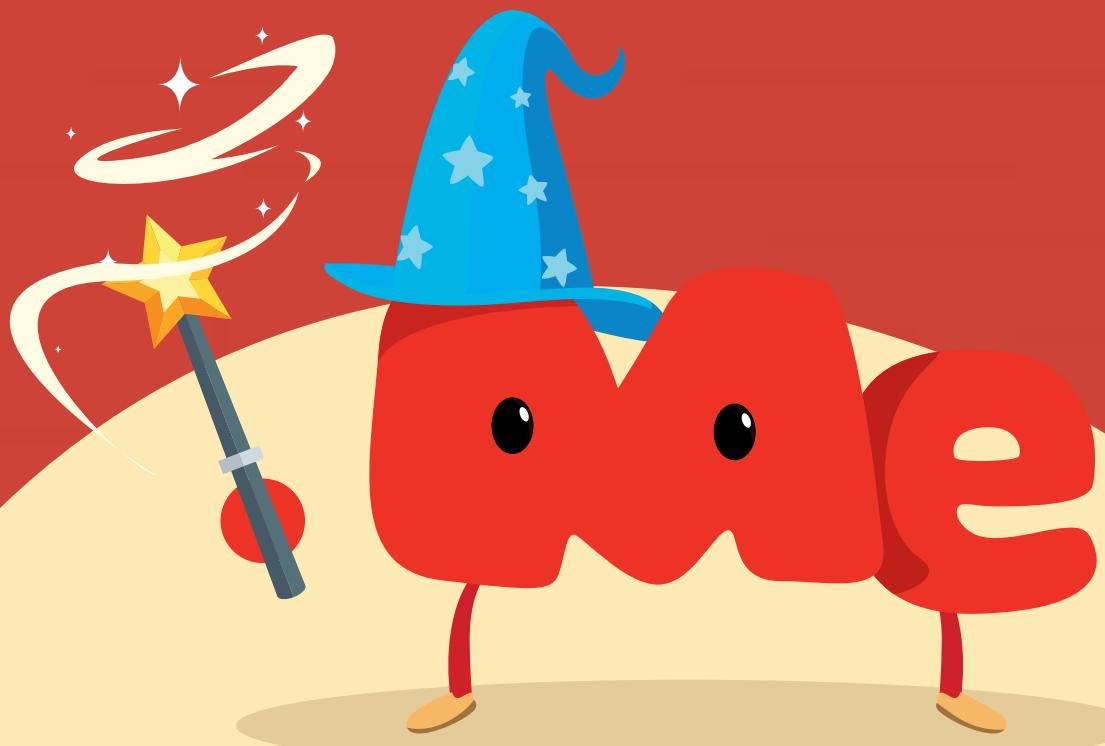


SCRATCH

DoMEn škola
programiranja:
**Kreiraj svoju
prvu igru**





Scratch

DoMEn škola programiranja:

Kreiraj svoju prvu igru

Dobro došli u DoMEn priručnik za osnovce!

Ovo je svijet u kom je sve moguće. I baš u ovom svijetu tvoja mašta i ideje imaju veliku moć. Zamisli žirafu koja piće čaj, slona veličine graška ili ljudi koji lete. Scratch ti nudi upravo to - mogućnosti da ostvariš svoje ideje, a onda ih podijeliš sa drugima. Ovaj priručnik će ti pomoći da zakoračiš u svijet programiranja, iako se možda do sada nisi susreo/la sa tim. :)

Sadržaj ovog priručnika je preuzet sa portala www.stemedukacija.me na kom se mogu naći svi radni materijali DoMEn škole programiranja.

S obzirom na to da ovo čitaš, sigurni smo da već imaš pregršt ideja koje jedva čekaš da sproveđeš u djelo, pa hajde da počnemo!

Šta je Scratch i zbog čega je poseban?

Scratch je besplatan programski jezik koji se koristi za kreiranje interaktivnih priča, igrica i animacija. Baš kroz učenje ovog programskog jezika stičeš i znanje o osnovnim pojmovima u programiranju kao i važnim strategijama za rješavanje problema. Osmišljen je prvenstveno za uzrast od 8 do 16 godina, ali ga mogu koristiti svi (jer kada su u pitanju mašta i učenje, godine stvarno nisu važne), a platforma scratch.mit.edu dostupna je na više od 40 jezika.

Sada kada znaš da možeš svoje ideje da pretvoriš u stvarnost, smjesti se udobno i krećemo. :)

1. Poglavlje

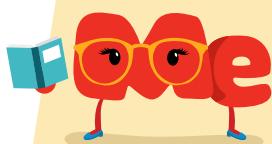
Imaj u vidu da je ovo praktični priručnik, pa dok čitaš ovo, ne zaboravi da otvorиш Scratch i uporedo sa nama gradiš igru. U svakom poglavlju ćemo naučiti nešto novo, pa ti preporučujemo da odvojiš vrijeme i za istraživanje. To će ti pomoći da lakše zapamtiš naučeno, ali i da do nekih saznanja dođeš samostalno i svoju Scratch igru učiniš jedinstvenom!

1.1 Kreiraj nalog

Kako bi zašli u svijet Scratch-a i počeli da se upoznajemo sa svim njegovim mogućnostima, potrebno je da prvo napravimo nalog, a da bi se to desilo, klikni na neki od linkova koji se nalaze u nastavku.

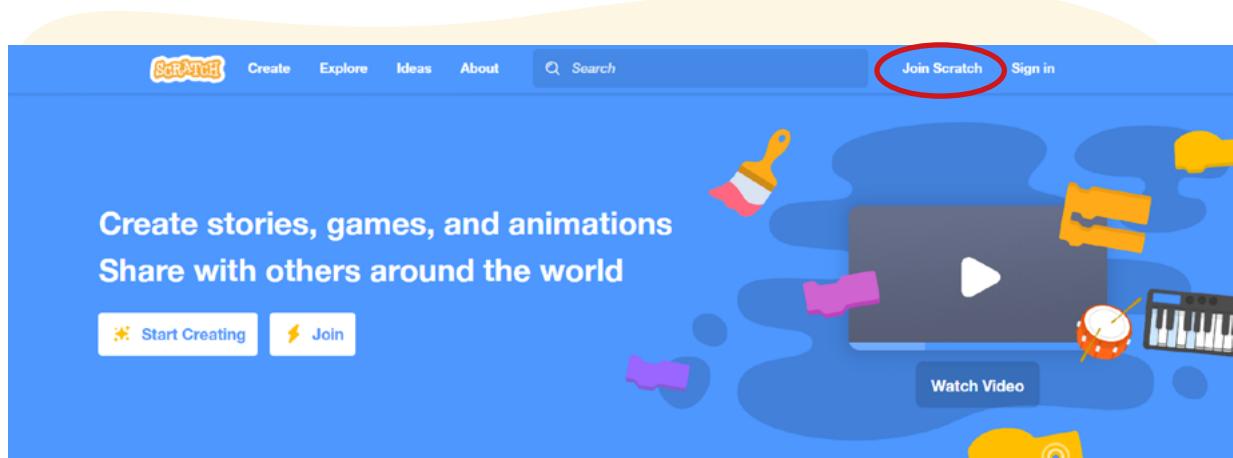
- Online verziju možeš naći na [ovom linku](#).
- Ukoliko misliš da nećeš uvijek imati internet na raspolaganju, ne brini, postoji rješenje i za to. Scratch 2 offline verziju možeš preuzeti [ovdje](#), dok novu verziju Scratch 3 možeš naći na [ovom linku](#).

Napomena: Scratch 3 offline editor zahtijeva OS Windows 10 ili više, pa ako na svom računaru imaš stariju verziju Windows-a moraš instalirati Scratch 2 offline editor.



Scratch program je dostupan offline i online, ali ti preporučujemo da kreiraš nalog na Scratch online editoru i radiš online na svom nalogu. Na taj način ne zavisiš od računara na kom radiš, a svi tvoji projekti će biti sačuvani pod tim online nalogom.

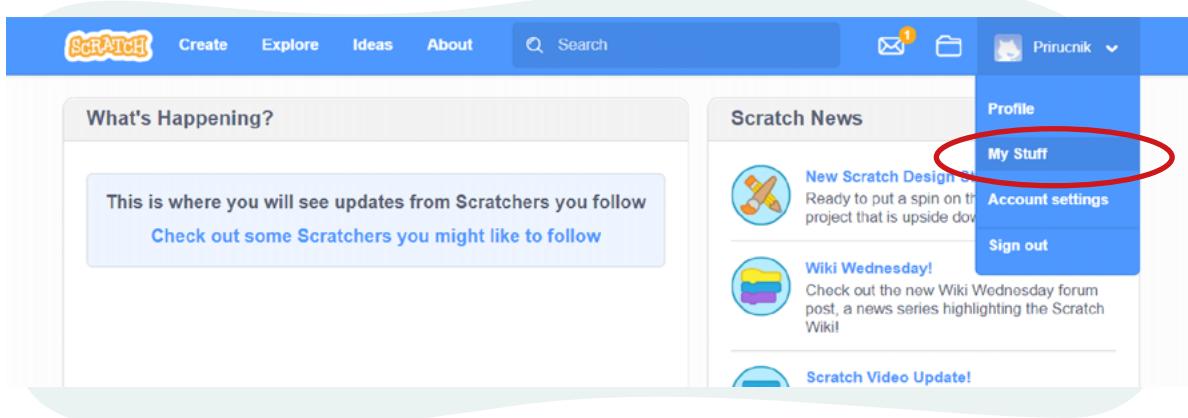
Sada kada smo odabrali verziju koja nam najviše odgovara, idemo dalje. Klikni na gore dati link i napravi nalog prateći uputstva koja ti program nudi. Bez brige, vrlo je jednostavno! :)



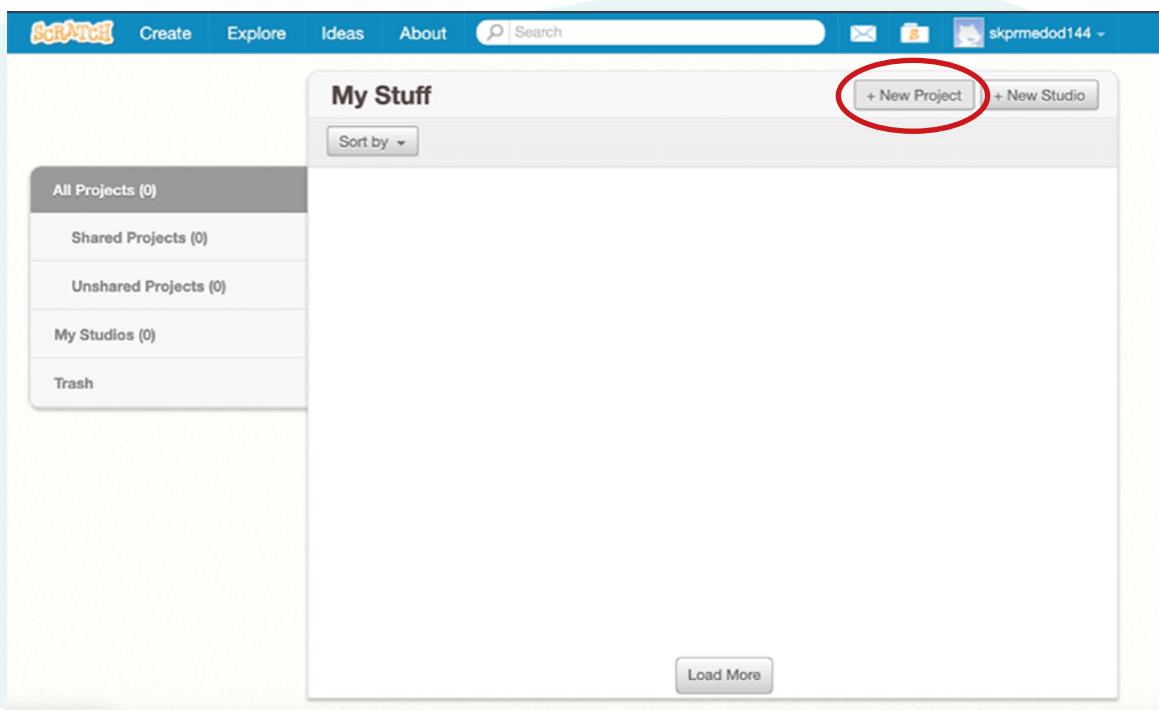


Ukoliko i dalje imаш problem sa kreiranjem naloga, [ovaj video](#) ti može pomoći u tome! Upoznaćeš se i sa osnovnim funkcijama Scratch-a, koje ćemo takođe objasniti u nastavku.

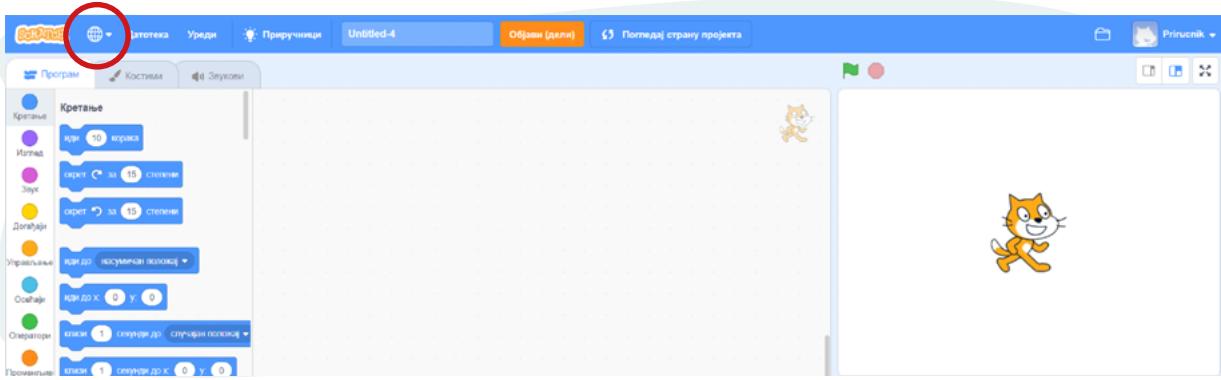
E sada smo konačno tu! Napravili smo nalog na Scratch-u i spremni smo za naš prvi projekat! Klikom na Username, u gornjem desnom uglu, otvorice se padajući meni. Pa kako bi započeli kreiranje projekta potrebno je da kliknemo na "My stuff".



Klikom na "New Project", koji se nalazi u gornjem desnom uglu, započinjemo rad na našem prvom projektu.



Kako bi se lakše snašli u programu, uvijek možemo promijeniti jezik. Opcija za promjenu jezika se nalazi u gornjem lijevom uglu ekrana. Sve što je potrebno jeste da kliknemo na globus i izaberemo jezik koji nam najviše odgovara! :)



1.2 Upoznaј se sa osnovnim funkcijama Scratch-a

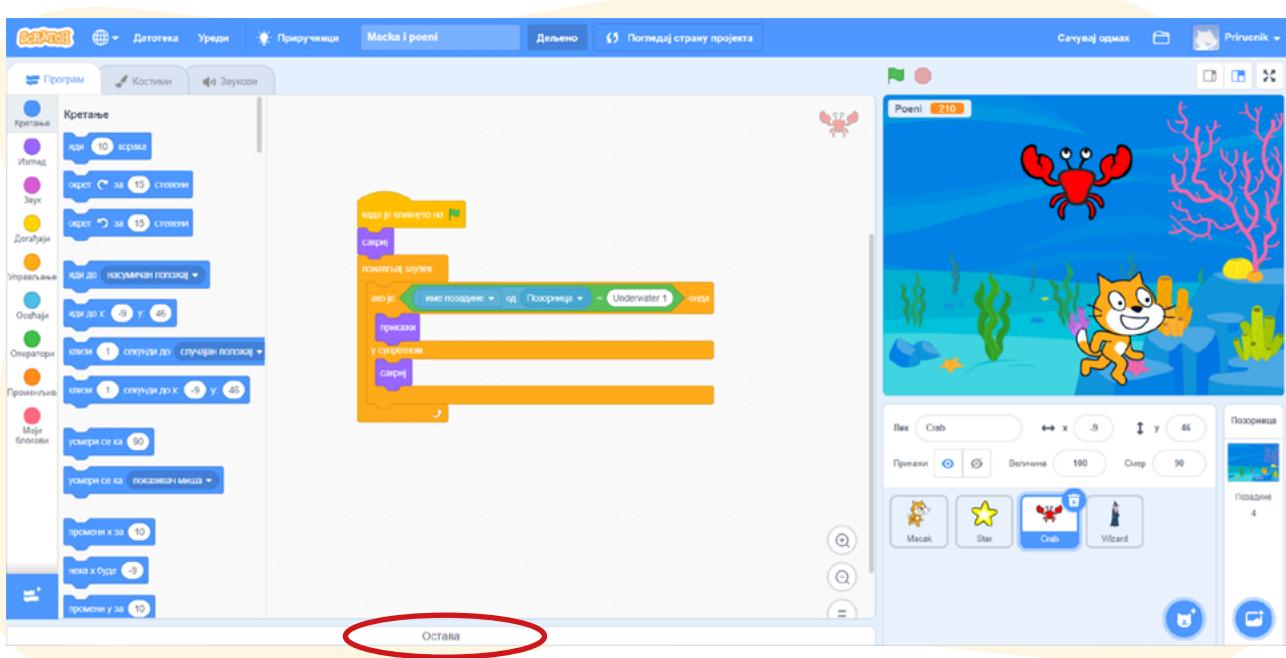


- 1. Dostupne komande** u Scratch-u su grupisani blokovi iz palete blokova komandi koji imaju različite funkcije i koji se spajaju poput slagalice. Ovo čini cijeli proces slaganja kodova veoma zanimljivim, a ubrzo ćeš to vidjeti i na djelu!
- 2. Blokovi komandi** su podijeljeni u kategorije različitih boja, kao na gornjoj slici, a vrste blokova su: kretanje, izgled, zvuk, događaji, osjećaji, operatori, promjenjive. Na njih moraš posebno обратити pažnju jer će ti pomoći da se lakše snađeš kada budemo radili zadatke. Svaki novi projekat u početku sadrži mačka kao **lika**, koji se naravno može promijeniti.

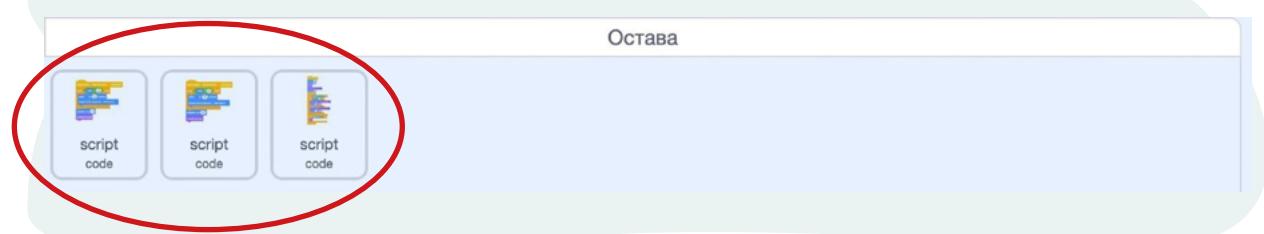
- 3.** Svakom liku možeš dati i **ime** kako bi ga učinio jedinstvenim!
- 4.** Neki likovi imaju više kostima, a kostim je slika izgleda našeg lika. Klikom na jezičak "Kostimi", iznad paleta blokova, možeš pronaći sve dostupne kostime. Njih možeš mijenjati uz pomoć ponuđenih alatki, što ti daje mogućnost da nekom kostimu promijeniš boju, izraz lica, pa čak i visinu! Naravno, uvijek možeš dodavati i nove kostime.
- 5. Komande** se koriste tako što se prevuku do prostora za program. One se slažu jedna na drugu, kao djelovi slagalice, a takođe se mogu obrisati ili duplirati tako što ćemo desnim klikom kliknuti na njih i zatim odabratи "Obriši blok" ili "Umnoži". Blokovi se mogu obrisati i prevlačenjem nazad u dio sa dostupnim komandama (nekad je ovako lakše obrisati više vezanih blokova).
- 6. Izbor novog lika** možeš naći u donjem desnom uglu, pa klikom na ikonicu mačke možeš vidjeti raznovrsne ilustracije koje ti stoje na raspolaganju.
- 7. Izbor pozadina** se nalazi pored ikonice za likove, pa poput njih, i ovdje imamo širok izbor.
- 8. Objavi (dijeli)** ti daje mogućnost da podijeliš svoj projekat sa internet zajednicom i time omogućiš vršnjacima da koriste, mijenjaju ili unapređuju kopiju tvog projekta.

1.3 Ostava

Važno je обратити пажњу на оставу. Како не би морали исти сет блокова изнова да градимо у следећем пројекту, можемо их једноставно сачувати у “Ostavi”. Код једном сачуван у остави, можемо да користимо и у сваком другом пројекту. Ово нам пружа много олакшице, посебно ако је код веома дугачак, као на слици испод, или ако се често користи.



Primjer: Set блокова које користимо за кретање лика ће нам засигурно бити од користи у неком од следећих пројеката. Потребно је да кликнемо на код и превучемо га у оставу, па када отворимо други пројекат, код превучемо из оставе у програм и одмах користимо.



