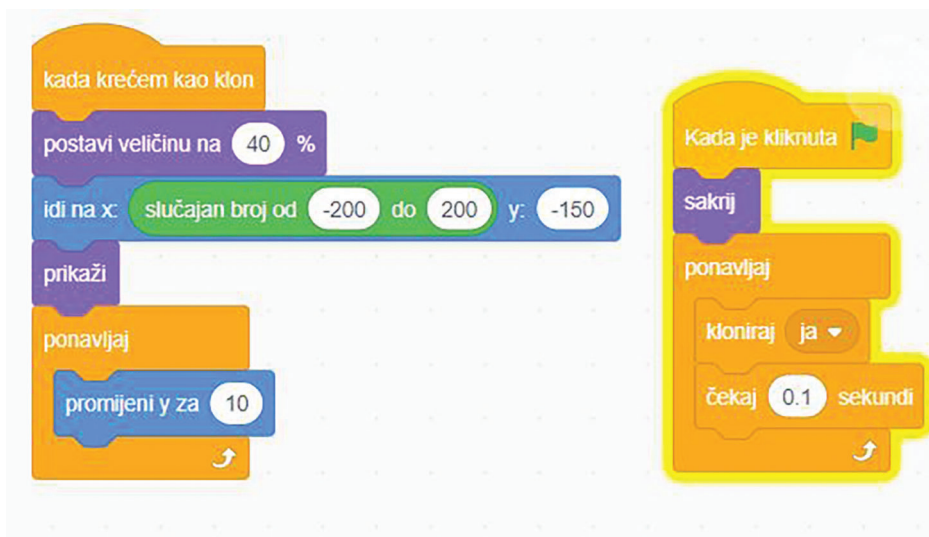
Tada kliknimo na video.

Sebe u igricu ubacujemo na sljedeći način.



Broj 50 označava broj piksela. Probajte pomaknuti lik 50 koraka unaprijed kako biste uvidjeli koliki je to pomak. Ukoliko se naše lice pomakne manje od 50 piksela, igrica neće reagirati.

Stvaranje i kretanje balona





3.9. Pričajmo sa Scratch-om

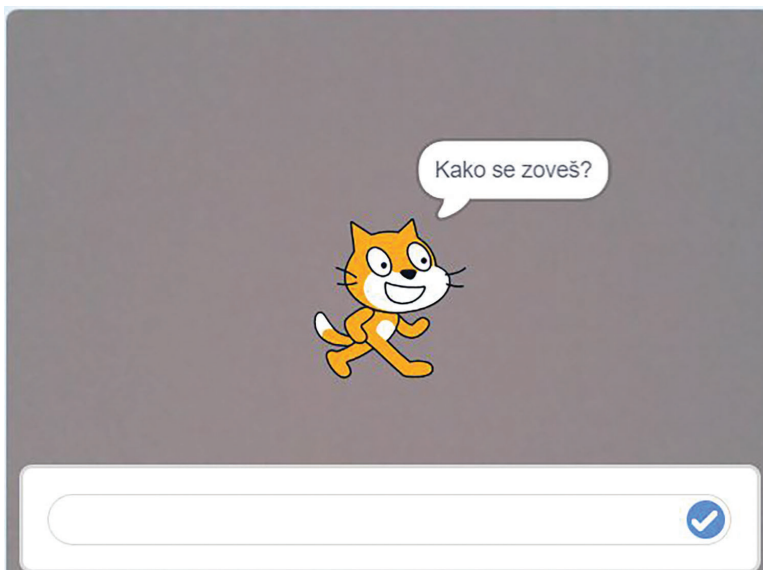
Razmislimo. Želimo da nam Scratch nešto utipkano izgovori.

Prvo je potrebno nadodati proširenje.



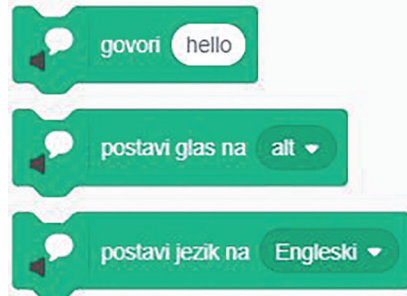
Rješenje.

Tada, vrlo jednostavno, upitajmo korisnika da nešto upiše.



Možemo postaviti i drugi strani jezik (hrvatskog nema).
Možemo i birati na koji način će se rečenica izgovoriti.