1.4 Remix

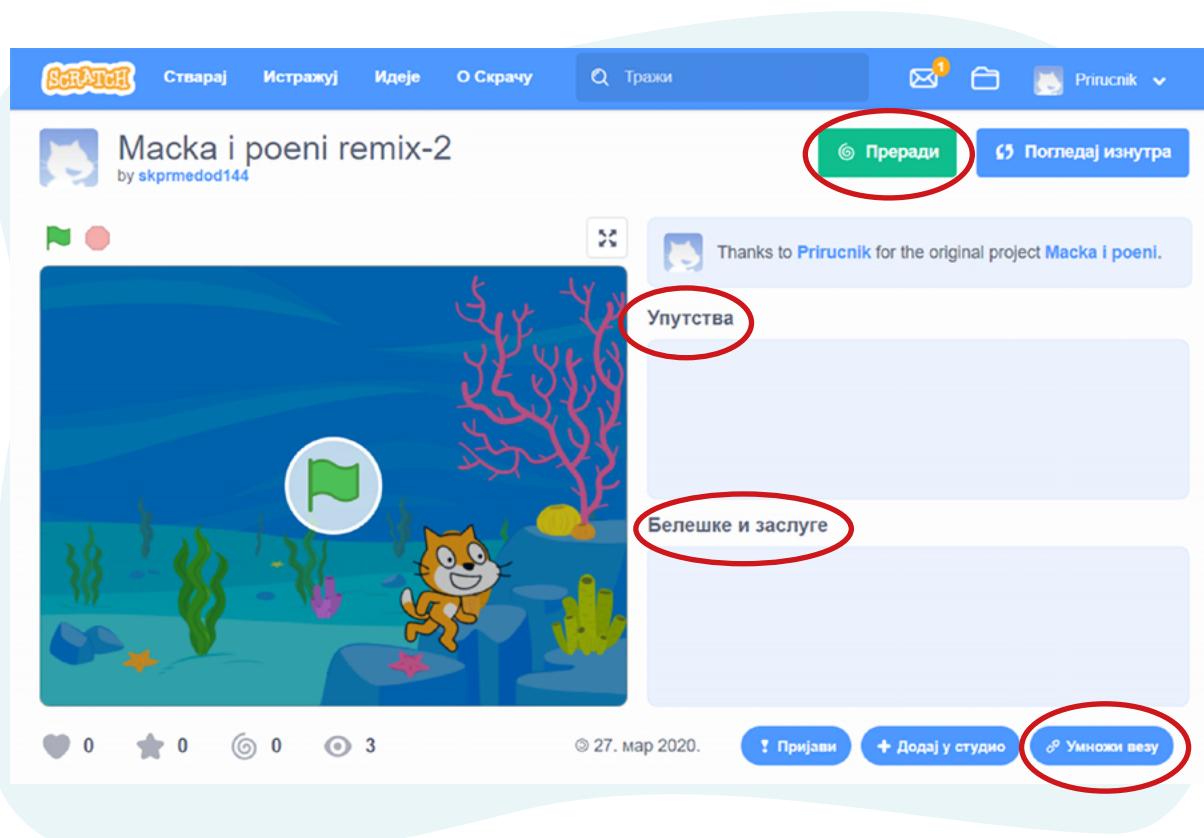
Scratch nudi jednu divnu opciju, a to je mogućnost dijeljenja projekata. Kao što mi možemo podijeliti svoje projekte sa drugima, tako možemo preuzeti i izmijeniti sve projekte koje su objavili naši vršnjaci! Ne smijemo zaboraviti da budemo fer pa u polju za **bilješke i zasluge** napišemo od koga je preuzet originalni projekat.

Unutar polja **uputstva** treba zapisati na koji način se pokreće lik i na koji način igrica funkcioniše. Na primjer: Pritisnite slovo X kako bi vam mačak postavio pitanje.

Klikom na **Преради**, Scratch pravi kopiju projekta i pruža mogućnost da postojeći projekat izmijenimo ili nadogradimo.

Nakon što podijelimo ili preuzmemosmo projekat, klikom na **умноžи везу** možemo poslati link našeg ili tuđeg projekta svima, nezavisno od toga da li imaju nalog na Scratch-u ili ne. Na taj način, naši drugari ili roditelji mogu da igraju igru koju smo mi pronašli ili kreirali.

Bitno: Kako bi se lakše snašao/la i bolje savlado/la gradivo, svaki zadatak za vježbu će ti biti dostupan za preuzimanje i uređivanje! :)



2. Poglavlje

Da li se sjećaš šta smo rekli na samom početku? Ovo je svijet u kom možeš da kreiraš zamisliš!

U ovom poglavlju ćemo naučiti Scratch mačka da mjaukne, da nas pozdravi i da se šeta po pozornici. Kasnije ćemo kroz zadatku za vježbu primjeniti naučeno uz malo maštete! Kako smo se u prethodnom poglavlju upoznali sa svim osnovnim funkcijama koje Scratch nudi, vrijeme je da otvorimo novi projekat i pređemo na praktičan rad! :)

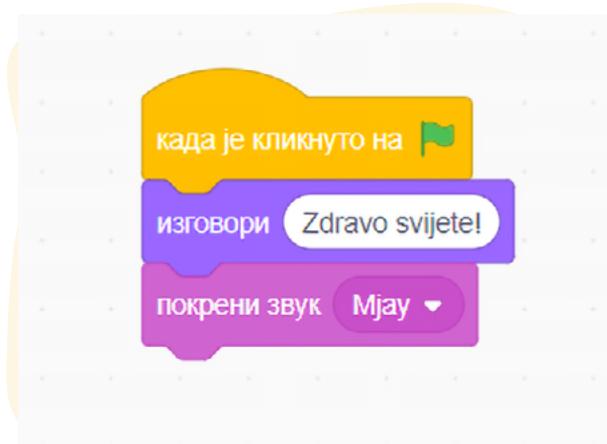
Primjer 1. – Pozdrav

Ideja: Kada kliknemo na zelenu zastavicu, Scratch mačak će izgovoriti “Zdravo, svijete!”, a zatim će mjauknuti.

Napomena: Tekst koji mačak treba da izgovori možeš izmijeniti klikom na bijelu površinu komande Izgovori.

Putem blokova komandi, koje se nalaze na lijevoj strani ekrana, možeš pronaći 3 komande sa slike i to: (događaji - žuta) **kada je kliknuto na**, (izgled - tamno ljubičasta) **izgovori i** (zvuk - svijetlo ljubičasta) **pokreni zvuk**.

Nasloži komande po uzoru na sliku (desno), tako što ćeš ih prevući na sredinu ekrana.



Napomena: Projekat je potrebno čuvati redovno kako ne bi izgubili izmjene.

Imenovaćemo ovaj projekat kao “Prvi projekt”, a zatim kliknuti **Saćuvaj odmah**. Ukoliko nekim slučajem ne vidiš to dugme, možeš pritisniti CTRL+S ili CMD+S (ako koristiš Mac). Ako klikneš narandžasto dugme **Objavi (dijeli)**, projekat će moći da vide svi korisnici Scratch-a!



Primjer 2. – Novi lik

Ovo je bilo prilično lako, zar ne? Sada je vrijeme da dodamo i novog lika na pozornicu!

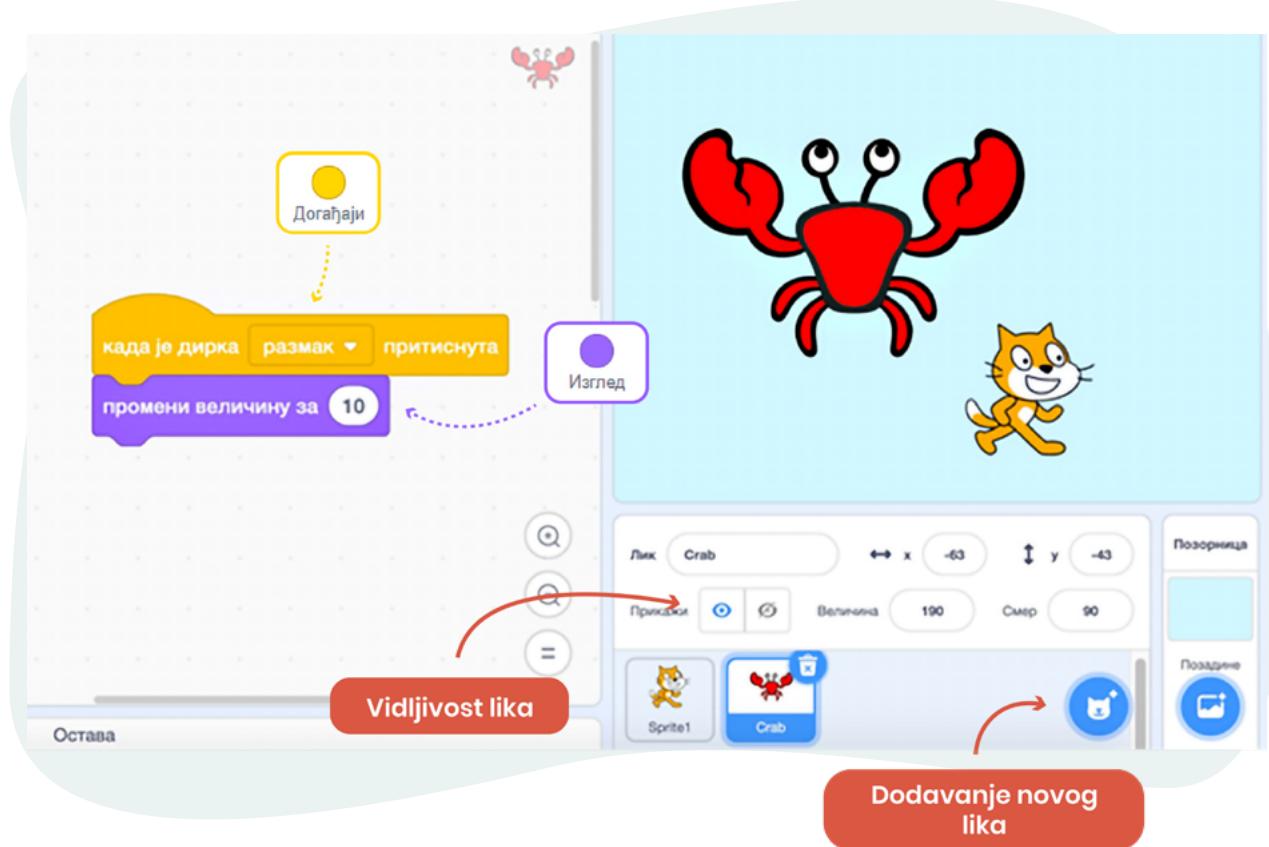
Ideja: Dodajemo novog lika, krabu, koji se povećava svaki put kada pritisnemo razmak.

Novog lika možeš naći klikom na ikonicu u donjem desnom uglu ekrana. Kada pronađeš krabu, koja se nalazi u sekciji “životinje”, prelazimo na komande koje će je navesti da se uvećava.

Prevuci komandu iz bloka **Događaji - kada je dirka razmak pritisnuta**. Njoj dodajemo komandu iz bloka **Izgled - promeni veličinu za 10**.

Šta se događa kada pritisnemo razmak?

Prilično zanimljivo, zar ne? :)



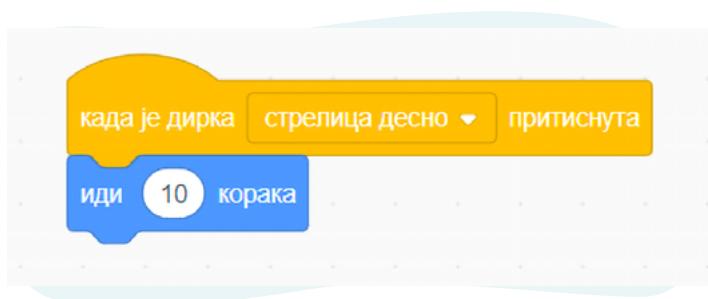
Primjer 3. – Kretanje

Na pozornici sada imamo dva lika, mačka koji nas pozdravlja i krabu koja se povećava kada pritisnemo razmak. Šta dalje? Pa nije to cijela igra koju si na početku zamislio/la, zar ne?

Primjetili smo da svaka komanda ima dio koji možemo da promjenimo, sada je vrijeme da to iskoristimo.

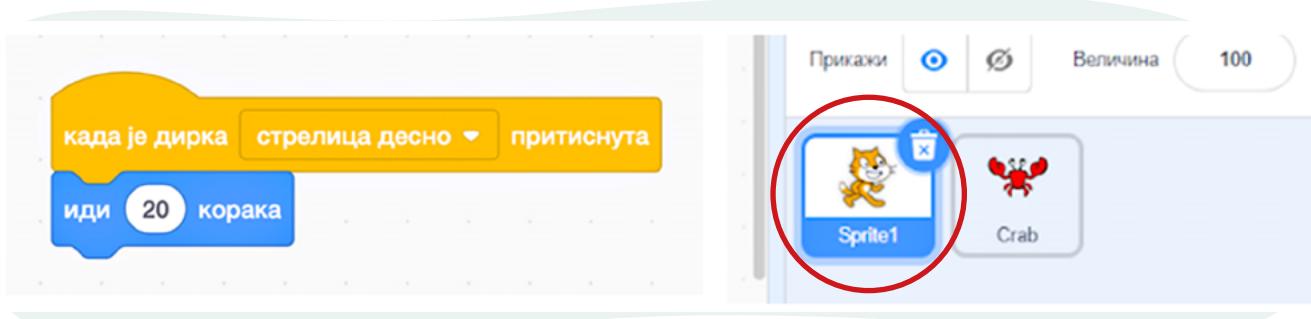
Ideja: Kada pritisnemo strelicu desno, kraba će se pomjeriti za 10 koraka naprijed, a mačak za 20.

Spreman/a?



Iz bloka događaji izvlačimo komandu *kada je dirka strelica desno pritisnuta*. Iz bloka kretanje prevlačimo *idi 10 koraka*.

Uspjeli smo da pomjerimo krabu, sada je na redu Scratch mačak! Kako bi mu dali komandu da se pomjeri, potrebno je da kliknemo na njegovog lika koji se nalazi ispod pozornice. Pa ga istim komandama, kojim smo pomjerili krabu za 10 koraka, pomjeramo za 20.

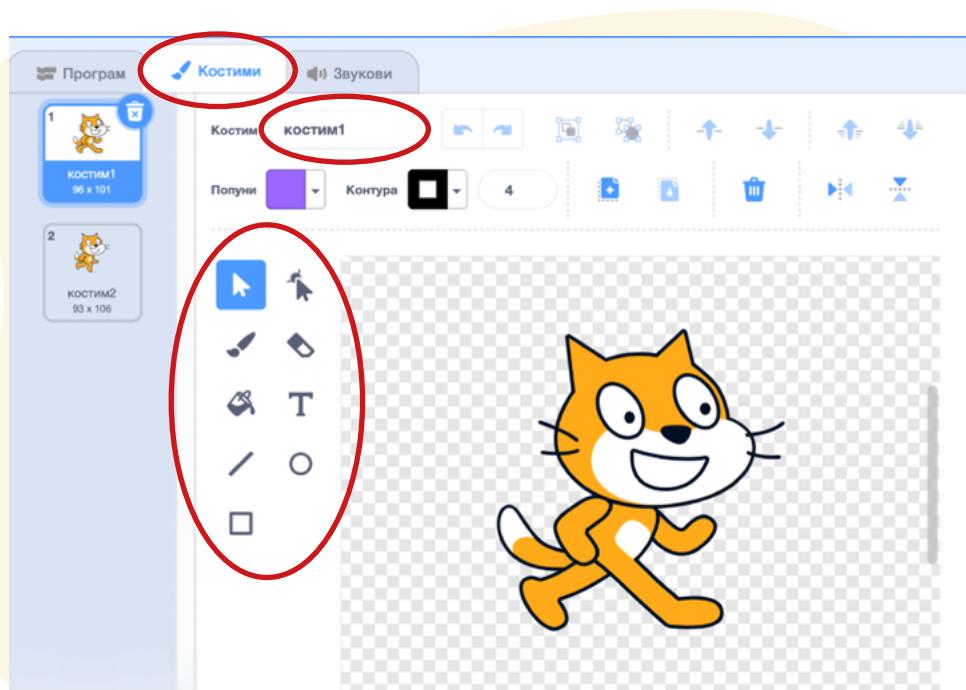


Primjer 4. – Promjena kostima

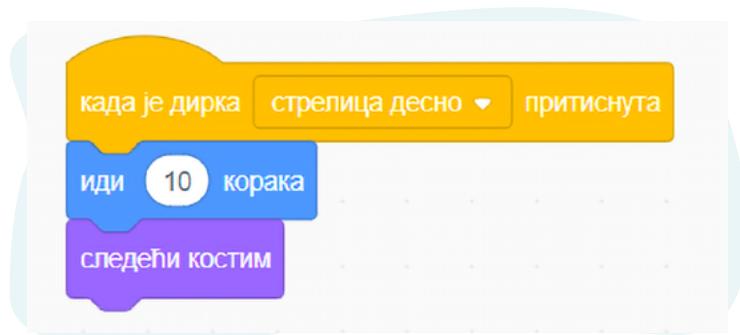
Kostimi postoje kako bi naši likovi postali dinamični. Uz pomoć kostima, likovi blago mijenjaju svoj izgled. Pa na primjer, na jednom kostimu će mačak imati ispruženu nogu, dok će na drugom njegova noga biti savijena. Kombinujući ta dva kostima i komande kretanja, posmatrači stiču utisak da se mačak zaista kreće po pozornici.

Takođe, ako обратимо pažnju na **alatke**, vidjećemo da svaki kostim možemo da izmjenimo! Uz pomoć alatki možemo da obojimo mačka u ljubičasto, da mu promijenimo izraz lica ili mu povećamo glavu!

Svaki kostim možemo **preimenovati**, što će nam biti od značaja kada ih budemo mijenjali i stvarali nove.



Kako bismo zadali ovaj uslov, potrebno je da već kreiranom kodu dodamo komandu iz bloka **izgled sledeći kostim**, po uzoru na sliku desno.



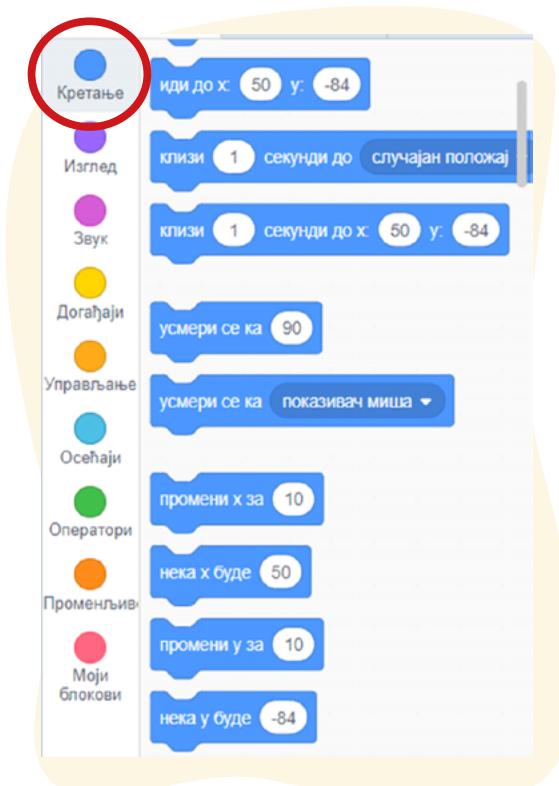
Primjer 5. – Snalaženje u prostoru

Za pisanje programa je veoma važno da razumijemo na koji način se naši likovi kreću po pozornici. Kad se započne novi projekat, lik mačka se nalazi na sredini ekrana $(x,y)=(0,0)$. Za pozicije lijevo od sredine, X je negativan, a za pozicije desno od sredine, X je pozitivan. Slično, za pozicije ispod sredine Y je negativan, a za pozicije iznad sredine, Y je pozitivan.

Postoji pozadina u Scratch-u, koja nam pomaže da lakše razumijemo koordinate, a zove se XY-grid i nalazi se u biblioteci pozadina. Na pozornici u Scratch-u X koordinata uzima najmanju vrijednost -240 i najveću vrijednost 240, dok Y koordinata uzima najmanju vrijednost -180 i najveću vrijednost 180.



Postoji šest blokova **Kretanja**, koji koriste koordinate da promijene poziciju lika. Kada kliknemo na broj unutar bloka, možemo unijeti novi negativan ili pozitivan broj.

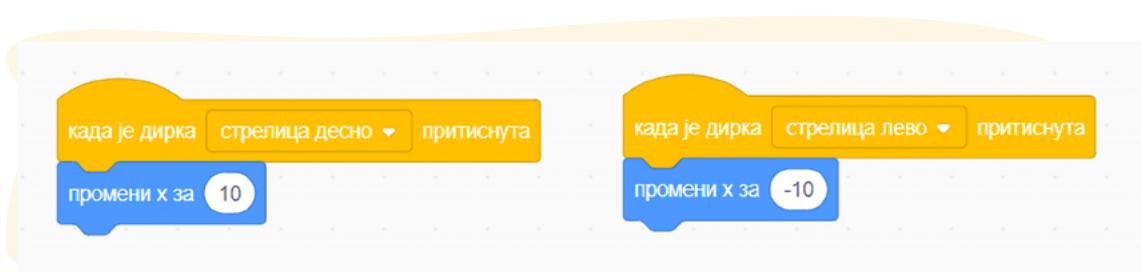


Pošto smo razumijeli snalaženje u prostoru, vrijeme je da saznamo kako se naš mačak može kretati lijevo-desno, ali i gore-dolje.

Kada komanda sadrži Y (**promeni Y za 10**), pozitivan broj će navesti mačka da se kreće gore, dok će ga negativan broj navesti da se kreće ka dolje.

Ideja: Kada se pritisne strelica lijevo, mačak će se kretati 10 mesta lijevo; kada se pritisne strelica desno, mačak treba da se kreće 10 mesta desno.

Za kretanje lijevo i desno koristimo X, pa će komanda biti **promeni X za 10** i **promeni X za -10**.



Kretanje mačka možemo upotpuniti i prethodnim znanjem. Na ove komande možemo dodati i kostim. Sada kretanje mačka postaje zanimljivije, zar ne?



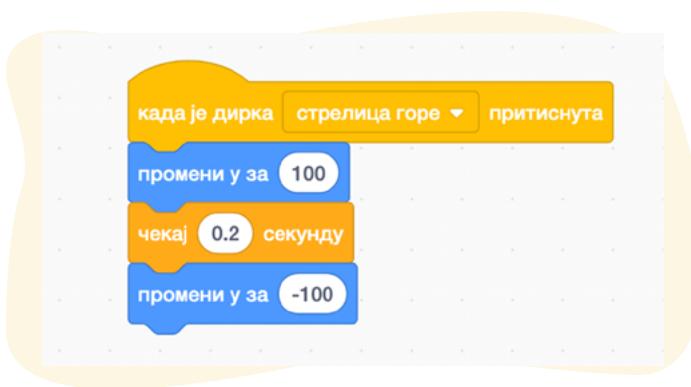
Primjer 6. - Skok

Pošto smo savladali kretanje, šta misliš o tome da pređemo na skok i time završimo ovo poglavlje?

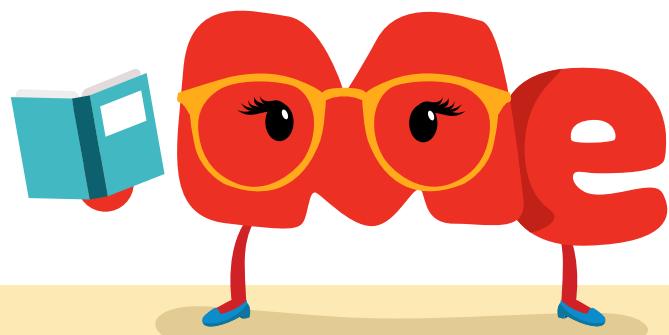
Kao što smo već pomenuli, kada se krećemo Y osom, mačak će ići gore ili dolje, samim tim, pri skoku moramo koristiti Y osu.

Ideja: Zadajemo komandu tako da, kada se pritisne strelica gore, mačak treba da skoči.

Vrijednost komandi mijenjamo u skladu sa slikom.



Sada naš mačak može i da skoči!



Remix (preradi)

Na kraju svakog zadatka za vježbu možeš naći link već odrađenog zadatka. Ukoliko želiš samostalno da radiš u već postojećem radu osobe koja je svoj projekat podijelila, potrebno je da klikneš na link.

Kada se projekat otvori, u gornjem desnom uglu vidjećeš dugme PRERADI, pa samim tim snimaš kopiju već postojećeg rada. Ovim putem dobijaš uvid u kodove datog projekta, ali i mogućnost da unosiš svoje promjene i ideje.

Zadatak za vježbu:

Sada je prava prilika da primjeniš naučeno! Hajde da svemu dodamo malo mašte i kreiramo nešto novo i zanimljivo!

1. Napravi novi Scratch projekat koji ćeš nazvati "Zadatak za vježbu 1"

Pomoć: Pogledaj primjer 1. drugog poglavlja.

2. Dodaj lika po izboru i daj mu ime "Veseljko", a zatim dodaj i pozadinu.

Pomoć: Pogledaj sliku iz prvog poglavlja "1. 2 Upoznaj se sa osnovnim funkcijama Scratch-a" i obrati pažnju na donji desni ugao ekrana.

3. Kada se pritisne strelica desno, lik treba da se kreće naprijed. Kada se pritisne strelica lijevo, lik treba da se kreće unazad, a da se pritom njegov kostim mijenja.

Pomoć: Pogledaj primjer 5.

4. Kada se pritisne tipka razmak, lik treba da skoči.

Pomoć: Pogledaj primjer 6.

5. Testiraj šta se dešava kada istovremeno pritisneš razmak i strelicu desno, ili razmak i strelicu lijevo.

6. Dodaj početno slovo svog imena koje se povećava svaki put kada prvi lik skoči.

Pomoć: Pogledaj primjer 2. drugog poglavlja, ali obrati pažnju i da je u skladu sa trećim korakom.



[Našu verziju zadatka možeš pronaći ovdje.](#)

3. Poglavlje

Kako ti se svidio prvi projekat? Da li je zadatak za vježbu bio izazov? Prilično smo sigurni, da ako ovo čitaš, već jedva čekaš da počneš sa narednim projektom, ali prvo hajde da se prisjetimo šta smo naučili u prethodnom poglavlju!

Naučili smo da Scratch ima različite vrste blokova: kretanje, izgled, zvuk, događaji, osjećaji, operatori, promjenjive. Takođe smo naučili i da možemo imati više likova, ali i pozadina. Za svakog lika treba da koristimo posebne blokove, a naučili smo i da se krećemo naprijed-nazad, skočimo, kao i da mijenjamo veličinu likova!

Sada je vrijeme da otvoriš novi projekat i imenuješ ga “Matematika”. U ovom poglavlju ćemo učiti o logičkim operatorima, ali hajde prvo da počnemo sa uslovima. Ne brini, sve ćemo jednostavno objasniti.

3.1 Uslovi

Kada govorimo o uslovima, nemoguće je izostaviti blokove koji se izvršavaju samo ako je neki uslov zadovoljen!

Ne dozvoli da te uplaši njihov drugačiji izgled, s razlogom su takvog oblika. Kroz ovo poglavlje i korake ćeš vrlo lako ispratiti svaku komandu!



Projekat “Matematika” Primjer 1.

Ideja: Kada je slovo **T** pritisnuto, neka Scratch mačak pita: "Koliko je $4 * 19$?" Ako je odgovor tačan, neka mačak kaže "Bravo!". Ukoliko je odgovor netačan, neka mačak kaže "Netačan odgovor".

Koristićemo komandu odgovor iz bloka **Osjećaj**.

Pronadi blok odgovor iz grupe „Osjećaji“. Ne zaboravi da promjeniš tekst unutar ovog bloka!

Nalazi se u bloku komandi “događaji”

“Odgovor” se nalazi u bloku “Osjećaji”.

Naći ćeš ga u bloku “Upravljanje”. Unutar njega se nalazi prazan dio koji kasnije popunjavaš.

Blok “Operatori”. Prvo polje je prazno u koje kasnije ubacuješ “odgovor”

```
када је дирка t притиснута
питај Колико је 4*19? и чекај
ако је одговор = 76 онда
изговори Bravo!
у супротном
изговори Netačan odgovor.
```

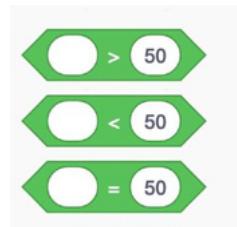
```
када је дирка t притиснута
питај Колико је 4*19? и чекај
ако је одговор = 76 онда
изговори Bravo!
у супротном
изговори Netačan odgovor.
```

U bloku “Izgled” ćeš pronaći izgovore.

3.1 Logički operatori

- Logički operatori su operatori logičkih izraza i mogu imati samo dvije vrijednosti: TAČNO i NETAČNO.

IZRAZ	VRIJEDNOST
$2 > 0$	TAČNO
$5 < 2$	NETAČNO
$7 = 16$	NETAČNO
$(34 : 2) > 15$?
$(14 \times 2) \leq 28$?



Logičko "i", "ili" i "ne" su novi operatori.

- "i" - da bi izraz bio tačan, **svi uslovi moraju biti tačni**.
- "ili" - da bi izraz bio tačan, **bar jedan uslov mora biti tačan**.
- "ne" - da bi izraz bio tačan, **uslov mora biti netačan, i obratno**.

IZRAZ	VRIJEDNOST
$(2 > 0) \text{ i } (5 < 3)$	NETAČNO
$(2 > 0) \text{ ili } (5 < 3)$	TAČNO
$(7 = 16) \text{ ili } (0 < 3) \text{ ili } (2 < 1)$?
$(7 = 16) \text{ ili } (0 < 3) \text{ ili } (2 < 16)$?
ne $(2 < 10)$	NETAČNO
ne $(12 < 10)$?



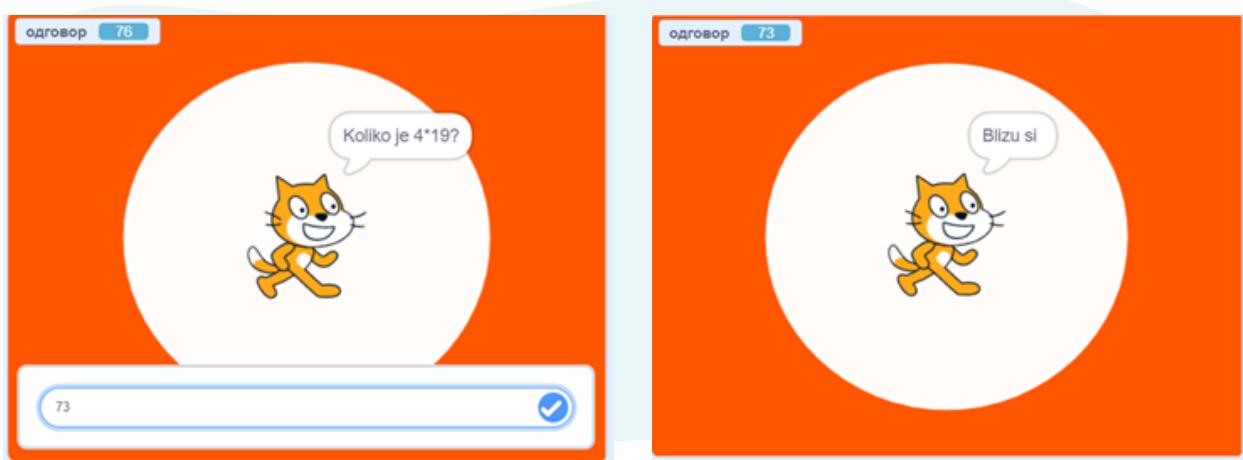
Primjer 2.

Za sada smo napravili da, kada je dirka **T** pritisnuta, naš lik pita "Koliko je $4*19$?" Ako je odgovor tačan, lik govori "Bravo". Prilično jednostavno, zar ne?

Hajde da našeg mačka napravimo malo zabavnijim.

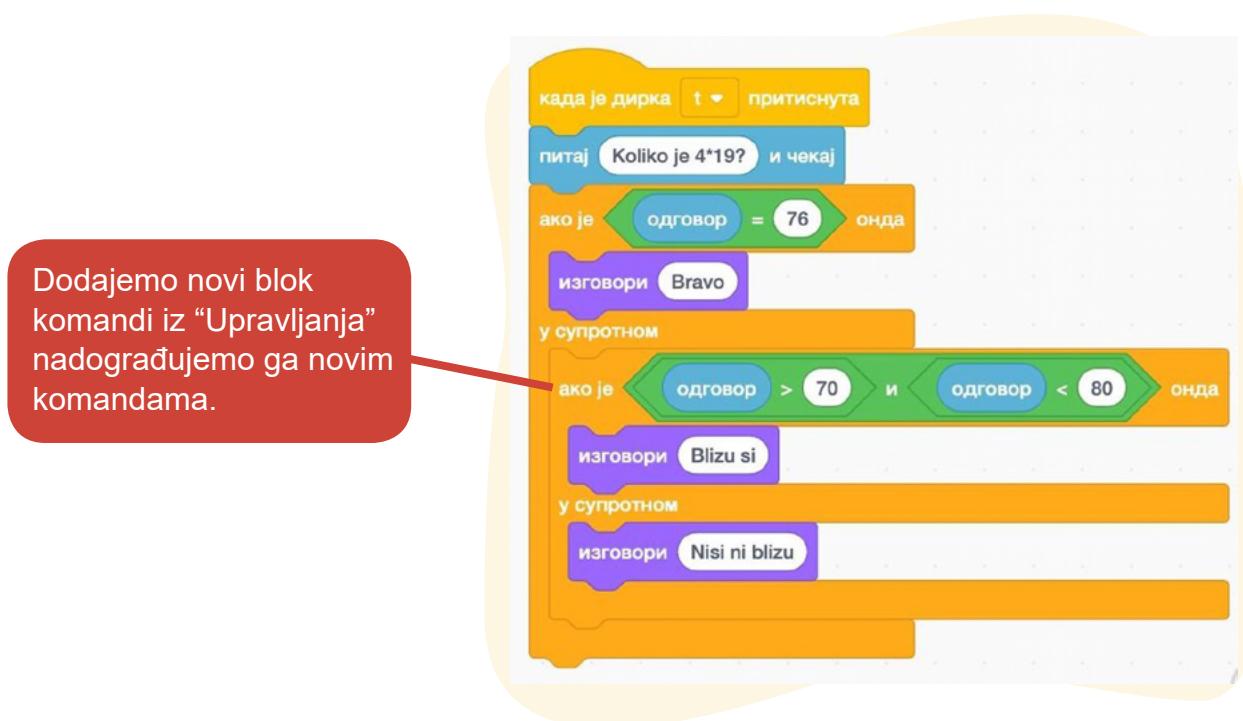
Ideja: Ukoliko odgovor nije tačan, a veći je od 70 i manji od 80, lik govori "Blizu si" (kao na slici ispod). Ako odgovor nije broj između 70 i 80, onda lik govori "Nisi ni blizu".

Kako to da postignemo?



Već napisanom kodu dodajemo novi blok komandi iz bloka **upravljanje**.

Taj blok nadograđujemo komandama iz blokova **operatori** i **izgled**.



Ovim kodom smo uspjeli da ispunimo sve zadate uslove i završavamo projekat "Matematika". Na isti način možeš postaviti razna pitanja i odgovore. Nakon što projekat objaviš, možeš ga **umnožiti**, pa link projekta možeš poslati svom društvu kako bi testirao njihovo znanje.

Projekat "Kretanje"

Primjer 1.

Vrijeme je za naredni projekat! Uzbuđenje raste zar ne?

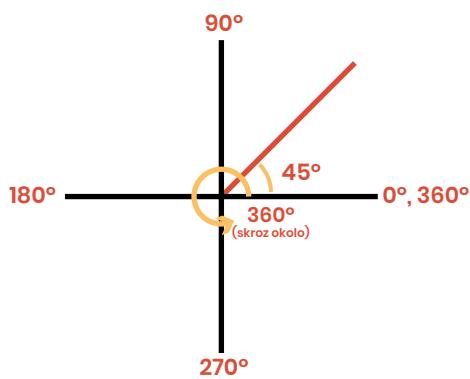
Imenuj prvo novi projekat "Kretanje" i za kratko se prisjeti onoga što smo radili u prethodnom poglavlju jer ćemo sada nadograditi taj kod. (Drugo poglavlje, Primjer 5.)

Ideja: Naš lik se okreće u smjeru svog kretanja.

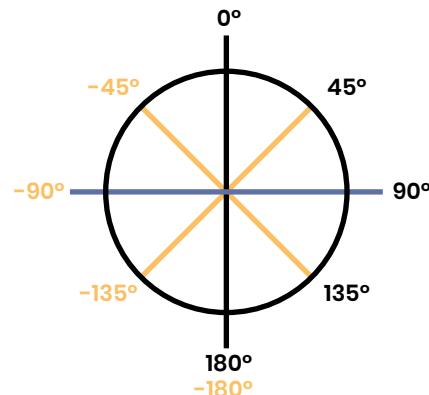
Da bismo to napravili, koristićemo jedan od dostupnih blokova kretanja: **smjer**. Da bismo vidjeli koja je vrijednost smjera, **štrikirajmo boks kao na slici**.



•Uglovi u Scratch-u



•Smjerovi u Scratch-u

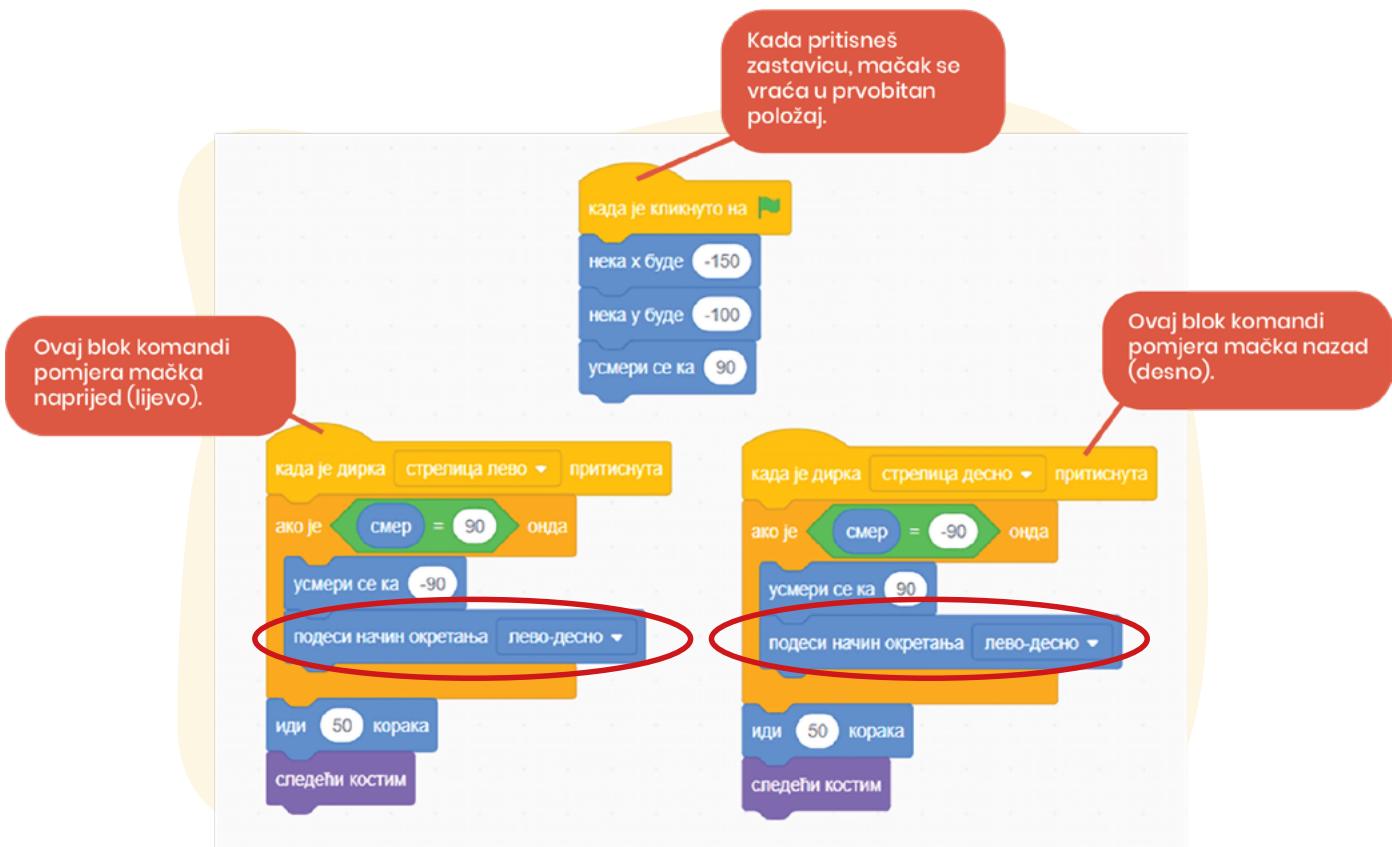


Na početku, **kada klikneš zelenu zastavicu, postavićeš mačka pri sredini ekrana** podešavajući njegove X i Y koordinate, i usmjerićeš ga naprijed (90 stepeni). Koordinate sredine ekrana su (0,0).

Ideja: Ukoliko je lik trenutno usmjeren naprijed, okrenimo ga i pomjerimo nazad 50 koraka. Istu logiku primjenjujemo i za kretanje unaprijed.

Kako bi se lik kretao pravo naprijed potrebno je da smjer bude podešen na 90 stepeni.

Zadaj komande kao sa slike ispod.



Primjetićeš da smo koristili blokove **иди 50 корака** umjesto **промјени X за 50** (или -50), kako smo to ranije činili. Zašto to sada možemo da uradimo? Zato što smo prije toga, lika usmjerili tako da može da nastavi da se pomjera u tom smjeru.

Primjer 2.

Ideja: Kada lik dođe do ivice pozornice, pozadina se mijenja.

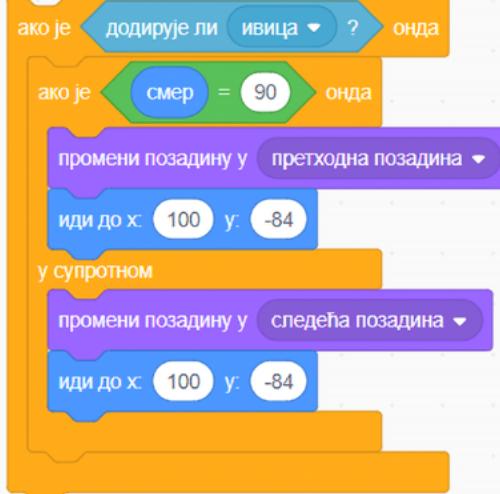
Nadogradimo kod tako da se pozadina mijenja kada lik dođe do ivice, tako da izgleda kao da se stvarno kreće. Za tu funkciju koristimo jedan od dostupnih blokova **kretanja: dodiruje li**. Možemo da odaberemo da lik dodiruje ivicu ili nekog od drugih likova našeg programa.

Napomena: Dodaj još 2 nove pozadine po izboru.



Važno: Ovi blokovi moraju da se sastave sa nekim **događajem** da bi se ikad pokrenuli! Pa zbog toga stavljamo ovaj blok na samom početku.

када је кликнуто на

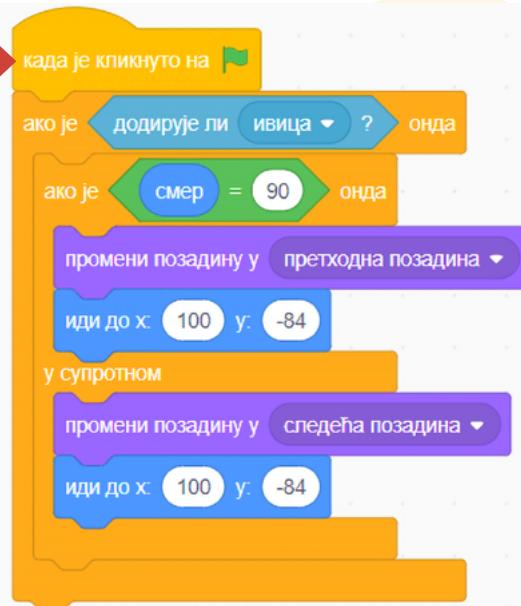


Zašto ne radi?

Kad je kliknuta zastavica, odmah se provjerava uslov **da li lik dodiruje ivicu**. Obično to nije slučaj, i onda se blokovi ne izvrše.

Šta se dešava kada lika dovedemo do ivice i kliknemo zastavicu?

Da li se pozadina promijenila?