Text to Speech



3.10. Zasvirajmo instrumente

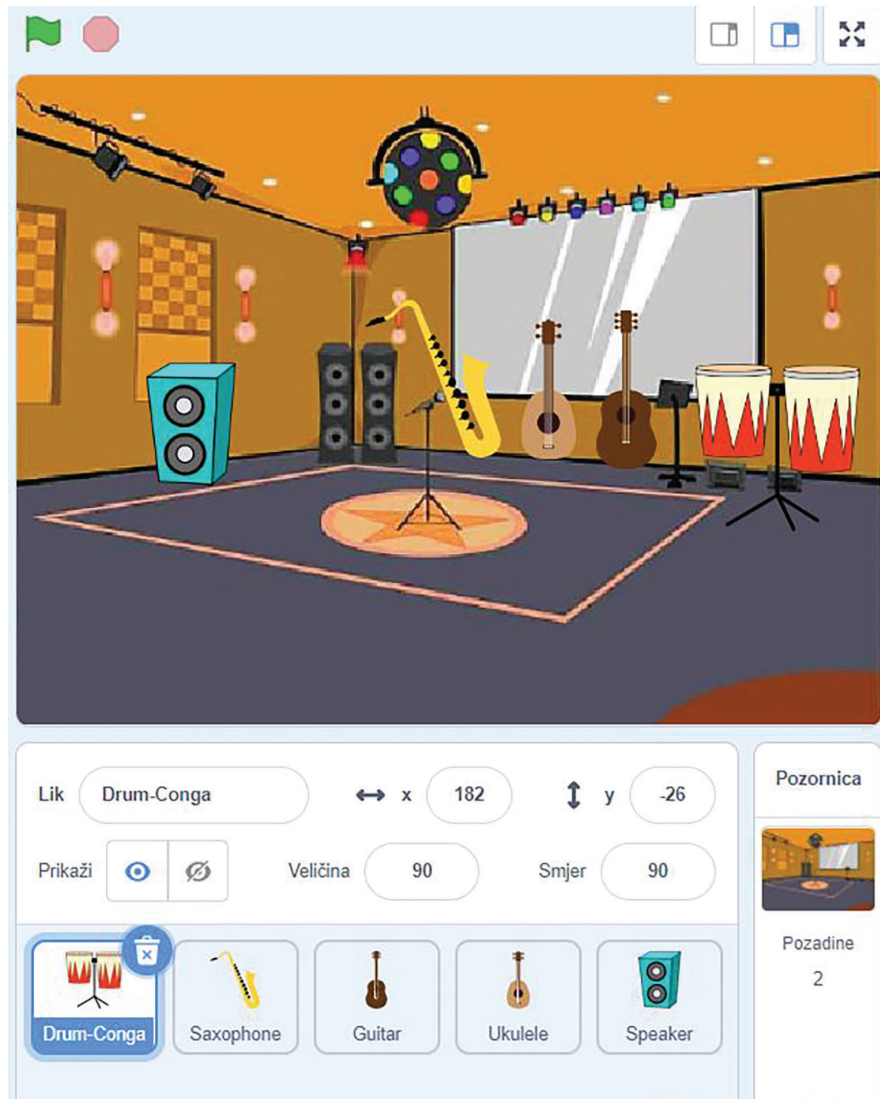
Razmislimo. Nakon govora, zašto ne bismo i zasvirali.

Potrebni su nam sljedeći instrumenti.

Rješenje.

Posložimo ih prema fotografiji.

Nadalje, svakom instrumentu dodijelimo da ga pokrene određene tipka.





Drum-Conga

```
kada je tipka razmaknica pritisnuta
ponovi 4
  sviraj zvuk high conga
  sljedeći kostim
  čekaj 0.125 sekundi
  sviraj zvuk low conga
  sljedeći kostim
  čekaj 0.125 sekundi
```

The code block for 'Drum-Conga' starts with a yellow 'kada je tipka' block set to 'razmaknica' and 'pritisnuta'. This is followed by an orange 'ponovi' block with the number '4'. Inside the loop, there are two purple 'sviraj zvuk' blocks: the first is 'high conga' and the second is 'low conga'. Each 'sviraj zvuk' block is followed by a blue 'sljedeći kostim' block and an orange 'čekaj' block set to '0.125 sekundi'. The loop ends with a small white arrow icon.

Gitara

```
kada je tipka strelica desno pritisnuta
sviraj zvuk slučajan broj od 1 do 8
promijeni efekt color za 25
```

The code block for 'Gitara' starts with a yellow 'kada je tipka' block set to 'strelica desno' and 'pritisnuta'. This is followed by a purple 'sviraj zvuk' block set to 'slučajan broj od 1 do 8'. The final block is a purple 'promijeni efekt' block set to 'color' and 'za 25'.

Saksofon

The image displays four Scratch code snippets arranged in a 2x2 grid, each triggered by a specific key press. Each snippet consists of a yellow 'when key pressed' block followed by purple 'play sound' blocks and orange 'wait' blocks.

- Top-left snippet:** Triggered by the 'up arrow' key. It plays 'C2 sax' (0.25s wait), 'G sax' (0.25s wait), and 'E sax' (0.25s wait).
- Top-right snippet:** Triggered by the 'g' key. It plays 'G sax', 'B sax', and 'D sax'.
- Bottom-left snippet:** Triggered by the 'down arrow' key. It plays 'F sax', 'A sax', and 'C2 sax'.
- Bottom-right snippet:** Triggered by the 'c' key. It plays 'C2 sax', 'G sax', and 'E sax'.

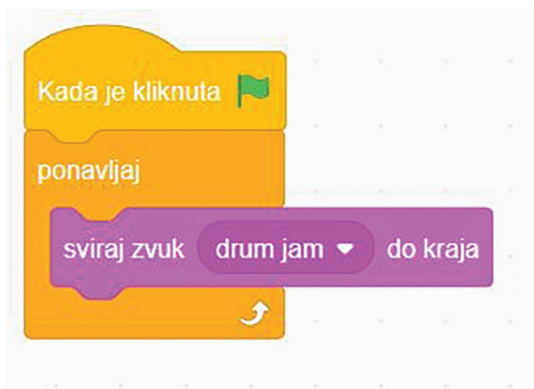


Ukulele



Zvučnik

On svira glazbu bez prekida čim pokrenemo program.



3.11. Tvoj projekt

Sada si na redu ti. Osmisli, poveži, promijeni. Napravi nešto zabavno i zanimljivo. Iznenadi svoje roditelje i učitelje.

Uživaj u programiranju!



4.

Literatura

Scratch stranica, <https://scratch.mit.edu/ideas> Zadnja posjeta: svibanj 2022.

Bilješke



Zagreb, lipanj, 2022.