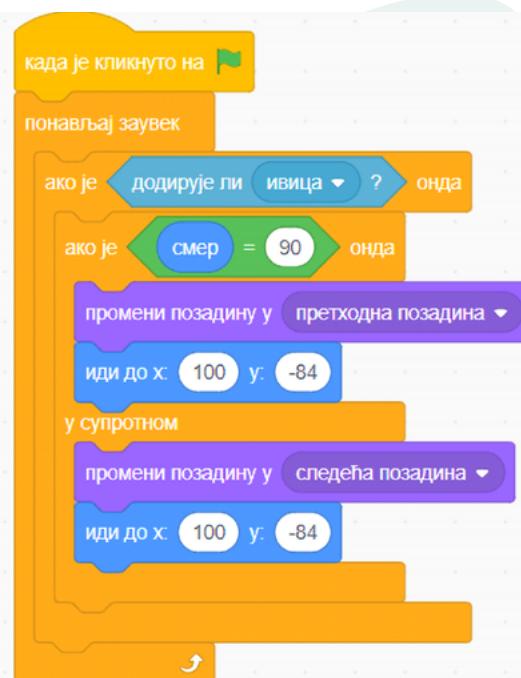


Zašto sada radi?

Od kada je kliknuta zastavica, stalno se provjerava da li lik dodiruje ivicu. Kad je to slučaj, blokovi unutra se izvršavaju i pozadina se mijenja. Kako bi se ova komanda uvijek izvršavala, dodajemo **ponavljam zauvijek**.

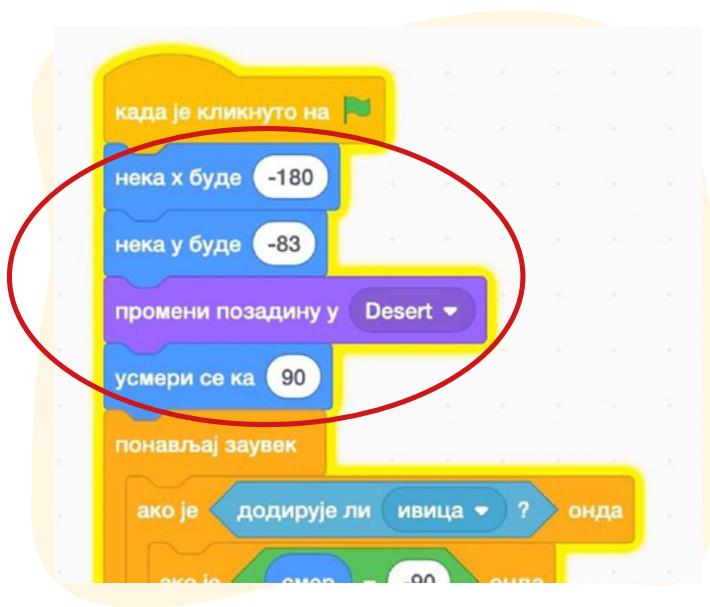
Kada kliknemo na zastavicu, da li se lik vraća na prvobitnu pozadinu i kreće odatle?

Šta je još potrebno dodati?

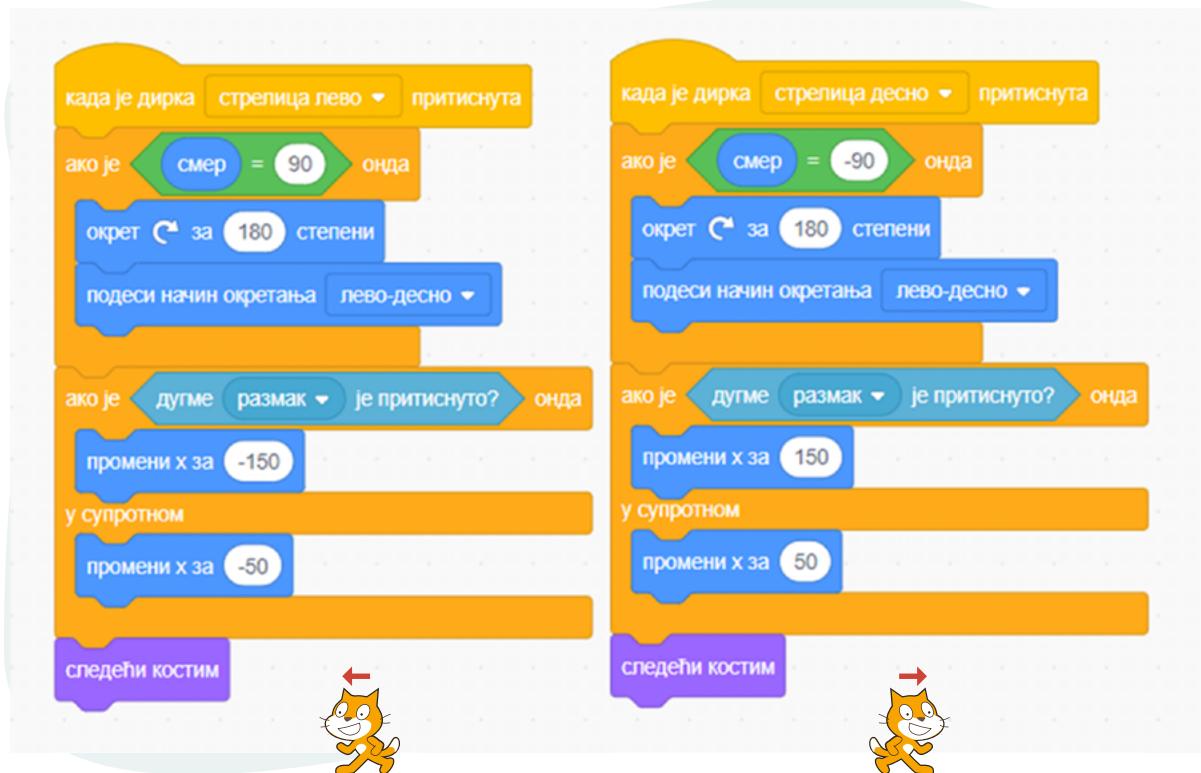


Zašto sada radi?

Zato što, svaki put kada kliknemo na zastavicu, resetujemo poziciju lika (X i Y). Na taj način postavljamo početnu pozadinu i usmjeravamo lika ka naprijed.



Promjena pozadine kada lik stigne do ivice zahtijeva komandu iz bloka **izgled** i bloka **kretanje**.



Uspjeli smo da završimo i treći projekat ovog priručnika. Hajde da sada na konkretnim zadacima primijenimo znanje! :D

Zadaci za vježbanje:

Projekat 1.

1. Napravi poseban projekat koji će nositi naziv "Računanje"
2. Neka tvoj lik bude čarobnjak "Merlin", koji će, kada pritisne X pitati "Koliko je $12 \cdot 8$ "?
3. Ako je odgovor tačan, neka kaže BRAVO!
4. Ako odgovor nije tačan, već je 95 ili 97, neka kaže BAŠ SI BLIZU!
5. Ako odgovor nije BAŠ BLIZU, a između je 80 i 90, neka kaže BLIZU SI!
6. Ako odgovor ponovo nije baš blizu, a veći je od 0, neka kaže MAKAR JE ODGOVOR POZITIVAN.
7. Ako odgovor nije ni pozitivan, neka kaže TO NIJE DOBRO.

Pomoć: Pogledaj primjer 2. drugog poglavlja, ali obrati pažnju i da je u skladu sa trećim korakom.



[Urađenu verziju zadatka možeš pronaći ovdje.](#)

Projekat 2.

1. Napravi poseban projekat koji će nositi naziv "Medved", pa za svog lika izaberi medveda koji šeta.
2. Neka se medved kreće naprijed i nazad uz okretanje.
3. Kada klikneš strelicu gore neka medved skoči.
Pomoć: Pogledaj drugo poglavlje, Primjer 6.
4. Projekat treba da ima makar 3 pozadine, i one treba da se mijenjaju kada medved stigne do ivice, tako da njegovo kretanje izgleda što realnije. U ovom projektu takođe promijenite i veličinu medveda na 75.
Pomoć: Pogledaj projekat "Kretanje" iz trećeg poglavlja.



[Urađenu verziju zadatka možeš pronaći ovdje.](#)

4. Poglavlje

Na samom početku ovog priručnika rekli smo da uz pomoć Scratch-a možemo da stvaramo igre. Došao je red da se polako upoznamo sa komandama koje će nam pomoći da na kraju ovog priručnika napravimo igru iz mašte!

Kako bismo što bolje ispratili sledeće primjere, a i olakšali sebi dalje stvaranje, sve veće blokove komandi iz prethodnih projekata prenesi u **Ostavu** kako bismo ih koristili u daljem radu.

Projekat “Igra”

1. Nadogradimo projekat “Kretanje” (Poglavlje 3.) sledećim koracima.
ILI → U programu, iz ostave, možemo prenijeti komande koje smo kreirali u projektu “Kretanje”, a zatim kreirati novi projekat i nazvati ga “Igra”.
2. Postavljamo 3 različite pozadine i to: Jurassic, Savanna i Jungle.
3. Dodajemo novog lika - zvijezdu.

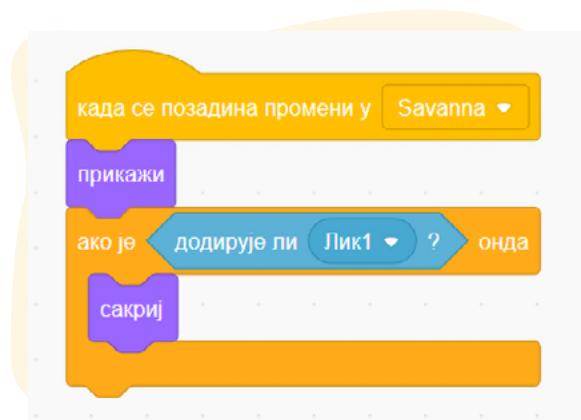
Ideja:

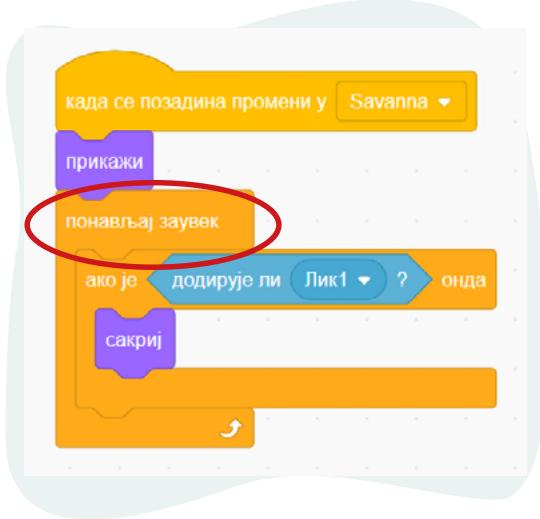
1. Dodajemo zvijezdu.
2. Kada se pozadina promjeni u Savanna, zvijezda se pojavljuje.
3. Ako je dodirne naš lik, zvijezda nestaje. → Na pozadini Savana se pojavljuje zvijezda. → Naš lik kupi zvijezdu ako je dotakne.
4. Nakon što glavni lik skupi 20 ili preko 20 poena, pojavljuje se čarobnjak.

Primjer 1.

Kako bismo komande mogli da slažemo u programu zvijezde, potrebno je da u donjem desnom dijelu ekrana kliknemo na lik zvijezde. Sada slažemo komande baš onako kako je dato na slici ispod.

Da li sve radi kako treba?
Kako to da popravimo?
Ne smijemo zaboraviti
ponavljanje zauvek!





Kako bismo ispravili ovu grešku, na samom početku treba uvijek da postavimo **sakrij** kako se zvijezda ne bi vidjela.

Ovakve greške se u programiranju nazivaju bug.

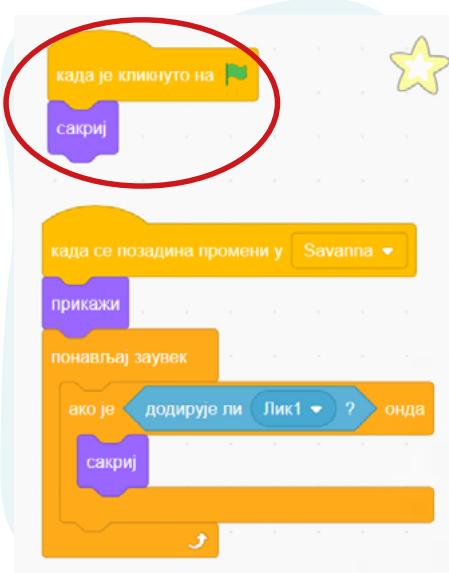
Da li znaš zašto?

Da li sada radi kako treba?

Dodi blizu zvijezde, ali je nemoj pokupiti.

Da li je zvijezda vidljiva na ostalim pozadinama?

Kako to da popravimo?

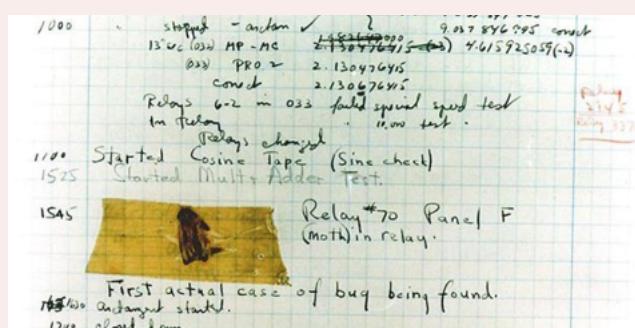


Prvi računarski bug

Septembra 1947. Grace Hopper je prijavila prvi računarski bug (grešku)! Nakon cjelodnevne istrage, zbog problema sa računarom, saznali su šta nije bilo u redu.

Ispostavilo se da računar nije radio kako treba, jer je u njemu bila zarobljena buba (na engleskom: bug) koja je remetila elektroniku računara.

Operativci su uklonili moljca i zaličepili ga u dnevnik rada. Sam unos u evidenciji glasio je: "Prvi stvarni slučaj nađene bube (bug)". Ubrzo potom je nastao i termin "debagovati" za proces popravljanja koda i programa koji ne rade kako bi trebali.

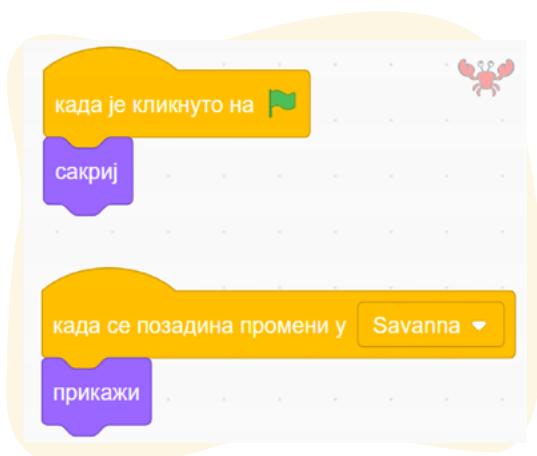


Primjer 2.

Nadogradimo kod iz Primjera 1. tako da se na pozadini Underwater 1 pojavljuje kraba. Naš lik nestaje ako dotakne krabu..

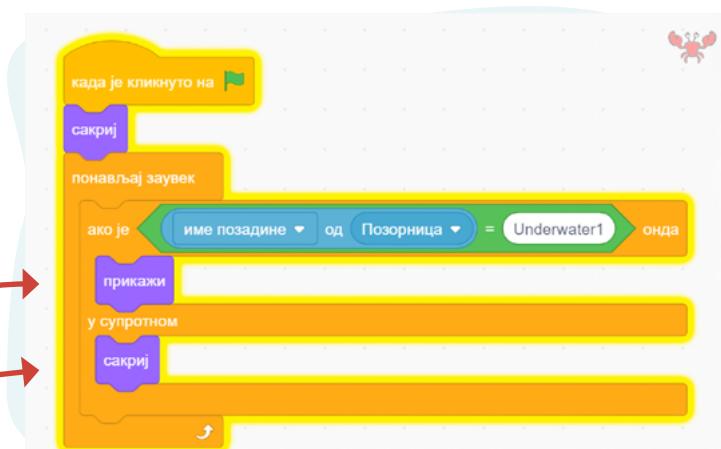
Ideja:

1. Dodaj pozadinu Underwater 1 (ako je već nemamo).
2. Dodaj krabu.
3. Kada se pozadina promjeni u Underwater 1, kraba se pojavljuje.
4. Ako ga dodirne, naš lik nestaje.

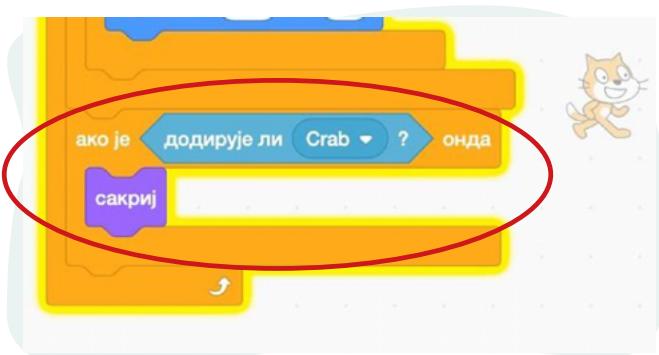


Da li se kraba pojavljuje i na svakoj pozadini nakon Underwater 1?

Bez brige! Samo treba dodati **понављaj заувек**. Ako je trenutna pozornica Underwater 1, koristimo **prikaži**. Ako nije, **sakrij**.



Kako bi se ideja ispunila u potpunosti, i kako bi naš lik nestao kada dotakne krabu, nadogradimo kod našeg glavnog junaka.

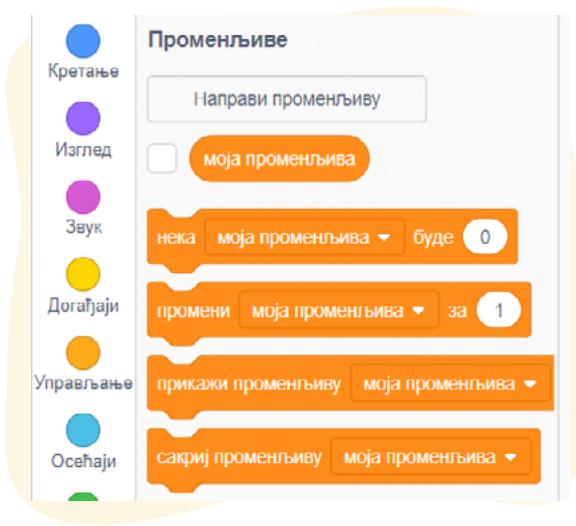


Unutar petlje **понављaj заувјек**, dodajemo uslove iz blokova **управљање, осјећаји и изглед**.

4.1. Promjenjive

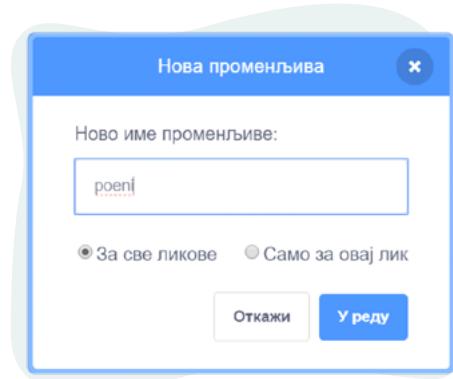
Jedna od grupa blokova u Scratch-u su i promjenljive. Vrijednost se obično mijenja u toku programa u zavisnosti od naših potreba i može biti bilo šta!

Sjeti se, rekli smo da želimo da napravimo igru. Pa kakva je to igra bez poena? Hajde da popravimo stvari!



Klikom na **Napravi promjenljivu** možemo napraviti promjenljivu.

Promjenjivoj dajemo ime. U ovom slučaju je to **poeni**. Promjenjiva može da se koristi za sve likove ili samo za ovaj lik. Ako se koristi za **samo ovaj lik** (mačak), kada kliknemo na raka ili zvijezdu, nećemo vidjeti poene među promjenljivima.

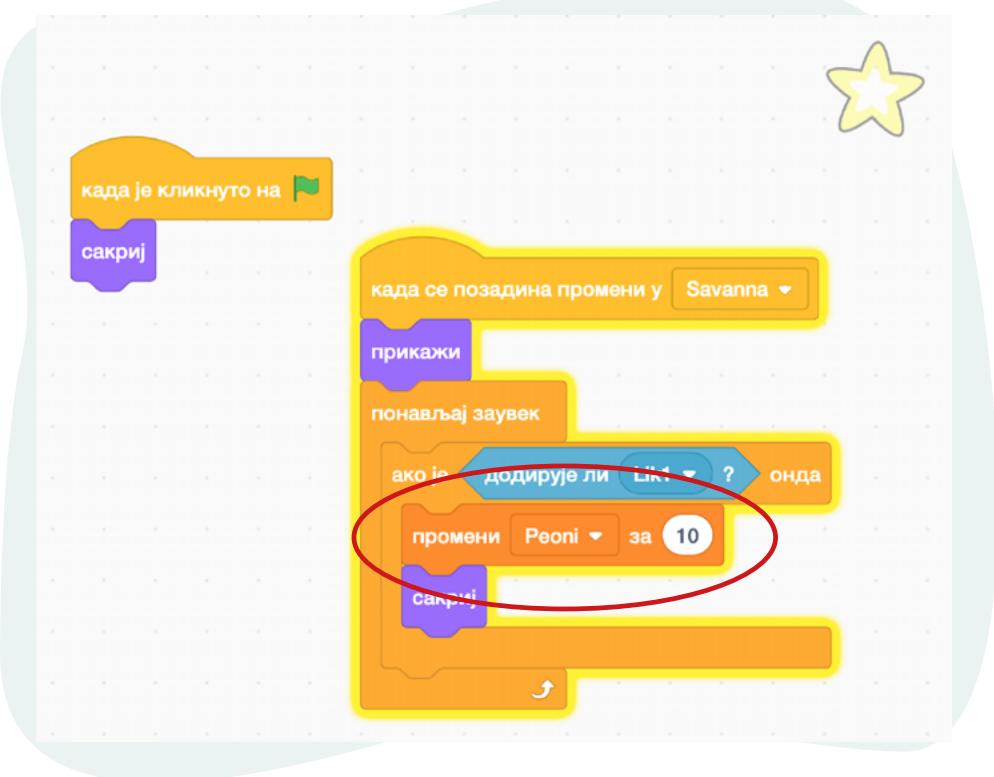


Primjer 3.

Sada kada imamo promjenjivu - poene, vrijeme je da ih iskoristimo.

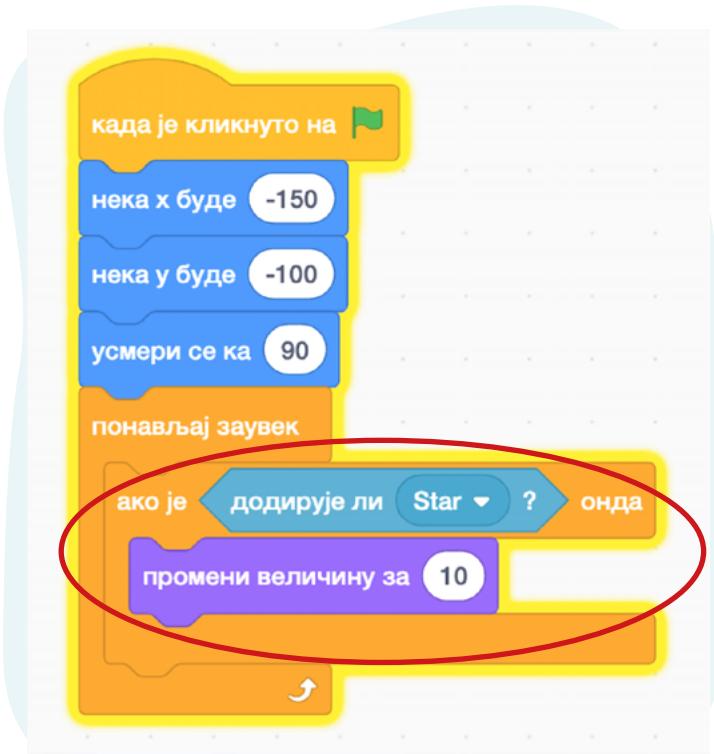
Ideja: Broj poena se povećava za 10 svaki put kada mačak kupi zvjezdicu, dok na početku mačak ima 0 poena.

Kako bi ovo postigli, nadograđujemo blok komandi koje smo prethodno postavili na lik zvijezde. Dodajemo komandu iz bloka **promjenjive**.



Na isti način možemo i na Krabin set komandi da promijenimo poene za -3 i naš lik će gubiti poene svaki put kada dotakne Krabu.

Primjer 4.



Ideja: Kako bismo igricu načinili zanimljivijom, hajde da dodamo mogućnost da se mačak povećava svaki put kada dotakne zvijezdu!

Za to je potrebno da nadogradimo kod našeg glavnog junaka i to u sklopu komande **ponavljam zauvijek**.

4. 2. Čekaj dok ne

Vrijeme je da kreiramo igru do kraja! Ideja je da dodamo čarobnjaka nakon što mačak sakupi 20 ili preko 20 poena, a za to koristimo novu komandu!

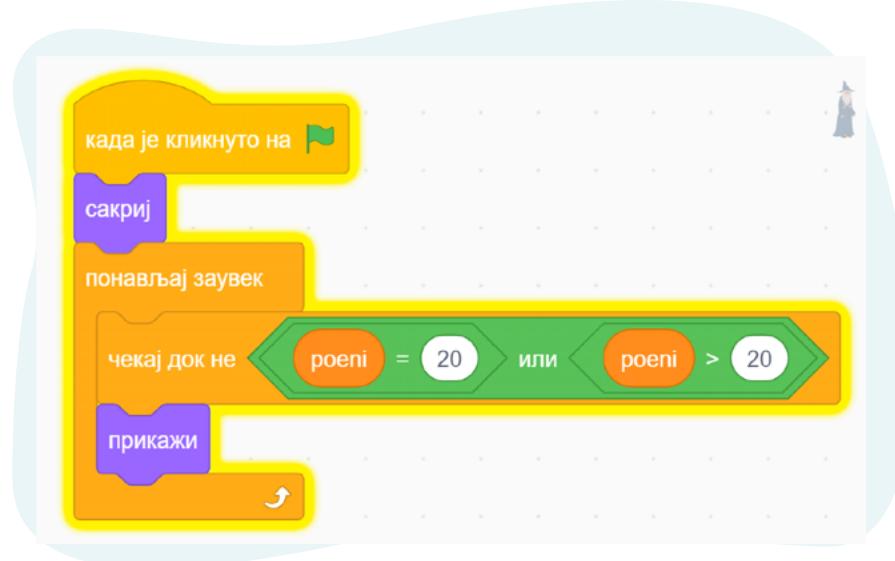


Ovaj blok se koristi zajedno sa uslovom (uslov se stavlja u prazni šestougao). Ono što slijedi se dešava tek kada je uslov ispunjen. **Čekaj dok ne** znači da se sve što slijedi **ne izvršava** ukoliko uslov na koji se čeka nije zadovoljen.

Primjer 1.

Kao što smo rekli, ideja je da dodamo novog lika (čarobnjaka) koji se pojavljuje ako je broj skupljenih poena makar 20. To znači da uslove moraš postaviti u okviru komande **ponavljaj zauvijek**, koristeći **čekaj dok ne**. Postavi dva uslova i to da poeni budu **jednaki 20**, ali i da budu **veći od 20** kako bi se naš novi lik prikazao.

Za ovaj uslov takođe koristimo i komandu iz bloka **operatori** kojima dodajemo promjenjivu **poeni**.



Ovim setom komandi završavamo našu prvu igru! Ne zaboravi da koristiš **Ostavu** kako bi u narednom poglaviju mogli da koristimo ove kodove uz dodatne izmjene. Takođe, kodovi koje budemo kreirali, biće odlična osnova za kreiranje tvoje jedinstvene igre!

Zadatak za vježbu:

Koristeći se instrukcijama iz četvrtog poglavlja, kreiraj sopstvenu igru, na isti način.

1. Ovaj projekat nazovi "Magičan svjet".
2. Neka tvoj glavni lik bude žabac Srećko, i neka njegova veličina bude 80.
3. Neka se kreće lijevo-desno, a kada stigne do ivice, neka se pozadina promjeni.
4. Kada pritisneš razmak, neka žabac skoči.
5. Dodaj 2 nova lika. Neka jedan bude čarobni napitak, a drugi kristal.
6. Izaberi po jednu pozadinu na kojoj će se naći ova 2 lika. I neka to budu različite pozadine.
7. Kada žabac dotakne kristal, neka dobije 10 poena, i neka kristal nestane.
8. Kada žabac dotakne čarobni napitak, neka izgubi 3 poena, i neka napitak nestane.
9. Kada žabac skupi 30 poena, neka se pojavi vještica koja se zove Sabrina.
10. Neka Sabrina kaže "Bravo!".



[Našu verziju zadatka možeš pronaći ovdje.](#)

5. Poglavlje

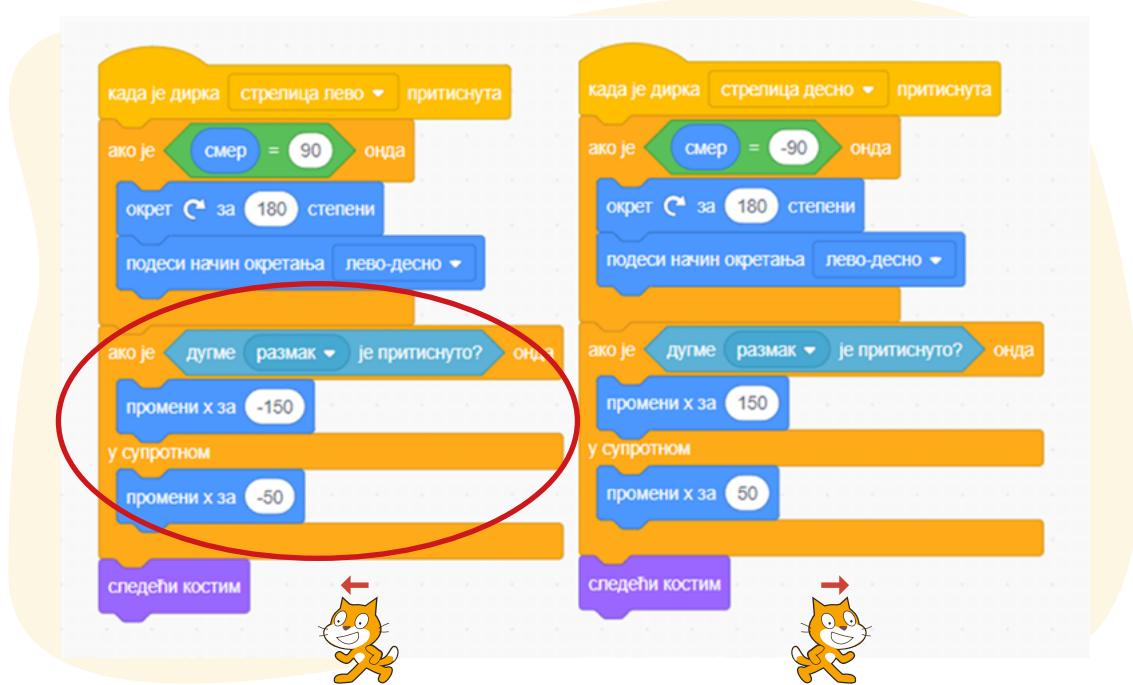
Sada kada smo kroz zadatak za vježbu obnovili znanje, vrijeme je da nastavimo dalje i susretnemo se sa novim komandama koje će nam pomoći da naše naredne igre budu još zanimljivije!

Primjer 1.

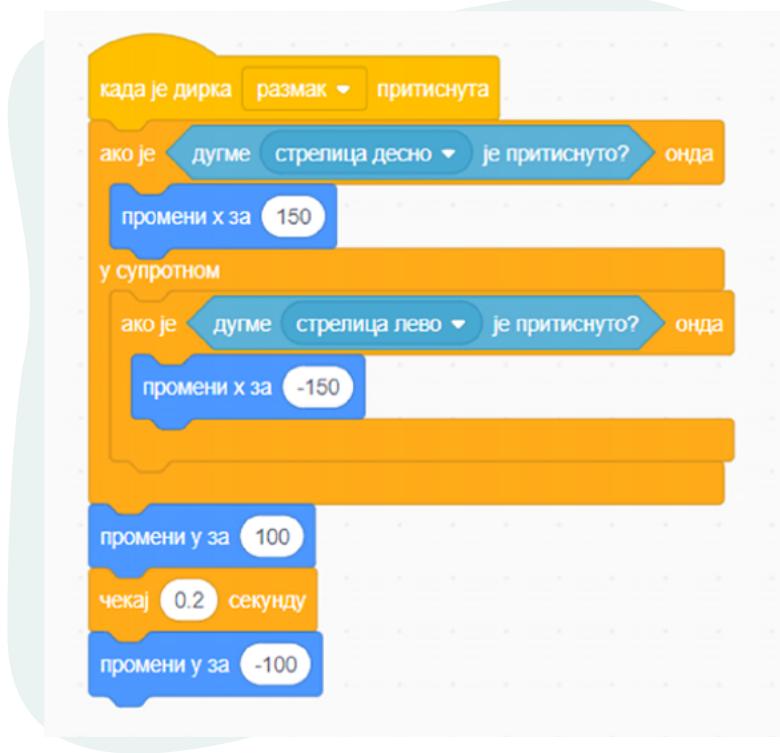
Kao i u svakoj igri, potrebno je da znamo kako možemo preskočiti ili zaobići negativnog lika. Za to je potrebno da naučimo kako da postavimo komande da naš lik u skoku može preskočiti neprijatelja i pri tome zaobići gubljenje poena. **Možeš koristiti projekat iz prethodnog poglavlja i to kod kretanja i nadograditi ga novim komandama.**

Ideja:

1. Mačak će, kada pritisnemo u isto vrijeme strelicu naprijed ili nazad i razmak, prekoći prepreku ispred sebe.
2. Važno je i da podesimo poziciju prepreke. Odnosno, možemo se igrati veličinom raka, pa ga zbog trenutnih potreba igre, možemo smanjiti i pozicionirati tako da mačak može da ga prekoči.



Međutim, to nije jedini način. **Skok možemo izgraditi i putem jednog bloka komandi**, kao sa slike desno.



Optimizacija

Većina problema u programiranju može da se riješi na nekoliko različitih načina. Optimizacija je biranje onog načina koji je jednostavniji i koji se brže izvršava (u našem slučaju, to je korišćenje manjeg broja blokova).

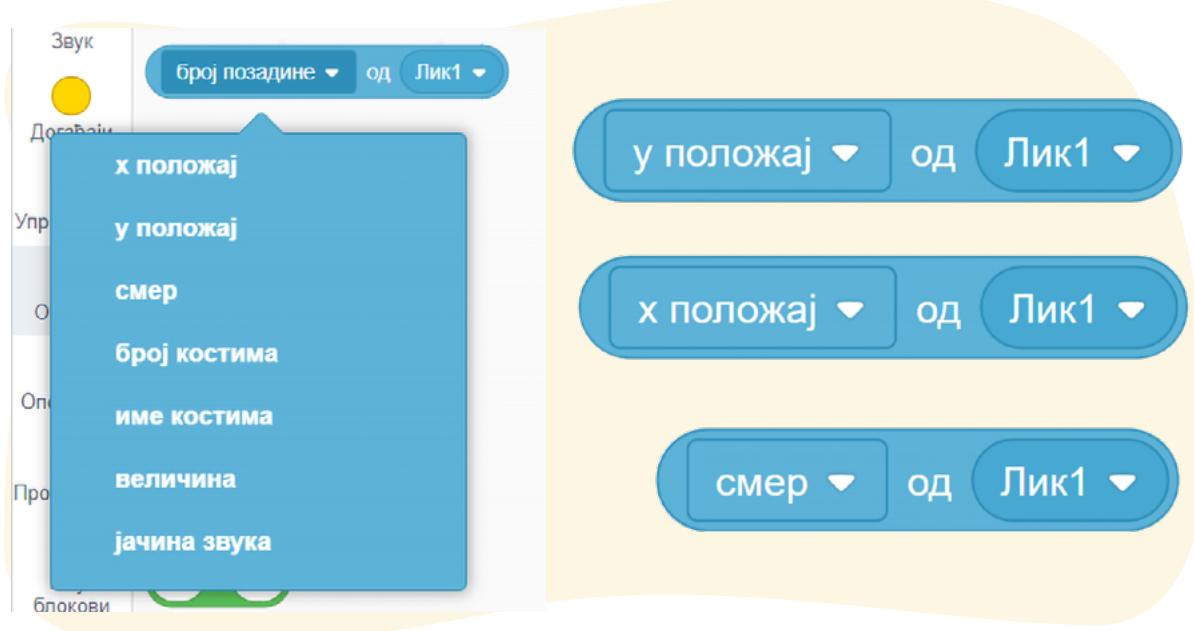
Primjer 2.

Do sada smo naučili kako se mijenjaju pozadine dok se naš lik kreće, kako da dobijamo i gubimo poene, ali i to kako da preskačemo neprijatelje. Došao je red da naučimo našeg mačka da se odbrani i to tako što će moći da gađa kada pritisnemo G.

Ideja:

1. Dodaćemo novog lika kojim će naš junak gađati neprijatelja, a taj lik može biti loptica.
2. Sakrićemo lopticu ukoliko je naš junak ne koristi.
3. Ako naš junak gađa, prikazaćemo lopticu.
4. Treba da razmislimo gdje bi trebalo da se pojavi kada mačak počne da gađa.
5. Gdje bi trebalo da se uputi?

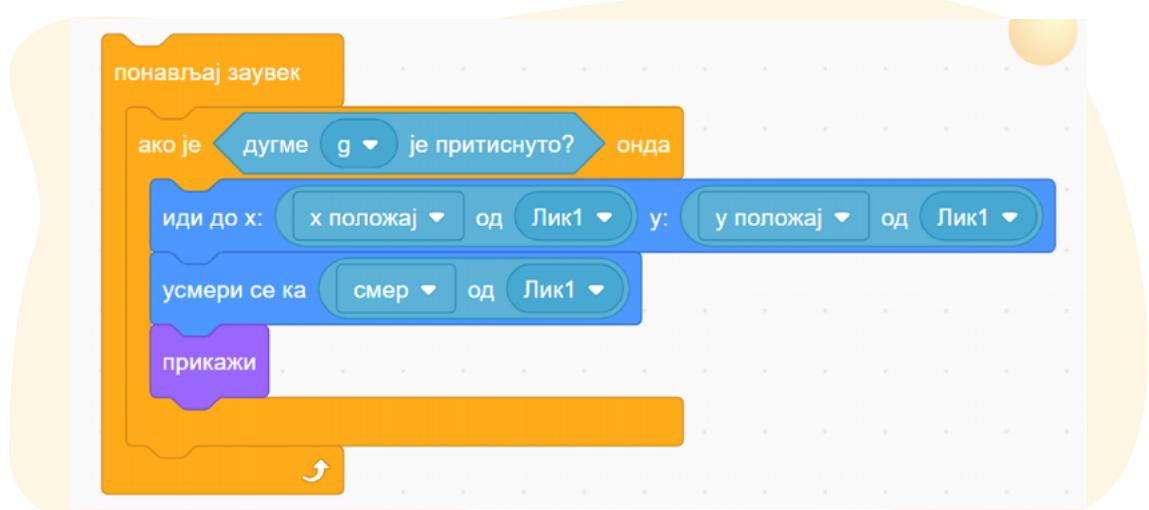
Sada kada smo na pozornici postavili loptu, počećemo da joj pravimo blok komandi. Koristićemo blokove iz grupe **osjećaj**.



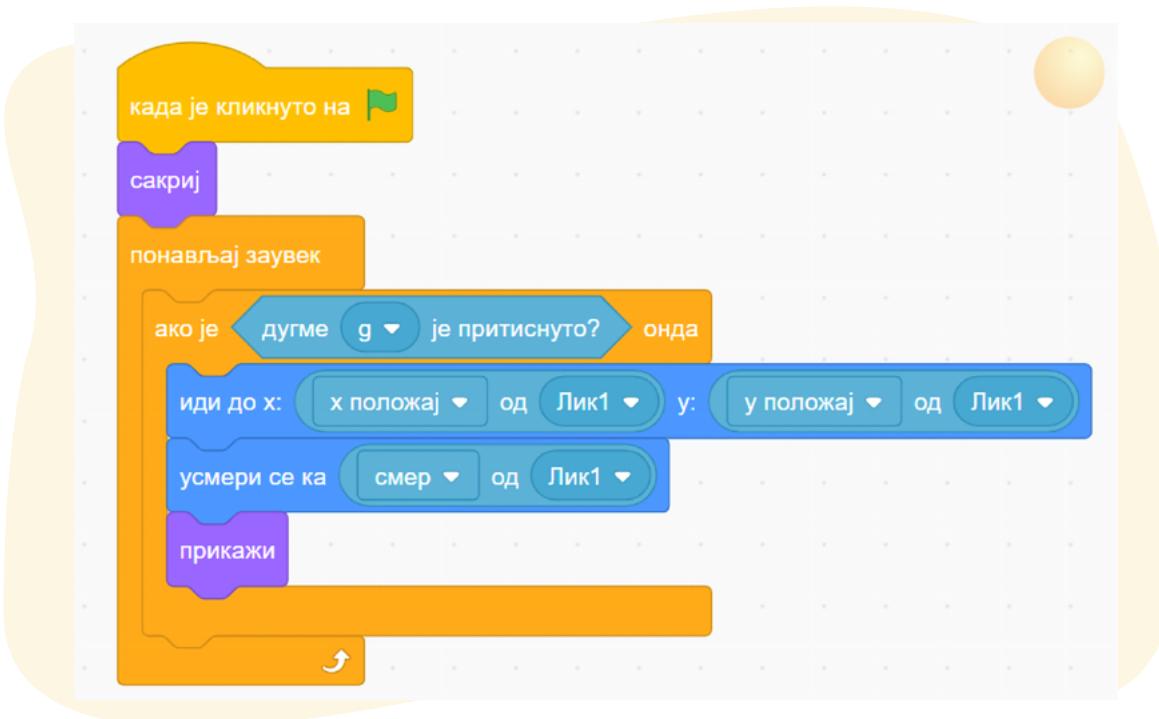
Prvo je potrebno da postavimo i usmjerimo lopticu, pa tek onda da je prikažemo. Postavi je preko mačka kako bi odatle počela da se kreće. Zatim kombinujemo komande iz blokova **osjećaj i kretanje**.



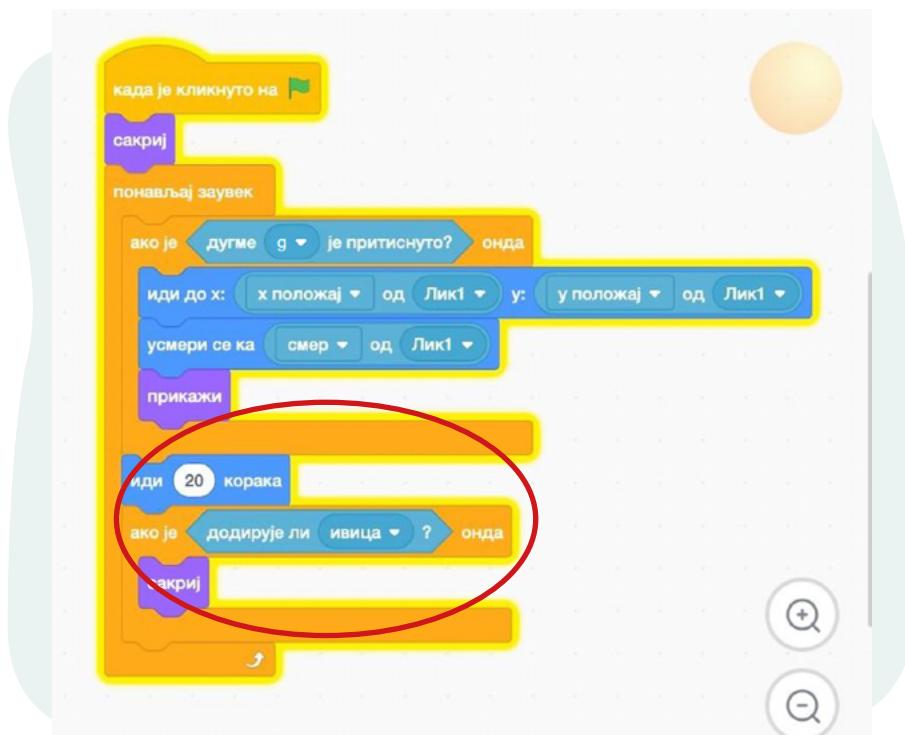
Međutim, uslov je da moguć jedino **ako je притиснуто дугме G**. Pa dodajemo i komandu iz bloka upravljanje koju kombinujemo sa komandom iz bloka **osjećaji**. Takođe, ne smijemo zaboraviti na **понављай заувек**.



Nakon što smo kreirali ovaj blok komandi, moramo biti sigurni da će **loptica na samom početku biti sakrivena** dok joj ne damo komandu da se kreće.

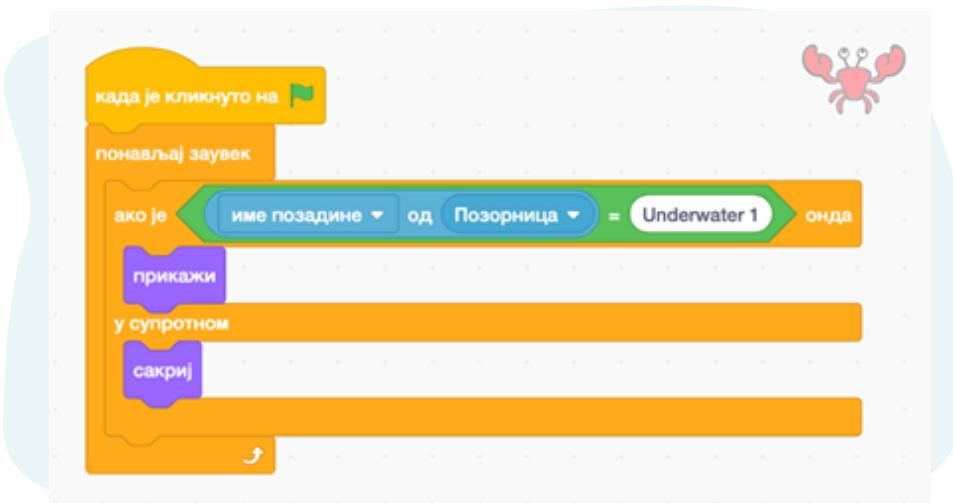


Sada kada smo je sakrili, želimo da se loptica negdje kreće, a ne samo da se prikaže. Loptica je već usmjerena, pa joj sada zadaj komandu da se kreće 20 koraka. Da ne bi ostajala zaglavljena, sakrij lopticu ako dođe do ivice.

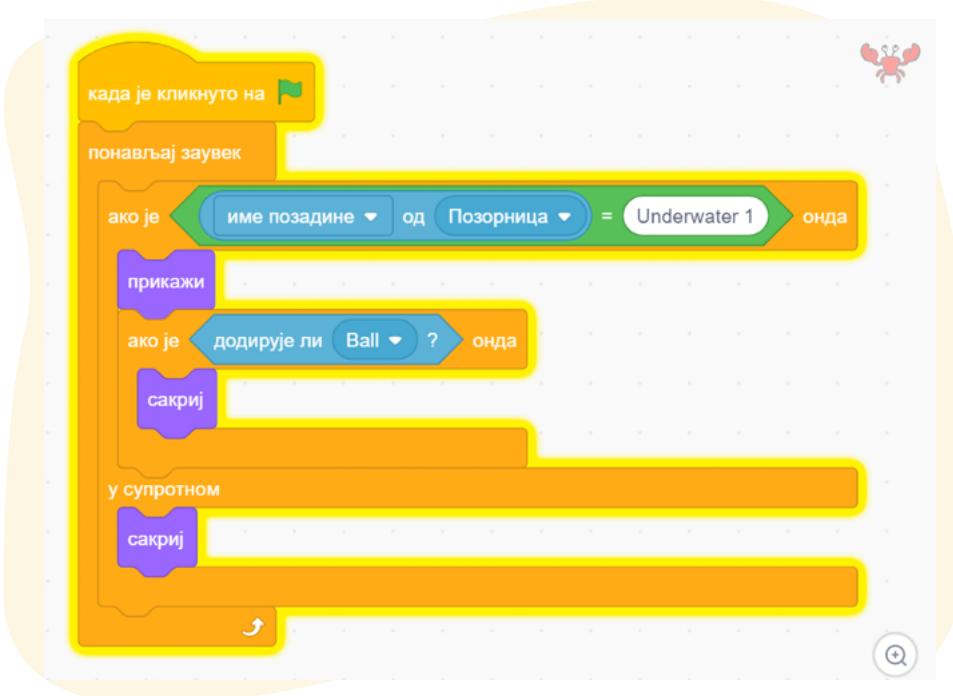


Primjer 3.

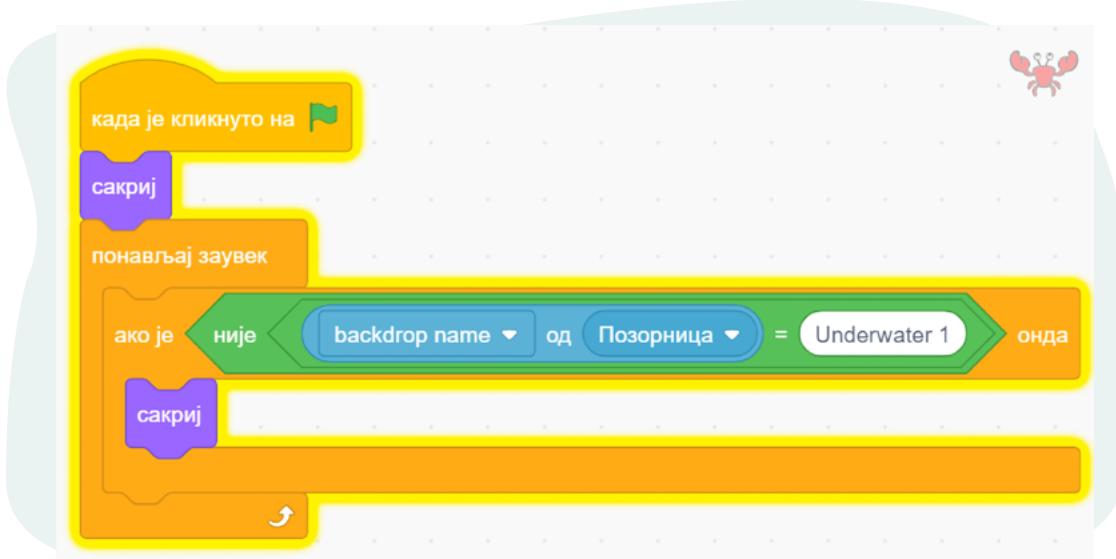
U sledećem zadatku treba da budeš programer spasilac i izmijeniš kod tako da rak nestaje ukoliko ga mačak pogodi lopticom. Ovo je njegov trenutni kod:



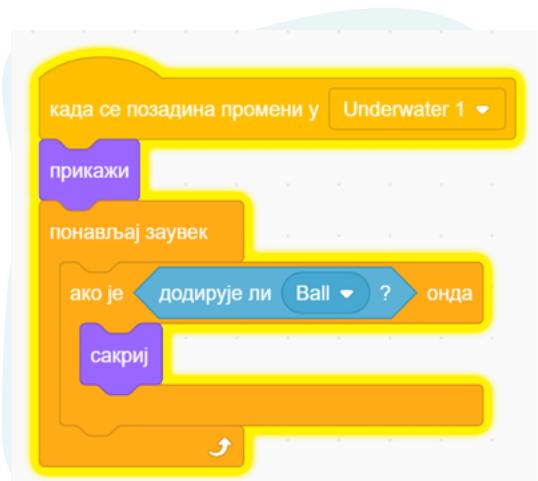
Šta ako, nakon što prikažemo raka, dodamo kod koji provjerava da li ga je loptica dotakla? Ovaj kod se izvršava u bloku **ponavlja zauvijek**, tako da, čim loptica pogodi raka, pitamo da li je pozadina i dalje *Underwater 1*, i ukoliko nije, rak se opet pojavljuje.



Na početku sakrij raka. Uvijek provjeravaj da li je pozadina neka druga osim Underwater 1, kako bi tada sakrili raka.

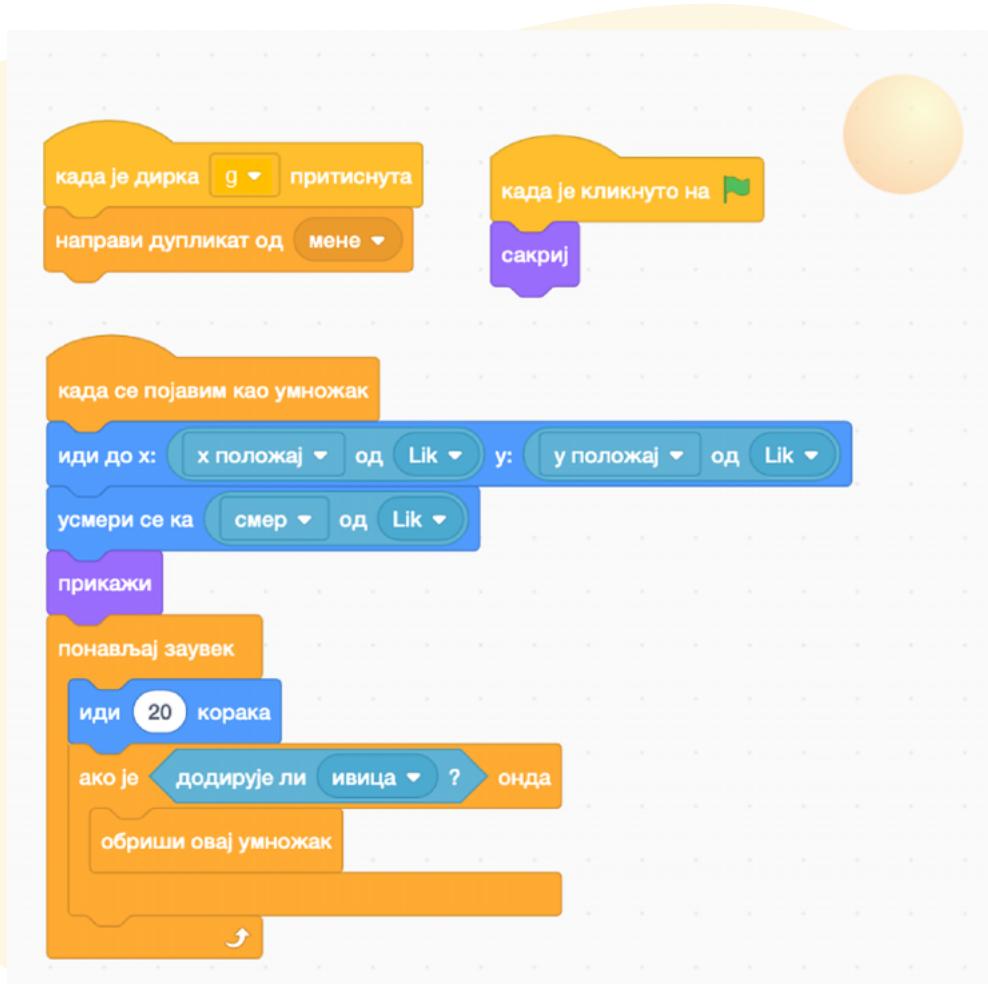


Kada se pozadina promijeni u *Underwater 1*, prvo prikažemo raka, a zatim uvijek pitamo da li rak "pogođen", i krijemo ga ako jeste.



Primjer 4.

Kako bi unaprijedili kod, **tako da lik može da ispali više loptica**, mijenjamo kod za Ball (opticu) na sledeći način:



Sada znaš sve što je potrebno kako bi kreirao/la pravu igru. Napravi malu pauzu, pa onda predi na finalni dio ovog priručnika. Spremi se! :D

6. Poglavlje

Prvi dio

Iskoristi dosadašnje znanje da napraviš finalnu igru. Postavićemo pravila i gradićemo igru po koracima, a na tebi je samo da ispratiš svaki korak.

Pravljenje igre će zapravo biti prilično lako, jer ćemo koristiti znanje iz prethodnih poglavlja, a kombinovaćemo ih tako da kreiramo jedan dinamičan projekat. Ne zaboravi na ostavu koja ti može značajno olakšati izradu igre, kao i mogućnost remiksovanja.

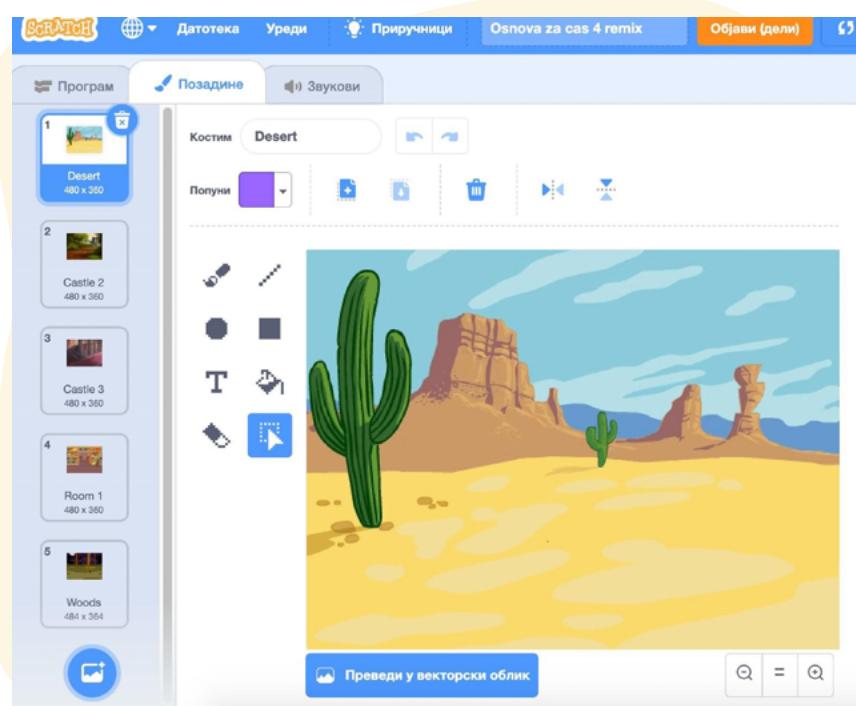
Naslovi ovaj projekat “Finalna igra” i počnimo!

Pravila

- 1.** Lik može da se kreće naprijed. Kada stigne do ivice, pojavi se sledeća pozadina.
- 2.** Lik može da se kreće unazad, ali samo do ivice.
- 3.** Lik može da skače i preskače.
- 4.** Lik može da čučne.
- 5.** Lik može da gađa.
- 6.** Na početku ima 3 života i 0 poena. Kada sakupi preko 30 poena, dobija novi život, a poeni se spuste na 0. Kada broj poena padne ispod 0, gubi život. Ako izgubi sve živote, igrica se zaustavlja i piše KRAJ IGRE.
- 7.** Imamo 6 pozadina. Na 2. i 4. se nalaze likovi koji donose po 5 poena kada se skupe. Na 3. i 5. se nalaze likovi koji oduzimaju po 3 poena ako se dotaknu, a donose 3 ako su pogodjeni.
- 8.** Na pozadini broj 6 se pojavljuje čarobnjak koji postavlja pitanje. Ako je odgovor tačan, čarobnjak nestaje i donosi 3 poena. Ako nije, gubimo 5 poena i čarobnjak nestaje. Nakon 3 tačna odgovora, prelazi se ovaj nivo.

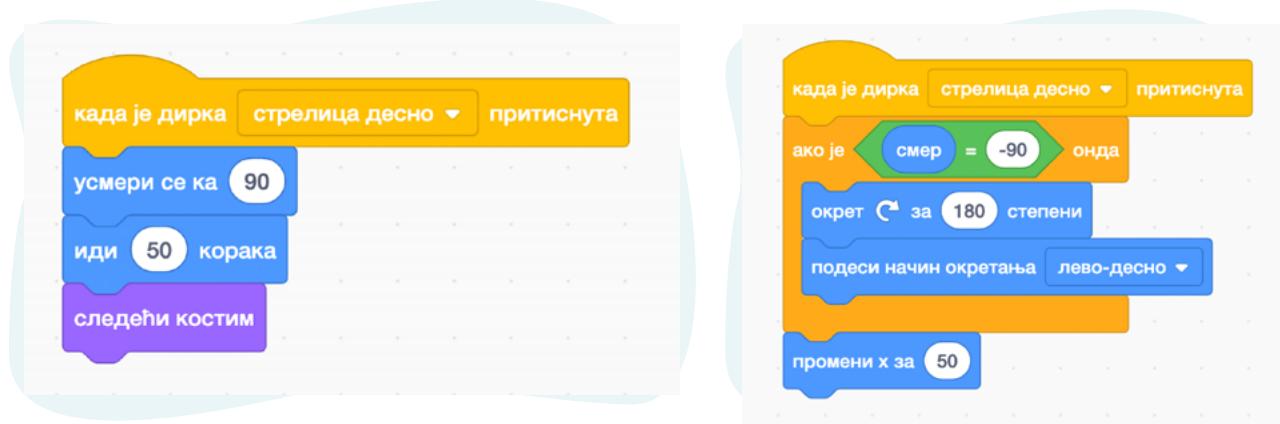
Korak 1. – Pozadine

Dodajemo 6 novih pozadina.



Korak 2. – Kretanje

Naš lik se kreće lijevo i desno uz okret. Na lijevoj strani nalazi se stari kod sa početka priručnika, a na desnoj novi kod za kretanje. Mnogo smo napredovali, zar ne?



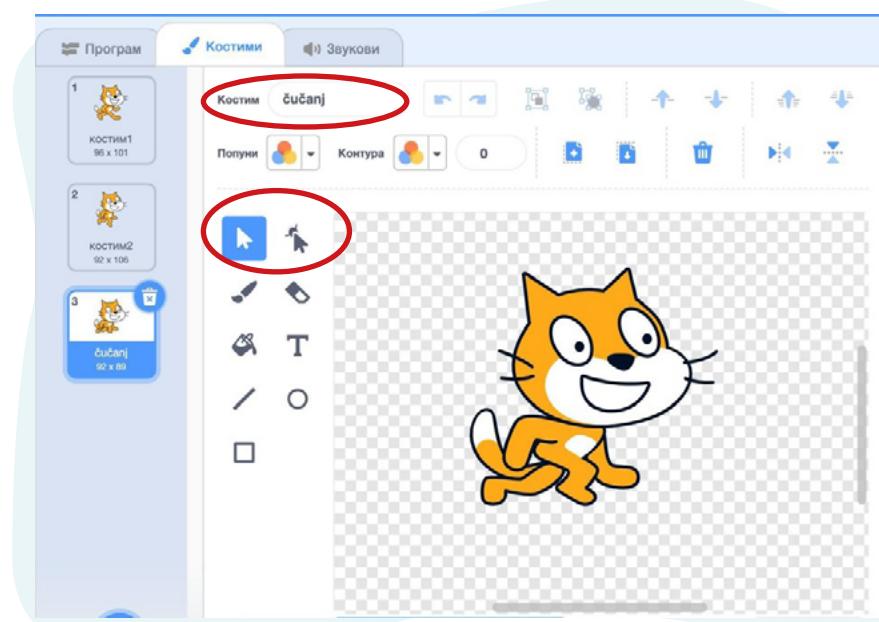
Korak 3. – Skok

Sjeti se optimizacije. Naš lik sada može da skače kada pritisnemo razmak. Kada u isto vrijeme pritisnemo razmak i strelicu lijevo ili desno, on može da preskoči prepreku.



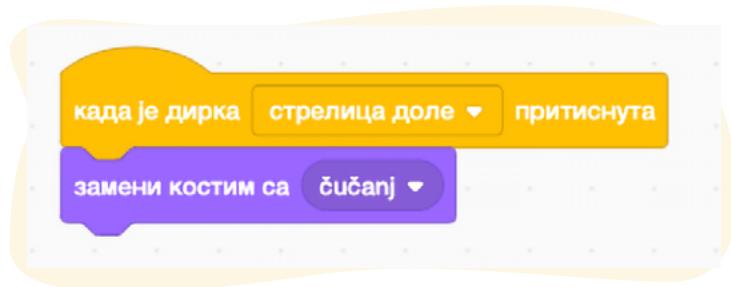
Korak 4. – Kostim

Pravimo novi kostim - čučanj. Klikom na karticu *Kostimi*, desnim klikom klinki na neki od kostima i izaberis *Umnoži*. Umnoženi kostim možeš da mijenjaš na razne načine. Međutim, nama je sada potrebno da oblikujemo tijelo mačka, na način da izgleda kao da čuči. Za to ćemo koristiti ponuđene alatke, a prije nego završimo, ne zaboravi da imenuješ kostim jer će nam kasnije biti potreban.

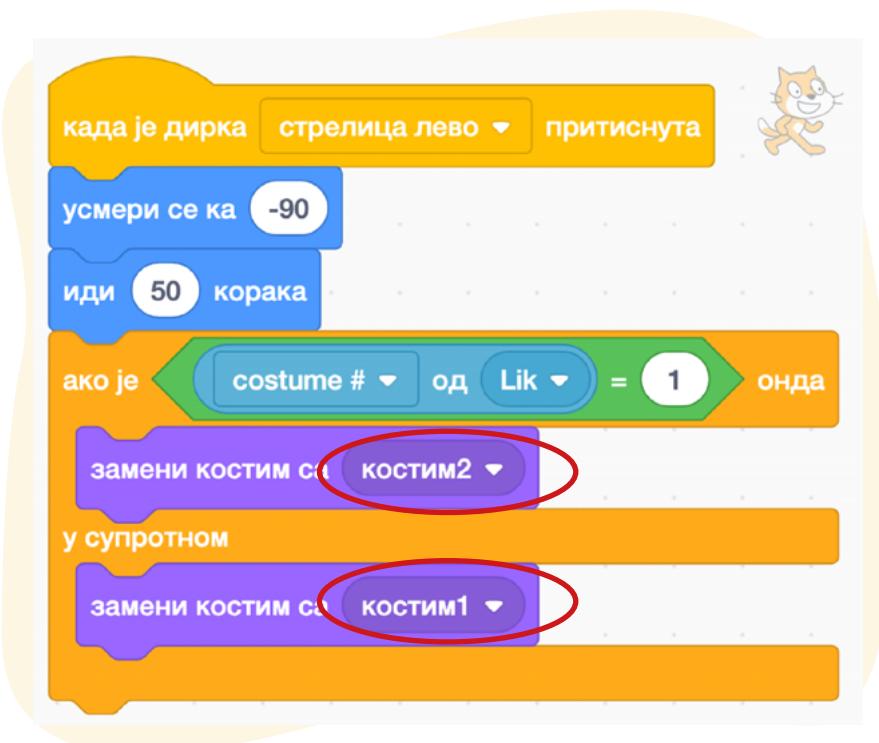


Korak 5. - Čučanj

Nakon što napravimo kostim čučanj potrebno je i da napravimo novi kod.

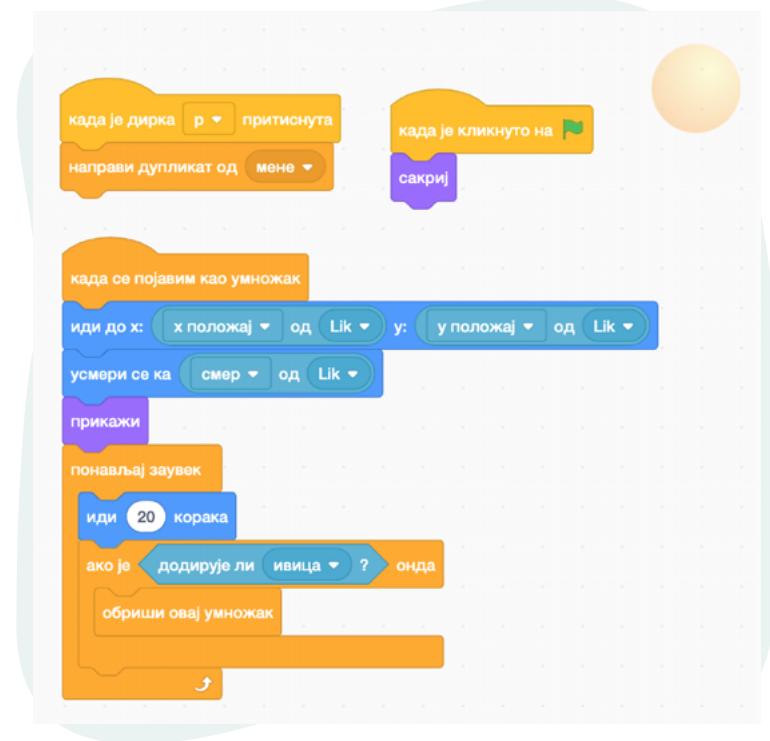


Ne smijemo zaboraviti da novi kostim može da utiče na izgled mačka dok šeta. Zbog toga moramo dati konkretnе komande pri promjeni kostima, kako to ne bi uključivalo kostim čučanj.



Korak 6. - Gadjanje

Sledeći korak jeste gađanje. Ako se osvrnemo na prethodno poglavlje, sjetićeš se da nam je za gađanje potreban novi lik - lopta. I da pritiskom na G naš junak može da pogodi negativnog lika.



Korak 7. – Promjenjive

Kako bi naša igra imala svrhu, pravimo i nove promjenjive koje ćemo kasnije da uvrstimo u kodove. Kao što smo na samom početku rekli potrebni su nam: broj života, poeni i tačni odgovori.

Broj života

Нова променљива

Ново име променљиве:
broj_zivota

За све ликове Само за овај лик

Откажи У реду

Poeni

Нова променљива

Ново име променљиве:
poeni

За све ликове Само за овај лик

Откажи У реду

Tačni odgovori

Нова променљива

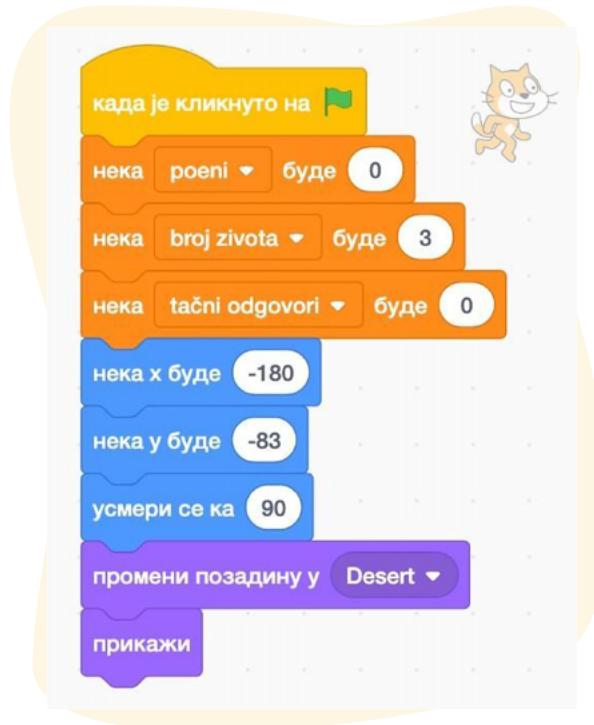
Ново име променљиве:
tačni_odgovori

За све ликове Само за овај лик

Откажи У реду

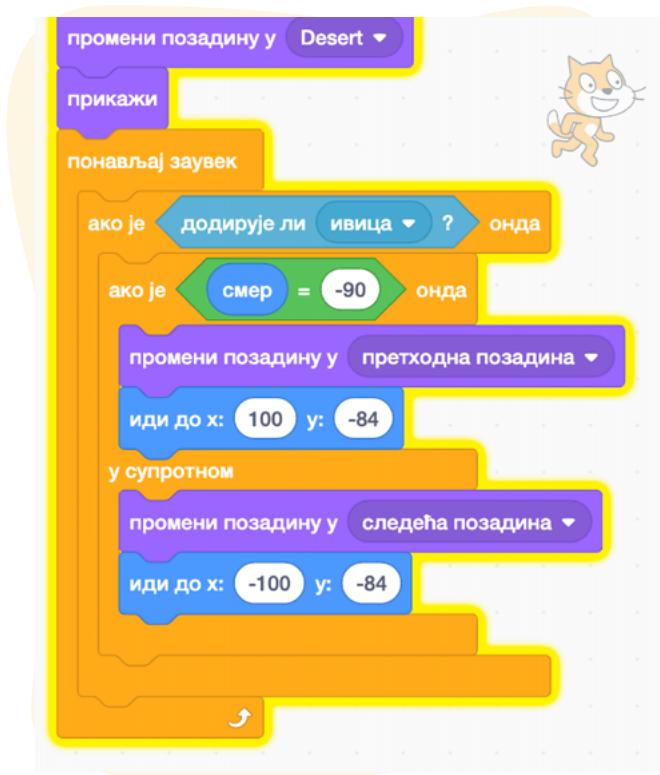
Korak 8. – Početne vrijednosti

Postavljamo početne vrijednosti kako bi sama igra mogla da počne. Pored svih komandi sa slike ispod, možemo podesiti i koji će biti početni kostim našeg lika, ali to je opcionalno.



Korak 9. – Promjena pozadine

Prisjetimo se koda za mijenjanje pozadine koji smo napravili tokom prethodnog poglavlja. On je izgledao ovako:



Međutim, potreban nam je kod koji će odgovarati pravilima koje smo postavili na početku. Lik se može kretati naprijed, a da se pritom pozadine mijenjaju, a nazad se može kretati samo do ivice. Zato, kada dođe do lijeve ivice vraćamo lika malo dalje od same ivice kako ne bi ostao tu zauvijek.



6. Poglavlje

Drugi dio

Prisjetimo se gdje smo stali i koja su pravila igre:

Do sada smo podesili sledeće:

- ✓ Lik može da se kreće naprijed. Kada stigne do ivice, pojavi se sledeća pozadina.
- ✓ Lik može da se kreće unazad, ali samo do ivice.
- ✓ Lik može da skače i preskače.
- ✓ Lik može da čučne.
- ✓ Lik može da puca.
- ✓ Na početku ima 3 života i 0 poena.

Ono što nam je ostalo da programiramo:

1. Kada sakupi preko 30 poena, dobija novi život, a poeni se spuste na 0. Kada broj poena padne ispod 0, gubi život. Ako izgubi sve živote, igra se zaustavlja i piše KRAJ IGRE.
2. Imamo 6 pozadina. Na 2. i 4. se nalaze likovi koji donose po 5 poena kada ih glavni lik dotakne. Na 3. i 5. se nalaze likovi koji oduzimaju po 3 poena ako se dotaknu, a donose 3 ako su pogodjeni.
3. Na pozadini broj 6 se pojavljuje čarobnjak koji postavlja pitanje. Ako je odgovor tačan, čarobnjak nestaje i donosi 3 poena. Ako nije, gubimo 5 poena i čarobnjak nestaje. Nakon 3 tačna odgovora, uspješno se prelazi ovaj nivo.

Ostali likovi

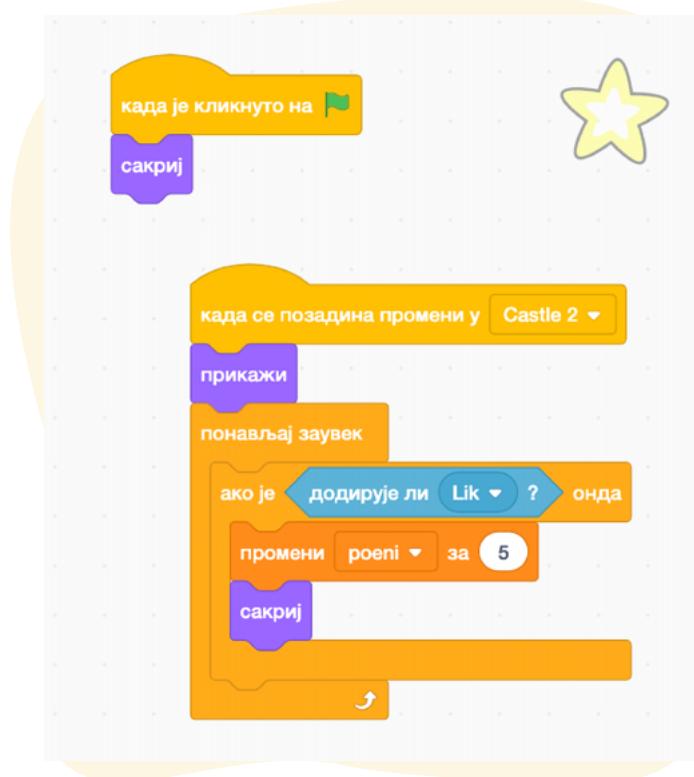
Već imamo glavnog lika i lopticu kojom gađamo. Potrebni su nam:

- 2 dobra lika (dodaju poene),
- 2 loša (oduzimaju poene),
- lik koji postavlja pitanje.

Korak 1. – Dobijanje poena

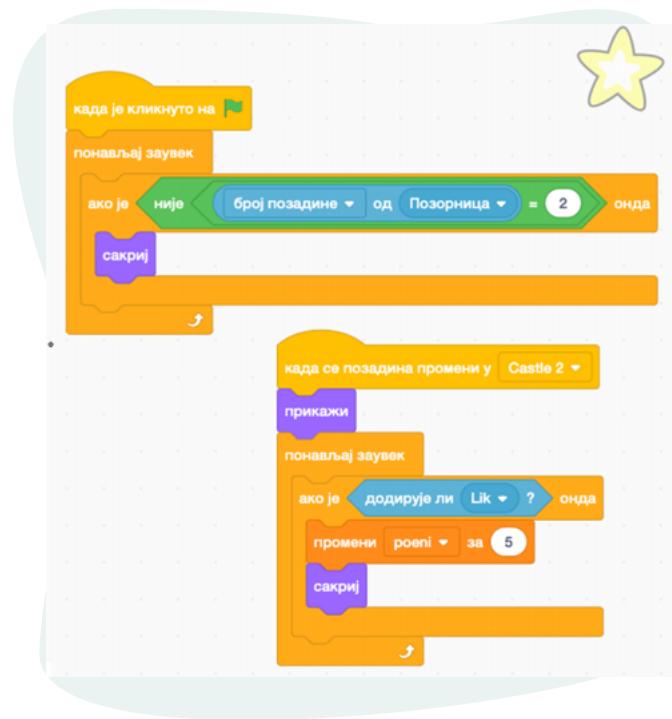
Neka je jedan od dobrih likova zvjezdica, i neka je druga pozadina Castle 2. **Kada glavni lik dodirne zvjezdicu dobija 5 poena, a ona nestaje.**

Isti ovaj princip treba da primijenimo za sledećeg dobrog lika. Slika ispod predstavlja stari kod koji smo kreirali u petom poglavlju i koji ćemo sada nadograditi.



Novi kod bi trebalo da izgleda baš kao sa slike desno.

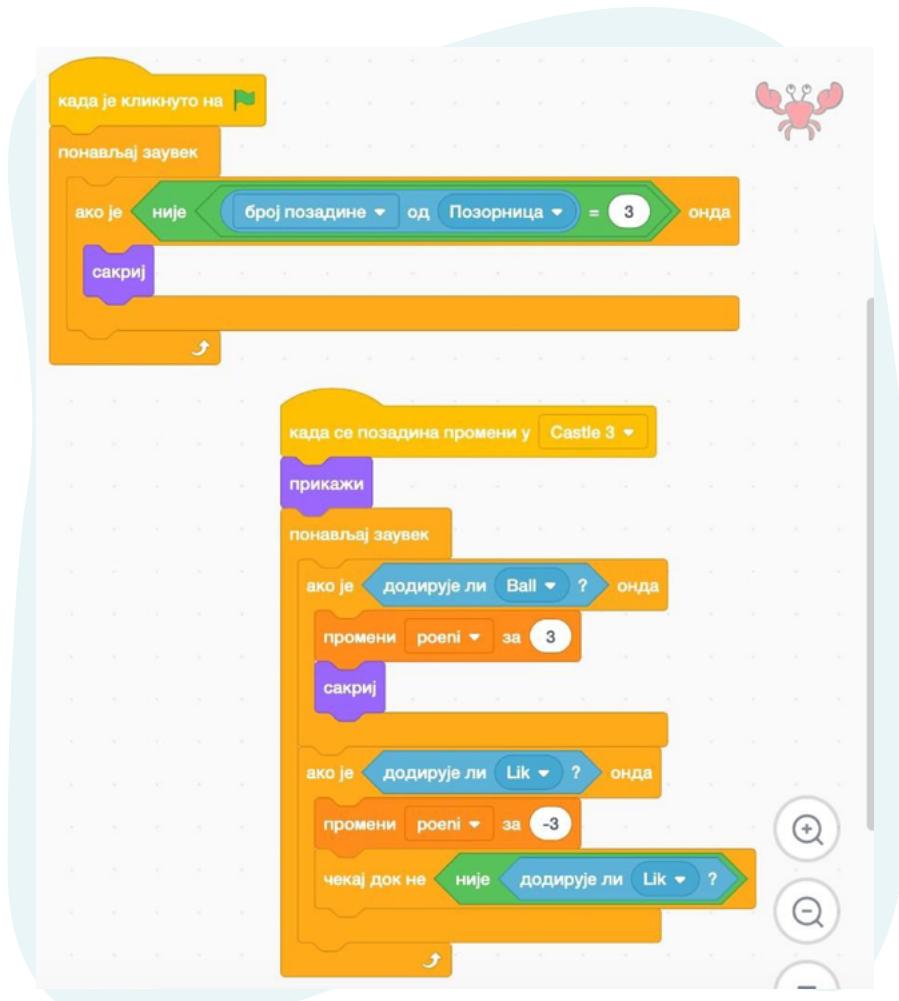
Moramo da obratimo pažnju na to da ne želimo da se zvjezdica vidi na svakoj pozadini. Već da se može naći samo na drugoj po redu. Ovo je potrebno uraditi i za sve ostale dobre likove.



Korak 2. – Zli likovi

Nakon što postavimo komande za svakog dobrog lika, prelazimo na zle likove.

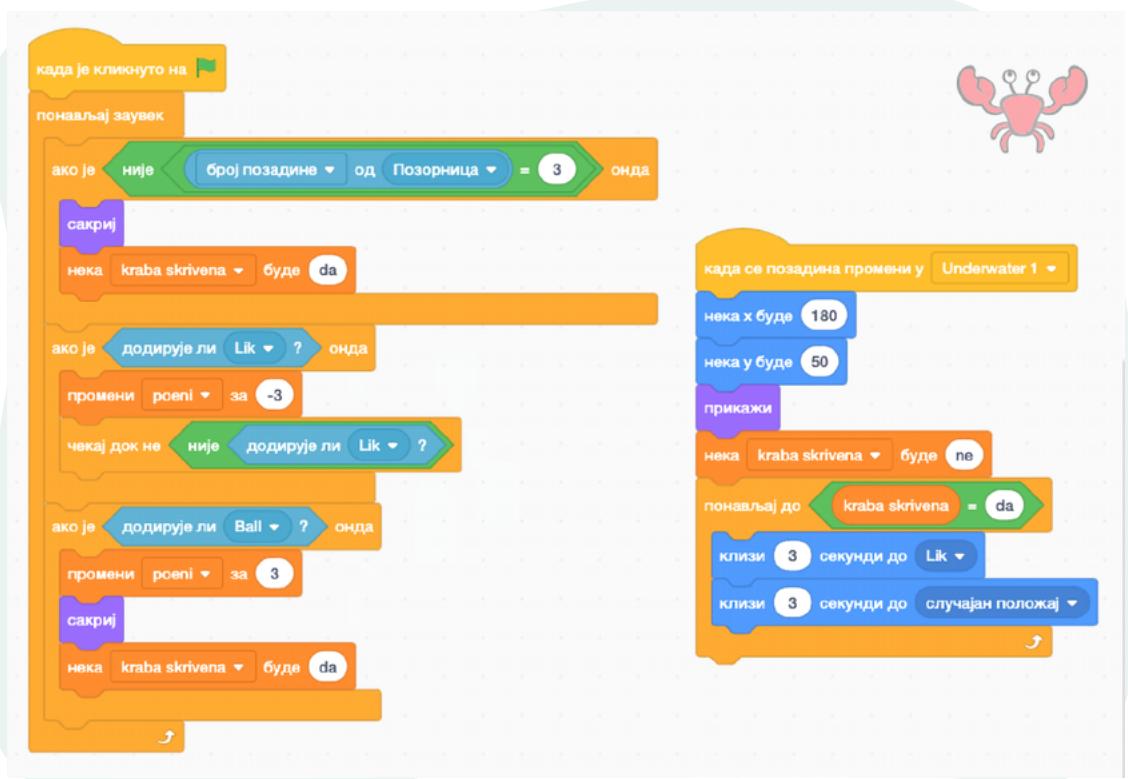
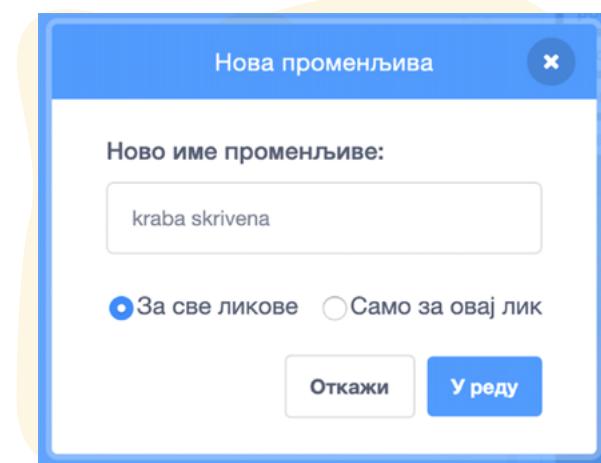
Stavićemo da je jedan od zlih likova kraba, i pojavljuje se na trećoj pozadini. Moramo se prisjetiti da je ideja sledeća - **kada главни lik додирне крабу, он губи 3 поена**, međutim, **ако је погоди лоптом, он добија 3 поена и краба nestaje**. Isti način postavljanja komandi treba primijeniti i na sledećeg “lošeg” lika.



Zli likovi (napredni nivo)

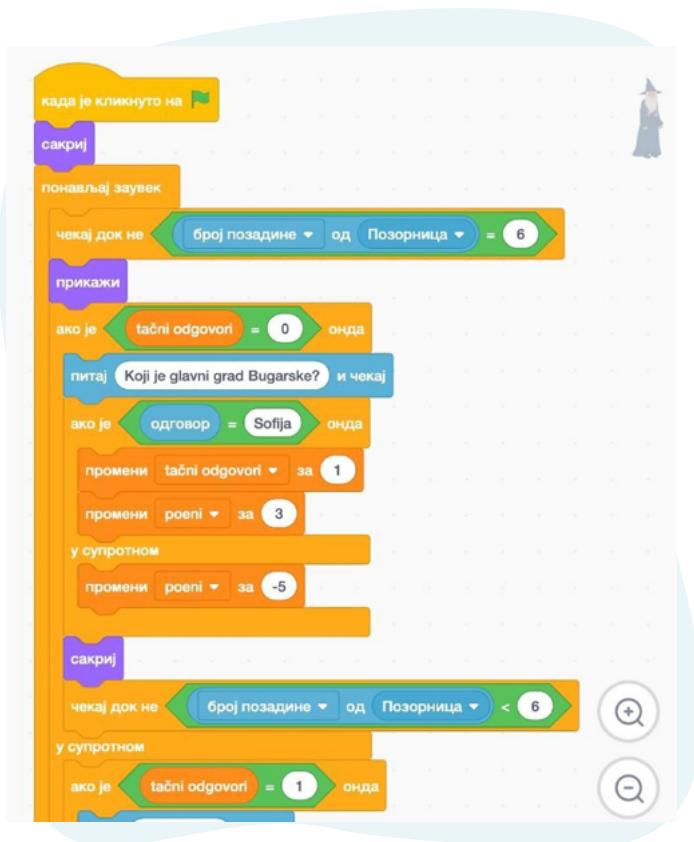
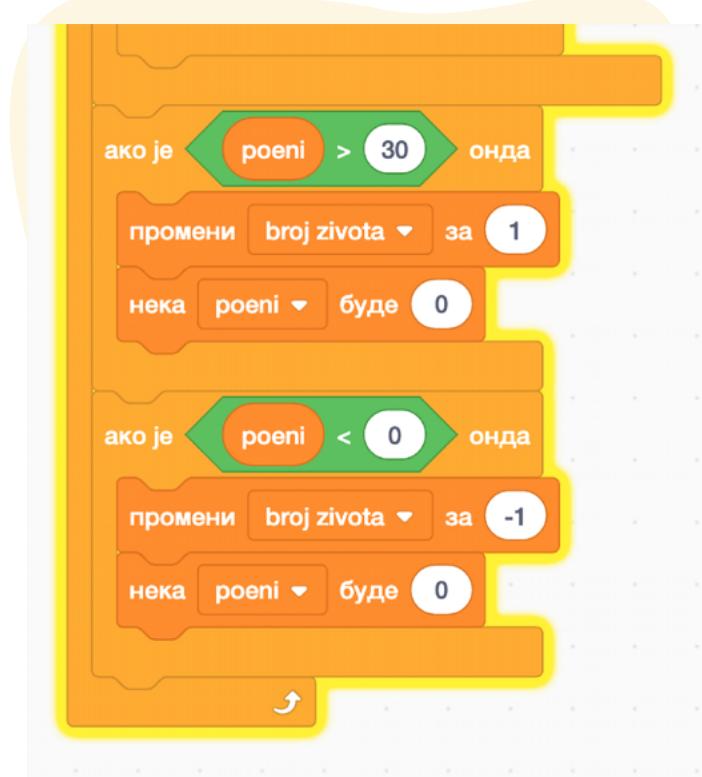
Ovaj korak nije neophodan, ali ako želiš dodatan izazov, stvari možeš učiniti zanimljivijim tako što ćeš dodati komandu da se naši zli likovi kreću. Ukoliko se odlučiš za ovaj korak, ovo možeš uraditi i za sve svoje zle likove, pazeći na broj i ime pozadine na kojoj treba da se pojave.

Koristimo i dodatnu promjenljivu **kraba vidljiva**, da se kraba ne bi, skrivena, uzalud šetala po ekranu. A zatim pratimo kod ispod.



Korak 3. – Poeni i životi

Kada smo završili sa dobrom i lošim likovima, podešavamo broj života i poene našem glavnom liku. Pravilo je glasilo: **Kada sakupi preko 30 poena, dobija novi život, a poeni se spuste na 0. Kada broj poena padne ispod 0, gubi život.**



Korak 4. – Čarobnjak

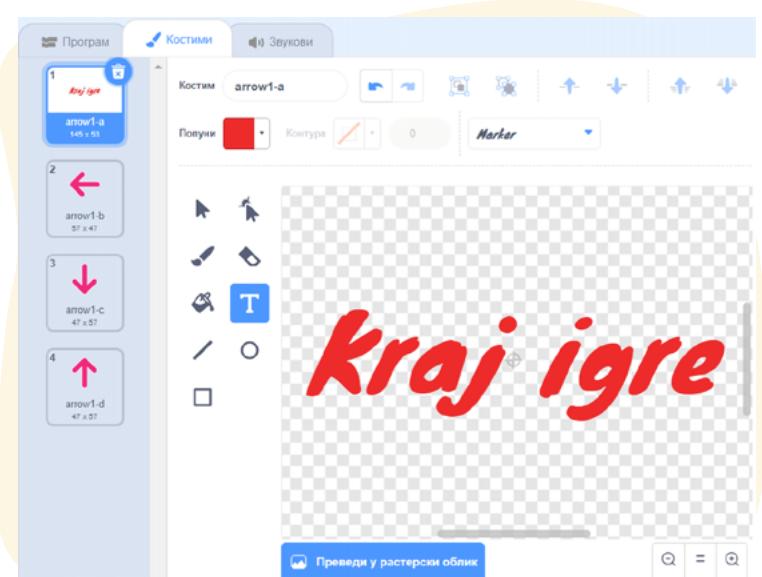
Na pozadini broj 6 se pojavljuje čarobnjak koji postavlja pitanje. **Ako je odgovor tačan, čarobnjak nestaje i donosi 3 poena. Ako nije, gubimo 5 poena i čarobnjak nestaje.** Moramo обратити pažnju da ne ponavlja pitanje na koje smo već tačno odgovorili. Zbog toga vodimo računa o broju tačnih odgovora.

Korak 5. – Kraj igre

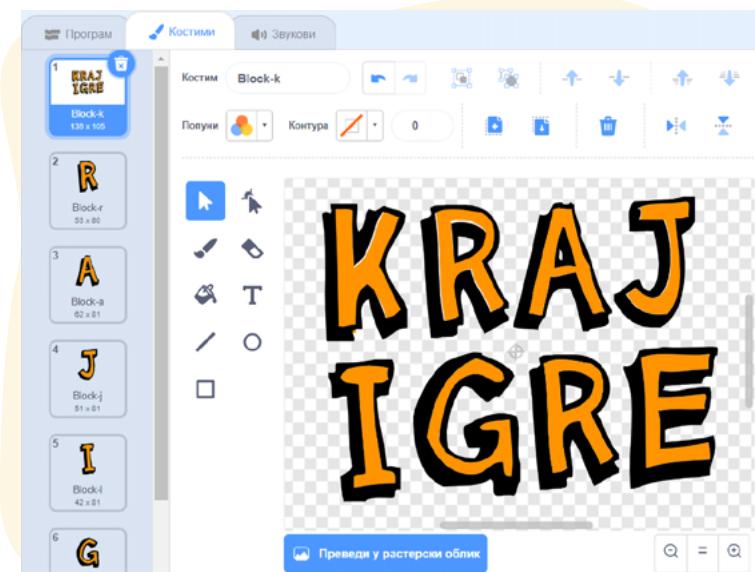
Kako bi na kraju igre, preko ekrana, imali ispisano KRAJ IGRE, izdvojili smo dva načina na koji to možemo da odradimo.

Prvi način: Izaberi bilo kog lika.

Mi smo nasuničnim izborom izabrali strijelu. Uđi na njegov prvi kostim i obriši ga. Potom, klikni na alatku za tekst i ispiši KRAJ IGRE. Kako bi riječi izgledale zanimljivije, slobodno promijeni boju i font po svojoj



Drugi način: KRAJ IGRE možemo napisati tako što ćemo u okviru otvorenog taba za prvo slovo, dodati jedan po jedan kostim koji će kreirati ove riječi. Slova možemo da iskopiramo ili čitav kostim da selektujemo i kopiramo na prvi kostim tako što ćemo kliknuti na plavu ikonicu sa bijelim plusom. Kako bi ga nalijepili na prvobitni kostim u koji ređamo slova, treba da kliknemo na plavu ikonicu sa bijelom strelicom ka dolje. Na kraju slažemo slova, uz dodatnu pažnju na to da smo selektovali sve njihove djelove.



Nakon što kreiramo kostim za završenu igru, potrebno je da izgradimo kod koji će odgovarati utvrđenim uslovima i to: **Ako naš lik izgubi sve živote, igrica se zaustavlja i piše KRAJ IGRE.**



Korak 6. – Pređen nivo

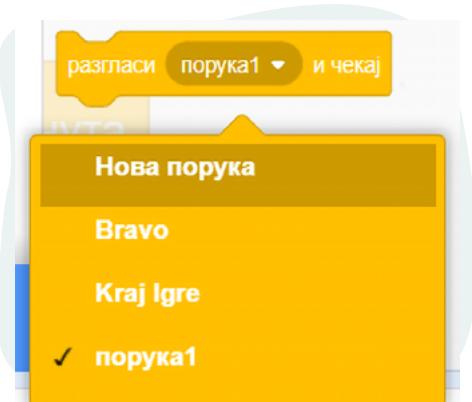
Prisjetimo se čarobnjaka i pravila da **ako damo tačan odgovor 3 puta, prelazimo nivo**. Pa u skladu s tim, kreiramo kod, kao na slici ispod.

Takođe, kada je nivo pređen, napisaćemo BRAVO. To radiš na sličan način kao što smo pravili KRAJ IGRE, samo, naravno, sa drugim slovima.



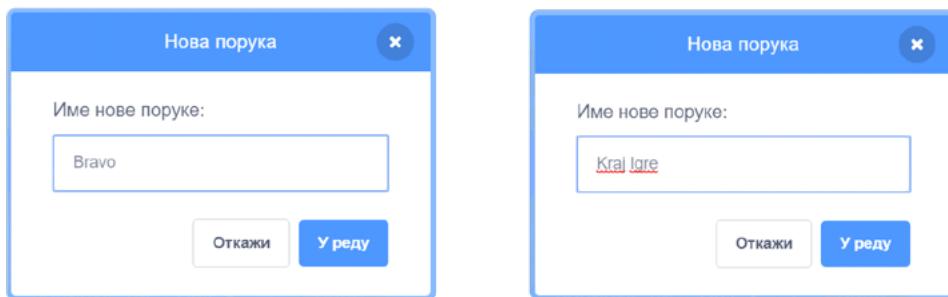
Korak 7. – Zvuk

Zar je igra potpuna bez zvuka? Hajde da upotpunimo doživljaj same igre i dodamo zvuk! Kako do tog koraka? Potrebno je razglasiti kad dođemo do novog nivoa ili do Kraja Igre-a. Na samom kraju ćemo takođe sakriti sve likove.

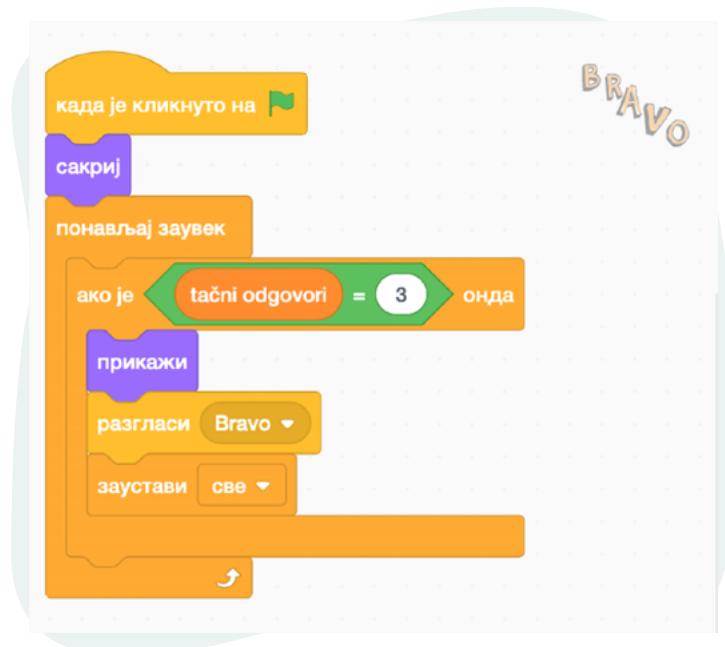


Koristimo blok iz grupe **Događaji - Razglaši**.

Ukoliko nemamo poruku, možemo je lako kreirati klikom na **Nova poruka**, pa zatim i ukucati njen sadržaj.

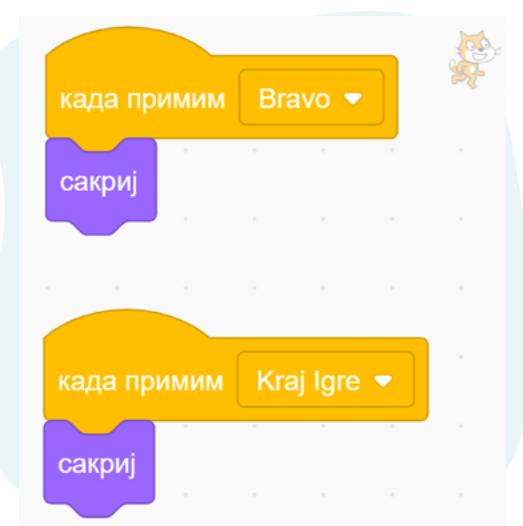


Kako bismo ubacili zvuk, potrebno je da nadogradimo kod pređenog nivoa, a po istom principu dodajemo zvuk i kada je u pitanju Kraj Igre.



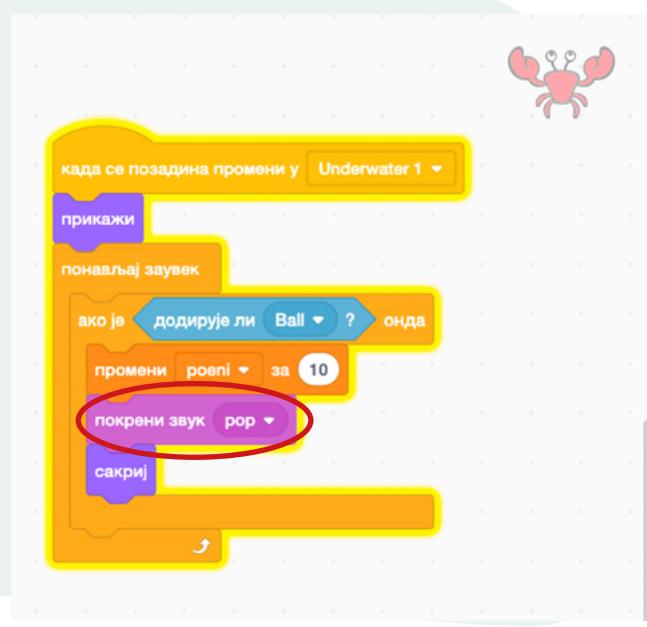
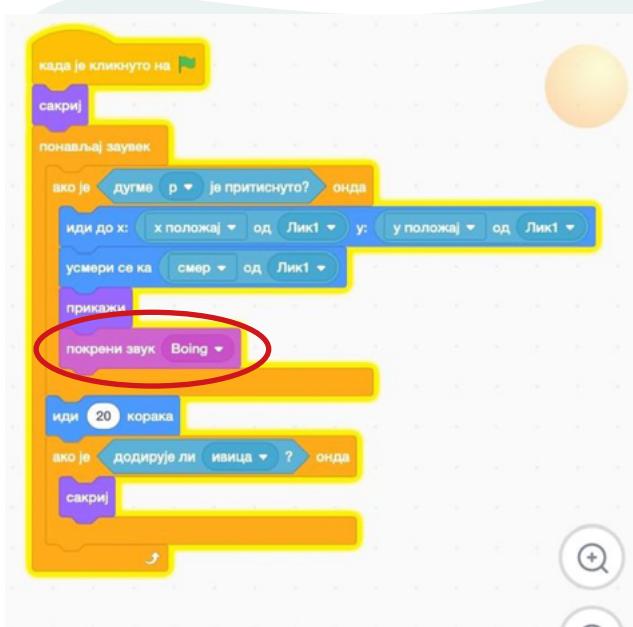
Korak 8. - Sakrivanje likova

Sve likove možemo sakriti na vrlo jednostavan način.



Korak 9. - Zvučni efekti

Zvuk takođe možemo dodati i drugim likovima kako bi igra bila dinamičnija. Kod možemo izmijeniti tako da se čuje neki zvuk kada je loptica bačena i kada je rak pogođen. Sada je vrijeme da sve to sjedimo i isprobamo našu novu igru!



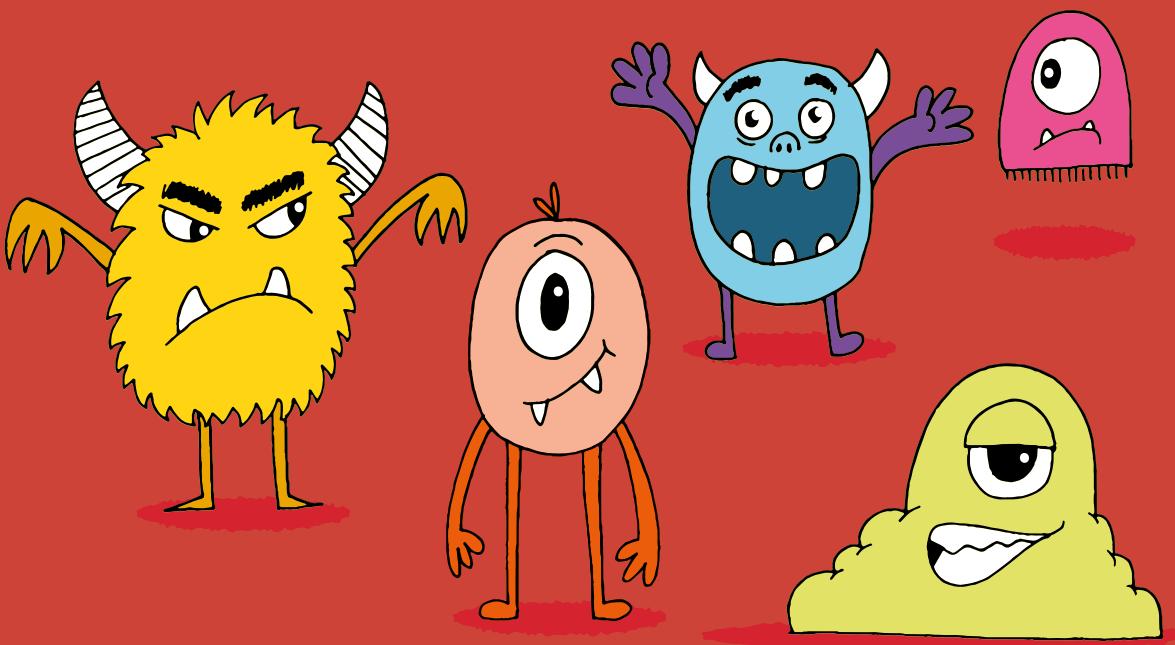
Stigli smo do kraja

Svaka čast! Uspio/la si da kreiraš svoju prvu igru! To što se ovdje završava naš priručnik, ne znači da se ovdje završava i sva zabava.

Ukoliko želiš da proširiš svoje znanje o kreiranju u programskom jeziku Scratch, izdvojili smo dva posebno zanimljiva tutorijala. Igra [Pong](#) je jednostavna za pravljenje, a dovoljno interesantna da je isproba i tvoje društvo i zajedno se zabavite. Takođe, ukoliko možeš ispratiti instrukcije na engleskom jeziku, pogledaj tutorijal za izradu poznate igre [Flappy Bird](#).

Kao što smo već rekli, Scratch je programski jezik koji razvija idejni proces i logičko razmišljanje. Sada kada smo zajedno kreirali jednu igru i naučili osnove ovog programskog jezika, sigurni smo da svaku svoju ideju od sada možeš sprovesti u djelo! Na portalu [STEM edukacija](#) možeš pronaći brojne radne materijale, i to ne samo iz ovog, već i iz drugih programskih jezika.

Podijeli svoje naredne projekte i nastavimo zabavu!



Autorka: Sara Dragojević

Editorke: Ivana Ćorović i Milica Matović

Dizajner: Goran Jokanović

Kako smo došli na ideju da napravimo ovaj priručnik?

Iznenadna situacija nastala usled pandemije virusa korona navela nas je da se zapitamo - šta je ono što bi djeci bilo korisno ali ujedno i zabavno da rade u svojim domovima?

Vrlo brzo smo se dosjetili [DoMEn škole programiranja](#) koja pored kurseva za C++ i Java programske jezike, ima i kurs programiranja namijenjen djeci, u vrlo intuitivnom i jednostavnom programskom jeziku Scratch.

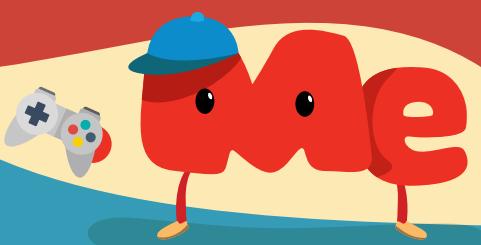
Scratch je nastao na MIT-ju (Massachusetts Institute of Technology) u Americi, kao besplatan vizuelni programski jezik namijenjen svim početnicima u programiranju. Tako lak i zabavan, štivo je koje može zabaviti cijelu porodicu u ovim teškim trenucima.

Ovaj program koriste najčešće djeca u osnovnim školama, ali i studenti i roditelji koji imaju želju da nauče kako da naprave animacije i igrice. Pa navodi mlade istraživače na osmišljanje i kreiranje ideja, kao i dijeljenje istih putem interneta, a savršen je za upoznavanje osnova programiranja i matematike!

Kako i sama [kompanija doMEn](#) već 11 godina sarađuje sa Prirodno-matematičkim fakultetom Univerziteta Crne Gore, ulažući napore da internet i nove tehnologije približi crnogorskom društvu, a posebno mladima, rodila se ideja za ovim Scratch priručnikom.

Kako bismo oživjeli priručnik, koristili smo i materijal koji se može naći na portalu [www.stemedukacija.me](#), koji je pokrenut od strane kompanije doMEn 2019. godine.

Ukoliko ti se priručnik svidio i želiš da proširiš svoje znanje, na ovom portalu možeš naći zanimljivosti iz STEM svijeta, ali i razne prilike za mlade koje kroz svoje društveno odgovorno poslovanje podržava kompanija doMEn. Pa ukoliko se tokom kreiranja ove igre, kod tebe razvila ljubav prema programiranju, [Škola programiranja](#) je baš ono što ti treba! Međutim, ako je ovaj priručnik za tebe ipak bio mačiji kašalj, online takmičenje u programiranju za osnovce i srednjoškolce poput [DoMEn Kupa](#), kao i Olimpijade znanja su prava stvar za tebe!





.Me

ome





STEM Edukacija
powered by: DoMEn